

REPORT DOCUMENTATION PAGE			Form Approved OMB NO. 0704-0188		
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 31-10-2016		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 1-Oct-2014 - 30-Sep-2016	
4. TITLE AND SUBTITLE Final Report: A universal framework for non-deteriorating time-domain numerical algorithms in Maxwell's electrodynamics			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER W911NF-14-C-0161		
			5c. PROGRAM ELEMENT NUMBER 665502		
6. AUTHORS alex Fedoseyev, Semyon Tsynkov			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Computational Sciences, LLC 8000 Madison Blvd., Suite D102-351 Madison, AL 35758 -2035			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSOR/MONITOR'S ACRONYM(S) ARO		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) 66138-MA-ST2.3		
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT The project will remove a key difficulty that currently hampers many existing methods for computing unsteady electromagnetic waves on unbounded regions. Numerical accuracy and/or stability may deteriorate over long times due to the treatment of artificial outer boundaries. We propose to develop a universal algorithm and software that will correct this problem by employing the Huygens' principle and lacunae of Maxwell's equations. The algorithm will provide a temporally uniform guaranteed error bound (no deterioration at all), and the software will enable robust electromagnetic simulations in a high performance computing environment. The methodology					
15. SUBJECT TERMS Unsteady electromagnetic waves; Maxwell's equations; unbounded regions; (grid) truncation; artificial outer boundaries; artificial boundary conditions (ABCs); non-reflecting boundaries; perfectly matched layer (PML); long-time deterioration; loss of accuracy,					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Edward Kansa
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU			19b. TELEPHONE NUMBER 256-270-0956

Report Title

Final Report: A universal framework for non-deteriorating time-domain numerical algorithms in Maxwell's electrodynamics

ABSTRACT

The project will remove a key difficulty that currently hampers many existing methods for computing unsteady electromagnetic waves on unbounded regions. Numerical accuracy and/or stability may deteriorate over long times due to the treatment of artificial outer boundaries. We propose to develop a universal algorithm and software that will correct this problem by employing the Huygens' principle and lacunae of Maxwell's equations.

The algorithm will provide a temporally uniform guaranteed error bound (no deterioration at all), and the software will enable robust electromagnetic simulations in a high-performance computing environment. The methodology will apply to any geometry, any scheme, and any boundary condition. It will eliminate the long-time deterioration regardless of its origin and how it manifests itself. Dr. Tsynkov who co-invented this method is the Academic partner on the project.

Phase I included development of an innovative numerical methodology for high fidelity error-controlled modeling of a broad variety of electromagnetic and other wave phenomena. Proof-of-concept 3D computations have been conducted that convincingly demonstrate the feasibility and efficiency of the proposed approach.

In Phase II our algorithms will be implemented as robust commercial software tools in a standalone module that can be combined with existing numerical schemes in computational electromagnetic codes.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

<u>Received</u>	<u>Paper</u>
-----------------	--------------

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

<u>Received</u>	<u>Paper</u>
-----------------	--------------

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received Book

TOTAL:

Received

Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Dr. Edward Kansa, the project PI in the Year 1, has been awarded the 2015 George Green Medal for his pioneering work on Meshless Methods.

He is on a way to the Wessex Institute of technology (UK) for the award ceremony on Sep. 21, 2015 (G.Green is the inventor of Green function).

Link:

<http://wessex-institute-of-technology.blogspot.jp/2015/02/2015-george-green-medal-g-e-orge-green.html>

or

<http://wessex-institute-of-technology.blogspot.jp/>

Graduate Students

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
FTE Equivalent:	
Total Number:	

Names of Under Graduate students supported

NAME

PERCENT SUPPORTED

FTE Equivalent:

Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:..... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:..... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):..... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:..... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:..... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME

PERCENT SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

(4) Statement of the problem studied

The project has removed a key difficulty that currently hampers many existing methods for computing unsteady electromagnetic waves on unbounded regions. Numerical accuracy and/or stability may deteriorate over long times due to the treatment of artificial outer boundaries. We developed a universal algorithm and software that will correct this problem by employing the Huygens' principle and lacunae of Maxwell's equations.

The algorithm provides a temporally uniform guaranteed error bound (no deterioration at all), and the software will enable robust electromagnetic simulations in a high-performance computing environment. The methodology is applicable to any geometry, any scheme, and any boundary condition. It eliminates the long-time deterioration regardless of its origin and how it manifests itself.

Phase I included development of an innovative numerical methodology for high fidelity error-controlled modeling of a broad variety of electromagnetic and other wave phenomena. Proof-of-concept 3D computations have been conducted that convincingly demonstrate the feasibility and efficiency of the proposed approach.

In Phase II our algorithms have been implemented as robust software tools and has been combined with existing numerical schemes in several computational electromagnetic codes.

(5) Summary of the most important results

The project has been a success:

(i) We have done a further theoretical development of the Lacuna method,

(ii) Successfully implemented the Lacuna method for three widely used Computational Electromagnetic Codes,

(iii) Demonstrated the accuracy, stability of the Lacuna method for a number of important for DoD codes and applications:

(5.1) LNL Overture code implementation (key developer Dr. William Henshaw, a collaborator on the project).

OVERTURE is a toolkit for solving partial differential equations on structured, overlapping, and hybrid grids. It provides a high level C++ interface for rapid prototyping of PDE solvers built upon optimized C and Fortran kernels, a library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations, and support for block structured adaptive mesh refinement (AMR). OVERTURE has extensive grid generation capabilities, CAD interfaces, interactive graphics and data base support, command scripts with embedded Perl commands. PDE solvers built upon OVERTURE include CGMX — a time domain Maxwell's equations solver, see [Henshaw 2006], CGMP — a multi-physics solver, see [Henshaw 2006], and others. The CGMX solver is designed for HPC and demonstrates an impressive linear parallel scaling (CPU time vs. number of nodes).

OVERTURE has 10 users at DOE Labs, 40 (heavy) users in the Academia, and about 10 users in the industry.

he key developer of the OVERTURE, Dr. William Henshaw, is a collaborator on the project.

Important practical applications demonstration presented:

(i) Electromagnetic scattering about a three-dimensional reentry vehicle, using (a) Lacuna method alone, and (b) Lacuna + PML combination (problem suggested by Dr. Richard Albanese).

(ii) Electromagnetic scattering about a buried obstacle (detection of IED in soil). This problem has been proposed by Dr. Traian Dogaru, Adelphi, MD U.S. Army Research Laboratory (traian.v.dogaru.civ@mail.mil). His team has conducted advanced research on Sensing-Through-the-Wall (STTW) radar technology for imaging complex building environments, static and moving human signature, see e.g. [Dogaru 2010].

(5.2) WARPX (Washington Approximate Riemann Plasma code) is a multi-fluid plasma/Maxwell simulation code from U. of Washington, the most advanced plasma simulation in the world. WARPX employs 5Momentum & 13Momentum models, and different discretization methods, finite element/DG and finite volumes [Meier 2012].

WARPX has 19 users at AFRL, R&D companies, plasma physics Labs (Princeton U. and others).

We have demonstrated that Lacuna method greatly improves the accuracy of (i) Quasi-spherical Electromagnetic Pulse and (ii) Rotated Quasi-spherical Electromagnetic Pulse simulations.

The key developer of the WARPX, Dr. Uri Shumlak, is a collaborator on the project.

(5.3) ICEPIC AFRL Particle-In-Cell code, a critical AFRL technology. TPOC: Dr. A. Greenwood.

ICEPIC is the Improved Concurrent Electromagnetic Particle-In-Cell code used to model high power microwave (HPM) devices: high-power microwave tubes, including magnetrons, klystrons, gyrotrons, and Magnetically Insulated Line Oscillators (MILOs). ICEPIC has been developed by numerous scientists and engineers since 1994 at the AFRL Kirtland AFB, Albuquerque, NM, and is considered as a critical AFRL technology. Dr. A. Greenwood is one of the key developers of ICEPIC and a TPOC for ICEPIC, has strongly supported our project on Lacuna implementation into ICEPIC.

ICEPIC has about 20 users at AFRL, many of them are experimentalists. The code is extensively used for virtual prototyping of high-power microwave devices. Besides AFRL, ICEPIC is used at the Air Force Institute of Technology and at Arizona State University;

- in-house modular/portable code,

We have successfully demonstrated Lacuna method with PML for the case of Vlasov antenna simulation.

(5.4) We have demonstrated that the prototype software development has been completed successfully, can be applied to a number of important to DoD problems, and outlined further research and development related to that. Defense uses of this technology include high fidelity numerical simulations for applications such as the propagation of electromagnetic waves in free space and in materials, and scattering of electromagnetic waves off the shapes and structures. This could be used for the virtual prototyping of detection and tracking systems, e.g., air surveillance radars or OTH radars.

The lacunae-based methodology appear well suited for advancing a number of applications of interest to the US Army. Specifically, one of the long-term projects pursued by the electric-field sensing team of the US Army Research Lab in Adelphi, MD (led by Dr. Dave Hull, 301-394-3140, david.m.hull6.civ@mail.mil) involves upgrading a key piece of the team's equipment, the E-field cage that is used for testing the various sensors [Hull 2006]. The upgrade is geared toward higher frequencies, addition of the magnetic field, and unsteadiness. Numerical simulations will provide a very efficient design tool for developing this upgrade.

All major results of the project are published in [Fedoseyev2016]

(6) Bibliography

[Henshaw 2006] William D. Henshaw. A high-order accurate parallel solver for Maxwell's equations on overlapping grids. SIAM J. Sci. Comput., 28(5):1730–1765 (electronic), 2006.

[Hull 2006] D.M. Hull, S.J. Vinci, and Yongming Zhang. ARL electric-field cage modeling, design and calibration. In 2006 12th Biennial IEEE Conference on Electromagnetic Field Computation, pages 89–89, Miami, FL, April–May 2006.

[Meier 2012] E. T. Meier, A. H. Glasser, V. S. Lukin, and U. Shumlak. Modeling open boundaries in dissipative MHD simulation. J. Comput. Phys., 231(7):2963–2976, 2012.

[Dogaru 2010] Traian Dogaru, Anders Sullivan, Calvin Le and Chris Kenyon. Radar Signature Prediction for Sensing-Through-the-Wall by Xpatch and AFDTD - Part II, High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC) 2010 DoD, pp. 401-406, 2010.

[UWB radar kit] Evaluation Kit GZ6EVK (Evaluation Kit for UWB Radar Application), GZ6EVK opt 31106 , 5,746.00 EUR/set, http://www.geozondas.com/Old_Site/Evaluation%20Kit/UWB_Radar_Evaluation_Kit.htm

[Fedoseyev 2016] A. Fedoseyev, E. J. Kansa, S. Tsynkov, S. Petropavlovskiy, M. Osintcev, U. Shumlak and W. D. Henshaw, \newblock AIP Conf. Proc. 1773, 020001 (2016); dx.doi.org/10.1063/1.4964955

Technology Transfer

Implemented our developed non-deteriorating Lacuna time-domain numerical algorithms into several highly popular software codes for solving electromagnetic and plasma problems, used by DOD, DOE, Academia and industry:

1. Implemented Lacuna method into OVERTURE code (developed by LLNL, DOE, key developed W.D.Henshaw, our collaborator on a project).

OVERTURE has 10 users at DOE Labs, 40 (heavy) users in the Academia, and about 10 users in the industry.

Important practical applications demonstration presented:

(i) Electromagnetic scattering about a three-dimensional reentry vehicle, using (a) Lacuna method alone, and (b) Lacuna + PML combination (problem suggested by Dr. Richard Albanese, DoD scientist, retired).

(ii) Electromagnetic scattering about a buried obstacle (detection of IED in soil). This problem has been proposed by Dr. Traian Dogaru, Adelphi, MD U.S. Army Research Laboratory (traian.v.dogaru.civ@mail.mil). His team has conducted advanced research on Sensing-Through-the-Wall (STTW) radar technology for imaging complex building environments, static and moving human signature

2. Implemented Lacuna method into ICEPIC AFRL Particle-In-Cell code, a critical AFRL technology.

ICEPIC has about 20 users at AFRL, many of them are experimentalists. The code is extensively used for virtual prototyping of high-power microwave devices. Besides AFRL, ICEPIC is used at the Air Force Institute of Technology and at Arizona State University.

We demonstrated a dramatic improvement of stability and accuracy for long time simulation for Vlasov antenna design.

3. Implemented Lacuna method into WARPX multi-fluid plasma/Maxwell simulation code. WARPX (Washington Approximate Riemann Plasma code) has 19 users at AFRL, R&D companies, plasma physics Labs (Princeton U. and others). We have demonstrated that Lacuna method greatly improves the accuracy of (i) Quasi-spherical Electromagnetic Pulse and (ii) Rotated Quasi-spherical Electromagnetic Pulse simulations.

A universal framework for non-deteriorating time-domain numerical algorithms in Maxwell's electrodynamics

Alex Fedoseyev, Principal Investigator, Semyon Tsynkov, Co-Investigator
Edward J. Kansa, Uri Shumlak, Sergei Petropavlovskiy,
William D. Henshaw, Michael Osintcev, Jordan Angel,
Computational Sciences LLC,
8000 Madison Blvd, STE D102,
Madison, AL 35758-2035

Final report #24

Submitted to
Dr. Joseph Myers
US Army Research Office
Research Triangle Park, NC 27709-2211
Contract W911NF-14-C-0161

October 31, 2016

Abstract

The project will remove a key difficulty that currently hampers many existing methods for computing unsteady electromagnetic waves on unbounded regions. Numerical accuracy and/or stability may deteriorate over long times due to the treatment of artificial outer boundaries. We propose to develop a universal algorithm and software that will correct this problem by employing the Huygens' principle and lacunae of Maxwell's equations. The algorithm will provide a temporally uniform guaranteed error bound (no deterioration at all), and the software will enable robust electromagnetic simulations in a high-performance computing environment. The methodology will apply to any geometry, any scheme, and any boundary condition. It will eliminate the long-time deterioration regardless of its origin and how it manifests itself. Dr. Tsynkov who co-invented this method is the Academic partner on the project. Phase I included development of an innovative numerical methodology for high fidelity error-controlled modeling of a broad variety of electromagnetic and other wave phenomena. Proof-of-concept 3D computations have been conducted that convincingly demonstrate the feasibility and efficiency of the proposed approach. In Phase II our algorithms will be implemented as robust commercial software tools in a standalone module that can be combined with existing numerical schemes in computational electromagnetic codes.

Keywords: Unsteady electromagnetic waves; Maxwell's equations; unbounded regions; (grid) truncation; artificial outer boundaries; artificial boundary conditions (ABCs); non-reflecting boundaries; perfectly matched layer (PML); long-time deterioration; loss of accuracy, loss of stability, error build-up; the Huygens' principle, aft fronts of the waves; lacunae of the solutions; quasi-lacunae; accumulation of charge; guaranteed accuracy; temporally uniform error bounds.

Monthly Technical Progress Report

Proposal number: A2-5691

Title of the project: A universal framework for non-deteriorating time-domain numerical algorithms in Maxwell's electrodynamics

Contract performance period: 30-Sep-2014 to 30-Sep-2016

Total contract amount: \$999,893.28

Amount of funds paid by DFAS to date: \$958231.06

Total amount expended/invoiced to date: \$999893.28

Number of employees working on the project: 5

Number of new employees placed on contract this month: 0

Contractor's Name: Computational Sciences LLC

Contractor's Address: 8000 MADISON BLVD, STE D-102, MADISON, AL 35758-2035

Progress Report ID: CLIN 0001AK

Report Date: 15-October-2016

Classified or sensitive information contained: No

Summary

The project has been successfully completed, a very substantial progress has been made by the team along all major research directions:

- (1) Further theoretical development of the lacuna method;
- (2) Numerical study of performance of the various ABCs/PMLs combined with lacunae-based time marching using the newly developed modular in-house Fortran code;
- (3) Implementation and testing of the lacunae-based time marching in the code **OVERTURE**;
- (4) Application of the developed tools to important practical problems: Electromagnetic scattering about a three-dimensional reentry vehicle;
- (5) Application of the developed tools to important practical problems: Electromagnetic scattering about a buried obstacle (e.g., an IED in soil);
- (6) Application of the lacunae-based approach to other types of boundaries and interfaces, beyond the artificial outer boundaries;
- (7) Implementation and testing of the lacunae-based time marching in the code **ICEPIC**;
- (8) Implementation and testing of the lacunae-based time marching in the multi-fluid plasma code **WARPX**.

Our effort, as it applies to the individual items on the foregoing list, includes the following:

(1) In our prior work, we have shown theoretically and corroborated numerically that the Huygens' principle and lacunae can help construct efficient far-field closures for the computer simulation of unsteady waves propagating over unbounded regions, Section 1. Those closures can be either standalone or combined with other techniques for the treatment of artificial outer boundaries. A standalone lacunae-based closure can be thought of as a special artificial boundary condition (ABC) that is provably free from any error associated with the domain truncation. If combined with a different type of ABC or a perfectly matched layer (PML), a lacunae-based approach can help remove any long-time deterioration (e.g., instability) that arises at the outer boundary regardless of why it occurs in the first place.

A specific difficulty associated with Maxwell's equations of electromagnetism is that classical lacunae in their solutions may be replaced by quasi-lacunae. Unlike in the classical case, the field inside the quasi-lacunae is not zero; instead, there is an electrostatic solution driven by the electric charges that accumulate over time. In the course of our current effort, we have shown that quasi-lacunae can also be used for building the far-field closures. However, for achieving a provably non-deteriorating performance over arbitrarily long time intervals, the accumulated charges need to be known ahead of time.

To remove this limitation, we have modified the algorithm in such a way that one can rather avoid the accumulation of charge altogether. This was accomplished by deriving special corrections to the currents that drive the electromagnetic field. Accordingly, the field inside the quasi-lacunae becomes equal to zero, which facilitates obtaining the temporally uniform error estimates as in the case of classical lacunae. The range of problems that the new version of the method can address includes important combined formulations, for which the interior sub-problem may be non-Huygens', and only the exterior sub-problem, i.e., the far field, is Huygens', Section 2.

(2) We went beyond our previously reported implementation of the complete radiation boundary conditions by Hagstrom and Warburton [53–55] that involved only one side of a 3D cubic computational domain, and adopted the full-fledged version that involves all six faces of the domain. It employs the double absorbing boundaries, see [56–58], and has recently been made available by the authors via www.rbcpack.org, Sections 3, 4, 5, 6, 7.

Computational complexity of ABCs, tested and used in our research, has been evaluated in Section 8. It is noted that non-solenoidal source is more challenging for the ABCs. Namely, the performance of both the unsplit PML and the DAB method deteriorates very considerably already for moderate simulation times.

(3) Initially, the lacunae-based time marching was implemented in *OVERTURE* in its pure form, i.e., it was not combined with any third party treatment of the artificial outer boundary and rather provided a standalone closure in its own right. This approach is unimprovable from the standpoint of accuracy and stability, yet it is relatively costly. More recently, we have focused on the combined implementation of lacunae with PML in *OVERTURE*. By now, we have developed and tested a comprehensive treatment of the artificial outer boundary in *OVERTURE* by means of a PML+lacunae. This approach results, in particular, in a very significant speedup of simulations, Section 9.

(4) To demonstrate the performance of the combined numerical methodology that we have developed, we applied it to the analysis of electromagnetic scattering about a 3D objects: perfectly electrically conducting (PEC) sphere, and the 3D model reentry vehicle (RV). This is a non-canonical yet practically important formulation. We found out that the standard approach in *OVERTURE*, which is based on the pure PML treatment of the outer boundary, may result in a rapidly divergent solution already at short simulation times (depending on the PML parameters and the impinging frequency). On the other hand, the application of the lacunae-based time marching yields a stable and accurate solution for arbitrarily long computation times, Section 9.

Numerical experiments that pertain to items (3) and (4) were conducted on the Dell PowerEdge 7300 Linux server at NCSU (tsynkov-srv01.math.ncsu.edu). The server has 32 CPU cores, two Intel Phi MIC cards, and 512 GB of fast RAM.

(5) Another important practical application is the electromagnetic scattering about an improvised explosive device (IED) buried in soil, for the purpose of assisting to the development of an ultra-wideband radar (UWB) detection technology. The Army research into countermeasures such as an UWB radar is a key weapon in the battle to protect the warfighters from the very real threat posed by IEDs and hidden ordnance. The work on the modeling of electromagnetic scattering from an underground embedded body using *OVERTURE* included the problem formulation, setting the far-field boundary conditions, and the condition on the air-soil interface. Problem parameters have been selected to simulate the realistic case, including the radar pulse shape (the chirp of a given rate and duration), as well as the initial and boundary conditions. Preliminary results are provided and discussed, Section 17.

(6) Using the lacunae-based approach for solving general non-Huygens’ initial boundary value problems (IBVPs), we have computed solutions to the interior Dirichlet and Neumann problems for the d’Alembert equation. Besides the lacunae per se, our approach employs the method of difference potentials by Ryaben’kii. Previously, it has been applied to a variety of steady-state and time-harmonic problems, in particular, to solving the Helmholtz equation [59–62]. The Huygens’ principle and lacunae [63–65] provide a key tool for computing the difference potentials in the unsteady framework. The Dirichlet solution of the d’Alembert equation that we have computed demonstrates the design (second order) rate of grid convergence, and the overall numerical method proves efficient and amenable to parallelization. The Neumann solution also shows the design rate

of convergence on any finite time interval, yet the maximum error grows linearly in time. We are interested in obtaining an interpretation of this phenomenon in the framework of the general well-posedness theory for the d'Alembert equation, Sections 10, 11, 12.

(7) The ICEPIC code has been successfully installed and tested on a number of testcases, Section 13. The analysis conducted on the ICEPIC source code led to the selection of a candidate module as a template for the lacunae implementation in ICEPIC — the PML module of ICEPIC. This provided a convenient structure for the solution of the auxiliary problem. The Lacunae module implementation in ICEPIC code has been completed. Two important test cases have been selected for the ICEPIC demonstration and testing: AFRL A6-3 magnetron, Section 14, and Vlasov antenna, Section 15.

(8) In the code WARPX, the progress centers on analyzing the quasi-spherical electromagnetic pulse and rotating quasi-spherical electromagnetic pulse, and on developing benchmark simulation cases to test the lacunae-based boundary condition implementation in WARPX. Simulating open boundary conditions in three dimensions offers the greatest potential computational cost savings. Initializations that exercise this capability must be derived to satisfy the physical constraints described by the governing Maxwell's equations. Benchmark data has been generated for the simulation on an extended domain that effectively removes any boundary condition interaction with the propagating pulse. Several simulations have been performed using the same initial conditions on a truncated domain surrounded by lacunae-based auxiliary boundary domains. The development of the Lacunae module in WARPX has been successfully completed, Section 16.

All major results of the project are presented in paper published by the American Institute of Physics CP [161].

Contents

1	Introduction	10
1.1	General background	10
1.2	Lacunae and the Huygens' principle	11
1.3	Relation between the Huygens' construction and the Huygens' principle.	11
2	Theoretical development	13
2.1	Temporally uniform error estimate for general non-Huygens' problems.	13
2.1.1	Quasi-lacunae of Maxwell's equations.	14
2.1.2	Lacunae-based time marching for Maxwell's equations.	15
2.1.3	Non-Huygens' problems and decomposition technique.	16
2.1.4	Making quasi-lacunae classical.	20
2.1.5	A temporally uniform error bound.	24
2.2	Application of lacunae to other types of boundaries	27
3	Implementation of the algorithm for computations by the in-house Fortran code	28
3.1	Algorithm of Section 2.1 applied to an unsplit PML.	28
3.1.1	Test solution.	29
3.1.2	Implementation.	31
3.2	Modular version of the in-house code	33
3.2.1	Programming language.	33
3.2.2	Parallel multi-processor computations.	34
3.2.3	Objects and structure.	35
3.2.3.1	Mesh and source.	35
3.2.3.2	Problem.	36
3.2.3.3	PML.	36
3.2.3.4	Buffer.	38
3.2.3.5	Solution.	38
4	Lacunae-based methodology optimization in the in-house Fortran code	41
5	Implementation of low order local ABCs with the in-house code	42
5.1	Sommerfeld boundary condition	42
5.2	Higdon ABC	43
5.3	Betz-Mittra ABC	43
5.4	Mur ABC	43
5.5	Numerical experiments	43
5.6	Intermediate conclusions	46
6	Implementation of high order local ABCs by Givoli & Neta with the in-house code	46
6.1	Second order accurate approximation of Givoli-Neta ABCs	47
6.1.1	Number of angles.	47
6.1.2	Size of the computational domain.	48
6.1.3	Choice of angles — three angles.	48
6.1.4	Choice of angles — six angles.	49
6.1.5	Grid size.	50
6.1.6	Intermediate conclusion.	50
6.2	First order accurate approximation of Givoli-Neta ABCs in time	51
6.2.1	Number of angles.	51
6.2.2	Size of the computational domain.	51
6.2.3	Choice of angles — three angles.	53

	6.2.4	Choice of angles — five angles.	53
	6.2.5	Intermediate conclusion.	54
	6.3	Combined implementation of lacunae and Givoli-Neta ABCs	55
	6.4	Conclusions on Givoli-Neta ABCs	56
7		Complete radiation boundary conditions	57
	7.1	Overview	57
	7.2	FDTD implementation	58
	7.3	Boundary conditions for auxiliary functions	59
	7.4	Long-time instability and lacunae-based time marching	61
	7.5	The role of unsplit PML	64
	7.6	Application of lacunae with original divergence-free currents	65
	7.7	Application of lacunae with auxiliary currents	67
	7.8	Intermediate conclusions on Hagstrom-Warburton NRBC	73
8		Computational complexity of ABCs	77
	8.1	Introduction	78
	8.2	Estimation of computational complexity	78
	8.3	Test solutions and exact boundary condition	78
	8.4	Local low-order ABC	79
	8.5	Unsplit uniaxial PML	79
	8.6	Lacunae-based time marching algorithm with PML	80
	8.7	Lacunae-based time marching algorithm with PML and divergence-free aux- iliary currents	83
	8.8	Lacunae-based time marching algorithm without PML	83
	8.9	Double absorbing boundary layer	85
	8.10	Numerical simulations - serial implementation	86
	8.11	Parallelization with OpenMP	89
9		Numerical experiments with OVERTURE	91
	9.1	Background	91
	9.2	Governing equations	92
	9.3	Reference solution for type I experiments	93
	9.4	Computational setup	94
	9.4.1	Interior problem and computational domain.	94
	9.4.2	Auxiliary problems.	94
	9.4.2.1	Sources for the auxiliary problem.	94
	9.4.2.2	Boundary closure for the auxiliary problems.	96
	9.5	Type I experiments	97
	9.5.1	Parameters of the computational setting.	97
	9.5.2	Grids.	98
	9.5.3	PML design and parameters.	98
	9.5.4	Results of computations.	100
	9.6	Type II experiments	103
10		Application of lacunae to other boundaries	120
	10.1	General	120
	10.2	The method of difference potentials	121
	10.3	Lacunae and the Huygens' principle	121
	10.4	Computing Calderon's operators with the help of lacunae	122
11		Numerical tests for lacunae and the Dirichlet problem	124

11.1	Formulation of the problem	124
11.2	Reference solution	124
11.3	Application of difference potentials	124
11.4	Partial auxiliary problems	126
11.5	Parameters of the experiment	126
11.6	Parallelization	126
11.7	Results of the experiments	126
12	Computations for lacunae and the Neumann boundary conditions	128
12.1	Notation and terminology	128
12.2	The Klein-Gordon equation	129
12.2.1	Results of the Klein-Gordon simulations.	130
12.3	Choosing ∇u instead of u for the wave equation	130
12.3.1	Results of simulations.	131
12.3.2	Caveat.	131
12.4	Lacunae-based implementation	132
12.5	Intermediate conclusions	132
13	Lacunae implementation in the AFRL ICEPIC code	134
13.1	Introduction to the AFRL ICEPIC code	134
13.2	Users of ICEPIC	134
13.3	Algorithms implemented in ICEPIC	134
13.3.1	Improved Yee finite-difference time-domain (FDTD) technique.	135
13.3.2	Altering the computational stencil in ICEPIC.	137
13.3.3	Filtering.	139
13.3.4	ICEPIC parallel performance.	140
13.3.5	Summary on parallel performance.	141
13.4	ICEPIC installation and testing	143
13.5	PML module in ICEPIC	145
13.6	Formulation	146
13.7	Tapering PML Parameters	148
13.8	Specification of PML to ICEPIC	148
13.9	Advanced Topics	151
14	Practical ICEPIC applications: magnetron characterization and design	153
14.1	AFRL A6-3 magnetron testcase for Lacuna method demonstration.	154
15	Practical ICEPIC applications: Vlasov antenna design	157
15.1	Vlasov antenna optimization	157
15.2	Vlasov antenna simulation for Lacuna method demonstration	159
16	Lacuna BC Implementation in WARPX multi-fluid plasma code	163
16.1	Quasi-spherical Electromagnetic Pulse	163
16.2	Benchmark Case	165
16.3	Rotated Quasi-spherical Electromagnetic Pulse	167
16.4	New Benchmark Case	167
16.5	Simulations with Lacunae-based Boundary Conditions	167
16.6	Porting and Running WARPX on ComScis Computer	173
17	Applications of the developed tools: Modeling of electromagnetic scattering from an underground embedded body (IED in soil)	175
17.1	Problem specification	178
17.2	Simulation Time	179

17.3	One-way Wave	179
18	Results for electromagnetic scattering from a buried body	182
18.1	Plane Wave Source and Initial Conditions	182
18.1.1	Periodic in x	182
18.1.2	Radiation Boundary Conditions.	184
18.2	Antenna Source	186
18.3	Solving for Scattered Field Directly	186
18.4	Square	187
18.5	Embedded Cylinder	188
18.6	Embedded Body	189
18.7	Sharpness Study	190
18.8	Convergence	193
18.9	Annular Domain	194
18.10	Discussion of the results for Buried Body	195
List of Figures		196
List of Tables		205
References		206

1 Introduction

1.1 General background

Artificial boundary conditions (ABCs) help truncate the original domain and obtain a finite-dimensional discretization for the numerical simulation of waves propagating over unbounded regions [1, 2]. The role of the ABCs is to make the outer boundary transparent for all the outgoing waves, and to eliminate the unphysical reflections. For unsteady problems, it is particularly important that the non-reflecting properties of a given ABC are maintained over long times.

In the field of computational electromagnetism (CEM), there is a number of established techniques to build the ABCs for Maxwell’s equations. There are popular ABCs that are local in space and time, and are typically derived using asymptotic considerations, e.g., those of Bayliss and Turkel [3–5], Engquist and Majda [6–8], or Higdon [9, 66] (see also the reviews [1] and [2]). However, some of these techniques, while demonstrating excellent grid truncation properties over limited time intervals, may suffer from a deterioration of performance in long time computations. The empirical study [10] demonstrates that the solution obtained using Higdon and Bayliss-Turkel fourth order ABCs in the two-dimensional Cartesian case becomes highly unstable already at moderate computational times; and that the instability most likely arises near the corners of the computational domain. Susceptibility of local ABCs to long time error growth is also discussed in [54].

Another very efficient approach to the absorption of waves at the outer boundaries, the so-called perfectly matched layers (PMLs), may also lead to long time instabilities. They first appear in the layer and subsequently propagate back into the domain. The original split-field PML by Bérenger [11, 12] turns out to be only weakly well-posed, and a linear growth of the split field components inside the PML is possible already at the analytical level.¹ In addition, it has been proven in [13] that the popular Yee scheme [71] becomes unconditionally unstable in the presence of the PML [11], with the powers of amplification matrices growing linearly as the number of time steps increases.

Subsequent developments gave rise to unsplit PMLs, see, e.g., [14, 76], but the latter have also been found susceptible to gradually developing instabilities. First, the instabilities were predicted theoretically, and then also corroborated experimentally, see, e.g., [15]. A systematic computational study of the long time performance of unsplit PMLs with some explicit second order schemes can be found in [16]. A similar study that compares the numerical performance of various ABCs and PMLs, but in the frequency domain, is presented in [17].

Several approaches have been proposed in the literature to correct the aforementioned long time instabilities. However, the “stabilized” versions of Higdon’s and Liao’s [18, Sec. 6.5] boundary conditions offered an improved performance but did not eliminate the instability completely [10]. Furthermore, efforts to stabilize higher order ABCs were unsuccessful [19]. And when the stability is achieved, it often compromises the accuracy over the lower end of the frequency spectrum [20].

In that regard we note that early versions (from 1990s) of high order local ABCs such as Engquist-Majda, Higdon, or Liao involved high order derivatives that were explicitly approximated by high order differences. This approach could lead to “mild” instabilities of the GKS type due to the generalized eigenvalues in the sense of H.-O. Kreiss, see [21, Chapter 13] or [22, Section 10.5.4]. It is this kind of instability that the paper [19] attempts to address. Implementation of local high order ABCs via auxiliary variables has apparently eliminated the GKS-type instabilities, although some methods still demonstrate an unsatisfactory performance over long times. It is also to be mentioned that necessary and sufficient conditions for the continuous well-posedness of local high

¹Weakly well-posed systems may become ill-posed under small perturbations, and the corresponding example is given in [13]. In the implementation of PMLs, however, this scenario does not get realized.

order ABCs of Engquist-Majda type are provided in [23]. Subsequent work on long time behavior includes [24, 25].

Stabilization techniques for PMLs include changing the governing equations in the layer [15] and introducing frequency-shifted [26–28] or non-linear [29] PMLs. All these modifications perform well computationally, but theoretically it is not clear whether the modified layer remains perfectly matched or absorbing, in particular, absorbing regardless of the frequency.

1.2 Lacunae and the Huygens’ principle

An efficient approach to long time stabilization is based on lacunae [109] and the Huygens’ principle. Its key advantage is that it does not require a priori knowledge of what actually causes the problem in each specific case, but provides a universal methodology that corrects it regardless of both the type of the ABC (or PML) and the nature of its deterioration. In doing so, the method preserves all the useful properties that a given ABC displays before the deterioration onset, such as the degree of transparency for a local method or the matching and absorption properties for a PML.

It is important that our algorithm requires no problem-specific fine-tuning and thus facilitates high fidelity modeling for a variety of electromagnetic and other wave dominated phenomena. It can also be used as an ABC on its own, in which case it allows for a quite general shape of the artificial boundary and remains provably free from any error associated with the domain truncation.

The lacunae-based technique of [63, 80, 110, 111] has been successfully applied to various wave propagation problems. It allows one to obtain temporally uniform error bounds, which is the main reason why it proves useful for eliminating the long time deterioration. In [112], we have used it for stabilizing the unsplit PMLs. In [64], we have introduced the concept of quasi-lacunae that generalizes that of classical lacunae for Maxwell’s equations. In the case of quasi-lacunae, there may be a steady-state electrostatic solution behind aft fronts of the propagating waves, whereas for genuine lacunae the solution behind the aft fronts is zero. In [65], we have constructed a new version of the long time stabilization algorithm based on quasi-lacunae as opposed to classical lacunae. This algorithm is considerably more versatile than those of [63] and [112]. We also emphasize that unlike the ABCs of [54], our lacunae-based algorithm does not require that the simulation interval be specified ahead of time, and rather guarantees that the error due to the treatment of the artificial boundary will be bounded for all times.

To be able to stabilize a given ABC or PML by means of the lacunae-based methodology, we need to make only one important assumption. Namely, *the original ABC or PML should maintain its design characteristics over a relatively short yet reasonable interval of time*. If, for example, the lacunae-based algorithm were applied along with the local high order ABCs of [54] that are optimized for accuracy and performance over a predetermined interval T , then the same accuracy would be guaranteed for arbitrarily long times. If, on the other hand, the original treatment of the outer boundary causes an immediate rapid increase of the error, then the lacunae-based approach will not be able to correct it. For example, following the methodology of [30], it was proven in [31] that Higdon’s and other similar ABCs are strongly stable in the sense that they guarantee the existence of a non-increasing energy functional. As, however, pointed out by the authors of [31], this requires sufficiently smooth data. Otherwise, Higdon’s ABCs may become “sharply” unstable, and then lacunae-based integration will not provide a remedy.

1.3 Relation between the Huygens’ construction and the Huygens’ principle.

A number of papers in the literature refer to the Huygens’ principle as a useful tool for applications, including the development of numerical methods. In most cases, however, what these publications

exploit is not the full fledged Huygens’ principle, but rather the well-known Huygens’ construction, which is common, e.g., in optics [32]. The difference is fundamental. The Huygens’ principle sets forth the existence of aft fronts of the waves and is accompanied by additional constraints, such as the odd number of space dimensions. The Huygens’ construction, on the other hand, deals only with the propagating fronts; it considers the front at every given moment of time as a collection of secondary sources so that the propagating field at subsequent moments of time can be interpreted as the field due to those sources.

The Huygens’ secondary sources and the Huygens’ construction have proven useful, in particular, in the area of active control of sound. The JMC method (Jessel–Mangiante–Canevet), see [33–38] and also [39, 40], uses the information about the noise to be canceled at the perimeter (surface) of the region to be protected, and explicitly builds the Huygens’ anti-noise sources in the form of acoustic monopoles, dipoles, and quadrupoles at the boundary (see [41] for a brief comparison of JMC with other active noise control methods). A similar idea was used by Charles, et. al. in [42] and Mangiante and Charles in [43] to build the absorbing boundary conditions for computational acoustics. Namely, the Huygens’ secondary sources in [42, 43] eliminate the outgoing acoustic waves in the special transition region that surrounds the computational domain.

In electromagnetism, the Huygens’ construction can be interpreted as a particular realization of the electromagnetic equivalence theorem by Schelkunoff [44], see also [18, Sec. 8.4]. This theorem says that the field on a given region (interior or exterior), regardless of its actual sources located outside of this region, can be reproduced as the field from the specially chosen auxiliary currents at the boundary of the region. If the boundary is identified with a “frozen” propagating front at a given fixed moment of time, then the equivalence theorem yields the Huygens’ construction.

In work [45, 46], Béranger uses the equivalence theorem to design the subgridding techniques for the numerical solution of Maxwell’s equations; in work [47], the subgridding approach of [45, 46] is extended by Costen and Béranger to the case of frequency dependent (i.e., dispersive) media; and in work [48], Béranger builds the Huygens’ absorbing boundary conditions for time-domain Maxwell’s equations. In this approach, the Huygens’ secondary sources cancel out the outgoing electromagnetic field and hence prevent it from getting reflected off the artificial outer boundary. Even though there is no formal connection, work [48] can be thought of as an extension of work [42, 43] from the scalar acoustic fields to the vector electromagnetic fields.

We re-emphasize though that all of the aforementioned papers exploit only the Huygens’ construction for forward propagation, and never make any use of the full Huygens’ principle, i.e., of the presence of aft fronts and lacunae. The latter represent a considerably more subtle phenomenon, and as mentioned by Gårding in [49], can be thought of as “the result of fancy cancellations between branches of analytic functions.” There are actually very few publications in the literature that do attempt to apply the ideas related to the aft fronts and lacunae in the numerical context, and among those we mention work by Warchall [50, 51]. The focus of this work is on the study of domains of dependence and domains of influence for constant coefficient hyperbolic systems, and on the development of the treatment of artificial boundaries that would take advantage of the fact that the data at a given time only inside the domain fully determine the propagation through the outer boundary at all subsequent times. This work by Warchall, however, presents only preliminary theoretical observations. Its practical implementation, if there is ever one, is contingent upon the availability of “an explicit method for propagation of waves through an artificial domain boundary, using only data at a single time inside the domain, in such a way as to duplicate exactly the propagation in unbounded space.” As of yet, such a method has not been found. Contrary to that, our lacunae-based algorithm [65] uses the right-hand sides (source terms of the governing equations), rather than the Cauchy data, to propagate the information through the artificial boundaries.

2 Theoretical development

2.1 Temporally uniform error estimate for general non-Huygens' problems.

The sources of electromagnetic field are the electric charges and currents. They are related by the continuity equation. In particular, when the currents are given, charges can be reconstructed by integrating the continuity equation in time.

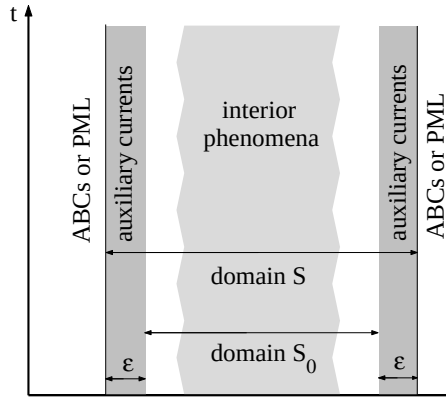


Figure 1: Computational setup. The phenomena of interest are inside S , outside is the free space, which is truncated off and replaced with an ABC or a PML. The auxiliary sources are in a narrow region of width ε right next to ∂S on the inside.

The algorithm of [65] relies on special auxiliary currents. They are needed in the case when the problem inside the computational domain is non-Huygens'. Then, the original problem is split into the interior and auxiliary sub-problems, see Figure 1. The resulting interior (sub-)problem is set on the bounded computational domain, whereas its auxiliary counterpart is first considered on the entire space, and then truncated by the chosen ABC (or PML). The two problems are connected to each other. Solution to the auxiliary problem (AP) provides the outer boundary data for the interior problem and thus renders its closure. In turn, the AP itself is driven by the source terms that are derived from the solution of the interior problem right next to the boundary.

The advantage of the foregoing split is that the interior problem contains all the essential near field components of the original formulation yet it does not need any ABCs. The AP is a plain Cauchy problem for constant coefficient Maxwell's equations; it is driven by the prescribed auxiliary currents, and truncated by the chosen ABC or PML. Hence, the AP is much easier to analyze than the original problem. First and foremost, the AP is a Huygens' problem and can therefore be solved with the help of lacunae, which leads to a non-deteriorating long time behavior, see [65].

However, the temporally uniform error bound was proven in [65] only for the case when both the currents and charges are given explicitly as functions of space and time. For most APs this is not the case. Typically, only the auxiliary currents can be considered given, whereas the corresponding charges cannot. The charges are rather reconstructed via the continuity equation, and the static contributions to the overall field that they generate inside the quasi-lacunae are summed up cumulatively. In this case, the argument that establishes the temporally uniform error bound in [65] no longer holds. In practice, though, we have never observed any error accumulation/growth when using the quasi-lacunae. It is therefore highly desirable to develop a version of the algorithm for which the non-deteriorating behavior in the general non-Huygens' case would be theoretically justified. In this section, we report on the corresponding development.

The key idea is to modify the auxiliary sources so that they will generate no static field, which is equivalent to transforming the quasi-lacunae back to classical lacunae.

In Section 2.1.1, we discuss the quasi-lacunae of Maxwell's equations that will be reduced to classical lacunae in our new algorithm. In Section 2.1.2, we discuss the time-marching technique for the Maxwell system in vacuum, which can be based on either quasi-lacunae or classical lacunae should the latter exist. In Section 2.1.3, we introduce a class of problems that our modified method will apply to, and describe the decomposition of those problems into the interior and auxiliary subproblems. We also emphasize that the auxiliary subproblems have quasi-lacunae as opposed

to classical lacunae. Section 2.1.4 introduces the key construct that we have introduced recently — a new form of the auxiliary currents that do not generate the static field behind the aft fronts and help reduce the quasi-lacunae to classical lacunae. To enable this reduction, one needs to build a divergence-free extension of the exterior field to the interior computational domain. In Section 2.1.4, we also investigate when this extension is possible. Section 2.1.5 presents a rigorous justification of the temporally uniform error bound in the non-Huygen’s case, which guarantees that no error due to the treatment of the artificial boundary will accumulate over time. This result is enabled by the use of classical lacunae.

2.1.1 Quasi-lacunae of Maxwell’s equations. Consider Maxwell’s equations in vacuum

$$\frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} - \text{curl } \mathbf{H} = -\frac{4\pi}{c} \mathbf{j}, \quad (1)$$

$$\frac{1}{c} \frac{\partial \mathbf{H}}{\partial t} + \text{curl } \mathbf{E} = -\frac{4\pi}{c} \mathbf{j}^M, \quad (2)$$

where $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields and c is the speed of light. Equations (1)-(2) are driven by the electric $\mathbf{j}$ and magnetic $\mathbf{j}^M$ currents respectively, although the latter is introduced artificially and does not exist in nature. It makes system (1)-(2) symmetric and, more importantly, appears crucial for the analysis of non-Huygens’ problems in Section 2.1.3. The currents are related to the respective electric ρ and magnetic ρ^M charge densities by the conventional continuity equations

$$\frac{\partial \rho}{\partial t} + \text{div } \mathbf{j} = 0, \quad \frac{\partial \rho^M}{\partial t} + \text{div } \mathbf{j}^M = 0. \quad (3)$$

According to (3), the charges ρ and ρ^M evolve over time at every spatial location where the currents are defined and may accumulate unless the currents are solenoidal, i.e., unless $\text{div } \mathbf{j} = 0$ and $\text{div } \mathbf{j}^M = 0$. By taking the divergence of both parts of equations (1) and (2) we obtain the expressions relating the fields and the respective charges:

$$\frac{\partial}{\partial t} \text{div } \mathbf{E} = 4\pi \frac{\partial \rho}{\partial t}, \quad \frac{\partial}{\partial t} \text{div } \mathbf{H} = 4\pi \frac{\partial \rho^M}{\partial t}, \quad (4)$$

or, integrating with appropriate initial conditions (which should satisfy $\text{div } \mathbf{E}_0 = 4\pi\rho_0$, $\text{div } \mathbf{H}_0 = 4\pi\rho_0^M$), we arrive at the time-dependent versions of the Gauss laws for electricity and magnetism:

$$\text{div } \mathbf{E}(t) = 4\pi\rho(t), \quad \text{div } \mathbf{H}(t) = 4\pi\rho^M(t). \quad (5)$$

Equations (1)-(2) can be reduced to the independent wave (d’Alembert) equations for $\mathbf{E}$ and $\mathbf{H}$:

$$\frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} - \Delta \mathbf{E} = -4\pi \left[\frac{1}{c^2} \frac{\partial \mathbf{j}}{\partial t} + \frac{1}{c} \text{curl } \mathbf{j}^M + \text{grad } \rho \right], \quad (6)$$

$$\frac{1}{c^2} \frac{\partial^2 \mathbf{H}}{\partial t^2} - \Delta \mathbf{H} = -4\pi \left[\frac{1}{c^2} \frac{\partial \mathbf{j}^M}{\partial t} - \frac{1}{c} \text{curl } \mathbf{j} + \text{grad } \rho^M \right]. \quad (7)$$

Now let us assume that the currents are compactly supported on some bounded space-time domain $Q = S \times [t_0, t_1] \subset \mathbb{R}^3 \times [0, \infty)$, i.e., they are localized in space (within the spatial domain S) and operate only for a finite time $t_1 - t_0$. It can be shown, see [64], that the solutions to equations (6) and (7) possess quasi-lacunae, i.e., regions behind the sharp aft fronts of the propagating waves

where the solution has already reached its steady-state value and does not evolve in time any more. Let $\mathbf{E}^{st}$, $\mathbf{H}^{st}$ denote those steady-state values for the electric and magnetic field, i.e., the electrostatic and magnetostatic components generated by the accumulated electric and magnetic charges, respectively. Then, we have the quasi-lacuna:

$$\mathbf{E}(\mathbf{x}, t) = \mathbf{E}^{st}, \quad \mathbf{H}(\mathbf{x}, t) = \mathbf{H}^{st}, \quad \forall \mathbf{x}, t \in \bigcap_{(\boldsymbol{\xi}, \tau) \in Q} \{(\mathbf{x}, t) : |\mathbf{x} - \boldsymbol{\xi}| < c(t - \tau), \quad t > \tau\}.$$

2.1.2 Lacunae-based time marching for Maxwell's equations. Suppose we want to solve the Maxwell system (1)-(2) on a truncated computational domain S . At first, we will assume that the currents $\mathbf{j}$ and $\mathbf{j}_M$ differ from zero on S only and operate for a finite time $[t_0, t_1]$. Due to the existence of quasi-lacunae, by the time $t = T_{\text{int}}$, where

$$T_{\text{int}} \equiv \frac{1}{c} \text{diam } S + T_0 \quad (8)$$

and $T_0 \equiv t_1 - t_0$, the computational domain S will fall entirely into the quasi-lacuna of the solution to (1)-(2). Hence, we need to compute the solution only for the finite time T_{int} of (8). As such, we can place the artificial outer boundary far enough so that the reflected wave will not come back to the domain of interest S before we stop the computations. Specifically, the outer boundary should be placed at a distance not closer than $cT_{\text{int}}/2$ from ∂S . In doing so, we completely eliminate the effect of the artificial outer boundary for equations (1)-(2) considered on the entire $\mathbb{R}^3$ (but integrated over the time interval T_{int}). As an alternative to placing the outer boundary at a distance from ∂S , we can set an ABC or PML right on ∂S provided that it will be able to maintain the desired non-reflecting properties for at least as long as the time interval T_{int} of (8).

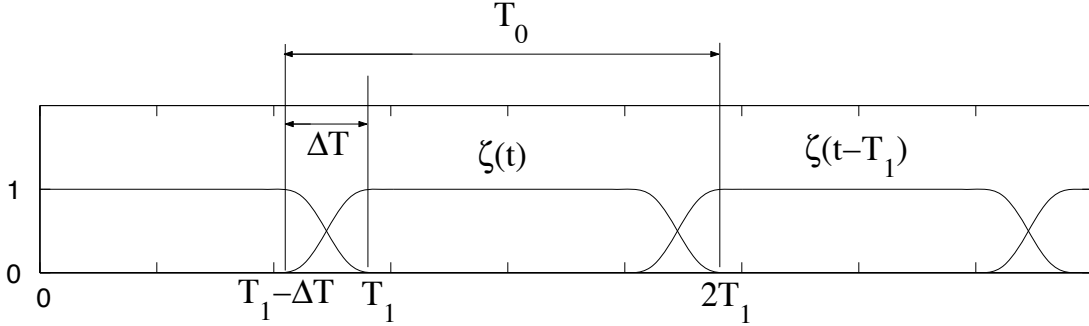


Figure 2: Temporal partition of unity.

Next, we turn to a more general case where the sources $\mathbf{j}$ and $\mathbf{j}^M$ operate continuously in time while remaining compactly supported in space (on S). We introduce a smooth partition of unity (see Figure 2):

$$\forall t \geq 0 : \quad 1 = \sum_{i=0}^{\infty} \zeta(t - (i-1)T_1), \quad \text{supp } \zeta(t) = [T_1 - \Delta T, 2T_1], \quad (9)$$

where T_0 is the size of each partition element, ΔT is the width of the overlapping interval, and $T_1 = T_0 - \Delta T$. By $\zeta(t)$ we denote a smooth compactly supported function such that $\zeta(t) = 1$ for $T_1 \leq t \leq 2T_1 - \Delta T$ and $\zeta(t) + \zeta(t + T_1) = 1$ for $T_1 - \Delta T \leq t \leq T_1$. All of the partition elements can be obtained from one another by a simple translation $t \mapsto t + (i-1)T_1$, $i = 0, 1, \dots$

Multiplying the current densities $\mathbf{j}$ and $\mathbf{j}^M$ by the right-hand side of (9), we transform these continuously operating currents into the sums of partial sources that operate only during a fixed finite time interval each:

$$\begin{aligned}\mathbf{j}(\mathbf{x}, t) &= \sum_{i=0}^{\infty} \mathbf{j}_i(\mathbf{x}, t), \quad \mathbf{j}_i(\mathbf{x}, t) = \mathbf{j}(\mathbf{x}, t) \zeta(t - (i-1)T_1), \\ \mathbf{j}^M(\mathbf{x}, t) &= \sum_{i=0}^{\infty} \mathbf{j}_i^M(\mathbf{x}, t), \quad \mathbf{j}_i^M(\mathbf{x}, t) = \mathbf{j}^M(\mathbf{x}, t) \zeta(t - (i-1)T_1).\end{aligned}\tag{10}$$

An individual i -th partial current is turned on at the moment

$$t_0^{(i)} = \begin{cases} 0, & i = 0, \\ iT_1 - \Delta T, & i = 1, 2, \dots \end{cases}\tag{11}$$

and is turned off at $t_1^{(i)} = (i+1)T_1$, thus operating only for T_0 “seconds.”

By linear superposition, the original system (1)-(2) can now be decomposed into a series of partial subsystems (indexed by $i = 0, 1, 2, \dots$) driven by the partial currents (10). The solutions $\mathbf{E}_i(\mathbf{x}, t)$ and $\mathbf{H}_i(\mathbf{x}, t)$ of these partial subsystems are turned on at the moments $t_0^{(i)}$ since their respective driving sources are zero prior to that. Those solutions have quasi-lacunae, i.e., they reach their (non-zero) steady-state values on S at $t_2^{(i)} = t_0^{(i)} + T_{\text{int}}$, where T_{int} is given by (8).

At every moment of time the overall solution to (1)-(2) can be represented as

$$\mathbf{E}(\mathbf{x}, t) = \sum_{i=0}^{N_0-1} \mathbf{E}_i^{\text{st}}(\mathbf{x}, t) + \sum_{i=N_0}^{N_1} \mathbf{E}_i(\mathbf{x}, t)\tag{12}$$

$$\mathbf{H}(\mathbf{x}, t) = \sum_{i=0}^{N_0-1} \mathbf{H}_i^{\text{st}}(\mathbf{x}, t) + \sum_{i=N_0}^{N_1} \mathbf{H}_i(\mathbf{x}, t).\tag{13}$$

The upper limit $N_1 = \lceil t/T_0 \rceil - 1$ in the sums (12)-(13) is finite because the solutions to the “future” problems, i.e., the problems whose sources have not started to operate yet, are zero. Here by $\lceil \cdot \rceil$ we denote the ceiling function, i.e., a function that yields the nearest integer greater or equal to its argument. In addition, the fields (12)-(13) are split into two sums according to the nature of the terms in these sums. The terms in the first sum in (12)-(13) represent the residual steady-state fields of the “mature” problems, i.e., the problems whose quasi-lacuna has already encompassed the entire domain of interest S . The second sum in (12)-(13) contains the non-stationary partial solutions to the relatively “fresh” subproblems whose quasi-lacuna has not covered the entire S yet. The first such problem is $N_0 = \left\lceil \frac{t - \text{diam } S/c}{T_0} \right\rceil$. It is also important that the number of terms in this second sum is finite and bounded for any t , namely $N_1 - N_0 + 1 \leq \left\lceil \frac{\text{diam } S/c}{T_0} \right\rceil + 2$.

2.1.3 Non-Huygens’ problems and decomposition technique. Our ultimate goal is to build an algorithm that would stabilize the existing ABCs/PMLs in long time simulations for a wide class of electromagnetic problems of almost arbitrary nature. We will refer to these problems as non-Huygens’ emphasizing the fact that they may have neither classical lacunae nor quasi-lacunae.

Let us consider the following problem on the entire $\mathbb{R}^3$:

$$\hat{\mathbf{F}}\left(\mathbf{x}, t, \mathbf{w}, \frac{\partial \mathbf{w}}{\partial t}, \frac{\partial \mathbf{w}}{\partial \mathbf{x}}, \dots\right) = \mathbf{0}, \quad \mathbf{x} \in S_0, \quad t > 0,\tag{14}$$

$$\frac{1}{c} \frac{\partial \mathbf{w}}{\partial t} + \hat{\mathbf{L}} \mathbf{w} = \mathbf{0}, \quad \mathbf{x} \in \mathbb{R}^3 \setminus S_0, \quad t > 0, \quad (15)$$

$$\mathbf{w}(\mathbf{x}, 0) = \mathbf{0}, \quad \mathbf{x} \in \mathbb{R}^3 \setminus S_0. \quad (16)$$

Here $\mathbf{w}(\mathbf{x}, t) = [\mathbf{E}, \mathbf{H}]^T$ is a vector containing the field components and

$$\hat{\mathbf{L}} = \begin{bmatrix} 0 & -\text{curl} \\ \text{curl} & 0 \end{bmatrix} \quad (17)$$

is the symbolic differential operator.

In formula (14), by $\hat{\mathbf{F}}(\dots)$ we denote an abstract operator defining the equations that govern the complex electromagnetic phenomena of quite general nature (propagation, scattering, absorption, interaction with non-linear media, etc.) on the domain of interest S_0 . We do not impose any restrictions on these equations except the following two: (i) we assume that equations (14) smoothly transition (no discontinuities in the coefficients, etc.) into the Maxwell system in vacuum outside S_0 given by formula (15) and (ii) the overall problem (14)-(16) is uniquely solvable and well-posed.

Problem (14)-(16) is formulated on $\mathbb{R}^3$. In practice, however, we have to solve this problem numerically on a bounded domain. A natural choice for this domain would be S_0 itself where the most interesting physics takes place. However, for the reasons that will become clear later, we surround S_0 with a thin layer of width ε . In other words, we choose $S = S_0 \cup S_\varepsilon$ where $S_\varepsilon = \{\mathbf{x} \in \mathbb{R}^3 \setminus S_0, \text{dist}(\mathbf{x}, \partial S_0) < \varepsilon\}$ as our computational domain, see Figure 1. As a boundary condition on ∂S , we can choose an available ABC or PML. The requirement is that should perform well (i.e., guarantee good transparency/absorption/stability properties) at least for moderate times that can still be much smaller than the desired duration of simulation.

As problem (14) on S_0 is almost arbitrary and, generally speaking, non-Huygens, we cannot use directly the quasi-lacunae approach from Section 2.1.2 which is valid only for the Maxwell system in vacuum. Therefore we decompose the overall problem (14)-(16) into the interior problem (IP) and auxiliary problem (AP). The interior problem has the form

$$\begin{aligned} \hat{\mathbf{F}}\left(\mathbf{x}, t, \mathbf{w}, \frac{\partial \mathbf{w}}{\partial t}, \frac{\partial \mathbf{w}}{\partial \mathbf{x}}, \dots\right) &= \mathbf{0}, \quad \mathbf{x} \in S_0, \quad t > 0, \\ \frac{1}{c} \frac{\partial \mathbf{w}}{\partial t} + \hat{\mathbf{L}} \mathbf{w} &= \mathbf{0}, \quad \mathbf{x} \in S_\varepsilon, \quad t > 0, \\ \mathbf{w}(\mathbf{x}, t) &= \mathbf{w}^{\text{aux}}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \quad t > 0. \end{aligned} \quad (18)$$

where $\Gamma = \partial S$ is the boundary of the computational domain. Problem (18) includes equations (14) on S_0 that smoothly transition to the pure Maxwell's equations on S_ε . At the outer boundary Γ , the IP (18) is terminated by the boundary conditions provided by the auxiliary problem.

The auxiliary problem is the pure Maxwell system considered on the entire space

$$\begin{aligned} \frac{1}{c} \frac{\partial \mathbf{w}^{\text{aux}}}{\partial t} + \hat{\mathbf{L}} \mathbf{w}^{\text{aux}} &= \mathbf{f}^{\text{aux}}(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^3, \quad t > 0, \\ \mathbf{w}^{\text{aux}}(\mathbf{x}, 0) &= \mathbf{0}, \quad \mathbf{x} \in \mathbb{R}^3, \end{aligned} \quad (19)$$

and driven by the auxiliary currents $\mathbf{f}^{\text{aux}}(\mathbf{x}, t) = -\frac{4\pi}{c} [\mathbf{j}, \mathbf{j}^{\text{M}}]^T$ of a special kind. Those currents are constructed as follows.

First, we introduce a smooth function $\mu(\mathbf{x})$, such that $\mu(\mathbf{x}) = 1$ for $\mathbf{x} \in \mathbb{R}^3 \setminus S$, i.e., outside the computational domain, and $\mu(\mathbf{x}) = 0$ for $\mathbf{x} \in S_0$, i.e., inside S_0 . In the near-boundary layer S_ε ,

this function increases smoothly from zero to one. Qualitatively, the profile of this function in the direction of its growth resembles a smoothed step (Heaviside) function. For this reason, we will sometimes refer to this function as to a “smoothed step function.”

Next, suppose we know the solution $\mathbf{w}(\mathbf{x}, t) = [\mathbf{E}, \mathbf{H}]^T$ to problem (14)-(16) on the entire space and for any $t \geq 0$. We multiply $\mathbf{w}(\mathbf{x}, t)$ by the scalar $\mu(\mathbf{x})$ and apply the operator of the pure Maxwell system to the resulting product on $\mathbb{R}^3 \times [0, +\infty)$:

$$\mathbf{f}^{\text{aux}}(\mathbf{x}, t) \stackrel{\text{def}}{=} \frac{1}{c} \frac{\partial \mu \mathbf{w}}{\partial t} + \hat{\mathbf{L}} \mu \mathbf{w}, \quad (20)$$

or, in the component form,

$$\mathbf{j} = -\frac{c}{4\pi} \left[\frac{1}{c} \frac{\partial \mu \mathbf{E}}{\partial t} - \text{curl} \mu \mathbf{H} \right], \quad (21)$$

$$\mathbf{j}^M = -\frac{c}{4\pi} \left[\frac{1}{c} \frac{\partial \mu \mathbf{H}}{\partial t} + \text{curl} \mu \mathbf{E} \right]. \quad (22)$$

Clearly, the quantities (20)-(22) are identically equal to zero on S_0 because $\mu(\mathbf{x}) = 0$ there. On the other hand, those quantities also turn into zero outside the computational domain, i.e., on $\mathbb{R}^3 \setminus S$, because the function $\mu(\mathbf{x})$ is equal to unity there and, according to (15), the solution $\mathbf{w}(\mathbf{x}, t) = [\mathbf{E}, \mathbf{H}]^T$ satisfies the pure Maxwell system already outside S_0 and, hence, outside S . Consequently, we conclude that: (i) the currents built this way differ from zero only in the near-boundary layer S_ε and, more importantly, (ii) to construct these currents, we need to know the solution $\mathbf{w}(\mathbf{x}, t)$ to the original problem (14)-(16) only on S_ε , i.e., near the boundary of the computational domain. In practice, it means that in order to generate the sources for the AP we can use the solution to the interior problem (18) rather than to the overall one (14)-(16).

Furthermore, substituting the source function (20) into the AP (19) one obviously obtains that $\mathbf{w}^{\text{aux}} \equiv \mu \mathbf{w}$ since the pure Maxwell system is uniquely solvable. Therefore, at the boundary of the computational domain Γ where $\mu = 1$ we have $\mathbf{w}^{\text{aux}} = \mathbf{w}$ and consequently, the solution to the AP on Γ can indeed provide the boundary data for the IP, see (18).

Hence, the IP (18) and the AP (19) are intertwined with each other as follows: the solution to the IP (which is, of course, of primary interest on its own) is also used for constructing the currents for the AP. The latter, in turn, provides the boundary closure for the IP. Note that the AP is still posed on the entire $\mathbb{R}^3$ and in practice it has to be truncated by means of an ABC/PML.

A key advantage of having the original problem (14)-(16) decomposed into the IP and the AP is that the AP is of the Huygens' type, i.e., its solution possesses quasi-lacunae. Therefore, we can employ the quasi-lacunae based time-marching technique to stabilize the behavior of the chosen ABC/PML in long runs.² This algorithm was presented in [65], and its performance has been corroborated by numerical experiments. Hereafter, we further develop this approach and show that due to a specific structure of the currents (21)-(22) one can construct a time-marching algorithm for the AP based on classical lacunae rather than quasi-lacunae. This, in turn, allows us to prove a temporally uniform error estimate for the non-Huygens' case, see Section 2.1.5.

Suppose that we know the solution $\mathbf{w}(\mathbf{x}, t)$ to problem (14)-(16) on the layer S_ε at any $t \geq 0$. As we have shown, this is sufficient for constructing the currents (21)-(22) that drive the AP. Then, we use the partition of unity and multiply (21)-(22) by the right-hand side of (9). This yields the individual terms in the sum (10):

$$\mathbf{j}_i(\mathbf{x}, t) = -\frac{c}{4\pi} \eta_i(t) \left[\frac{1}{c} \frac{\partial \mu \mathbf{E}}{\partial t} - \text{curl} \mu \mathbf{H} \right], \quad (23)$$

²Alternatively, lacunae-based time marching can also be used as a standalone exact ABC.

$$\mathbf{j}_i^M(\mathbf{x}, t) = -\frac{c}{4\pi}\eta_i(t) \left[\frac{1}{c} \frac{\partial \mu \mathbf{H}}{\partial t} + \text{curl } \mu \mathbf{E} \right], \quad (24)$$

where following (9) we have introduced the notation (see also Figure 2):

$$\eta_i(t) = \zeta(t - (i - 1)T_1). \quad (25)$$

Now let us calculate the divergence of the partial currents (23)-(24). Since $\text{div curl} \equiv 0$ and $\eta_i(t)$ depends only on t we have:

$$\text{div } \mathbf{j}_i(\mathbf{x}, t) = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \text{div } \mu \mathbf{E} = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \{ \mu \text{div } \mathbf{E} + (\nabla \mu, \mathbf{E}) \}, \quad (26)$$

$$\text{div } \mathbf{j}_i^M(\mathbf{x}, t) = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \text{div } \mu \mathbf{H} = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \{ \mu \text{div } \mathbf{H} + (\nabla \mu, \mathbf{H}) \}, \quad (27)$$

where we have taken into account that $\text{div } c\mathbf{a} = c \text{div } \mathbf{a} + (\nabla c, \mathbf{a})$ for a scalar c and a vector $\mathbf{a}$.

Recall that $\mathbf{E}$ and $\mathbf{H}$ in (26)-(27) solve the homogeneous Maxwell's equations with the currents vanishing on $\mathbb{R}^3 \setminus S_0$. Since the domain $\mathbb{R}^3 \setminus S_0$ is free of currents, the electric and magnetic charges do not accumulate there either staying equal to their initial values (zero) for any $t \geq 0$, according to the continuity equations (3) with zero initial conditions. In turn, the Gauss laws (5) imply that

$$\text{div } \mathbf{E} = \text{div } \mathbf{H} = 0 \quad \text{on } \mathbb{R}^3 \setminus S_0 \quad (28)$$

for any instance of time. This is not the case for the domain S_0 itself where the governing equations (14) may be complex and in general $\text{div } \mathbf{E} \neq 0$ and $\text{div } \mathbf{H} \neq 0$ there. However, on this domain the factor $\mu \equiv 0$ by design. Altogether, we conclude that the terms $\mu \text{div } \mathbf{E}$ and $\mu \text{div } \mathbf{H}$ in (26) vanish everywhere on $\mathbb{R}^3$. Note also that if $\mathbf{H}$ is the actual physical magnetic field, then $\text{div } \mathbf{H} \equiv 0$.

Consequently, the divergence of the currents (23)-(24) takes the form:

$$\text{div } \mathbf{j}_i(\mathbf{x}, t) = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \text{div } \mu \mathbf{E} = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} (\nabla \mu, \mathbf{E}), \quad (29)$$

$$\text{div } \mathbf{j}_i^M(\mathbf{x}, t) = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \text{div } \mu \mathbf{H} = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} (\nabla \mu, \mathbf{H}), \quad (30)$$

and differs from zero only in the near-boundary layer S_ε , where the gradient of the smoothed step function μ is non-zero. Hence, the currents (23)-(24) are, generally speaking, not solenoidal, and according to the continuity equations (3) the electric and magnetic charges may accumulate on S_ε . For each partial current (23)-(24), this accumulation goes on for as long as the current operates, i.e., for T_0 “seconds,” and results in a non-zero charge distribution on S_ε . We emphasize that we are talking about the currents and charges of the AP — a formal auxiliary construct that does not necessarily represent a physical reality. Therefore, the words “currents” and “charges” in the context of the AP should not be understood literally, as physical quantities (especially, the non-existing in nature magnetic currents). Those words rather represent a conventional terminology to describe the sources for Maxwell's system (19). Note also that even if $\mathbf{H}$ is a solenoidal physical field, i.e., $\text{div } \mathbf{H} = 0$, the magnetic charge for the AP may still accumulate according to (30).

As shown in our work [64], the accumulated charges generate a non-zero steady-state contribution to the solution of every partial subproblem inside its quasi-lacuna. Those steady-state components cannot be dropped at later times; they rather need to be kept for the entire duration

of computation because they are used in the sums (12)-(13) for reconstructing the overall solution to the AP (19). The latter, in turn, provides a boundary closure for the IP (18).

This strategy, however, may introduce an additional error for the practical implementation of the algorithm. The difficulty is that the accumulating charge is not given by any analytic expression. At every time step the currents (23)-(24) are rather constructed from the solution to the IP (18), which is not known ahead of time. Then, the steady-state components of the the solution to each partial subproblem can be computed using either (i) the Poisson equation with the charges obtained numerically by means of the continuity equations (3) or (ii) steady-state numerical solution by means of the same scheme taken at a late time when the quasi-lacuna is already covering the entire computational domain.

However, neither of these approaches guarantees that the error associated with the numerical calculation of the increasing number of steady-state contributions will not accumulate over time. Indeed, each of those steady-state contributions is evaluated approximately by the nature of its numerical computation. Then, we add them together and use for subsequent moments of time as a part of the boundary data for the IP. The number of these approximate steady-state contributions increases over time, which makes it difficult to obtain a temporally uniform error bound.

2.1.4 Making quasi-lacunae classical. In this section we show that the partial currents (23)-(24) can be redefined in such a way that the charge accumulated during the operation period of $\mathbf{j}_i$, $\mathbf{j}_i^M$ will vanish at every point of the computational domain:

$$\rho_i = - \int_{iT_1-\Delta T}^{(1+i)T_1} \operatorname{div} \mathbf{j}_i dt = 0, \quad \rho_i^M = - \int_{iT_1-\Delta T}^{(1+i)T_1} \operatorname{div} \mathbf{j}_i^M dt = 0. \quad (31)$$

At the same time, the overall solution to the AP obtained by summing up the solutions to individual subproblems will remain unaffected as if it were driven by the original currents (23)-(24). This means that the quasi-lacunae of the partial subproblems turn into classical lacunae, and the steady-state components in (12)-(13) can be dropped. The latter circumstance, in turn, paves the way to the rigorous justification of the algorithm as shown in Section 2.1.5.

We emphasize that the redefined partial currents are not required to be solenoidal on $\mathbb{R}^3$. A theoretical possibility of constructing the solenoidal currents for the AP was pointed out in [65], but its practical implementation faces some unresolved technical issues and computational overheads.

To eliminate the accumulation of charges, we modify the partial currents as follows:

$$\dot{\mathbf{j}}_i(\mathbf{x}, t) = -\frac{c}{4\pi} \eta_i(t) \left[\frac{1}{c} \frac{\partial \mu \mathbf{E}}{\partial t} - \operatorname{curl} \mu \mathbf{H} \right] + (1 - \mu) \dot{\eta}_i(t) \mathbf{E}', \quad (32)$$

$$\dot{\mathbf{j}}_i^M(\mathbf{x}, t) = -\frac{c}{4\pi} \eta_i(t) \left[\frac{1}{c} \frac{\partial \mu \mathbf{H}}{\partial t} + \operatorname{curl} \mu \mathbf{E} \right] + (1 - \mu) \dot{\eta}_i(t) \mathbf{H}', \quad (33)$$

that is, we add the extra terms

$$(1 - \mu) \dot{\eta}_i(t) \mathbf{E}', \quad (34)$$

$$(1 - \mu) \dot{\eta}_i(t) \mathbf{H}', \quad (35)$$

to the original expressions (23)-(24). The dot in (34)-(35) denotes differentiation with respect to time, and the auxiliary fields $\mathbf{E}'$ and $\mathbf{H}'$ are defined as follows:

$$\mathbf{E}' = \begin{cases} \mathbf{E}, & x \in S_\varepsilon, \\ \mathcal{E}, & x \in S_0, \operatorname{div} \mathcal{E} = 0, \end{cases} \quad \mathbf{H}' = \begin{cases} \mathbf{H}, & x \in S_\varepsilon, \\ \mathcal{H}, & x \in S_0, \operatorname{div} \mathcal{H} = 0. \end{cases} \quad (36)$$

In other words, the fields $\mathbf{E}'$ and $\mathbf{H}'$ are specified on the computational domain S , and coincide with the fields $\mathbf{E}$ and $\mathbf{H}$ of the interior problem (18) in the near-boundary layer S_ε . On the bulk of the computational domain, i.e., on S_0 , they are obtained by a divergence-free extension: $\text{div}\mathcal{E} = \text{div}\mathcal{H} = 0$ on S_0 for any t . Due to this composite structure, we will refer to the fields (36) as to the composite fields. The divergence of the composite fields is zero both on S_0 and S_ε :

$$\text{div}\mathbf{E}' = \text{div}\mathbf{H}' = 0, \quad \mathbf{x} \in S_0 \quad \text{or} \quad \mathbf{x} \in S_\varepsilon, \quad (37)$$

because $\text{div}\mathbf{E} = \text{div}\mathbf{H} = 0$ on S_ε due to (28) and $\text{div}\mathcal{E} = \text{div}\mathcal{H} = 0$ on S_0 by design. We postpone the discussion of a subtle question to defining and treat the divergence exactly on the boundary ∂S_0 until we have introduced the matching conditions for the fields on ∂S_0 and pointed out the way of actually constructing the extensions $\mathcal{E}$, $\mathcal{H}$.

In the meantime, assuming that the fields (36) are available, let us analyze the properties of the redefined sources (32)-(33):

- (a) The extra terms (34)-(35) introduced into the partial currents do not affect the overall solution of the AP. Indeed, the additional terms (34)-(35) differ from zero only on the time intervals where two consecutive hat-shaped functions $\eta_i(t)$ and $\eta_{i+1}(t)$ overlap and where the time derivative of these functions does not vanish (see Figure 2). Moreover, on such intervals these consecutive hat functions are related to each other by $\eta_{i+1}(t) = 1 - \eta_i(t)$, so for their time derivatives we have: $\dot{\eta}_{i+1}(t) = -\dot{\eta}_i(t)$. Therefore, the extra terms (34)-(35) in two consecutive partial currents $\mathbf{j}_i$, $\mathbf{j}_i^M$ and $\mathbf{j}_{i+1}$, $\mathbf{j}_{i+1}^M$ have equal absolute values but differ in sign. Due to the linearity, these extra terms generate the additional fields (compared to the “unperturbed” currents (23)-(24)) that cancel out one another in the sums (12)-(13).
- (b) The charge accumulated for the operational period of each partial current (32)-(33) is equal to zero. To prove it, consider the divergence of, say, the magnetic current (33):

$$\text{div}\mathbf{j}_i^M(\mathbf{x}, t) = -\frac{\eta_i(t)}{4\pi} \frac{\partial}{\partial t} \text{div}\mu\mathbf{H} + \dot{\eta}_i(t) \text{div}(1 - \mu)\mathbf{H}' = \dot{\eta}_i(t) \text{div}\mathbf{H}' - \frac{\partial}{\partial t} \{ \eta_i(t) \text{div}\mu\mathbf{H}' \}. \quad (38)$$

In (38), we again used the identity $\text{div}\text{curl} \equiv 0$ and isolated the complete time derivative at the last step. In doing so, we have used the fact that $\mu\mathbf{H} = \mu\mathbf{H}'$ everywhere on the computational domain S : by design on S_ε (see (36)) and due to $\mu = 0$ on S_0 . Moreover, $\text{div}\mathbf{H}' = 0$ due to (37) and thus the divergence (38) takes the form of the complete time derivative. Integrating (38) over the support of the hat function $\eta_i(t)$ at any spatial point on S , we obtain:

$$\rho_i^M = - \int_{iT_1 - \Delta T}^{(1+i)T_1} \text{div}\mathbf{j}_i^M dt = \eta_i(t) \text{div}\mu\mathbf{H}' \Big|_{iT_1 - \Delta T}^{(1+i)T} = 0, \quad (39)$$

because $\eta_i(t)$ vanishes at the endpoints $t = (1 + i)T$ and $t = iT_1 - \Delta t$. The same applies to the electric current (32).

- (c) The redefined currents (32)-(33) are no longer localized inside the near-boundary layer S_ε (as opposed to the original currents (23)-(24)) because the additional terms (34)-(35) do not, generally speaking, vanish on S_0 . Indeed, the extensions $\mathcal{E}$, $\mathcal{H}$ are basically non-zero on S_0 and besides, $1 - \mu = 1$ there. Still, in order to construct the currents (32)-(33) at any moment of time, one needs to know the values of the fields $\mathbf{E}$, $\mathbf{H}$ of the interior problem only on S_ε and extend them in a certain way inside S_0 to obtain the fields $\mathcal{E}$, $\mathcal{H}$ in (36).

- (d) While the currents (32)-(33) are, generally speaking, non-zero on the computational domain S , they turn into zero outside S (including its boundary Γ) because the original currents (23)-(24) do so and the extra terms (34)-(35) vanish on $\mathbb{R}^3 \setminus S$ as $1 - \mu = 0$ there.

Let us now explain how to construct the extensions $\mathcal{E}$, $\mathcal{H}$ in practice. Given the values of $\mathbf{E}$ and $\mathbf{H}$ on the boundary ∂S_0 for any t , we seek the extensions $\mathcal{E}$, $\mathcal{H}$ in the form of anti-gradients of the respective potentials that solve the Neumann problem for the Laplace equation on S_0 :

$$\begin{cases} \Delta\varphi = 0, \\ \left. \frac{\partial\varphi}{\partial n} \right|_{\partial S_0} = -E_n, \end{cases} \quad \begin{cases} \Delta\varphi^M = 0, \\ \left. \frac{\partial\varphi^M}{\partial n} \right|_{\partial S_0} = -H_n, \end{cases} \quad \mathbf{x} \in S_0, \quad (40)$$

$$\mathcal{E} = -\nabla\varphi, \quad \mathcal{H} = -\nabla\varphi^M. \quad (41)$$

In (40)-(41), φ , φ^M are the unknown potentials, $\frac{\partial}{\partial n}$ is the normal derivative on ∂S_0 , and E_n , H_n are the normal components of the fields $\mathbf{E}$, $\mathbf{H}$ of the interior problem (18) on ∂S_0 .

Clearly, the fields $\mathcal{E}$, $\mathcal{H}$ constructed according to (40)-(41) are solenoidal because

$$\begin{aligned} \operatorname{div}\mathcal{E} &= -\operatorname{div}\nabla\varphi \equiv -\Delta\varphi = 0, \\ \operatorname{div}\mathcal{H} &= -\operatorname{div}\nabla\varphi^M \equiv -\Delta\varphi^M = 0. \end{aligned} \quad (42)$$

Relations (42) hold everywhere on S_0 up to the boundary.

The boundary conditions in (40) are set for the normal derivative of the potential rather than for the potential itself because it is the fields $\mathbf{E}$, $\mathbf{H}$ that are assumed known on ∂S_0 . These conditions guarantee the continuity of the normal components of the composite fields (36) on the interface ∂S_0 but say nothing about their tangential components. In general, the tangential components will be discontinuous on ∂S_0 because specifying both the normal and tangential component of the field on ∂S_0 will make the corresponding elliptic problem for the potential on S_0 overdetermined.³

The discontinuity of the tangential components though does not lead to discontinuity of the divergence of the composite fields $\mathbf{E}'$ and $\mathbf{H}'$ on the boundary ∂S_0 , or, more precisely, does not make the divergence singular. Indeed, when calculating the divergence, normal components of the fields are differentiated along the normal direction whereas the tangential components are differentiated along the tangential directions. Therefore, even as the tangential components undergo jumps at ∂S_0 , all differentiations can be carried through independently on different sides of the interface ∂S_0 , and the resulting quantities will be bounded. Hence, the divergence of $\mathbf{E}'$ or $\mathbf{H}'$ can formally be considered a dual-valued function on ∂S_0 . As, however, both values coincide, see (37), we can define this divergence on ∂S_0 merely by continuity, and consider $\operatorname{div}\mathbf{E}' = \operatorname{div}\mathbf{H}' = 0$ for all $\mathbf{x} \in S$.

Each of the interior Neumann problems for the Laplace equation, see (40), is known to have a solution only if the corresponding compatibility condition holds for the boundary data [52]:

$$\Phi = \int_{\partial S_0} E_n dS = 0, \quad \Phi^M = \int_{\partial S_0} H_n dS = 0. \quad (43)$$

In other words, problems (40) are solvable only if the fluxes of the fields $\mathbf{E}$ and $\mathbf{H}$ through the surface ∂S_0 are equal to zero. The solutions to (40) are unique but, obviously, up to an additive

³Unless the data E_n or H_n already pertain to an electrostatic or magnetostatic field. We, however, do not expect that in general, because the fields $\mathbf{E}$ and $\mathbf{H}$ whose normal components are used as the boundary data in (40) represent a solution to the unsteady interior problem (18) and may be quite complex, e.g., may not even have a potential.

constant. The extensions $\mathcal{E}$ and $\mathcal{H}$ are unique in the strict sense because the differentiation of φ , φ^M in (41) eliminates that arbitrary constant. Due to the constraint (43), the solenoidal (divergence-free) extension that we have described does not exist for any given interior problem (18). Hereafter, we discuss some specific cases where it can still be done.

To begin with, we emphasize that conditions (43) are necessary for the solenoidal extensions of the given vector fields $\mathbf{E}$ and $\mathbf{H}$ onto the domain S_0 to exist in the form (40)-(41). In the context of our discussion, $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic field of a potentially sophisticated interior problem (18). Those fields may display some complex behavior, and may, in particular, not even be defined everywhere on S_0 if, e.g., there are impenetrable scattering objects inside the computational domain. Therefore, equalities (43), which can be checked if the fields are known only outside S_0 , is all that tells us about the possibility of building the desired extensions $\mathcal{E}$, $\mathcal{H}$. If, however, the interior problem is made a little more specific, then it appears possible to identify some narrower yet perfectly meaningful classes of formulations to which conditions (43) will apply.

First of all, as the extension will be needed at every time step of the algorithm, we supply the argument t to all the quantities involved, e.g., $\mathbf{E}(t)$ and $\mathbf{H}(t)$. Next, keeping in mind that $\mathbf{E}(t)$ and $\mathbf{H}(t)$ are the electric and magnetic fields, we recast conditions (43) in terms of the total charge of the system:

$$0 = \Phi(t) = \int_{\partial S_0} E_n(t) dS = \int_{S_0} \operatorname{div} \mathbf{E}(t) d\mathbf{x} = 4\pi \int_{S_0} \rho(t) d\mathbf{x} \equiv 4\pi Q(t), \quad (44)$$

$$0 = \Phi^M(t) = \int_{\partial S_0} H_n(t) dS = \int_{S_0} \operatorname{div} \mathbf{H}(t) d\mathbf{x} = 4\pi \int_{S_0} \rho^M(t) d\mathbf{x} \equiv 4\pi Q^M(t), \quad (45)$$

where we have used the Gauss theorem applied to the domain S_0 and the Gauss laws (5). Hence, the possibility to obtain a solenoidal extension at any given moment of time depends on whether or not the total electric charge $Q(t)$ or magnetic charge $Q^M(t)$ is zero.

Furthermore, recall that we are considering only “isolated” systems of type (18) surrounded by vacuum, see (15). Hence, there can be no transport of charge (i.e., no current) from the exterior of S_0 . Therefore, the total charge is conserved, i.e., $Q(t) = Q(0) = \text{const}$, $Q^M(t) = Q^M(0) = \text{const}$.

Let us now make the IP more specific by identifying it with a lossy Maxwell’s system driven by the extraneous current $\mathbf{j}$ and having the conductivity σ :

$$\operatorname{curl} \mathbf{H} = \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} + \frac{4\pi}{c} (\mathbf{j} + \sigma \mathbf{E}), \quad (46)$$

$$\operatorname{curl} \mathbf{E} = -\frac{\partial \mathbf{H}}{\partial t}. \quad (47)$$

Both $\mathbf{j}$ and σ in the Ampère law (46) are almost arbitrary except that they are required to turn into zero outside S_0 . Taking the divergence of (46) and (47) and integrating over S_0 we obtain

$$\frac{\partial}{\partial t} \int_{\partial S_0} E_n dS = -\frac{4\pi}{c} \int_{\partial S_0} (j_n + \sigma E_n) dS, \quad (48)$$

$$\frac{\partial}{\partial t} \int_{\partial S_0} H_n dS = 0, \quad (49)$$

where we have also used the Gauss theorem. Since under our assumption the current $\mathbf{j}$ and conductivity σ vanish on ∂S_0 , we have from (48):

$$\int_{\partial S_0} E_n(t) dS = \text{const} \equiv 4\pi Q(0). \quad (50)$$

Similarly, taking into account that there is no magnetic charge in a real life physical setting, $\rho^M = 0$, we obtain from (49):

$$\int_{\partial S_0} H_n(t) dS = 0. \quad (51)$$

Comparing (50) to the first condition (43), we conclude that for a realistic IP of the type (46)-(47) the possibility of building a solenoidal extension of the electric field $\mathbf{E}$ onto S_0 in the form (40)-(41) is time invariant and is determined only by whether the conserved total electric charge is zero. At the same time, the magnetic field can always be extended onto S_0 in a divergence-free way due to (51). Indeed, as there are no physical magnetic charges, the second condition (43) written in the form of (45) will always hold.

Let us point out some special yet practically relevant situations when a divergence-free extension of the electric field can be constructed:

- A wide class of problems with zero initial conditions for $\mathbf{E}$. Clearly, if the total charge were not zero the IP would generate an electrostatic field contrary to the assumption $\mathbf{E}(0) = \mathbf{0}$. Mathematically, integrating $\text{div} \mathbf{E}(0) = 4\pi \rho(0) = 0$ over S_0 we obtain zero total charge.
- On the other hand, the condition $\mathbf{E}(0) = \mathbf{0}$ is sufficient but not necessary for building a divergence-free extension. Indeed, we can consider a system with zero total charge but non-zero higher multipole moments (dipole, quadrupole, etc). Such system still generates the electrostatic field yet as there is no total charge the divergence-free extension is possible.

We conclude this section by summarizing the results on solenoidal extension (40)-(41):

- For a general IP, one can obtain a divergence-free extension of $\mathbf{E}$ or $\mathbf{H}$ onto S_0 only if the flux of the respective field through S_0 is zero at any t .
- For isolated physical IPs, a divergence-free extension of the electric field can be built in the form (40)-(41) provided that the conserved total electric charge is zero.

2.1.5 A temporally uniform error bound. In this section, we prove a temporally uniform estimate for the error which is due to the treatment of the artificial outer boundary in the case where the lacunae-based time marching is applied to a non-Huygens' problem. If the computational domain is terminated by an ABC or a PML, then the lacunae based time marching facilitates its non-deteriorating behavior over arbitrarily long times. Qualitatively, the mechanism of the non-deteriorating performance is very simple. The partial subproblems that compose the AP need to be integrated over a limited interval of time only during which the intrinsic instabilities of the ABC/PML, if any, do not develop.

The proof of the temporally uniform error bound relies on the modified auxiliary currents of Section 2.1.4 or, equivalently, on the use of classical lacunae as opposed to quasi-lacunae. In that regard we note that the time-marching strategy based on quasi-lacunae, see [65], allows to prove

a temporally uniform error estimate only when the currents and charges that drive the field are known ahead of time, which is not the case for a non-Huygens' setting.

We first note that under our approach an ABC/PML is applied to the auxiliary problem (19). Hence, it is sufficient to demonstrate that the error introduced by the ABC/PML into the solution of the AP will stay bounded for all times. This, in turn, implies that the same will be true for the solution of the IP (18), which takes the solution of the AP (19) as the boundary data.

For convenience, we reproduce here the equations of the AP (19):

$$\begin{aligned} \frac{1}{c} \frac{\partial \mathbf{w}^{\text{aux}}}{\partial t} + \hat{\mathbf{L}} \mathbf{w}^{\text{aux}} &= \mathbf{f}^{\text{aux}}(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^3, \quad t > 0, \\ \mathbf{w}^{\text{aux}}(\mathbf{x}, 0) &= \mathbf{0}, \quad \mathbf{x} \in \mathbb{R}^3, \end{aligned} \quad (52)$$

where

$$\mathbf{f}^{\text{aux}}(\mathbf{x}, t) = -\frac{4\pi}{c} [\mathbf{j}, \mathbf{j}^{\text{M}}]^T$$

and

$$\hat{\mathbf{L}} = \begin{bmatrix} 0 & -\text{curl} \\ \text{curl} & 0 \end{bmatrix}.$$

Even though problem (52) is formulated on the entire $\mathbb{R}^3$, its practical implementation requires reduction to some bounded domain, which we have chosen to be S , see Figure 1:

$$\begin{aligned} \frac{1}{c} \frac{\partial \mathbf{w}^{(S)}}{\partial t} + \hat{\mathbf{L}} \mathbf{w}^{(S)} &= \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x} \in S, \quad t > 0, \\ \hat{\mathbf{\Gamma}} \mathbf{w}^{(S)}(\mathbf{x}, t) &= \mathbf{0}, \quad \mathbf{x} \in \Gamma, \quad t > 0, \\ \mathbf{w}^{(S)}(\mathbf{x}, 0) &= \mathbf{0}, \quad \mathbf{x} \in S. \end{aligned} \quad (53)$$

The operator $\hat{\mathbf{\Gamma}}$ in formula (53) models some ABC or PML set on the outer boundary $\Gamma = \partial S$.

For an ideal ABC or PML, the solutions to problems (52) and (53) coincide on S :

$$\mathbf{w}^{\text{aux}}(\mathbf{x}, t) \equiv \mathbf{w}^{(S)}(\mathbf{x}, t), \quad \mathbf{x} \in S, \quad (54)$$

for any moment of time $t > 0$. In practice, however, no ABC or PML is ideal, and instead of equality (54) the following inequality holds:

$$\|\mathbf{w}(\mathbf{x}, t) - \mathbf{w}^{(S)}(\mathbf{x}, t)\| \leq \delta(\hat{\mathbf{\Gamma}}), \quad \mathbf{x} \in S. \quad (55)$$

The constant δ on the RHS of (55) depends on the operator $\hat{\mathbf{\Gamma}}$, i.e., on the quality of the ABC or PML at ∂S . In the ideal case, $\delta = 0$, and then (55) transforms into (54). Otherwise, δ is non-zero; in the case of an ABC it may depend, for example, on the order of the boundary condition (see, e.g., [74, 124, 126]), whereas in the case of a PML it may depend, in particular, on its thickness and the absorption profile. In all cases, however, a non-deteriorating long-time performance implies that δ does not depend on time, or more precisely, remains bounded as the time elapses.

Unfortunately, this is not always the case. To account for the possible adverse behavior, we follow the suggestion of [15] and interpret any long-time error growth as the growth of small perturbations in the data that drive the problem.⁴ Specifically, along with the actual solution $\mathbf{w}^{(S)}$

⁴These perturbations may have arbitrary nature, although in the numerical context it is convenient to associate them with the residual terms of the approximation, i.e., with the truncation error.

that satisfies system (53) on the bounded region S , we consider its perturbed counterpart $\tilde{\mathbf{w}}^{(S)}$ that satisfies the system with the perturbed boundary data, initial data, and the RHS:

$$\begin{aligned} \frac{1}{c} \frac{\partial \tilde{\mathbf{w}}^{(S)}}{\partial t} + \hat{\mathbf{L}} \tilde{\mathbf{w}}^{(S)} &= \mathbf{f}(\mathbf{x}, t) + \boldsymbol{\mu}, \quad \mathbf{x} \in S, \quad t > 0, \\ \hat{\mathbf{\Gamma}} \tilde{\mathbf{w}}^{(S)}(\mathbf{x}, t) &= \boldsymbol{\epsilon}, \quad \mathbf{x} \in \partial S, \quad t > 0, \\ \tilde{\mathbf{w}}^{(S)}(\mathbf{x}, 0) &= \boldsymbol{\xi}, \quad \mathbf{x} \in S. \end{aligned} \quad (56)$$

The difference between the perturbed and unperturbed solutions may grow on the domain S as the time t elapses:

$$\|\tilde{\mathbf{w}}^{(S)}(\mathbf{x}, t) - \mathbf{w}^{(S)}(\mathbf{x}, t)\| \leq \xi(t) \|[\boldsymbol{\mu}, \boldsymbol{\epsilon}, \boldsymbol{\xi}]\|'. \quad (57)$$

The rate of growth $\xi(t)$ is determined by a particular ABC or PML; for example, some well-known PMLs have been reported to demonstrate a linear or quadratic growth [13, 15, 125]. The notation $[\boldsymbol{\mu}, \boldsymbol{\epsilon}, \boldsymbol{\xi}]$ on the RHS of inequality (57) represents a symbolic vector of all perturbations of the data in system (56). The choice of norms $\|\cdot\|$ and $\|\cdot\|'$ for some specific cases is discussed in [13, 15, 125].

We emphasize that inequality (57) is quite general in the sense that it allows the perturbations of the data to grow with a prescribed rate, yet the reason for the growth may be arbitrary and it does not need to be known.

Inequality (57) implies that the perturbations that are small initially may increase with no bound for large integration times. If, however, the plain integration of system (53) is replaced by the integration that uses quasi-lacunae (see Section 2.1.2), then estimate (57) can be drastically improved and, in fact, made *uniform in time*. Indeed, both solutions, $\mathbf{w}^{(S)}(\mathbf{x}, t)$ and $\tilde{\mathbf{w}}^{(S)}(\mathbf{x}, t)$, can be partitioned similarly to how it is done in expressions (12)-(13):

$$\mathbf{w}^{(S)}(\mathbf{x}, t) = \mathbf{w}_{N_0}^{(S), \text{st}}(\mathbf{x}) + \sum_{i=N_0}^{N_1} \mathbf{w}_i^{(S)}(\mathbf{x}, t), \quad (58)$$

$$\tilde{\mathbf{w}}^{(S)}(\mathbf{x}, t) = \tilde{\mathbf{w}}_{N_0}^{(S), \text{st}}(\mathbf{x}) + \sum_{i=N_0}^{N_1} \tilde{\mathbf{w}}_i^{(S)}(\mathbf{x}, t), \quad (59)$$

where

$$\mathbf{w}_{N_0}^{(S), \text{st}} = \sum_{i=0}^{N_0-1} \mathbf{w}_i^{(S), \text{st}}, \quad \tilde{\mathbf{w}}_{N_0}^{(S), \text{st}} = \sum_{i=0}^{N_0-1} \tilde{\mathbf{w}}_i^{(S), \text{st}}, \quad (60)$$

represent the corresponding steady-state contributions, whereas $\mathbf{w}_i^{(S)}(\mathbf{x}, t)$, $\tilde{\mathbf{w}}_i^{(S)}(\mathbf{x}, t)$, $i = N_0, \dots, N_1$, represent the time-dependent terms.

It is very important that the stationary contributions (60) are now equal to zero:

$$\mathbf{w}_{N_0}^{(S), \text{st}}(\mathbf{x}) = \mathbf{0}, \quad \tilde{\mathbf{w}}_{N_0}^{(S), \text{st}}(\mathbf{x}) = \mathbf{0},$$

because the solutions to the partial subproblems governed by the redefined currents (32)-(33) possess classical lacunae with zero residual solution behind the aft fronts.

Since each unsteady term needs to be integrated only for a limited time T_{int} , see formula (8), estimate (57) yields for $i = N_0, \dots, N_1$:

$$\|\tilde{\mathbf{w}}_i^{(S)}(\mathbf{x}, t) - \mathbf{w}_i^{(S)}(\mathbf{x}, t)\| \leq C_0 \|[\boldsymbol{\mu}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\xi}_i]\|', \quad (61)$$

where $C_0 = \xi(T_{\text{int}})$ is a constant, and $[\boldsymbol{\mu}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\xi}_i]$ denotes perturbations of the data for the i -th partial problem. As the overall number of unsteady terms $N_1 - N_0 + 1$ in either sum (58) or (59) does not exceed the value $\left\lceil \frac{\text{diam } S/c}{T_0} \right\rceil + 2$ for all times, we can use the triangle inequality and obtain the following estimate for the functions (58) and (59):

$$\left\| \sum_{i=N_0}^{N_1} \tilde{\mathbf{w}}_i^{(S)}(\mathbf{x}, t) - \sum_{i=N_0}^{N_1} \mathbf{w}_i^{(S)}(\mathbf{x}, t) \right\| \leq C_1 \sup_i \|[\boldsymbol{\mu}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\xi}_i]\|'. \quad (62)$$

In formula (62), $C_1 = C_0 \cdot \max\{N_1 - N_0 + 1\}$, and the supremum on the RHS of the inequality can naturally be assumed bounded.

Finally, combining estimates (55) and (62) with the help of the triangle inequality, we arrive at the following result.

Theorem 1 *Let $S \subset \mathbb{R}^3$ be a bounded computational domain, and let problem (52) be solved on S using an appropriate ABC or PML that reduces it to problem (53). Let problem (53) be time marched with the help of classical lacunae, i.e., using the modified auxiliary sources (32)-(33). Then, assuming that $\sup_i \|[\boldsymbol{\mu}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\xi}_i]\|' < \infty$, the error on the computational domain S will remain uniformly bounded for all times:*

$$\|\tilde{\mathbf{w}}^{(S)}(\mathbf{x}, t) - \mathbf{w}(\mathbf{x}, t)\| \leq \delta(\hat{\Gamma}) + C_1 \sup_i \|[\boldsymbol{\mu}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\xi}_i]\|'. \quad (63)$$

Theorem 1 guarantees that regardless of what causes the original long-time deterioration, time-marching with the help of classical lacunae will prevent any possible error growth.

2.2 Application of lacunae to other types of boundaries

Our most recent theoretical development enables the application of lacunae to other types of boundaries and interfaces, beyond the artificial outer boundaries. In general, the treatment of boundaries and interfaces that are not aligned with the discretization grid (real shapes, as opposed to simple model shapes) may present substantial difficulties for the numerical solution of PDEs and, in particular, for the numerical simulation of waves. It becomes especially challenging when high order accuracy is required, which is often the case in numerical wave propagation because of the pollution effect, see [87, 88] and [89, Section 4.6.1].

The method of difference potentials by Ryaben'kii [101–104] proves to be a very efficient tool for solving this type of problems. This method can be viewed as a discrete analog of Calderon's potentials and Calderon's boundary equations with projections in functional analysis [105, 106]. It is a very general approach that allows one to handle general boundaries not aligned with the grid and still maintain high order accuracy. We have used it for solving a variety of transmission and scattering problems for time-harmonic (i.e., single-frequency) waves, see [59–62].

A key component of the method of difference potentials is the auxiliary problem. It can be “loosely” thought of as an analogue of the convolution with the fundamental solution in the methods based on boundary integral equations. However, it is a substantially more general construct that does not require the explicit knowledge of the fundamental solution (or any Green's function at all), allows for variable coefficients of the governing equation(s), and does not involve any singular integrals or numerical approximation thereof.

Our central idea is to extend the approach based on difference potentials (that enables the geometric universality and high order accuracy) from the frequency domain to the time domain,

and to use the auxiliary problems that satisfy the Huygens' principle. This is possible because the method of difference potentials allows for a substantial flexibility in choosing the auxiliary problem, and it is typically possible to use the one that would be the easiest and most efficient to solve. Using Huygens' auxiliary problems makes the extent of their temporal non-locality fixed and bounded for all times. This carries a great potential for computational efficiency, without having to compromise any of the attractive features, such as the capability of handling the non-conforming boundaries on regular structured grids with no loss of accuracy. The approach based on the combined use of difference potentials and lacunae will be in the main focus of our theoretical analysis in the future.

3 Implementation of the algorithm for computations by the in-house Fortran code

In this section of the report, we present the numerical results that attest to the efficiency and universality of the proposed lacunae-based methodology. The results are obtained using our in-house Fortran code written specifically for demonstrating the performance of the method.

Currently, there are two versions of the in-house code that exist and are used for simulations. The first one is the original code that dates several years back. Its development has started in order to carry out the numerical experiments of [65]. In the course of the current project (both Phase I and Phase II), it was modified to include the more recent advances outlined in Section 2. The code provides a straightforward Fortran implementation of the lacunae-based algorithm. Its advantage is the high computational efficiency. In Section 3.1, we use this first version of the code to show how the lacunae-based time marching can stabilize an unsplit PML (specifically, that of [72]) in the case where the interior problem may be non-Huygens'.

The second version of the in-house code uses modern programming tools, such as classes and dynamic data arrays, as well as OpenMP for parallelization. It has modular structure that implements several major components of the algorithm as independently of the others as possible. This allows for easy modifications within individual parts and hence for a more straightforward testing of a broader range of combined formulations. Specifically, we are interested in testing a wider variety of artificial boundary conditions and perfectly matched layers along with the lacunae-based time marching. The design, implementation, and initial computational results obtained with the help of the second, modular, version of the in-house code are described in Section 3.2.

We emphasize that unlike in [65] or in our Phase I report, all the computations that we have performed with the help of the in-house code were conducted in a full three-dimensional setting, without assuming any additional symmetries.

3.1 Algorithm of Section 2.1 applied to an unsplit PML.

In this section, we describe the implementation of the most recent version of the lacunae-based algorithm presented in Section 2.1. It guarantees the non-deteriorating behavior, i.e., provides a temporally uniform bound for the error due to the treatment of the outer boundary, even in the case where the interior problem may be non-Huygens'. The implementation is rendered by the original (first) version of the in-house Fortran code.

As the interior problem, we have chosen Maxwell's equations in vacuum with the electric currents generating a pure transverse magnetic (TM) mode of electromagnetic field. This means that in the spherical coordinates (r, θ, φ) the only non-zero components of the field are E_r , E_θ and H_φ . The solution is axially symmetric with respect to the z -axis so that none of the functions involved depends on the azimuthal angle φ . This choice was made primarily for the reason of simplicity and

does not imply any loss of generality. Our numerical solver does not rely on any special properties of the solution though, and computes it in a general three-dimensional setting.

In the Cartesian coordinates, the full Maxwell system of equations in vacuum reads:

$$\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \frac{1}{c} \frac{\partial E_x}{\partial t} + \frac{4\pi}{c} j_x, \quad (64)$$

$$-\frac{\partial H_z}{\partial x} + \frac{\partial H_x}{\partial z} = \frac{1}{c} \frac{\partial E_y}{\partial t} + \frac{4\pi}{c} j_y, \quad (65)$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{1}{c} \frac{\partial E_z}{\partial t} + \frac{4\pi}{c} j_z, \quad (66)$$

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{1}{c} \frac{\partial H_x}{\partial t}, \quad (67)$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{1}{c} \frac{\partial H_y}{\partial t}, \quad (68)$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{1}{c} \frac{\partial H_z}{\partial t}. \quad (69)$$

We are solving this system subject to zero initial conditions. In doing so, we do not explicitly take advantage of either the axial symmetry of the solution or the absence of the axial magnetic field: $H_z \equiv 0$. We are rather solving the full three-dimensional system of equations (64)-(69) on the Cartesian grid. While this obviously leads to certain computational overheads, it also allows us to put a clear emphasis on the main objective of the current work, which is to demonstrate in practice the regularization capabilities of the lacunae-based method over long times. For that purpose, we need a specific example of the ABC or PML that would maintain its satisfactory domain truncation properties for short/moderate simulation times but lead to an error build-up at longer runs. It is known that some unsplit PMLs are prone to precisely this type of behavior in full 3D. The long time deterioration in this case may be attributed to the near-edge and near-corner regions of the computational domain in three dimensions. Hence, we define the computational domain S as a cube with side $2a$ centered at the origin:

$$S = \{(x, y, z) : -a \leq x, y, z \leq a\}. \quad (70)$$

3.1.1 Test solution. As has been mentioned, our test solution is a propagating TM mode of electromagnetic field. Specifically, in the spherical coordinates (r, θ, φ) its non-zero components are

$$E_r(r, \theta, t) = \psi(r) \frac{2d_0 \cos \theta}{r^3} \left(\chi + \frac{\dot{\chi} r}{c} \right) - \frac{d_0 \cos \theta}{r^2} \left(\chi + \frac{\dot{\chi} r}{c} + \frac{\ddot{\chi} r^2}{c^2} \right) \frac{\partial \psi}{\partial r}, \quad (71)$$

$$E_\theta = \psi(r) \frac{d_0 \sin \theta}{r^3} \left(\chi + \frac{\dot{\chi} r}{c} + \frac{\ddot{\chi} r^2}{c^2} \right), \quad (72)$$

$$H_\varphi = \psi(r) \frac{d_0 \sin \theta}{cr^3} \left(\dot{\chi} r + \frac{\ddot{\chi} r^2}{c} \right), \quad (73)$$

where

$$\psi(r) = 1 - e^{-\alpha r^5}, \quad (74)$$

$$\chi(t - r/c) = \sin^4[\omega_1(t - r/c)] + \sin^4[\omega_2(t - r/c)], \quad (75)$$

and the upper dot denotes differentiation with respect to t . The fields (71)-(73) are driven by the following non-zero components of the electric current density:

$$j_r = \frac{d_0 \cos \theta}{cr^2} \left(\dot{\chi} + \frac{\ddot{\chi}r}{c} + \frac{\ddot{\chi}r^2}{c^2} \right) \frac{\partial \psi}{\partial r}, \quad (76)$$

$$j_\theta = -\frac{d_0 \sin \theta}{cr^3} \left(\dot{\chi}r + \frac{\ddot{\chi}r^2}{c} \right) \frac{\partial \psi}{\partial r}. \quad (77)$$

In formulae (71)-(73) and (76), (77), for the factor $\psi(r)$ defined by (74) we have $\psi \propto r^5$ and $\frac{\partial \psi}{\partial r} \propto r^4$ as $r \rightarrow 0$. This behavior suppresses the singularities that would have otherwise appeared in the corresponding expressions at $r = 0$. On the other hand, for larger r the derivatives of ψ decay very rapidly (exponentially). This effectively turns the currents into zero (to machine precision) outside a given finite region around the origin. By adjusting the parameter α , one can guarantee that the support of the currents (the set where the latter are non-zero) is confined to $S_0 = S \setminus S_\varepsilon$, see Figure 1. We also point out that solution (71)-(73) is a regularized version (at $r = 0$) of the field radiated by the elementary dipole $d(t) = d_0 \chi(t)$ located at the origin and aligned with the z axis.

Two comments are in order regarding the test solution (71)-(73):

- By design, $\text{grad } \mu$ has non-vanishing r and θ components, and so does the electric field (71)-(72). Therefore, the inner product $(\nabla \mu, \mathbf{E})$ in (29) is non-zero on S_ε . As such, the electric charge will, generally speaking, accumulate, and we expect to have the quasi-lacunae of the electric field. The corresponding steady-state contributions will be eliminated by introducing the extra term (34) into the electric current (32).
- The necessary condition (44) for the existence of a divergence-free extension of the electric field is satisfied. This can be directly verified by calculating the flux of the electric field (71)-(72) through a sphere centered at $r = 0$. Physically, the flux is equal to zero because the solution is driven by a dipole-like source which is characterized by the zero total charge.

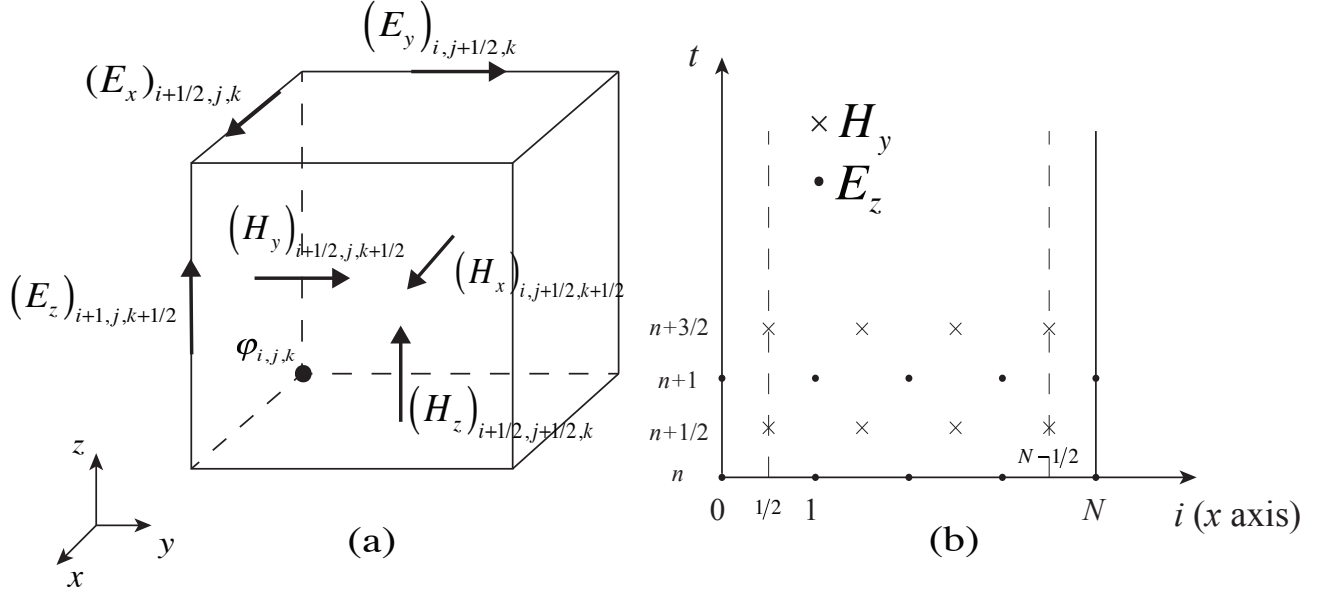


Figure 3: (a) Allocation of field components; (b) Temporal grid.

3.1.2 Implementation. Numerical implementation of the algorithm is based on the very popular Yee scheme [71]. The Yee scheme employs a staggered discretization of the fields in both space and time, see Figure 3. To illustrate the idea of the Yee scheme which is otherwise well-known, we present the discrete counterparts of the two governing equations, (66) and (68):

$$\frac{(H_y)_{i+1/2,j,k+1/2}^{n+1/2} - (H_y)_{i-1/2,j,k+1/2}^{n+1/2}}{h_x} - \frac{(H_x)_{i,j+1/2,k+1/2}^{n+1/2} - (H_x)_{i,j-1/2,k+1/2}^{n+1/2}}{h_y} = \frac{1}{c} \frac{(E_z)_{i,j,k+1/2}^{n+1} - (E_z)_{i,j,k+1/2}^n}{\tau} + \frac{4\pi}{c} (j_z)_{i,j,k+1/2}^{n+1/2}, \quad (78)$$

$$-\frac{(E_z)_{i+1,j,k+1/2}^{n+1} - (E_z)_{i,j,k+1/2}^{n+1}}{h_x} + \frac{(E_x)_{i+1/2,j,k+1}^{n+1} - (E_x)_{i+1/2,j,k}^{n+1}}{h_z} = -\frac{1}{c} \frac{(H_y)_{i+1/2,j,k+1/2}^{n+3/2} - (H_y)_{i+1/2,j,k+1/2}^{n+1/2}}{\tau}. \quad (79)$$

When time marching the solution by means of the Yee scheme, we first advance the electric component E_z to the time level $n+1$ with the help of (78) and then update of the magnetic field to the level $n+3/2$ using (79). However, to update of the near-boundary magnetic fields $(H_y)_{i+1/2,j+1/2,k}^{n+1/2}$, $i = 0, N-1$, and $(H_x)_{i+1/2,j+1/2,k}^{n+1/2}$, $j = 0, N$, from the previous time level one needs the electric field components $(E_z)_{i,j,k+1/2}^{n+1}$ at the boundary, i.e., at $i = 0, N$ and $j = 0, N-1$, respectively, see Figure 3(b). These components are not generated by the Yee scheme itself and should rather be supplied as the boundary data. It is the AP that provides these boundary data at every time step.

In general, due to the staggered nature of the spatial grid used by the Yee scheme, there are two options to provide a boundary closure for the interior problem. One can specify either the

tangential components of the electric field or those of the magnetic field. We choose the first option for definiteness.

If the aforementioned boundary values of the electric field are not known, we can still update the electric field of the IP one time step further, but not on the entire grid. In our example, E_z can be updated everywhere up to the second to last node leaving the boundary nodes $i = 0, N$ and $k = 0, N$ undefined. Accordingly, the magnetic field H_y of the IP can be updated to next time level $n + 3/2$ for all $i = 3/2, \dots, N - 3/2$. This partial update is still sufficient though to create the auxiliary currents (32) and (24).

Indeed, suppose we have done the foregoing partial update of the fields. After that we can compute the electric current (32) at the time level $n + 1/2$, as well as the magnetic current (24) at the level $n + 1$. When computing these currents, we do not need to know the non-updated boundary values of the fields because we know the exact values of currents (32), (24) at the boundary Γ of the computational domain S : they turn to zero there, see note (d) in the list on page 22. We also reiterate that we are using the original version of the magnetic current (24) without the extra term (35) because the boundary conditions for the IP involve only the electric field. The magnetic field is not needed for the boundary conditions and hence it is not important whether the magnetic steady state involves or does not involve any non-zero contributions.

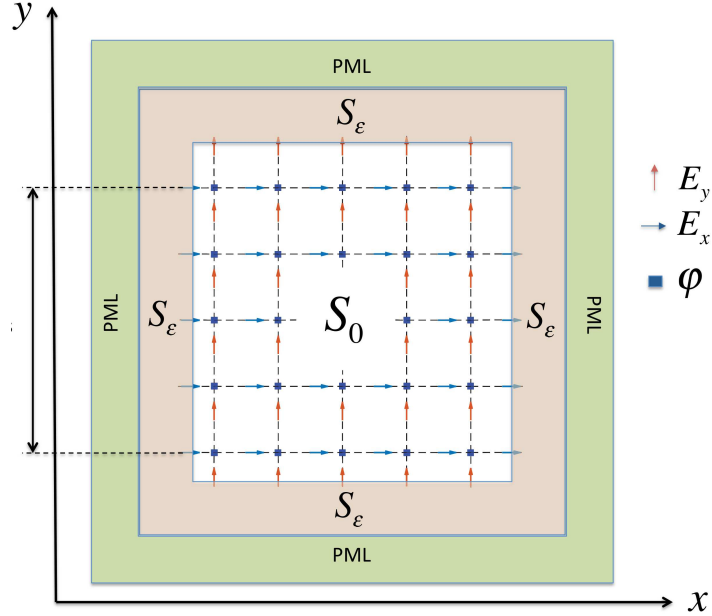


Figure 4: Grid for the electric potential.

The electric current (32) with the extra term (34) requires the continuation $\mathcal{E}$ of the electric field $\mathbf{E}$ onto S_0 from its boundary ∂S_0 , see (36). This continuation is computed as the anti-gradient of the solution to the discrete counterpart of the Neumann problem (40) for the electric potential φ on S_0 . The electric potential is defined at the corners of the elementary cell, see Figure 3(a), so a new grid for φ on S_0 has to be introduced, see Figure 4. This grid has the same dimension in each direction. The boundary values of the normal derivative of φ at ∂S_0 are simply the values of the electric field $\mathbf{E}$ at time level $n + 1$ computed during the foregoing partial update.

3.2 Modular version of the in-house code

The main goal of developing a new version of the in-house code was to achieve the modular capability. The new code implements a number of key components of the lacunae-based algorithm as independent modules:

- A time-domain solver for Maxwell's equations (currently employs the Yee scheme);
- A procedure for the treatment of artificial outer boundaries (ABCs/PMLs);
- Lacunae-based time marching;
- Partition into the interior and auxiliary problems, generation of auxiliary currents;
- Generation of primary sources and test solutions, computation of errors.

The modules communicate with one another through a certain well-defined mechanism that involves classes and objects. Independence of modules means that modifications can be introduced into a given module without having to touch any other parts of the code at all. This allows for an easier and more straightforward testing of the various implementation strategies and treatments of the artificial outer boundary (standalone or combined with lacunae-based time marching). It also guarantees a clear and unbiased comparison of performance for different test cases in the sense that an observed change in performance can be unambiguously attributed, for example, to a particular change in the treatment of the outer boundary and nothing else. In contradistinction to that, in the previous version of the code changing, say, an ABC or a PML would typically require making substantial changes through the entire program.

3.2.1 Programming language. One of the most important considerations in building the in-house Maxwell's/lacunae code with a modular capability is the choice of the programming language. The original version of the code that did not include the modular capability was written in Fortran 77. There are, however, dozens of modern programming languages nowadays that offer various advantages and have certain disadvantages. The language to be used for further development of the code should meet the following requirements:

- Support of standard mathematical functions and libraries;
- High performance;
- MacOS X support;
- Support of parallel multiprocessor computing;
- Backward compatibility.

The language should also provide high flexibility and support for a number of programming standards, such as the functional and object-oriented paradigms, as well as modular programming technique.

We have been choosing from the following languages: MATLAB, Python, C++, Fortran 2008.

MATLAB has a number of well-known advantages: simplicity of the language, an extensive set of built-in mathematical functions, as well as tools for data visualization. However, there are some significant drawbacks.

Sophisticated programs with large number of modules and functions are slow. We have developed programs for the numerical solution of Maxwell’s equations using MATLAB and Fortran. The MATLAB program ran 250 times slower than the Fortran program for the exact same case. It turned out that in the MATLAB program the time was spent mostly on execution of the functions from different modules and on transfer of the global parameters. Another drawback is the difficulty of porting MATLAB codes to such industrial languages like C++ or Java. In fact, the entire program has to be rewritten.

Python is a well-known programming language with simple syntax. The default Python interpreter was specifically designed with simplicity in mind, and has a thread-safe mechanism, the so-called GIL (Global Interpreter Lock). In order to prevent conflicts between threads, it executes only one statement at a time (serial processing, or single-threading). There is a number of libraries to circumvent this restriction, as well as the ability to run C++ code in the body of the Python program. However, it makes the structure of the program substantially more complex. In that regard we note that support of parallel execution of the code on multiple processors, which is basically missing in Python, is one of the key venues toward fast computations.

As such, the two remaining options are C++ and Fortran 2008. Both languages support standard mathematical libraries and parallel computing using the OpenMP standard. Both are suitable for high performance computations. The final choice was made in favor of Fortran 2008. This language has lots of mathematical functions as a part of the standard. The Intel compilers that we use are optimized for the architecture. Besides, Fortran 2008 offers perfect backward compatibility with our “legacy” programs and modules written in Fortran 77. At the same time, it has modern tools for object-oriented programming. In addition to Fortran, we will still use MATLAB for visualization of the data.

3.2.2 Parallel multi-processor computations. The simulation of electromagnetic waves propagating in three space dimensions requires substantial computational resources. The total amount of floating point operations depends on the grid dimension and the simulation time. Our goal is to investigate the long-time behavior of the solution with various types of boundary conditions. That is why it is very important to have an efficient computational algorithm. To attain high efficiency (fast execution speeds) we use parallel implementation of the code on a multi-processor platform with shared memory.

Fortran 2008 supports the OpenMP technology that is designed for multi-platform shared memory parallel programming. OpenMP has a number of advantages:

- The concept of “incremental parallelization” allows one to quickly parallelize programs with loops.
- OpenMP allows controlling the behavior of threads easily.
- OpenMP compiler directives make it easy to use one and the same program on both single-processor and multi-processor platforms. OpenMP directives are merely ignored in the case of a single processor and the program executes in the serial mode. Hence, there is no need to support two types of programs, serial and parallel.

We employ the loop-level parallelism for computing the electromagnetic fields. The explicit Yee scheme that we use for computing the solution to Maxwell’s equations allows for a straightforward parallelization of the loops to any number of processors. For any given time step, the result of calculation in each processor does not depend on the results of calculation in other processors.

In addition, we use OpenMP for parallelization of the loops needed for computing the reference analytical solution.

The electromagnetic field is computed on a three-dimensional domain. As such, all the loops for updating the field components are triple nested. To effectively partition the workload between all available processors, one could implement a balancer. As a simpler solution, we use the OpenMP tool called “automatic parallelization of loops” for the outer loop only. In this case, the efficiency of parallelization depends on the number of available processors and the number of steps in the outer loop. For example, if there are eight processors and the loop contains nine steps, one of the processors will make twice as many operations as the others. Nonetheless, the speed of calculation increases by a factor of 4.5 even in this case.

Table 1 summarizes the results on both serial and parallel program performance on an eight-core computer and a twelve-core computer. We emphasize that in all the cases we use exactly the same Maxwell’s/Yee code in the same setting (domain size, grid dimension/size, duration of the time interval, etc.). The most important conclusion is that we can achieve a substantial acceleration of computations by adding simple compiler directives with no additional coding. OpenMP directives do not clutter the code and do not decrease its readability.

Mac Pro 4.1, Quad-Core Intel Xeon, 2x2.93 GHz, 8 cores			
Grid	Serial	Parallel	Acceleration
$16 \times 16 \times 16$	9.02	1.54	5.86
$18 \times 18 \times 18$	14.51	3.15	4.61
$24 \times 24 \times 24$	46.36	8.38	5.53
$32 \times 32 \times 32$	148.69	22.59	6.58
Mac Pro 6.1, 12-Core Intel Xeon E5, 2.7 GHz, 12 cores			
Grid	Serial	Parallel	Acceleration
$16 \times 16 \times 16$	7.06	1.31	5.39
$18 \times 18 \times 18$	11.44	1.78	6.43
$24 \times 24 \times 24$	36.88	4.8	7.68
$32 \times 32 \times 32$	116.22	18.4	6.32

Table 1: Comparison of the single-thread and parallel execution times.

3.2.3 Objects and structure. The most important feature of the new code is its flexible structure that was built using principles of object oriented programming. This means that the algorithms are implemented using data structures that contain data and methods. A distinct feature of objects is that their procedures can access and modify the data fields of the object that they are associated with. In this case the program consists of the objects that interact with each other, but similar objects are implemented only once. It is possible to upgrade the program by upgrading the objects or adding other data structures without changing of the previous implementation.

3.2.3.1 Mesh and source. We have built the class `tMesh` that contains the description of the field components implemented as three-dimensional dynamic data arrays. The methods of this class allow one to allocate and free the dynamic memory. Control of the access to the cells of dynamic arrays is integrated as well. We use additional methods to control the update history of each cell to be sure that each cell is updated once per time step.

The class `tSource` has been implemented and tested as well. This class provides methods for computing the analytical test solution to Maxwell’s equations. Two test solutions were constructed previously without using the object-oriented principles. Yet we have developed a flexible structure that allows one to include these test solutions quickly with the help of a common programming interface. The class `tSource` is used to get the exact values of the electric and magnetic fields at any space location and any moment of time. Thus, we can obtain the exact analytical solution for Maxwell’s equations at all nodes of the chosen discretization grid and compare it with the numerical solution at every time step. This class also allows one to get the values of the currents that drive the solution at any point in space and every moment of time. Those currents provide the source terms for Maxwell’s equations when they are solved numerically (the right-hand side for the scheme).

3.2.3.2 Problem. The class `tProblem` is the main class of the entire program. It is developed for the numerical simulation of Maxwell’s equations with different boundary conditions. This class incorporates the descriptions of the numerical and analytical solutions using the `tMesh` and `tSource` classes.

When computing the solution to Maxwell’s equations, it is necessary to separate the processes of updating the fields inside the computational domain and at its boundaries. We have developed unified procedures for updating the field components at the interior grid nodes using the Yee scheme (a second order central difference staggered discretization). These procedures are called `updateEinterior(t)` and `updateHinterior(t)` for the electric and magnetic field, respectively. The word “interior” refers to the grid nodes inside the computational domain everywhere except at its boundaries.

Unified methods for updating the boundary nodes have been implemented as well. They allow one to use different types of boundary conditions. Regardless of the boundary conditions, the program updates the same boundary nodes in the same order. The size of the computational domain and its position with respect to the origin, the grid dimension in each direction, and the type of the boundary condition are the initialization parameters of the `tProblem` object. A schematic illustration of a 2D cross-section of the grid for the electric field is shown in Figure 5. Each cell in Figure 5 represents a location where the electric field is calculated. Red cells are the interior nodes and white cells are the boundary nodes (updated via the chosen boundary condition).

The `tProblem` class includes the method `DoStep(t)` to update all the nodes inside the computational domain. First, the program specifies the source terms (currents). Next, it updates the electric and magnetic fields at the interior nodes using the Yee scheme. Then, the values of the electric field at the boundary are updated on the boundaries using the chosen boundary condition. Finally, the values of the magnetic field are calculated at the boundaries using the same Yee scheme.

One of the methods of the `tProblem` class is `getAnalyticSolution(t)`. It allows one to obtain the exact values of the electric and magnetic fields on the entire discretization grid. After that, using the method `getProblemErrors(t)`, one can calculate the relative and absolute errors of the numerical solution. The block diagram of the function `DoStep(t)` is shown in Figure 6.

3.2.3.3 PML. We have developed a separate class `tPML` that describes the perfectly matching layer. This layer provides a boundary closure for the problem. The class computes the electromagnetic field, as well as all the auxiliary quantities needed for an unsplit PML in a layer that surrounds the computational domain. This class also stores all the variables for the PML.

First, the numerical solution is calculated in the `tProblem` class. Then, the values of the fields are transmitted to the object of the `tPML` class. Using this object, the solution is calculated in the

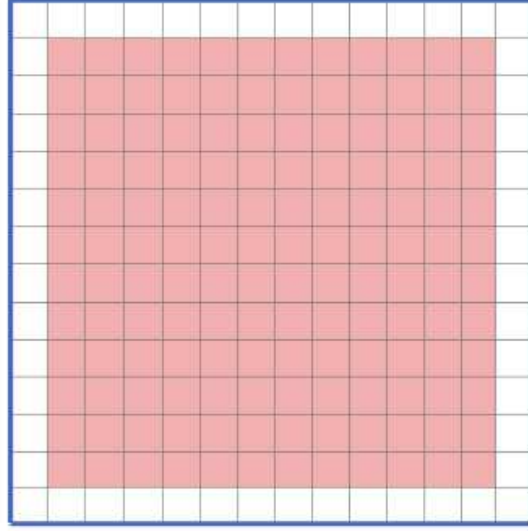


Figure 5: Schematic 2D cross-section of the grid for the electric field. Red cells are the nodes in the interior of the computational domain and white cells are the nodes that lie at the boundary.

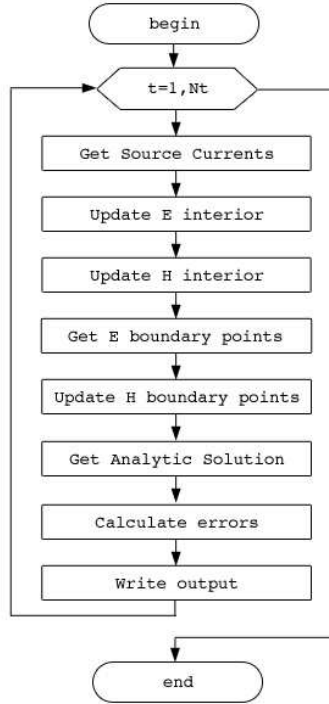


Figure 6: Block diagram of the method `DoStep(t)` that updates all nodes of the grid.

PML, including the boundary nodes of the grid. After that, the values of the fields at the boundary nodes are transmitted back to the object of the `tProblem` class. A schematic illustration of a 2D cross-section of the grid for the electric field is shown in Figure 7.

We emphasize that there is no need to change the size of the arrays that describe the problem in this implementation. PML is implemented as an add-on to the original problem, which allows one

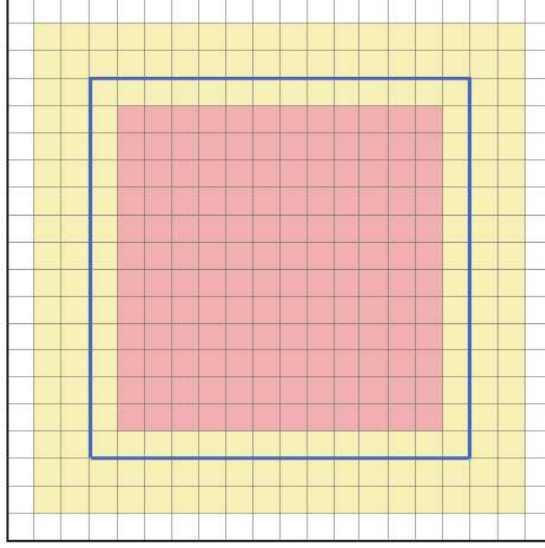


Figure 7: Schematic 2D cross-section of the grid for the electric field. Red cells are the nodes in the interior of the computational domain, yellow cells lie in the PML, and white cells are the nodes that lie at the outer boundary of the PML.

to calculate the fields at the boundary points. The procedure that updates the boundary points remains the same. In the case of using the PML, the values at the boundary nodes are taken from the PML. We note that the class `tProblem` contains the object of the class `tPML`.

It is possible to change the thickness of the perfectly matching layer, as well as the absorption profile of the PML. We have implemented methods for updating the interior nodes in the PML. As the values of the fields are equal to zero at the boundaries of the PML, there is no need to update the boundary points of the perfectly matching layer. The block diagram of the function `DoStep(t)` in the case of using the PML is shown in Figure 8.

3.2.3.4 Buffer. The class `tProblem` lets one create any number of problems with specified parameters. This is necessary for the implementation of the lacunae-based time marching algorithm. Nevertheless, the objects of the `tProblem` class cannot interact with one another directly.

The class `tBuffer` is implemented to enable interactions between the objects of the same class and transfer the data between them. All objects of the `tProblem` class can read and write the data from a `tBuffer` object. In fact, the object of the `tBuffer` class consists of a number of objects of the `tMesh` class and provides initialization and access to them in order to transfer the values of the currents and fields between different problems. It is important that the implementation of the `tBuffer` class allows one to use exactly the same procedures for getting the source terms (i.e., currents) and boundary values for the objects of the `tProblem` class. Depending on the preferences, an object of the `tProblem` class can get the values of the fields for the boundary nodes from the `tSource` class (in the case where the boundary condition is obtained from the exact solution), from the perfectly matching layer, or from the object of the `tBuffer` class. Likewise, `tProblem` objects can get the source terms from a `tSource` object or from a `tBuffer` object.

3.2.3.5 Solution. Yet one more class, `tSolution`, has been implemented in order to control and coordinate the actions between the interior problem and auxiliary problems. This class contains

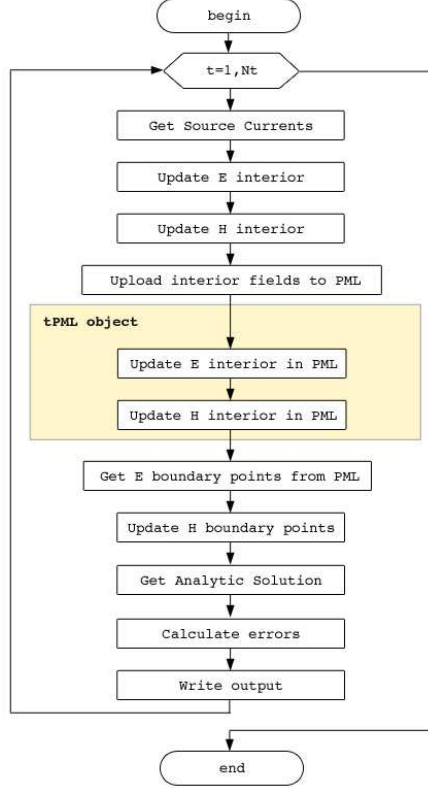


Figure 8: Block diagram of the method $\text{DoStep}(t)$ in the case where the PML is used in the capacity of a boundary closure. E denotes the electric field and H denotes the magnetic field.

methods for initialization and dropping of the individual problems. It enables the interaction between the problems and executes their functions in the right order.

Objects of the `tSolution` class are used to solve the problem with predetermined options. For example, the problem can be solved with the PML in the capacity of a boundary condition. In this case, the object of the `tSolution` class creates one object of the `tProblem` class and initializes the $\text{DoStep}(t)$ method of this object on every time step to compute the numerical solution.

The full-fledged lacunae-based time marching algorithm is somewhat more complicated. It is implemented as follows. An object of the `tSolution` class contains one object of the `tProblem` class that is called the main (i.e., interior) problem. In addition to that, the `tSolution` object contains a dynamic array of `tProblem` objects that are called the auxiliary problems. It is important that both the main problem and auxiliary problems are objects of the same class with the same implementation of the data and methods, including perfectly matching layers. The PML is used as a boundary condition for the auxiliary problems. The domain of the auxiliary problem is larger than that of the main problem while their centers coincide.

First, the interior nodes of electric field are updated in the main problem. After that we need to solve the Neumann problem for the Laplace equation (40) to build the auxiliary currents that prevent the accumulations of charge. A schematic illustration of a 2D cross-section of the grid for the electric field is shown in Figure 9. The Neumann problem is solved by an additional module that performs the fast Fourier transform in 3D. The composite divergence-free electric field is then formed from the solution of the Neumann problem and the original field of the interior problem according to the first formula (36).

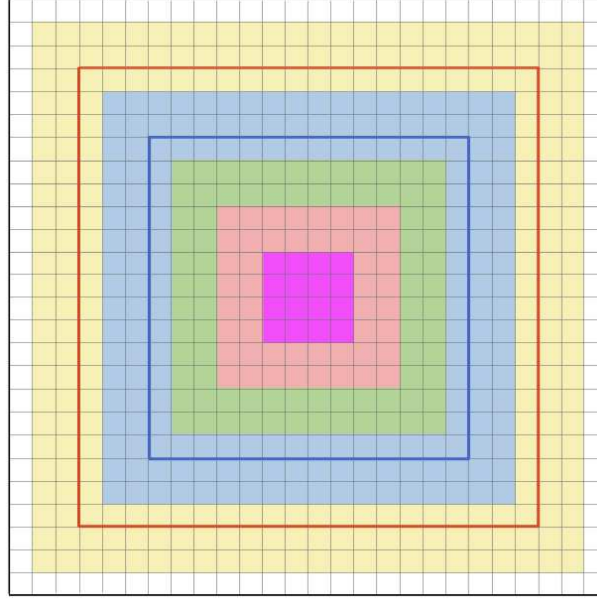


Figure 9: Schematic 2D cross-section of the grid for the electric field. Purple cells represent the domain of the Neumann problem the prevents the accumulation of charge. Red and green cells are the interior nodes of the main problem; green area contains the cells where the auxiliary currents are generated. Blue cells represent the interior nodes of the auxiliary problem. Yellow cells lie in the PML. White cells are at the outer boundary of the PML. Blue line is the outer boundary of the interior (i.e., main) problem; red line is the boundary for the auxiliary problem.

The auxiliary currents are formed near the inner boundary of the main problem. The genuine electromagnetic field in this area is governed by Maxwell's equations in vacuum. The resulting effective currents are saved to the `tBuffer` object.

Each auxiliary problem reads the values of the effective currents from the `tBuffer` class. The auxiliary currents are partitioned in time by the function $\nu(t)$ according to the index of the auxiliary problem, see Figure 2 and formulae (23), (24) and (32), (33). After that, the `tSolution` object executes the `DoStep(t)` function with the perfectly matching layer for each auxiliary problem. The auxiliary problems add the resulting electric field to the `tBuffer` object. The boundary values of the main problem are updated using the sum of the electric fields from the auxiliary problems that are stored in the `tBuffer` object. The last step is to update the values of the magnetic field at the boundary of the main problem. The block diagram of the overall algorithm is shown in Figure 10.

Each auxiliary problem has a predetermined finite “lifespan.” The `tSolution` object drops the auxiliary problems and adds new problems at proper times to render the time marching algorithm.

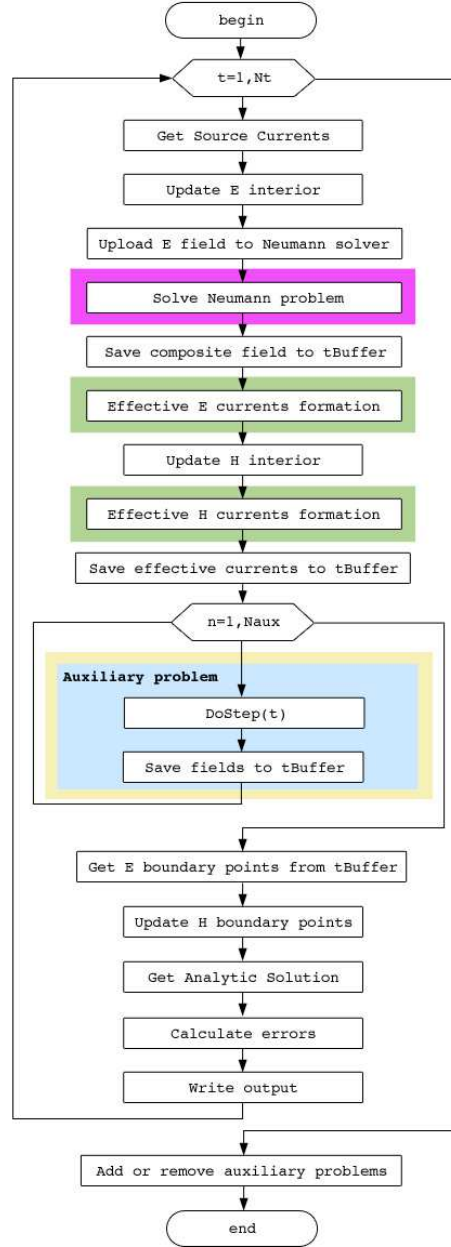


Figure 10: Block diagram for the implementation of the lacunae-based time marching algorithm. The colors are the same as in Figure 9; they represent different parts of the computational domain.

4 Lacunae-based methodology optimization in the in-house Fortran code

The goal of developing a modular in-house code was to enable efficient testing of the lacunae-based methodology along with a variety of ABCs/PMLs and for a variety of simulation regimes.

After the initial development was complete, the in-house code was optimized for performance. The optimization included several stages. First, the code was streamlined. All its unnecessary components, such as the temporary variables, functions, arrays, etc., that were introduced for development yet were not needed for production, have been removed. As a result, the execution

speed has noticeably increased.

Then, all the quantities that define a given simulation, such as domain sizes, grid dimensions and sizes, parameters of the source of electromagnetic waves (frequencies, polarizations, etc.) were identified and implemented as input variables. This enables a substantial degree of automation, i.e., it potentially allows us to run series of simulations for different values of the parameters without having to recompile any part of the code and restart the computation. Certain important parameters that pertain to the closure at the artificial outer boundary, e.g., the width or number of nodes in the PML, have also been implemented as input variables.

Next, the output module has been substantially improved. A number of options have been added to it and implemented as parameters that also define the simulation:

- Output of the relative and absolute errors for each component of the field;
- Output of the grid location where the maximum absolute or relative error is reached at each time step;
- Output of the divergence of the auxiliary field at each time step.

This further increases the degree of automation in running series of numerical experiments.

Finally, the module for the solution of the Poisson equation (needed for preventing the accumulation of charge by the auxiliary currents) has been updated. We have replaced all the static arrays by the dynamic ones. It allows us to allocate and deallocate the dynamic memory for the arrays that are used for constructing the auxiliary currents.

5 Implementation of low order local ABCs with the in-house code

We have implemented a number of low order local ABCs to study their behavior when combined with the lacunae method. The flexible structure of our software package allows us to implement different boundary conditions as methods of the object `tProblem` without having to change the rest of the code. The boundary conditions we have implemented can be used to truncate the computational domain for the original problem or for the auxiliary problems in the lacunae method. In the current implementation, these boundary conditions are set for the electric field.

5.1 Sommerfeld boundary condition

Sommerfeld radiation boundary condition in the time domain is the simplest possible ABC. It is given by the following first order hyperbolic PDE:

$$\frac{\partial E}{\partial x} = \frac{1}{c} \frac{\partial E}{\partial t}. \quad (80)$$

Boundary condition (80) is transparent for the outgoing plane waves of the type $E = E(x - ct)$ traveling along the x axis in the positive direction toward the artificial boundary $x = \text{const}$. For other boundaries, such as $y = \text{const}$ and $z = \text{const}$, Sommerfeld boundary conditions are formulated similarly. In all the cases, they are designed to absorb the waves at the normal incidence only. That's why it is expected that this boundary condition will produce a significant overall reflection error.

5.2 Higdon ABC

In work [66], Higdon has proposed a differential annihilator for the sum of N plane waves:

$$\prod_{n=1}^N \left(\cos \phi_n \frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right) E = 0. \quad (81)$$

Boundary condition (81) is exact for the plane waves traveling at the angles of incidence ϕ_n , $-\pi/2 < \phi_n < \pi/2$, toward the artificial boundary $x = \text{const}$. The order of the Higdon boundary operator is determined by the number of angles N . The second-order case, $N = 2$, of the Higdon boundary condition (81) has been implemented into our in-house code.

5.3 Betz-Mittra ABC

When an electromagnetic wave is interacting with an artificial outer boundary, both propagating and evanescent waves may get scattered. The Higdon ABC (81) is designed to annihilate the propagating waves, but is deficient in absorbing the evanescent waves. To circumvent this problem, Betz and Mittra in work [67, 68] have proposed a modified form of the Higdon ABC that also annihilates the evanescent waves:

$$\left(\frac{\partial}{\partial t} - \frac{c}{\cos \phi_1} \frac{\partial}{\partial x} + \alpha_1 \right) \left(\frac{\partial}{\partial t} - \frac{c}{\cos \phi_2} \frac{\partial}{\partial x} \right) E = 0. \quad (82)$$

Boundary condition (82) is of second order; other orders can also be considered. The quantity α_1 in (82) is the attenuation constant chosen so that $\alpha_1 \Delta x = 0.1$, where Δx is the grid size. A similar attenuation constant can also be introduced into the second factor of the operator on the left-hand side of (82). Boundary condition (82) has been implemented into our in-house code. For boundary conditions (80), (81), and (82), we have used the specific form of the FDTD implementation proposed in the book by Gedney [69].

5.4 Mur ABC

In work [70], Mur has introduced a simple and convenient finite difference approximation for the second order Engquist-Majda ABC [6]:

$$\frac{\partial^2 E}{\partial x \partial t} - \frac{1}{c} \frac{\partial^2 E}{\partial t^2} + \frac{c}{2} \frac{\partial^2 E}{\partial y^2} + \frac{c}{2} \frac{\partial^2 E}{\partial z^2} = 0. \quad (83)$$

to be used specifically with the Cartesian Yee scheme [71]. The artificial boundary for (83) is $x = \text{const}$. Similar boundary conditions can be easily derived for other artificial boundaries, $y = \text{const}$ and $z = \text{const}$. As of yet, we have implemented the Mur boundary conditions (83) at the faces of the computational domain, while at the edges and corners we have used the Sommerfeld ABC (80), as proposed in [69]. This approach leads to significant reflection errors, so at a later stage we plan to improve the implementation of (83) by employing the one-sided second order differences near edges and corners.

5.5 Numerical experiments

We started testing the foregoing local low order ABCs in the capacity of a standalone closure for the original problem, i.e., without the lacunae-based time marching. For the Higdon ABC (81) and

Betz-Mittra ABC (82) we took the values of $\phi_1 = 0$ and $\phi_2 = \pi/4$. In general, low order ABCs are expected to produce fairly significant spurious reflections from the artificial outer boundary back to the computational domain. The level of those reflections depends on the design of a given ABC, as well as on how close the artificial boundary is placed to the source of electromagnetic radiation.

In Figure 11, we are showing the profiles of the relative error for the various ABCs on the domain of size 5 discretized by the grid of dimension $30 \times 30 \times 30$ nodes. The plots demonstrate that the type of the ABC strongly affects the accuracy of computation. The highest level of error is generated by the Sommerfeld ABC (80), and the best ABC in this experiment is the Betz-Mittra ABC (82), yet even this ABC still yields a larger error than that of the PML [72].

In Figure 12, we are showing similar results but for the domain which is twice as large (size 10) and discretized by the grid of dimension $60 \times 60 \times 60$. In other words, while leaving the effect of the discretization the same, we are placing the outer boundary further away from the waves' source. By comparing the results in Figure 12 with those in Figure 11 we see that for the larger domain (Figure 12) all the error profiles, including that for the PML [72], have reached the same bottom (although the least accurate Sommerfeld ABC (80) still generates the highest peaks). This indicates that for the setting of Figure 12, the overall accuracy is dominated by the discretization error on the grid, whereas the reflection error due to the ABCs has decreased because the computational domain has become larger.

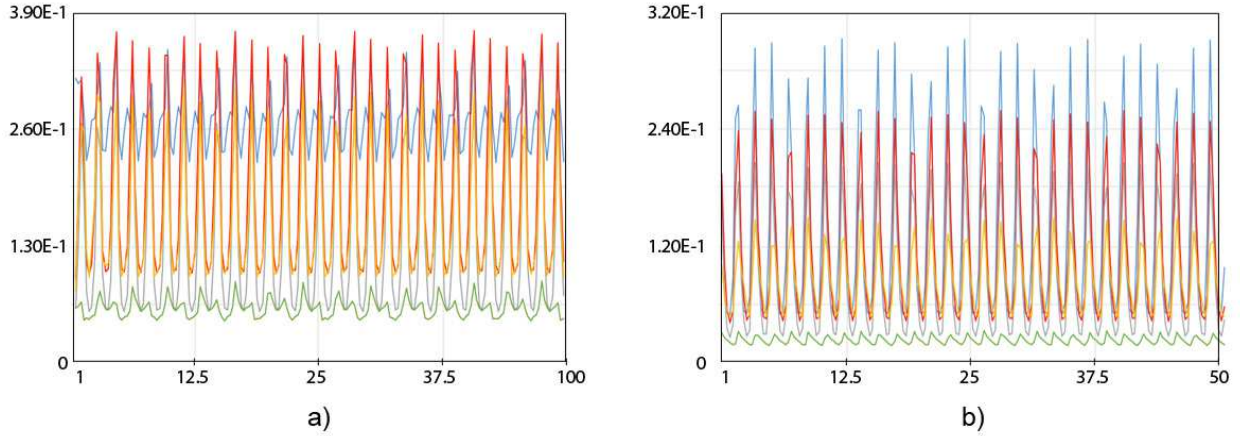


Figure 11: Relative error vs. time for the field components E_y (a) and H_x (b). The size of the computational domain in each direction is 5, and the grid dimension is $30 \times 30 \times 30$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].

Given that the error for the larger domain has reached the limit imposed by the discretization, we next study the grid convergence (see Figure 12), i.e., refine the grid while keeping the domain size (equal to 10) and the ABCs the same. The results are presented in Figure 13. We see that the Sommerfeld ABC (80), as well as the Mur ABC (83) (which still uses the Sommerfeld ABC at the edges and corners of the domain) demonstrate practically no grid convergence at all. This means that when the grid is refined and the discretization error decreases, the overall error for these boundary conditions is dominated by reflections that are independent of the discretization. The best grid convergence is demonstrated by the Betz-Mittra ABC (82), but it still flattens after the first refinement of the grid. This means that on finer grids the reflection error again becomes dominant. The error behavior for the Higdon ABC (81) is similar to that of the Betz-Mittra ABC

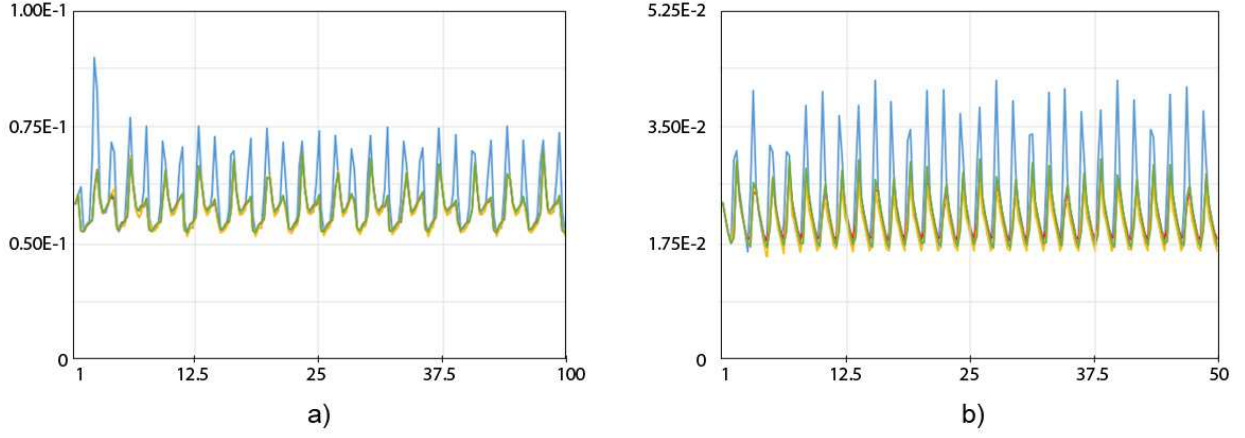


Figure 12: Relative error vs. time for the field components E_y (a) and H_x (b). The size of the computational domain in each direction is 10, and the grid dimension is $60 \times 60 \times 60$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].

(82) albeit the overall convergence is somewhat slower. The unsplit PML of [72] shows robust grid convergence on the entire sequence of grids.

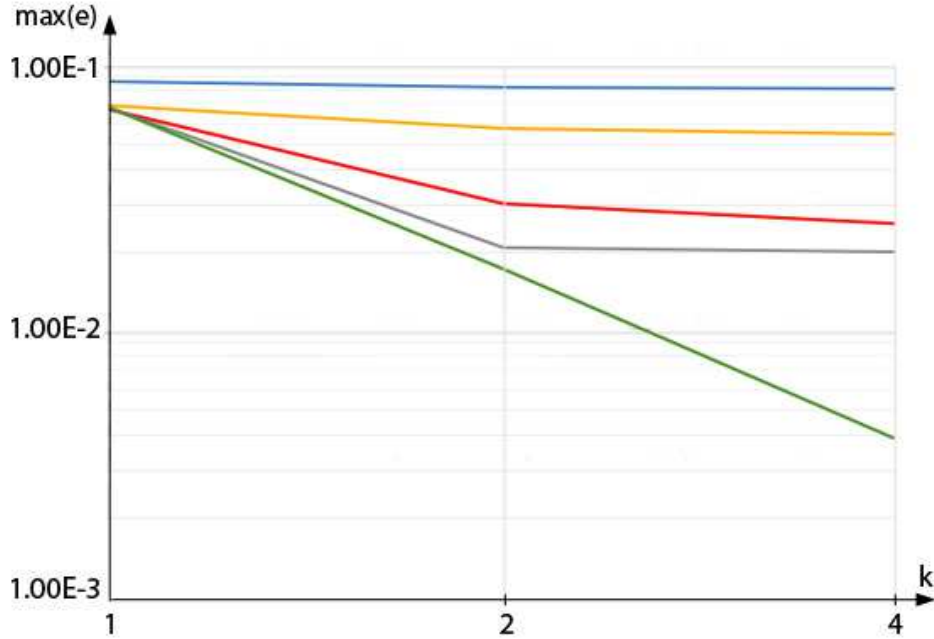


Figure 13: The grid convergence on the domain of size 10 for the field component E_y . The coarsest grid dimension is $60 \times 60 \times 60$; subsequent grids are refined by a factor of k , where $k = 2, 4$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].

5.6 Intermediate conclusions

The two main components of numerical error in our simulations are due to the discretization on the grid and due to the reflections from the artificial outer boundary. Those two components are independent from one another and controlled by different factors: the grid size and the type of ABC/location of the boundary, respectively. The overall accuracy of computations provided by low order local ABCs is typically not very high because of a substantial level of reflections from the outer boundary. Even the best performing ABC out of those we have considered (the Betz-Mittra ABC (82)) still appears inferior to the PML.

On the other hand, in those numerical experiments that we have conducted, we did not observe any long time deterioration of performance. In other words, the solution obtained with the help of low order local ABCs is stable yet not very accurate. The performance of local low order ABCs was compared against that of the unsplit PML [72]. For those simulation regimes that we have chosen, the PML did not show any long time deterioration either,⁵ yet has overperformed the low order ABC from the standpoint of accuracy.

Most likely, local low order ABCs will not require any lacunae-based stabilization as their performance does not seem to deteriorate over long simulation times. However, the accuracy that they provide is not satisfactory. Therefore, our next step will be to implement local high order ABCs into our in-house code.

6 Implementation of high order local ABCs by Givoli & Neta with the in-house code

For the initial implementation, we have chosen the Givoli-Neta ABCs [73]. These boundary conditions are among the first local high order ABCs that appeared in the literature. While substantial advances have been made in this area already after the publication of work [73] in 2003, the Givoli-Neta ABCs apparently remain one of the easiest in their class to implement. Moreover, in those cases where their implementation proves stable they can provide a noticeably better accuracy than the low order methods of Section 5.

The Givoli-Neta ABCs [73] are essentially equivalent to higher order Higdon ABCs (81) but implemented via the auxiliary variables at the artificial outer boundary. The use of the auxiliary variables allows one to avoid the numerical approximation of high order derivatives. The accuracy of the Givoli-Neta ABCs depends on their order, i.e., the number of terms in the product (81), as well as on the actual values of the angles ϕ_n .

The lacunae-based time marching helps improve the long time performance of other boundary conditions. However, to combine a given ABC with lacunae, one first needs to make sure that the standalone performance of the former remains acceptable over the time interval of a “reasonable” duration (not necessarily very long yet not too short either). Hence, we begin with testing the Givoli-Neta ABCs on their own. Our goal is to experimentally determine the parameters of the ABCs (as well as of the computational domain) that would enable the longest interval of their stable performance.

The computational domain we are using is shaped as a Cartesian cube with side length $2a$ centered at the origin. The cube contains a divergence-free compactly supported source of electromagnetic radiation. A Givoli-Neta ABC is set at the face $x = -a$. At the other five faces of the cube, we specify the values of the field from the exact solution. Ideally, we would like to see whether a Givoli-Neta ABC can provide stable performance for about 90 time units. This duration

⁵There are many cases for which the PML performs well altogether; otherwise, even if an instability develops it may manifest itself only after a rather long period of satisfactory performance.

of partial sub-problems has been successfully used previously when lacunae-based time marching was combined with an unsplit PML.

6.1 Second order accurate approximation of Givoli-Neta ABCs

6.1.1 Number of angles. We first study how the interval of stable performance depends on the number of angles ϕ_n chosen for a Givoli-Neta ABC. To do so, we fix the computational domain and the grid and compute several different solutions with the Givoli-Neta ABCs that differ by the number of angles. The angles are specified as follows: the first angle is 0° and the other angles are distributed uniform across the interval from 0° to 90° . For example, for three angles we have $(0^\circ, 30^\circ, 60^\circ)$; for four angles we have $(0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ)$, etc.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Number of Givoli-Neta angles ϕ_n .

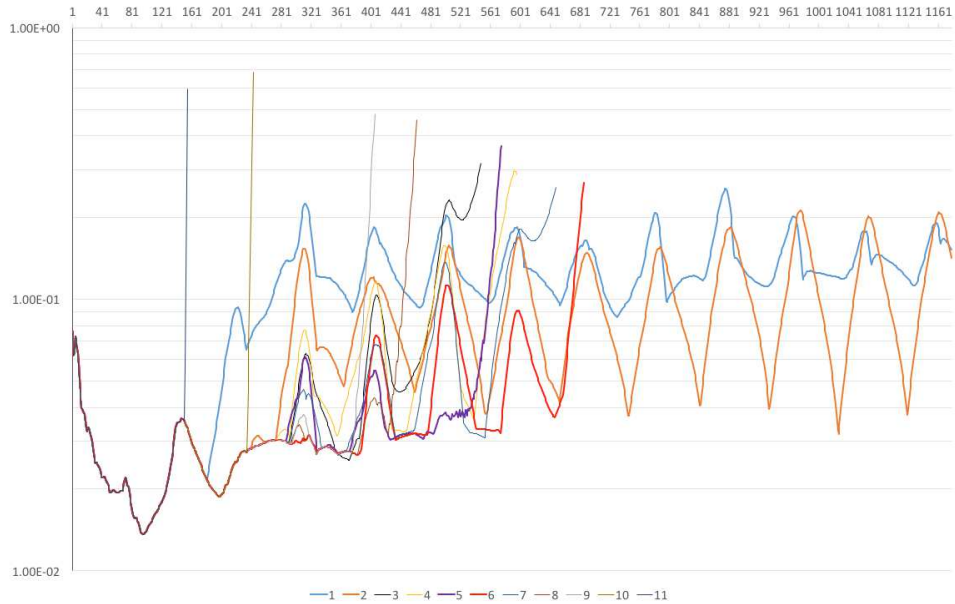


Figure 14: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the number of Givoli-Neta angles.

From Figure 14 we see that the Givoli-Neta ABCs are stable for the case of one or two angles. For larger numbers of angles, the longest period of stable performance corresponds to six angles. If the number of angles exceeds 11, then the solution blows up as quickly as for 11 angles, i.e., almost immediately after the wave reaches the outer boundary. Altogether, regardless of the number of angles (greater than 2), the interval of stable performance is relatively short, about 600 time steps or 20 time units, and we have not been able to increase it noticeably.

6.1.2 Size of the computational domain. Next, we study the dependence of the interval of stable performance on the size of the computational domain. To do so, we fix the number of Givoli-Neta angles at 6, as it was the best value derived from the previous experiment. The size of the computational domain $2a \times 2a \times 2a$ varies between $4 \times 4 \times 4$ and $10 \times 10 \times 10$. In doing so, the grid size in space remain fixed, while the grid dimension changes accordingly.

Fixed parameters:

Number of Givoli-Neta angles: 6;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Domain size: from $4 \times 4 \times 4$ to $10 \times 10 \times 10$;

Grid dimension: varies accordingly.

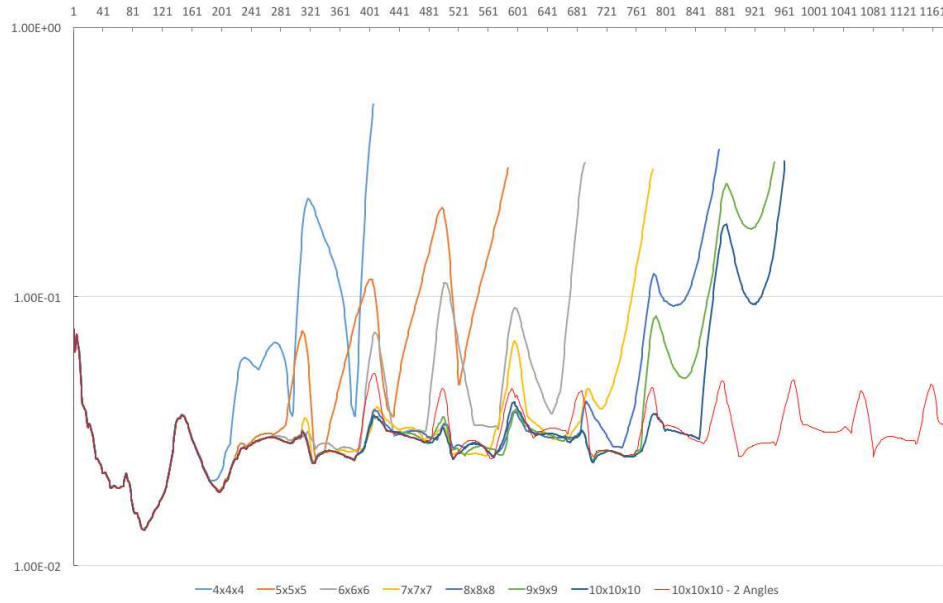


Figure 15: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the size of the computational domain.

From Figure 15 one can easily see that the larger the computational domain the longer the period of stable performance of the chose Givoli-Neta ABC (six evenly distributed angles). One possible explanation is that for larger domains that waves that reach the outer boundary are closer to plane waves and hence the boundary condition performs better. Moreover, for larger domains it takes longer for the waves to reach the outer boundary. Intuitively, one would therefore want to choose the large domain: $10 \times 10 \times 10$. However, the red curve in From Figure 15 corresponds to the $10 \times 10 \times 10$ and only two Givoli-Neta angles (rather than six). The difference in accuracy on the large domain of size $10 \times 10 \times 10$ between the ABC with six angles (high order) and the ABC with two angles (low order) is not large. In other words, the advantage of using higher order ABCs pretty much vanishes as the size of the computational domain increases.

6.1.3 Choice of angles — three angles. Let us now investigate how the interval of stable performance depends on the actual values of the Givoli-Neta angles ϕ_n . First, we take the overall

number of angles equal to 3. Then, using the same domain and the same grid, we compute the solution with the help of Givoli-Neta ABCs for which the angles ϕ_n assume different values.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Number of Givoli-Neta angles: 3;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Values of Givoli-Neta angles ϕ_n .

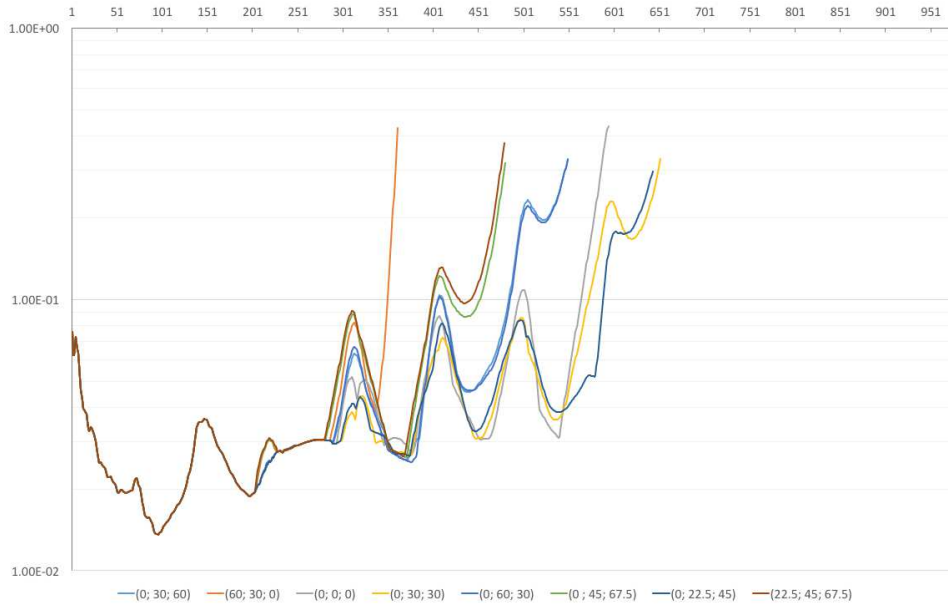


Figure 16: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles.

Figure 16 indicates that the interval of stable performance is the longest when the values of the Givoli-Neta angles are close to zero. It is interesting that the curves in Figure 16 have split into several groups. The worst performance is demonstrated by the decreasing sequence of angles: $(60^\circ, 30^\circ, 0^\circ)$. Next is the group of the brown and green curves that corresponds to the choice of large angles $(0^\circ, 45^\circ, 67.5^\circ)$ and $(22.5^\circ, 45^\circ, 67.5^\circ)$. Then comes the group of blue curves that correspond to the uniform choice of angles first of which is 0° : $(0^\circ, 30^\circ, 60^\circ)$ and $(0^\circ, 60^\circ, 30^\circ)$. Finally, there is a group that corresponds to the angles closer to 0° . The best performing ABC in the group is the one for which the angles are $(0^\circ, 22.5^\circ, 45^\circ)$. We still note that that in spite of the existing differences in performance, the choice of the Givoli-Neta angles does not play a decisive role as far as the interval of stable performance is concerned. For any choice of angles, this interval does not exceed 700 time steps.

6.1.4 Choice of angles — six angles. Our previous experiments show that the size of the computational domain should not be too small, because then the boundary condition quickly leads

to an instability, and should not be too large either, because in that case the advantage of using higher order ABCs vanishes. Moreover, the best performing number of angles is six. Let us therefore continue the investigation of the previous section, take the overall number of angles equal to six, and allow their values to vary.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Number of Givoli-Neta angles: 6;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Values of Givoli-Neta angles ϕ_n .

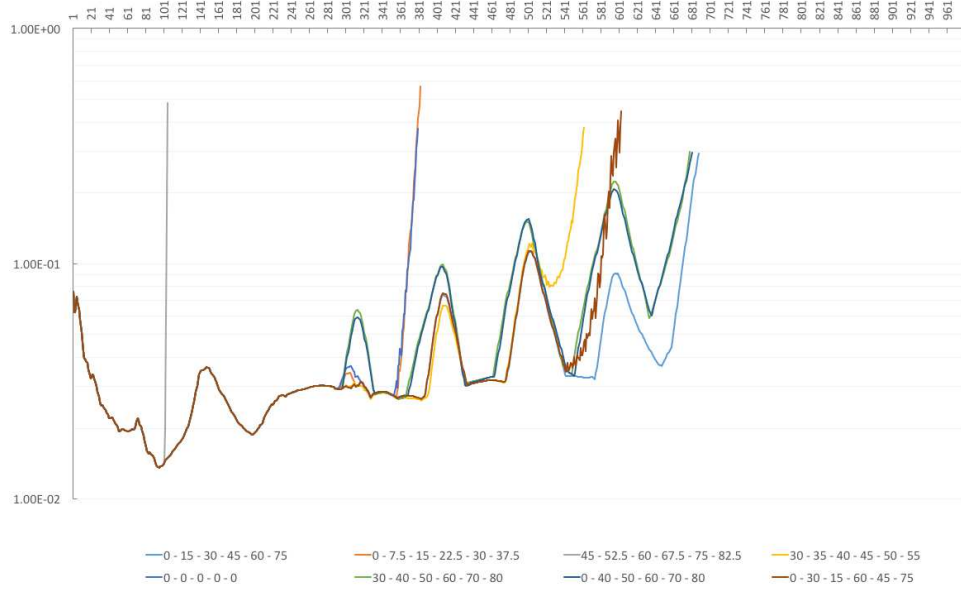


Figure 17: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of six Givoli-Neta angles.

Figure 17 shows that the optimal choice of six angles corresponds to the angles that are evenly distributed between 0° and 90° .

6.1.5 Grid size. The investigation in this section shows that as the grid size decreases the interval of the stable performance also decreases, whereas when the grid size increases the error becomes worse, see Figure 18. Hence, we cannot expect that by varying the grid size we will be able to achieve a better overall performance of the Givoli-Neta ABCs.

6.1.6 Intermediate conclusion. The optimal Givoli-Neta ABC for the domain of size $6 \times 6 \times 6$ corresponds to six evenly distributed angles between 0° and 90° . Even for this choice, the interval of their stable performance is only about 20 time units, which is far shorter than what we have used previously for the study of the unsplit PML.



Figure 18: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the grid size.

6.2 First order accurate approximation of Givoli-Neta ABCs in time

It is well known that often the approximation accuracy at the boundary for a hyperbolic problem can be chosen one count lower than inside the computational domain. This does not hurt the overall accuracy yet may improve the stability. Therefore, we decided to conduct a similar series of experiments with Givoli-Neta ABCs as that described in Section 6.1, but approximate the boundary conditions with only first order accuracy in time while keeping the second order accuracy in space.

6.2.1 Number of angles.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Number of Givoli-Neta angles ϕ_n .

The behavior of the solution curves in Figure 19 is similar to that in Figure 14 (second order accurate approximation of the ABCs); the differences are not substantial. High order Givoli-Neta ABCs always lead to instability. The best choice in this case is five angles (green curve); it guarantees the longest period of stable performance. For comparison, we also present the graph that correspond to an unsplit PML.

6.2.2 Size of the computational domain. Next, we study the dependence of the interval of stable performance on the size of the computational domain. We fix the number of Givoli-Neta

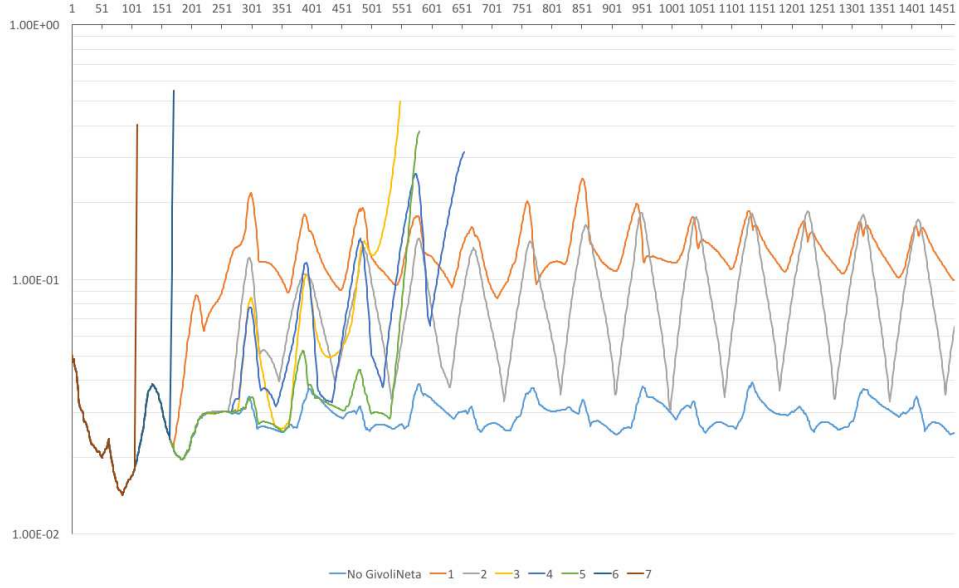


Figure 19: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the number of Givoli-Neta angles. First order accurate approximation of the ABCs in time.

angles at 5, which corresponds to the best performance the previous experiment. Then, we vary the size of the computational domain between $4 \times 4 \times 4$ and $10 \times 10 \times 10$. In doing so, the grid size in space remain fixed, while the grid dimension changes accordingly.

Fixed parameters:

Number of Givoli-Neta angles: 5;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Domain size: from $4 \times 4 \times 4$ to $10 \times 10 \times 10$;

Grid dimension: varies accordingly.

Similarly to Section 6.1.2, Figure 20 shows that by increasing the size of the computational domain one can make the duration of the interval of stable performance longer, but the advantage of using a higher order ABC vanishes.

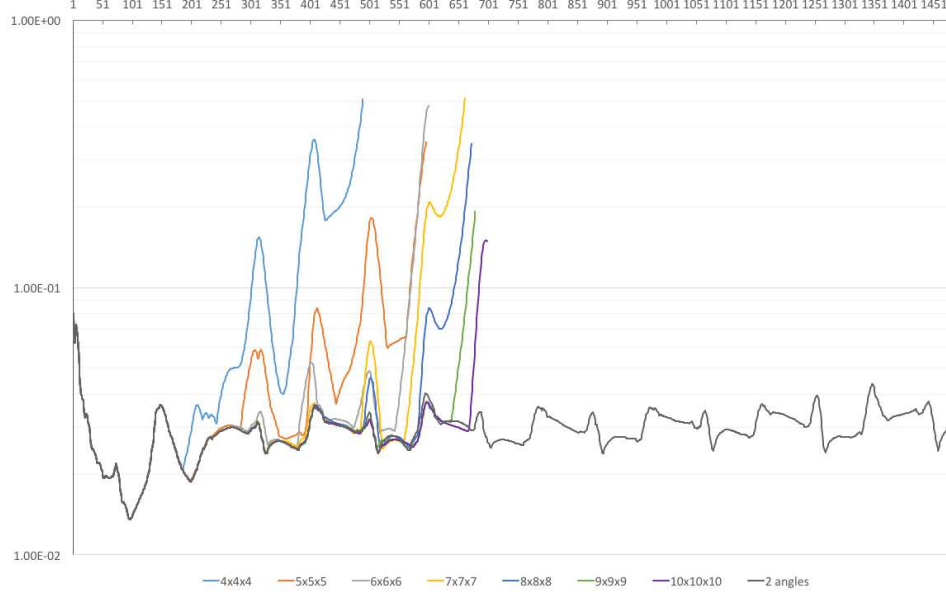


Figure 20: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the size of the computational domain. First order accurate approximation of the ABCs in time.

6.2.3 Choice of angles — three angles.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Number of Givoli-Neta angles: 3;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Values of Givoli-Neta angles ϕ_n .

Figure 21 indicates that in this case the difference between various sets of three Givoli-Neta angles is insignificant.

6.2.4 Choice of angles — five angles.

Fixed parameters:

Domain size: $6 \times 6 \times 6$;

Grid dimension: $61 \times 61 \times 61$;

Number of Givoli-Neta angles: 5;

Grid size in space: 0.1;

Grid size in time: 0.0333;

Variable parameters:

Values of Givoli-Neta angles ϕ_n .

Figure 22 shows that as in the case of three angles, the difference between various choices are insignificant. The best choice corresponds to evenly spaced angles between 0° and 90° .

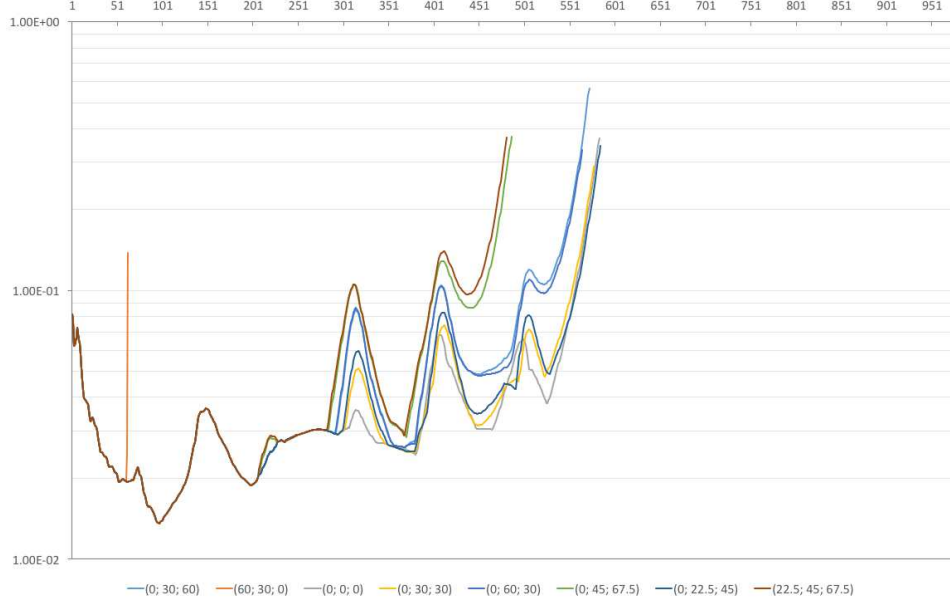


Figure 21: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles. First order accurate approximation of the ABCs in time.

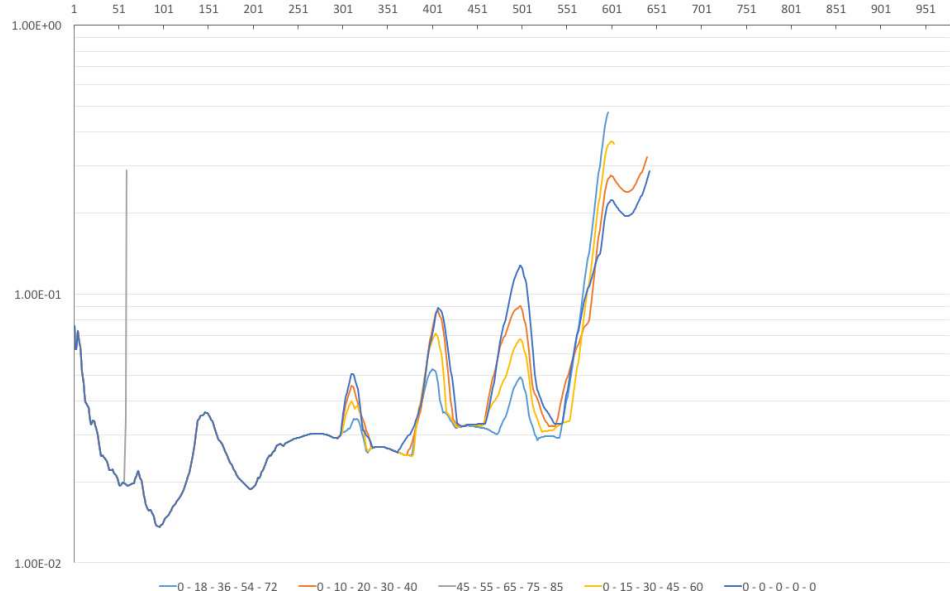


Figure 22: Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles. First order accurate approximation of the ABCs in time.

6.2.5 Intermediate conclusion. Changing the approximation order does not noticeably affect the performance of the Givoli-Neta ABCs.

6.3 Combined implementation of lacunae and Givoli-Neta ABCs

For the combined numerical implementation of the Givoli-Neta ABCs with lacunae we are using the computational setup that has previously been determined optimal. The domain size is $6 \times 6 \times 6$, the grid dimension is $61 \times 61 \times 61$, and the time step is 0.0333. The speed of light has been normalized to 1. A compactly supported source of electromagnetic radiation is located near the center of the domain. It takes 89 time steps for a wave from this source to reach the outer boundary.

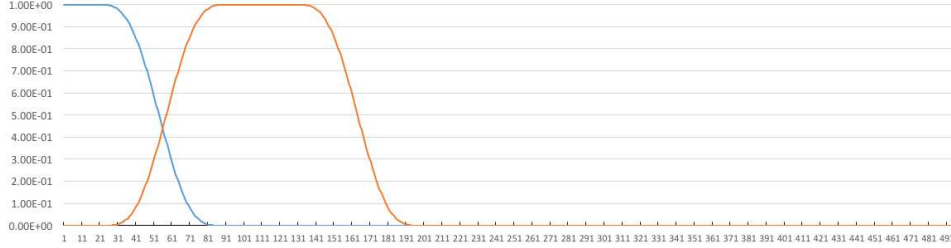


Figure 23: The functions $\Theta(t)$ that help partition the sources.

To implement the lacunae-based time marching, one needs to partition the source of radiation into elements of finite duration. This is done via multiplication by the specially chosen functions $\Theta(t)$. For the first two partition elements, those functions are shown in Figure 23. At the outer boundary, each partial sub-problem is terminated by the Givoli-Neta ABC at the face $x = -3$ and by the unsplit PML at the remaining five faces of the computational domain (cube). The Givoli-Neta ABC is built on the six angles evenly distributed between 0° and 90° . The experimentally determined interval of its stable performance is approximately 650 time steps. This interval provides an upper bound for the integration time of each partial sub-problem. Accordingly, the latter time is chosen approximately equal to 500 time steps. It is fairly short, and necessitates choosing short partition elements. In the continuous setting, it would still be sufficient though for all the waves generated by the corresponding partial source to leave the computational domain. In the discrete setting, however, the “bottom” of the lacuna is not exactly zero. The residual solution there eventually reduces the accuracy of the lacunae-based treatment of the artificial outer boundary. The nature of this residual solution is actually the plain approximation error of the finite difference scheme. It would, of course, go down if the discretization grid were refined. As, however, we have seen previously, refining the discretization grid immediately decreases the interval of stable performance of the Givoli-Neta ABC. Therefore, grid refinement cannot be used as a means for improving the overall accuracy.

In Figure 24, we are showing error profiles for the combined implementation of a high order Givoli-Neta ABC and lacunae. For comparison, we are also including the cases of low order ABCs. One can clearly see that low order ABCs (one and two angles — blue and green curves) are stable. A high order ABC (six angles — purple curve) develops an instability at about 650 time steps. At the same time, throughout the interval of its stable performance, it provides a better accuracy than low order ABCs, which is expected. The lacunae-based time marching can stabilize the long-time behavior of this ABC — red curve in Figure 24. However, because of the residual error inside the lacuna, the accuracy of the stabilized solution is not better than that provided by low order ABCs. This residual error could have been efficiently reduced by grid refinement. It is not possible though because grid refinement would also cause the interval of stable performance of the ABC to shrink.

Figures 25 and 26 present similar error profiles, but for the field components H_x and H_z , respectively. The behavior is qualitatively the same as that in Figure 24. For the H_x component

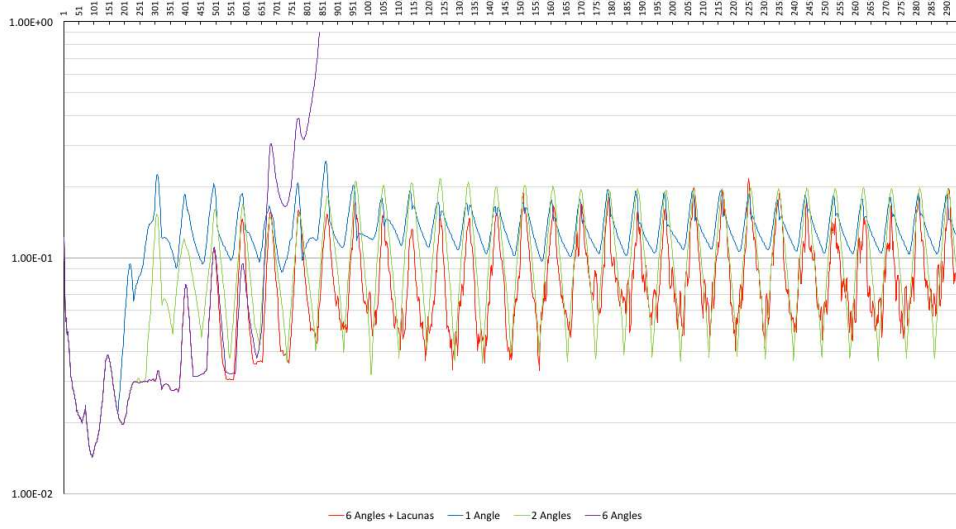


Figure 24: Relative error for the E_y component of the electromagnetic field.

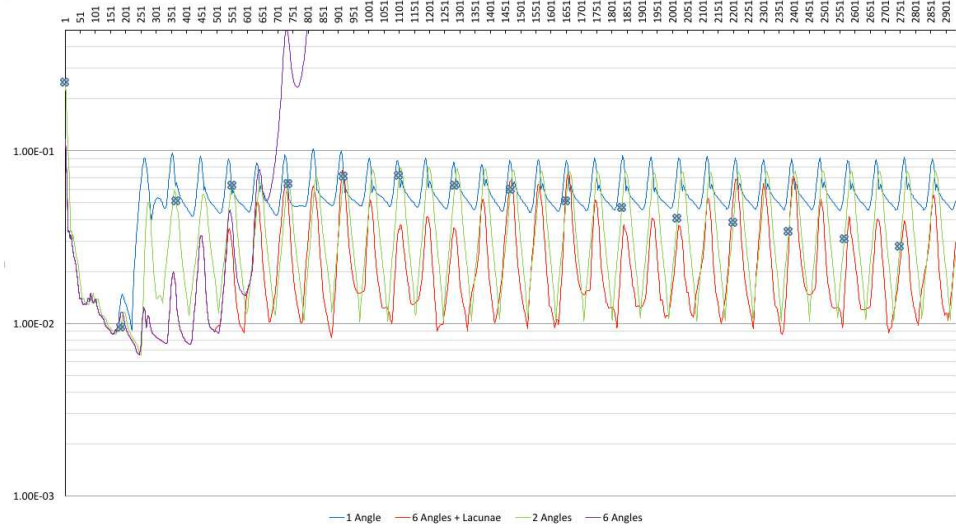


Figure 25: Relative error for the H_x component of the electromagnetic field.

(Figure 25) the lacunae-based treatment provides a moderate accuracy gain over the entire long time interval. For the H_z component (Figure 26), the long-time accuracy of the stabilized solution with a high order ABC is in-between the accuracy given by the first and second order ABCs.

6.4 Conclusions on Givoli-Neta ABCs

High order local ABCs by Givoli and Neta [73] are capable of providing a better accuracy compared to that obtained by low order local ABCs. However, the interval of their stable performance appears relatively short, and we have not been able to extend it by varying either the parameters of the ABCs or those of the computational setting. Moreover, this interval depends on the grid size and decreases as the grid is refined, which is typical for numerical instabilities. This short interval of stable performance necessitates choosing short partition elements for lacunae-based time marching.

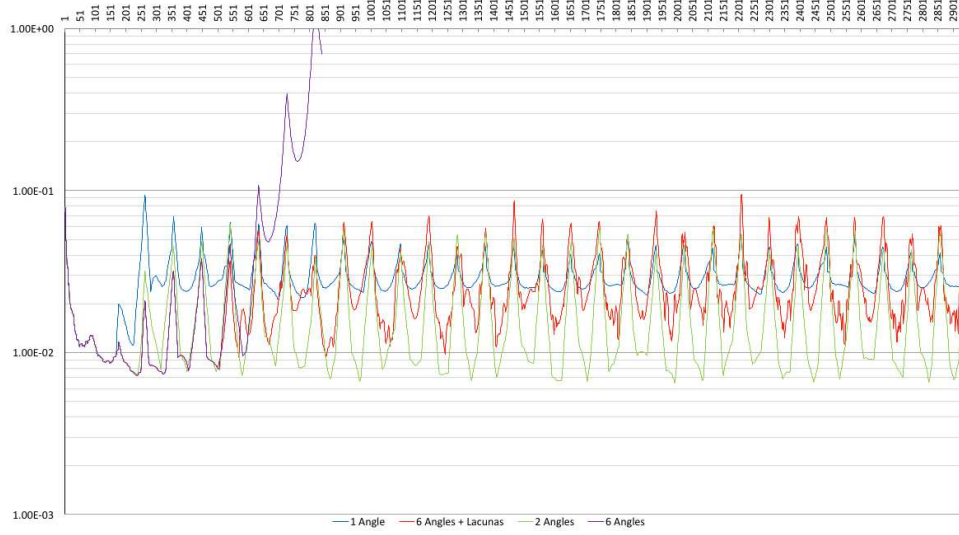


Figure 26: Relative error for the H_z component of the electromagnetic field.

The latter, in turn, leave residual error at the bottom of the numerical lacunae. As a result, the accuracy of the solution that is stabilized by lacunae over long integration times decreases compared to the accuracy provided by the high order ABC on its own. This deterioration of accuracy could have been easily corrected by refining the grid, because the residual error at the bottom of the lacunae is the conventional discretization error of the scheme. One, however, cannot adopt this approach because the grid refinement exacerbates the instability of the ABC. Altogether, we must conclude that the instability of the high order Givoli-Neta ABCs is too strong so that this treatment of the outer boundary cannot derive any noticeable benefits for the lacunae-based time marching.

7 Complete radiation boundary conditions

The next development in the area of local high order ABCs is represented by the methodology of [53] which eventually led to the radiation boundary conditions based on complete wave expansions [54, 55].

7.1 Overview

Hagstrom-Warburton non-reflecting boundary condition (NRBC) is a high order local artificial boundary condition based on complete wave expansions [53–55, 74, 75]. Our effort during the month of March 2016 has focused on implementing this type of NRBC into our in-house software package. The boundary conditions of [53–55, 74, 75] can be thought of as a reformulation of the high order Higdon conditions [66] (see Section 5.2) that introduces auxiliary variables to replace a high order differential operator of type (81) by a recursive system of first order differential operators. The high order absorbing boundary conditions proposed in [74] are given by:

$$\begin{aligned}
 a_0 \partial_t u + c \partial_x u &= a_0 \partial_t \phi_1, \\
 a_j \partial_t \phi_j + c \partial_x \phi_j &= a_j \partial_t \phi_{j+1} - c \partial_x \phi_{j+1}, \quad j = 1, \dots, P, \\
 \sigma_j \phi_{P+j} + \partial_x \phi_{P+j} &= \sigma_j \phi_{P+j+1} - \partial_x \phi_{P+j+1}, \quad j = 1, \dots, E, \\
 \phi_0 &\equiv u, \quad \phi_{P+E+1} = 0.
 \end{aligned} \tag{84}$$

Boundary conditions (84) are set at the $x = x_E$ boundary of a two-dimensional wave-guide. The auxiliary variables ϕ_j , $j = 1, \dots, P$, represent the propagating modes, and the variables ϕ_j , $j = P + 2, \dots, E$, represent the evanescent modes of the overall solution. The quantities a_j and σ_j are parameters of the ABC. Since the auxiliary functions ϕ_j , $j = 1, \dots, P + E + 1$, are required to satisfy the homogeneous scalar wave equation near the boundary, the derivatives with respect to x can be eliminated in (84). Two additional equations are needed at the “seam” between the propagating and evanescent modes. The resulting (P, E) -order ABC takes the following form:

$$\begin{aligned}
a_0 \partial_t u + c \partial_x u &= a_0 \partial_t \phi_1, \\
l_{j,j-1} \partial_t^2 \phi_{j-1} + l_{j,j} \partial_t^2 \phi_j + l_{j,j+1} \partial_t^2 \phi_{j+1} \\
&= c^2 (m_{j,j-1} \Delta \phi_{j-1} + m_{j,j} \Delta \phi_j + m_{j,j+1} \Delta \phi_{j+1}), \quad j = 1, \dots, P, \\
l_{P+1,P} \partial_t^2 \phi_P + l_{P+1,\psi} \partial_t^2 \psi + l_{P+1,P+1} \partial_t^2 \phi_{P+1} &= c^2 (m_{P+1,P} \Delta \phi_P + m_{P+1,P+1} \Delta \phi_{P+1}), \\
n_{1,1} \partial_t^2 \phi_{P+1} + n_{1,2} \partial_t^2 \phi_{P+2} - c \partial_t \psi &= c^2 (v_{1,1} \Delta \phi_{P+1} + v_{1,2} \Delta \phi_{P+2}) + c^2 (s_{1,1} \phi_{P+1} + s_{1,2} \phi_{P+2}), \quad (85) \\
n_{j,j-1} \partial_t^2 \phi_{P+j-1} + n_{j,j} \partial_t^2 \phi_{P+j} + n_{j,j+1} \partial_t^2 \phi_{P+j+1} \\
&= c^2 (v_{j,j-1} \Delta \phi_{P+j-1} + v_{j,j} \Delta \phi_{P+j} + v_{j,j+1} \Delta \phi_{P+j+1}) \\
&\quad + c^2 (s_{j,j-1} \phi_{P+j-1} + s_{j,j} \phi_{P+j} + s_{j,j+1} \phi_{P+j+1}), \quad j = 2, \dots, E, \\
\phi_0 &\equiv u, \quad \phi_{P+E+1} = 0,
\end{aligned}$$

where Δ is the Laplace operator at the boundary (i.e., on the surface, as opposed to in the volume) and ψ is the seam function between the propagating mode conditions and evanescent mode conditions. The coefficients $l_{j,j}$, $m_{j,j}$, $n_{j,j}$, $s_{j,j}$, and $v_{j,j}$ are calculated using the parameters a_j , σ_j , see [74]. Boundary conditions (85) were implemented for a 2D wave-guide in [74].

7.2 FDTD implementation

Implementation for a waveguide, see [74], does not require any special treatment for the corners of the computational domain, which is known to be one of the most delicate issues related to local high order ABCs. The current goal of our work is to have the Hagstrom-Warburton NRBCs implemented for an open 3D domain rather than for a waveguide, and subsequently combine the resulting implementation with the lacunae-based time marching. An open domain shaped as a cube has both edges and vertices that may require special treatment. We, however, leave it for the future study and in the meantime implement the NRBC [74] on one face of the cube only, while terminating the other five faces with the exact boundary conditions that are available for the test solution that we are using. This way we can have an open domain in 3D and at the same time avoid the need to handle the edges and vertices in a special way.

In our implementation we set the complete radiation boundary conditions at the boundary $x = x_E$ of the cube for the component $E_y(x_E, y, z)$ of the electromagnetic field. To discretize the boundary condition, we employ the following second-order finite difference approximations:

$$\begin{aligned}
\Delta \phi_j^n|_{x_E, y, z} &\simeq \frac{\phi_j^n|_{x_E, y+1, z} - 2 \phi_j^n|_{x_E, y, z} + \phi_j^n|_{x_E, y-1, z}}{h_y^2} \\
&\quad + \frac{\phi_j^n|_{x_E, y, z+1} - 2 \phi_j^n|_{x_E, y, z} + \phi_j^n|_{x_E, y, z-1}}{h_z^2}, \quad (86)
\end{aligned}$$

$$\partial_t \phi_j^n|_{x_E, y, z} = \frac{3 \phi_j^{n+1}|_{x_E, y, z} - 4 \phi_j^n|_{x_E, y, z} + \phi_j^{n-1}|_{x_E, y, z}}{2h_t}, \quad (87)$$

$$\partial_t^2 \phi_j^n|_{x_E, y, z} = \frac{\phi_j^{n+1}|_{x_E, y, z} - 2 \phi_j^n|_{x_E, y, z} + \phi_j^{n-1}|_{x_E, y, z}}{h_t^2}, \quad (88)$$

$$\partial_x u^n|_{x_E, y, z} = \frac{3 u^n|_{x_E, y, z} - 4 u^n|_{x_{E-1}, y, z} + u^n|_{x_{E-2}, y, z}}{2h_x}, \quad (89)$$

In formulae (86)–(89), h_x , h_y , h_z , and h_t are the grid sizes in each coordinate direction and in time, respectively. Using the vectors:

$$Z^{n+1} = \begin{pmatrix} u^{n+1} \\ \phi_1^{n+1} \\ \vdots \\ \phi_P^{n+1} \\ \phi_{P+1}^{n+1} \\ \vdots \\ \phi_{P+E}^{n+1} \\ \psi^{n+1} \end{pmatrix}, \quad Z^n = \begin{pmatrix} u^n \\ \phi_1^n \\ \vdots \\ \phi_P^n \\ \phi_{P+1}^n \\ \vdots \\ \phi_{P+E}^n \\ \psi^n \end{pmatrix}, \quad Z^{n-1} = \begin{pmatrix} u^{n-1} \\ \phi_1^{n-1} \\ \vdots \\ \phi_P^{n-1} \\ \phi_{P+1}^{n-1} \\ \vdots \\ \phi_{P+E}^{n-1} \\ \psi^{n-1} \end{pmatrix},$$

we recast system (85) in the vector form:

$$AZ^{n+1} + BZ^n + CZ^{n-1} = -2ch_t \partial_x^n u \mathbf{e}_1 + D\Delta(Z^n). \quad (90)$$

From (90), we obtain a formula for the time marching algorithm:

$$Z^{n+1} = A^{-1}(D\Delta(Z^n) - BZ^n - CZ^{n-1} - 2ch_t \partial_x^n u \mathbf{e}_1).$$

7.3 Boundary conditions for auxiliary functions

The discrete Laplacian (86) cannot be applied at the edges and corners of the face. We have tried several alternative strategies. A second order accurate one-sided difference approximation leads to an instability of the boundary conditions. A first order accurate one-sided difference approximation provides a stable algorithm but also causes a substantial overall loss of accuracy. We recall, however, that the auxiliary functions ϕ_j satisfy the homogeneous wave equation along the face. Thus, we can use one of the previously studied low order boundary conditions for those auxiliary functions. We have implemented the Sommerfeld boundary condition (80) first. Figure 27 shows the maximum relative error for the field component $E_y(x, y, z, t)$.

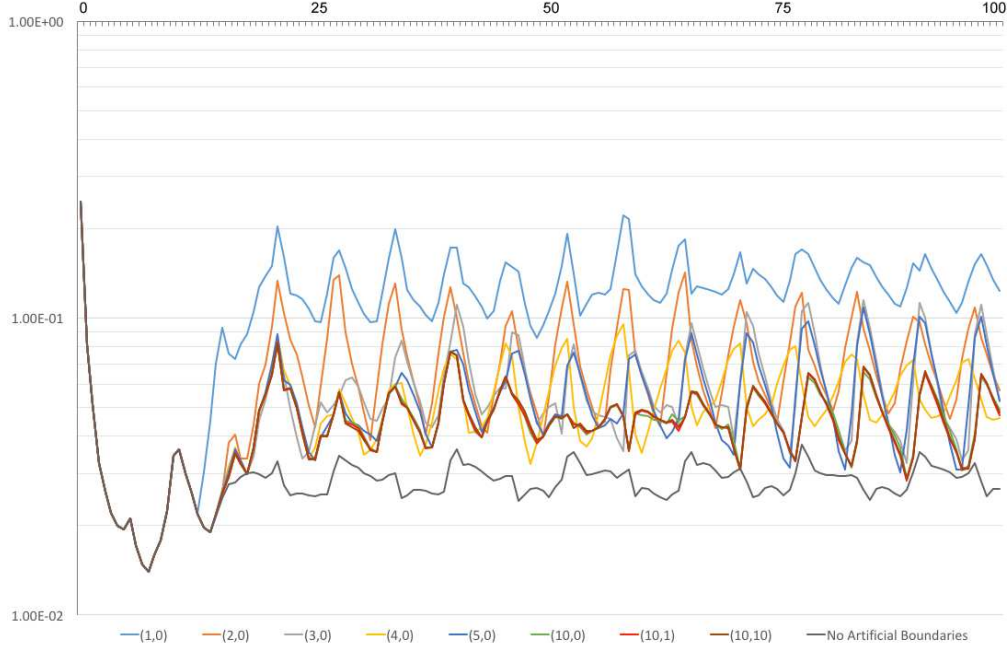


Figure 27: Maximum relative error of the field component $E_y(x, y, z, t)$. Sommerfeld ABC (80) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E) .

The computational domain is $[-3.25, 3.25] \times [-3.25, 3.25] \times [-3.25, 3.25]$, and the source of electromagnetic waves is located around the center of the domain. We use a uniform Cartesian grid with $60 \times 60 \times 60$ nodes. As one can see from Figure 27, the error generated by the Hagstrom-Warburton NRBC of order (P, E) decreases as the number of angles P in the ABC increases. The best result is achieved when we use ten angles. The corresponding accuracy is still not as good as that provided by the exact boundary conditions on all six faces (grey curve in Figure 27). The reason that the accuracy does not improve any further as P increases is that the Sommerfeld ABC generates substantial spurious reflections for auxiliary functions and hence it appears not possible to get a better accuracy by using more auxiliary functions. Of course, the overall performance of the complete radiation boundary conditions is affected by the values of $0 < a_j \leq 1$ and $\sigma_j > 0$. As of yet, we have only used the simplest choice of these parameters: $a_j = 1$ and $\sigma_j = 1$. It is apparently not optimal, and the recipes proposed in [74] may, in principle, help achieve better results by judiciously choosing the parameters of the NRBC.

Beyond the Sommerfeld ABC of section 5.1, we have successfully implemented the second-order Higdon ABC with two angles for the auxiliary functions (see Section 5.2):

$$\left(\partial_t + \frac{\cos \alpha}{c} \partial_x \right) \left(\partial_t + \frac{\cos \beta}{c} \partial_x \right) \phi_j = 0, \quad (91)$$

where α and β are the two angles to be chosen. Figure 28 shows the maximum relative error for $E_y(x, y, z, t)$ in this case.

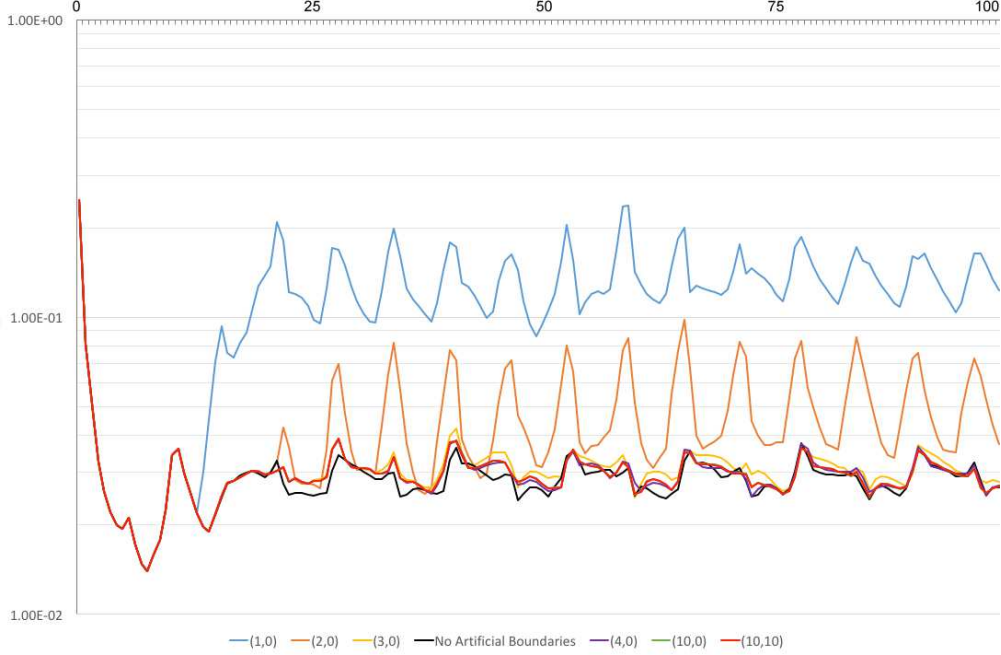


Figure 28: Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E) .

To enable a direct comparison of the results, we use the same computational domain and grid as those that correspond to the computations presented in Figure 27. The black curve in Figure 28 represents the error for the case of exact boundary conditions on all six faces. It is the smallest error that can be achieved on a given grid. Note that the error of the Hagstrom-Warburton NRBC is practically indistinguishable from this black curve when only four auxiliary functions are used.

Moreover, when Higdon ABC (91) is used for the auxiliary functions, one can achieve the design second-order rate of grid convergence. Figure 29 compares the results obtained on the previously chosen grid $60 \times 60 \times 60$ and the twice as fine grid of dimension $120 \times 120 \times 120$ (while the size of the computational domain remains the same). The second-order grid convergence is achieved by using the Hagstrom-Warburton NRBC of order (5,0) and Higdon ABC for auxiliary functions. For simplicity, both angles, α and β , in the Higdon ABC (91) are equal to zero.

7.4 Long-time instability and lacunae-based time marching

Even though the Hagstrom-Warburton boundary conditions have performed well in the computations that correspond to Figures 28 and 29, they still appear prone to error growth in long-time simulations. Furthermore, the Higdon ABC (91) that is used for auxiliary functions in our experiments may also cause long-time instabilities. In Figure 30, the error growth is shown for two grids. The interval of stability depends neither on the number of auxiliary functions that we use in the Hagstrom-Warburton NRBC nor on the grid dimension. However, this interval of stability appears sufficiently long for the lacunae method to be efficiently applied. This method suppresses the long-time instabilities. As it partitions the original problem into a sequence of auxiliary sub-problems for which no analytical solution is available, we use the unsplit PML [72] instead of the exact boundary conditions to terminate the computational domain at those five faces where we

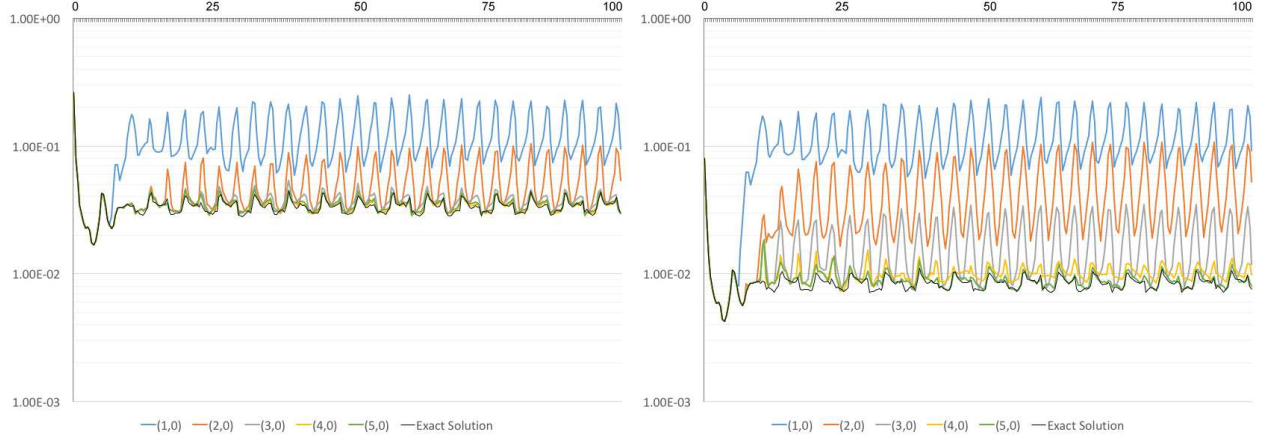


Figure 29: Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order $(5, 0)$. The solution in the left panel is computed on the grid with $60 \times 60 \times 60$ and the solution in the right panel is computed on the grid with $120 \times 120 \times 120$ nodes.

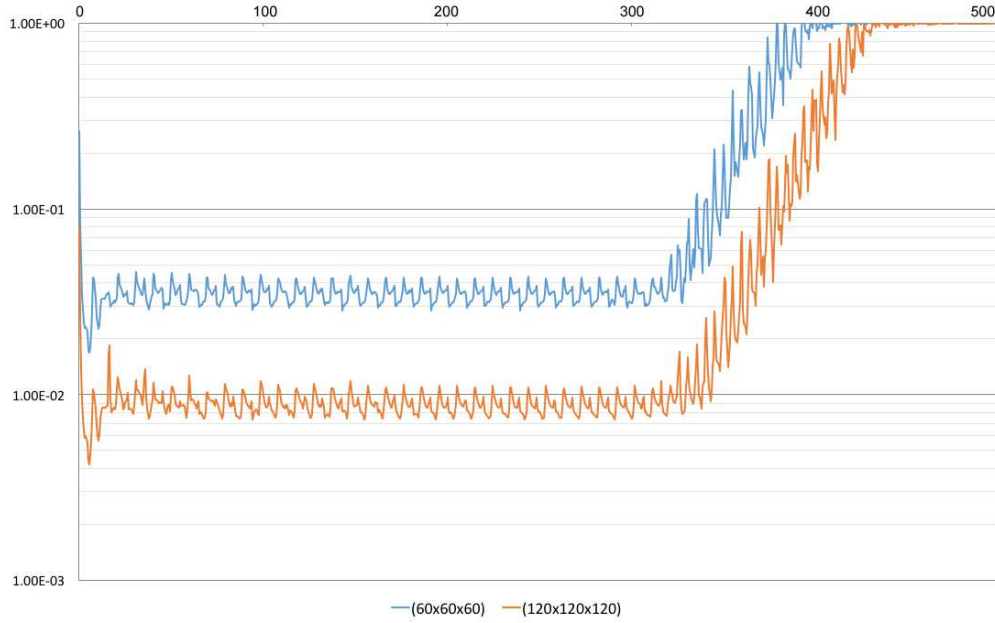


Figure 30: Long-time growth of maximum relative error for the field component $E_y(x, y, z, t).t$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E) . The blue curve corresponds to the solution computed on the grid with $60 \times 60 \times 60$ and the orange curve corresponds to the solution computed on the grid with $120 \times 120 \times 120$ nodes.

do not set the complete radiation boundary condition.⁶ Figure 31 presents the initial results of implementation of the Hagstrom-Warburton complete radiation boundary condition on one face of the cube, supplemented by the unsplit PML on the remaining five faces and lacunae-based time marching that mitigates the long-time error growth. One can see that the method of lacunae is

⁶The interval of stable performance for the PML is at least as long as that for the complete radiation boundary condition.

indeed capable of restoring the non-deteriorating performance over long time intervals.

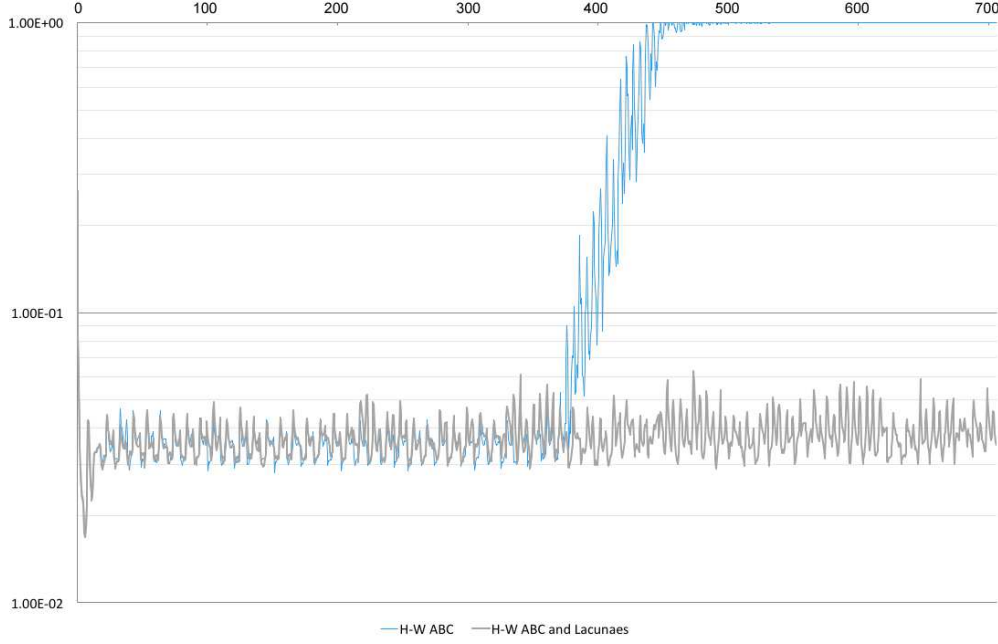


Figure 31: Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (5, 0). Long-time stability is enabled by the method of lacunae.

These initial positive results warrant a more detailed investigation of the joint implementation of the complete radiation boundary conditions along with the lacunae-based time marching.

Let us first note that even before the lacunae-based time marching is implemented, it is not immediately obvious how to combine a high order local ABC and a PML on two neighboring faces of the cube. To avoid the potential difficulties related to handling a common edge between the ABC and the PML, we concurrently solve two different problems. Figure 32 schematically depicts two layers of the electric field for the first problem (red) and the second problem (blue). The second problem is terminated by the PML on all six faces. In the first problem, we set the Hagstrom-Warburton NRBC on one face and take the boundary data from the second problem on the other five faces. For both problems, we use the same source term that drives the Maxwell equations. As long as it remains the original source (as opposed to the auxiliary sources needed for some versions of lacunae implementation, see Section 7.7), there is no feedback from the first problem to the second problem.

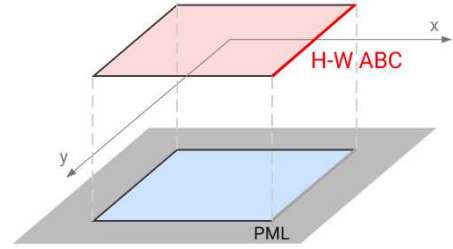


Figure 32: Schematic for two problems that allow us to combine the Hagstrom-Warburton NRBC and a PML on neighboring faces.

Similarly to the case of Figure 30 where the complete radiation boundary condition on one face of the cube was combined with exact boundary conditions on the remaining faces, the implementation that replaces the exact boundary conditions with an unsplit PML also appears prone to error growth in long-time simulations. We found that the error and the total time of stable calculations depends on the domain size. In doing so, we have determined that the key factor that affects the

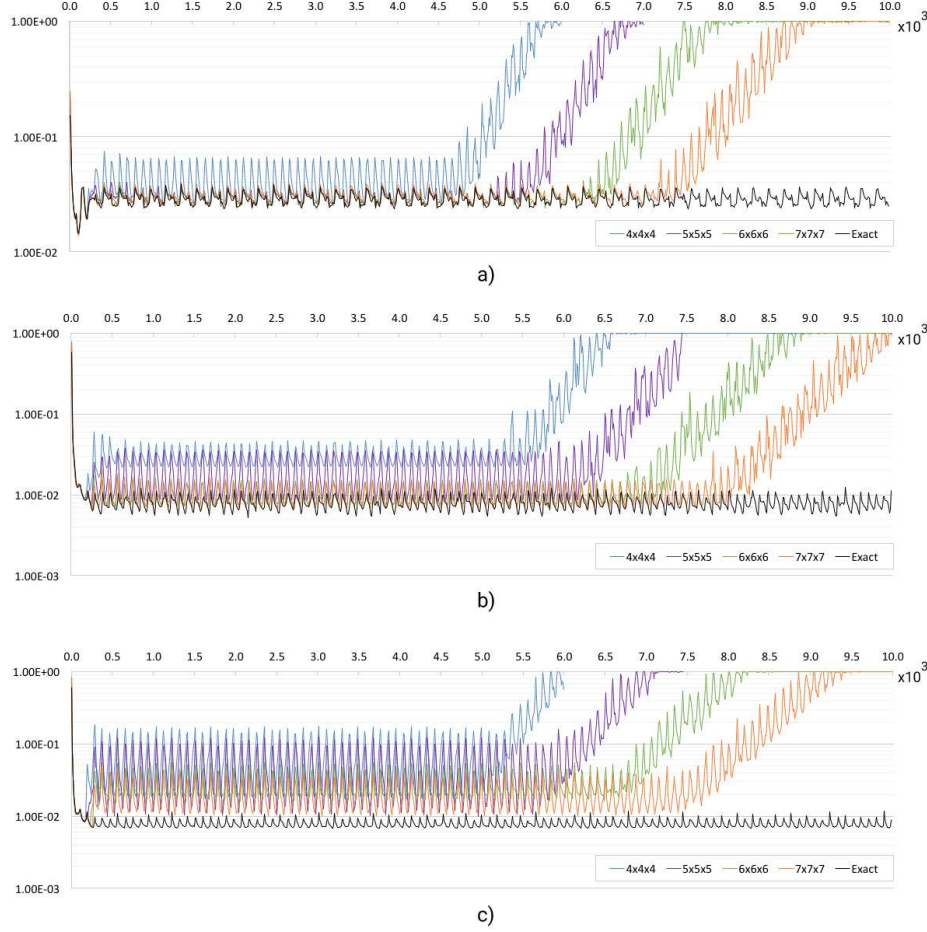


Figure 33: The performance of Hagstrom-Warburton NRBC on one face of the cube combined with the unsplit PML of [72] on the other five faces as it depends on the size of the computational domain. Curves of different color represent maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c).

maximum relative error for the field components $E_y(x, y, z, t)$, $H_x(x, y, z, t)$, and $H_z(x, y, z, t)$, as well as the interval of stable performance, is the size of the computational domain or, equivalently, the distance from the source of the waves to the outer boundary. The corresponding results are presented in Figure 33. We use five auxiliary functions for the Hagstrom-Warburton NRBC. The spatial and temporal grid sizes are 0.1 and 0.0333, respectively. The source is compactly supported near the center of the computational domain. The thickness of the PML in the second problem (see Figure 32) is 15 grid cells. The computational domain size varies between $4 \times 4 \times 4$ and $7 \times 7 \times 7$. Figure 33 shows that the boundary condition works longer and the error gets smaller as the boundaries move further away from the source.

7.5 The role of unsplit PML

Given that the long-time deterioration of solution shown in Figure 33 takes place when the computational domain is terminated by the combination of a high order ABC and a PML, we need to identify a particular component of the algorithm that is responsible for the observed error growth.

It turns out that the error generated by the unsplit PML alone is rather small. Figure 34 shows the maximum relative error for the field components $E_y(x, y, z, t)$, $H_x(x, y, z, t)$, and $H_z(x, y, z, t)$ when the unsplit PML with thickness of 15 grid cells is used to terminate the computational domain everywhere. The domain size is $6.5 \times 6.5 \times 6.5$ and the spatial and temporal grid sizes are 0.1 and 0.0333, as before. We see that the PML generates very low spurious reflections, as its error curves basically coincide with those that correspond to the exact boundary conditions (so that all of the error can be attributed to discretization rather than to the outer boundary). Moreover, the performance of this unsplit PML in this particular setting (source, domain, etc.) appears to remain stable over long simulation times. Figure 35 corroborates that by showing the maximum relative error of $E_y(x, y, z, t)$. Comparing Figures 34 and 35 with Figure 33, we can unambiguously attribute the long-time deterioration of accuracy in Figure 33 to the use of the Hagstrom-Warburton NRBC on one face of the cube.

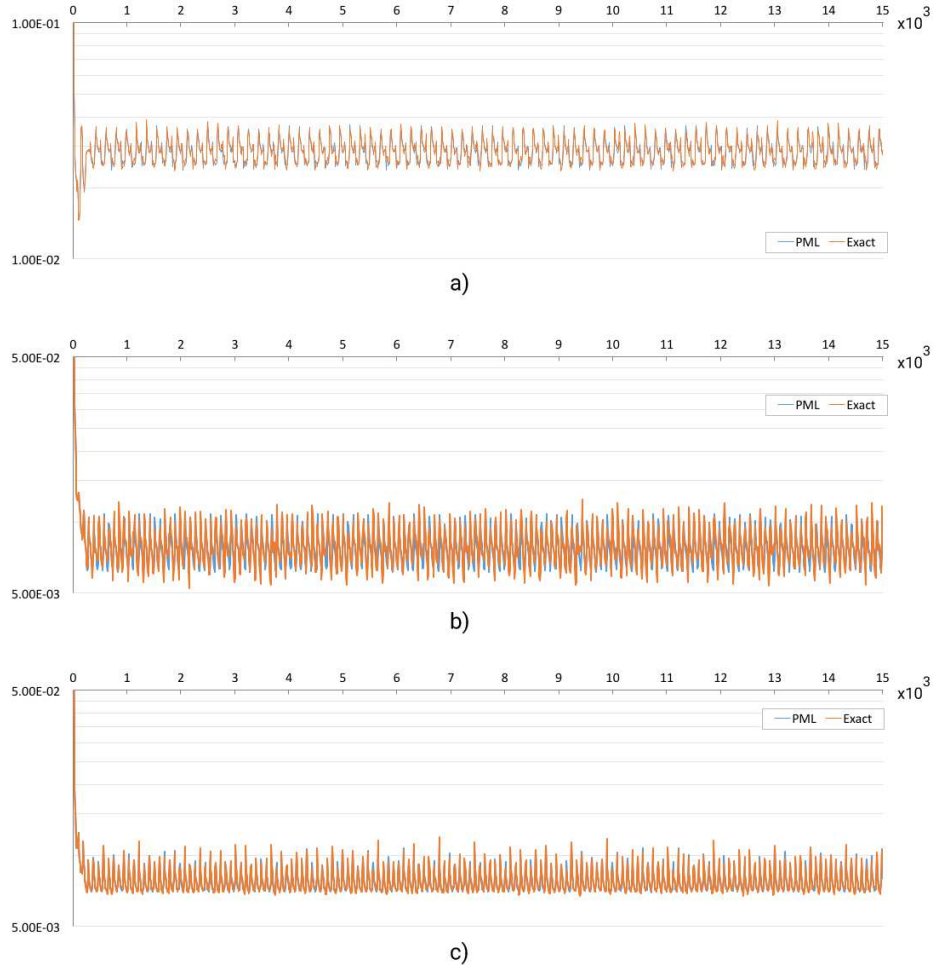


Figure 34: Maximum relative error for $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) components. Orange curve – exact boundary condition, blue curve – unsplit PML [72].

7.6 Application of lacunae with original divergence-free currents

To suppress the long-time error growth demonstrated by the algorithm with Hagstrom-Warburton NRBC, we employ the lacunae-based time marching. Its implementation involves having the origi-

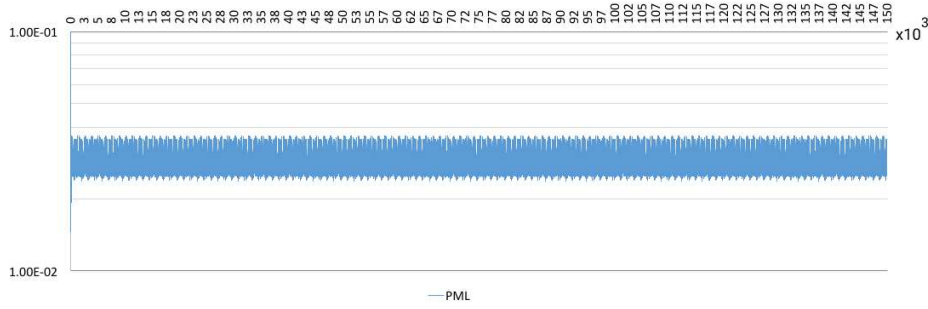


Figure 35: Maximum relative error for $E_y(x, y, z, t)$ in long-time simulation.

nal (main) problem split into a series of sub-problems driven by partial sources (currents) of finite duration. The latter are obtained with the help of the following smooth partition of unity:

$$\forall t \leq 0 : 1 = \sum_{i=0}^{\infty} \zeta(t - (i-1)T_1), \quad \text{supp}\zeta(t) = [T_1 - \Delta T, 2T_1]. \quad (92)$$

In formula (92), $T_1 \stackrel{\text{def}}{=} T_0 - \Delta T$, where T_0 is the size of each partition element, and

$$\Delta T = \sigma T_0, \quad (93)$$

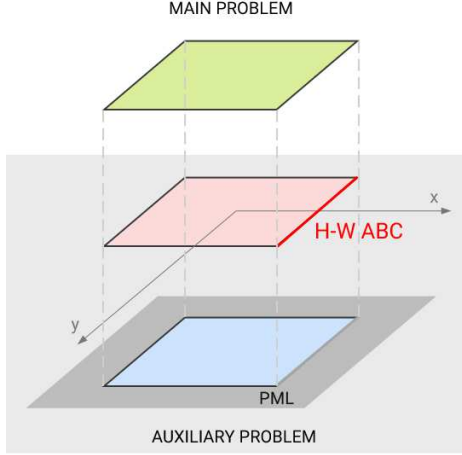


Figure 36: Implementation of the Hagstrom-Warburton NRBC with lacunae and the original divergence-free source. The main problem (green) is partitioned into sub-problems. Each of the latter consists of the first problem (red) with Hagstrom-Warburton NRBC on the one face of the domain and the second problem (blue) terminated by the PML.

is the width of the overlapping interval. Multiplying the electric and magnetic current densities j and j^M of the original source by the functions $\zeta(\cdot)$ of (92), we transform the continuously operating currents into the sums of partial currents that each operate only during a fixed finite time interval. These partial currents drive the sub-problems of the main problem. To enable the use of the complete radiation BC along with the PML on neighboring faces, each of the sub-problems, in its own right, consists of the first and second problems according to the schematic in Figure 36 (see also Figure 32). As we use the partition of unity (92) to split the original divergence-free currents into a sequence of partial currents, we avoid the accumulation of electric and magnetic charges in the resulting sub-problems. Figure 37 shows the evolution of the maximum relative error for the electric field component E_y of the main problem. We use five auxiliary functions for the Hagstrom-Warburton NRBC; the domain size is $6.5 \times 6.5 \times 6.5$, and the spatial and temporal grid sizes are 0.1 and 0.0333, respectively. The size of each partition element is $T_0 = 40$, and $\sigma = 0.8$, see formula (93) is used to define the overlapping interval. The method of lacunae eliminates the long-time error growth. The simulation presented in Figure 37 is similar to that presented in Figure 31.

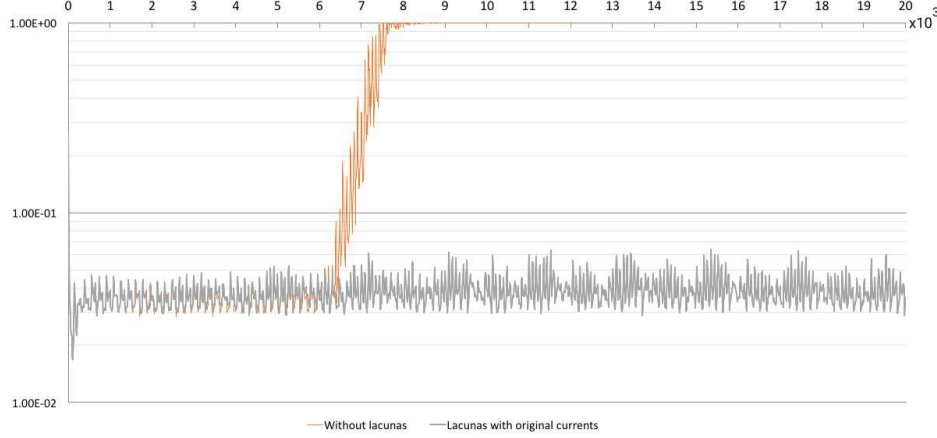


Figure 37: Maximum relative error for the electric field component $E_y(x, y, z, t)$: grey curve — computation with lacunae and partition of the original currents; orange curve — computation with no use of lacunae. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.

7.7 Application of lacunae with auxiliary currents

The formulation of the main problem may be more general than merely having a compactly supported source of electromagnetic radiation, or equivalently, the right-hand side to Maxwell's equation. In this case, the foregoing straightforward implementation of lacunae-based time marching that employs direct partitioning of the original currents in time may not be possible. Instead, we first decompose the overall problem into the interior problem formulated on a bounded computational domain and auxiliary problem formulated on the entire space. In doing so, the interior problem may even be non-Huygens', but the auxiliary problem still consists of Maxwell's equation driven by compactly supported auxiliary currents. Then, any treatment of the artificial outer boundary (including lacunae) applies to the auxiliary problem, which is Huygens', and its solution provides the boundary closure to the interior problem. At the same time, the auxiliary currents are derived based on the solution to the interior problem. In other words, the two problems are coupled with one another. This approach was first introduced in [65] (see Section 5 of the paper) and successfully tested along with an unsplit PML in the capacity of an outer boundary closure.

In our implementation of the algorithm with auxiliary currents, we do not change the original source. It still remains divergence-free. This does not present a considerable loss of generality, as we are primarily interested in testing the component that involves the modified currents. The key steps of the algorithm are schematically shown in Figure 38. They are:

- (a) The original source is used to update interior points of the main problem.
- (b) The auxiliary currents are obtained by solving a specially formulated Neumann problem for the Laplace equation. They are compactly supported within the interior domain.
- (c) The auxiliary currents are partitioned in time according to (92).
- (d) Each second problem for all auxiliary sub-problems is updated. The boundary values are then used for the corresponding first problem on five faces of the cube, and the Hagstrom-Warburton NRBC is used to update the boundary values on the sixth face of the cube.

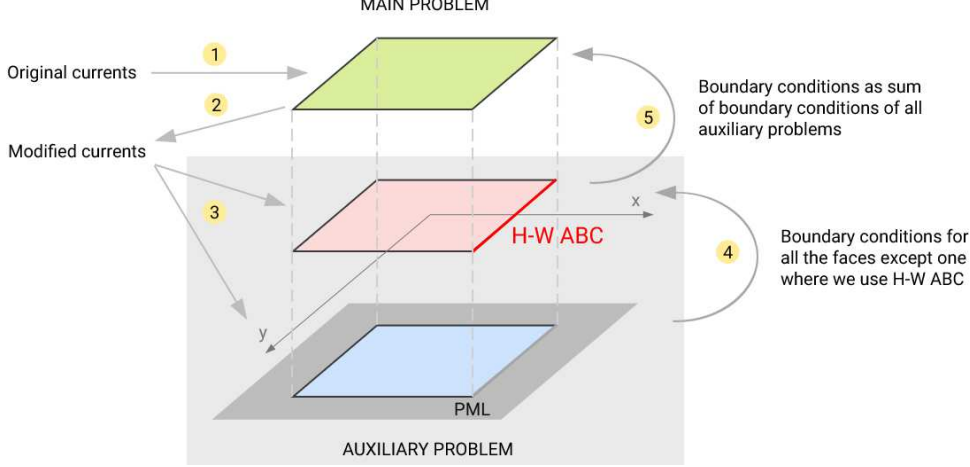


Figure 38: Schematic for the implementation of Hagstrom-Warburton NRBC with lacunae and auxiliary currents.

- (e) The boundary values of the main problem are updated by summing up the boundary values of all the auxiliary problems that contribute at a given time.

Figure 39 compares the maximum relative error for the field components $E_y(x, y, z, t)$, $H_x(x, y, z, t)$, and $H_z(x, y, z, t)$ in the case of lacunae-based time marching with original currents against that in the case of lacunae-based time marching with auxiliary currents. In every case, the thickness of PML is 15 cells, the domain size is $6.5 \times 6.5 \times 6.5$, and the spatial and temporal grid sizes are 0.1 and 0.0333, respectively. The length of each partition element is $T_0 = 40$ and we use $\sigma = 0.8$ for the overlapping interval, see formulae (92) and (93). We see that while in the case of original currents the lacunae-based time marching demonstrates a perfectly stable performance, in the case of auxiliary currents the performance still deteriorates, although much later in time (order of magnitude) compared to the computations with no lacunae at all (Figures 33 and 37).

The unstable behavior shown in Figure 39 is clearly triggered by the combination of auxiliary currents used for lacunae-based time marching (as opposed to the original currents) and the Hagstrom-Warburton NRBC (because if only the PML is used along with lacunae at the outer boundary, the performance always remains stable). To understand the mechanism of the long-time deterioration in Figure 39 and, perhaps, propose a strategy for mitigating it, we first need to see how the interval of stable performance depends on the various parameters of the algorithm.

The total time of computation for each partial sub-problem depends on the size of the corresponding partition element in (92) and the size of the computational domain. Clearly, we must continue to update each sub-problem until all the waves generated by its partial current leave the computational domain. There is a parameter T_g in our implementation that allows us to increase the total simulation time for each sub-problem by T_g steps beyond the required minimum, i.e., beyond the moment when the waves leave the domain. Normally, we take $T_g = 10$ and this value works perfectly well in case of Hagstrom-Warburton NRBC and lacunae applied to original currents (Section 7.6). By increasing T_g we drop each auxiliary problem later. In Figure 40, we compare the errors for $T_g = 10$, $T_g = 750$, and $T_g = 1500$. It turns out that increasing the computation time for each sub-problem (i.e., keeping it longer) reduces the overall time of stable computation.

To further investigate the effect of the increased computation time for individual auxiliary

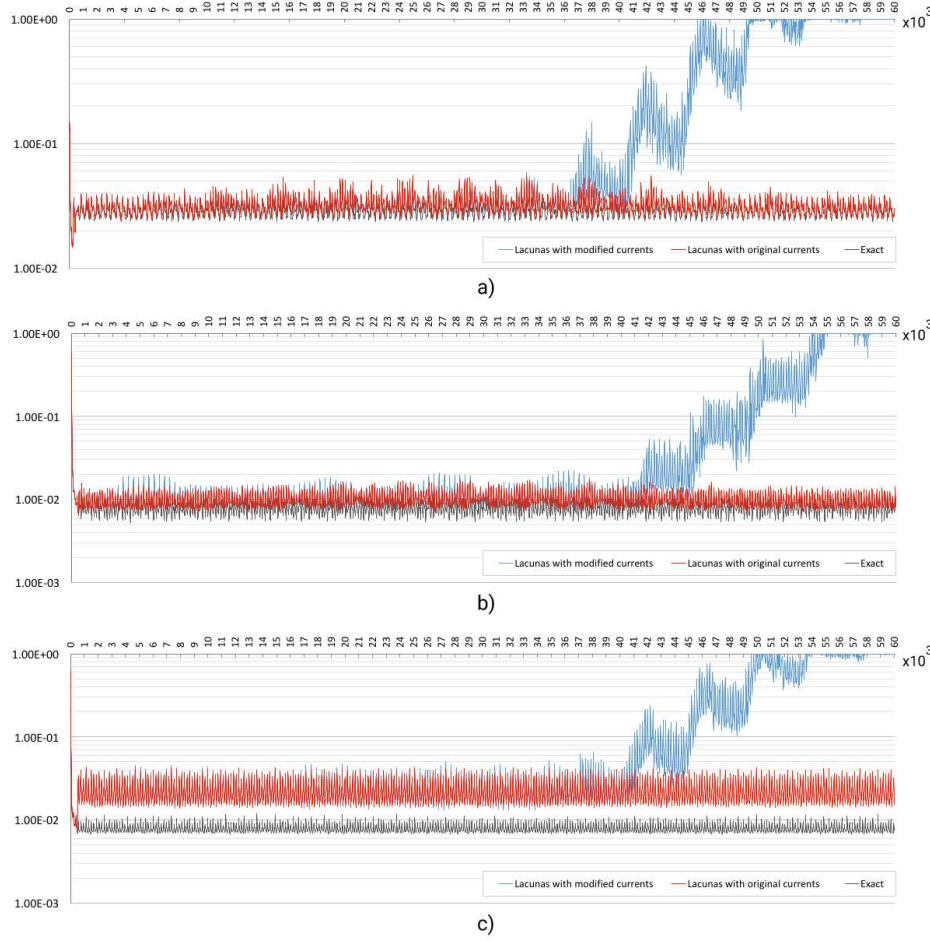


Figure 39: Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c): blue curve — lacunae applied to auxiliary currents; red curve — lacunae applied to original currents. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile for the case where the computational domain is terminated by the exact boundary conditions.

sub-problems on overall performance,⁷ we plot the maximum absolute values of their respective solutions as functions of time, see Figure 41. The left column in Figure 41 corresponds to the case of no additional simulation time, i.e., $T_g = 10$. The value of the solution at the terminal point of every given curve represents the maximum absolute value of the residual field inside the lacuna. One can see that those values increase with the number of sub-problem, which indicates that there is a slow accumulation of error. The right column in Figure 41 corresponds to the case where the computation time for each subproblem is increased by $T_g = 1500$ beyond the required minimum. We see that in this case the error accumulates considerably faster. Moreover, by comparing the respective plots in the left and right columns of Figure 41, one can conclude that previous sub-problems definitely affect subsequent sub-problems. Indeed, if there were no interaction between

⁷We are considering the case where the Hagstrom-Warburton NRBC and unsplit PML at the outer boundary are combined with auxiliary currents for lacunae-based time marching.

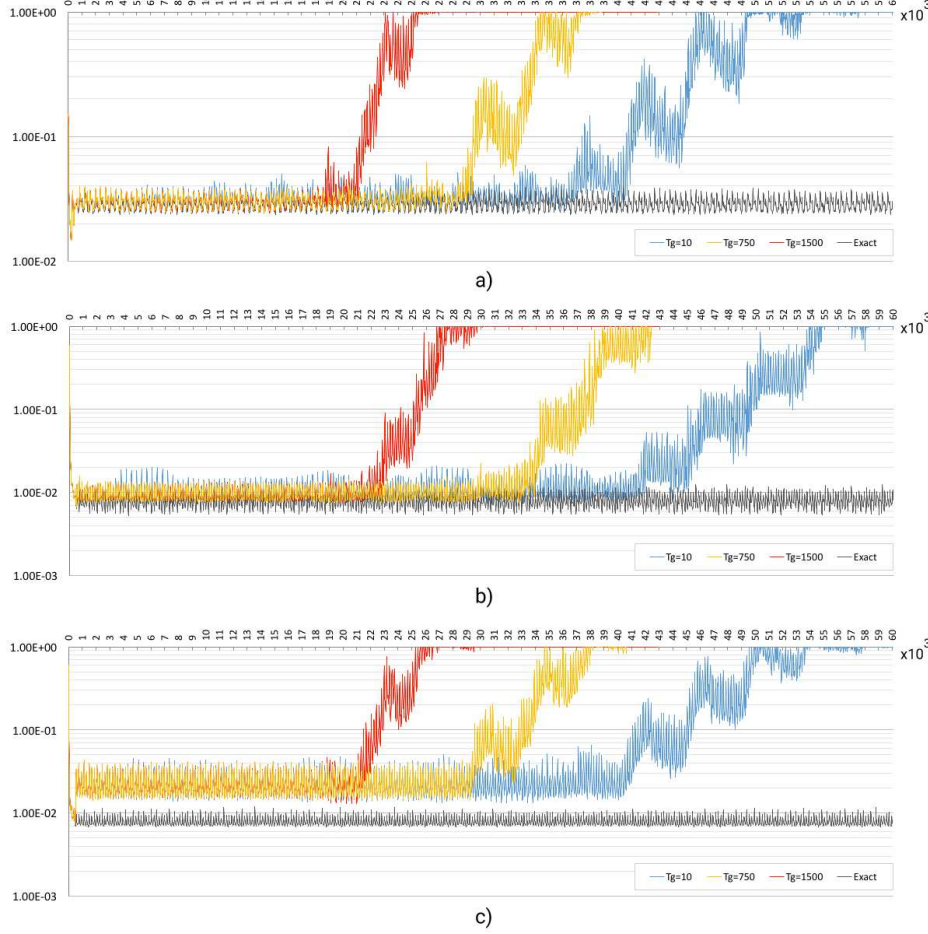


Figure 40: Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c), when the additional time of each auxiliary problem is T_g steps. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile for the case where the computational domain is terminated by the exact boundary conditions.

the subproblems, then the curves that correspond to a shorter overall simulation time would look as exact fragments of the respective curves that correspond to a longer simulation time. Yet the exact mechanism of interaction that leads to the observed slow deterioration is still not fully understood. Apparently, the reflections due to Hagstrom-Warburton NRBC that are not large although not negligible either “contaminate” the interior solution and hence the auxiliary currents. This gives rise to a residual error that gets carried over from one auxiliary sub-problem to another via the interior problem, even though when a given sub-problem is dropped all of its solution is completely removed including the residual field inside the lacuna. We will study this phenomenon more thoroughly and explore the possible strategies to alleviate its adverse effect in the coming months.

Another important factor that affects the performance of the lacunae method is the smoothness of the function $\zeta(t)$ that creates the partition of unity on the right-hand side of equation (92). In doing so, smoothness shall be understood not only literally, as the number of continuous derivatives

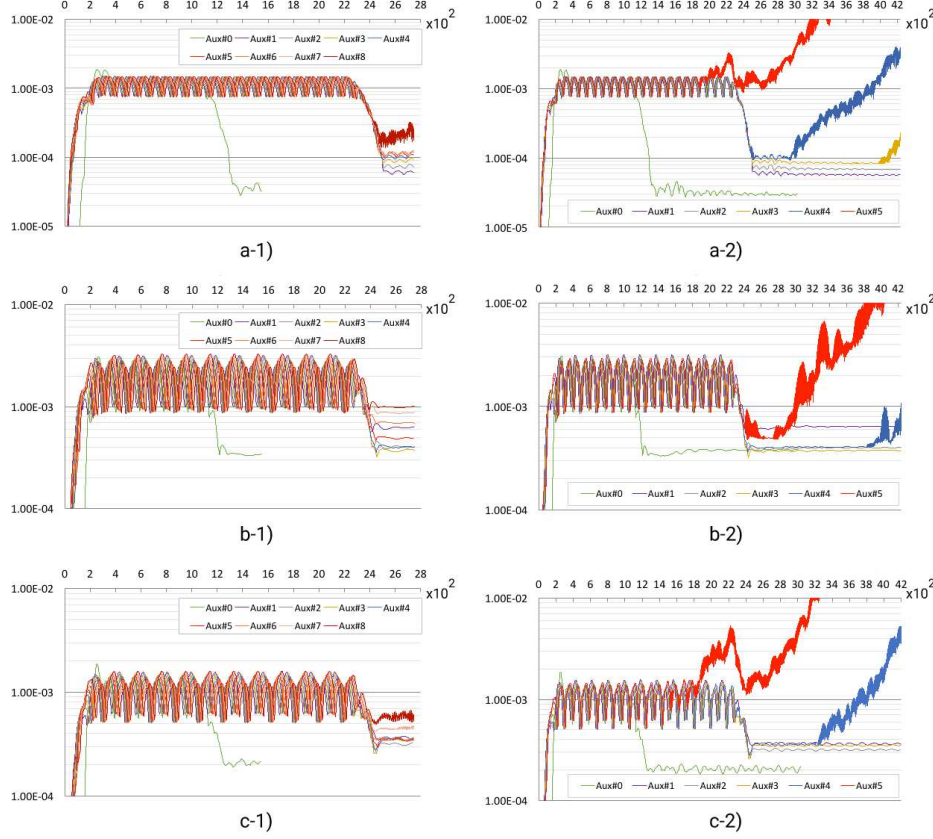


Figure 41: Maximum absolute value of the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c). Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. The first column represents the results when an additional time of each auxiliary problem is zero and the second column represents the results when $T_g = 1500$ steps.

that the function $\zeta(t)$ has, but also as how gradual or sharp its rate of variation is on the interval between 0 and 1. It is expected that the quality of numerical lacunae will be better for more gradually varying $\zeta(t)$. The rate of change of $\zeta(t)$ can be controlled by the parameter σ in (93). For larger σ the function $\zeta(t)$ will vary more gradually, and hence the overall partition of unity (92) can be considered “smoother.” Figure 42 compares the maximum relative error for the field components $E_y(x, y, z, t)$, $H_x(x, y, z, t)$, and $H_z(x, y, z, t)$ and three values of the smoothing parameter: $\sigma = 0.8$, $\sigma = 0.5$, and $\sigma = 0.2$. One can clearly see that the deterioration starts sooner for smaller σ ’s.

The parameter that is known to have a major impact on the performance of Hagstrom-Warburton NRBC [74] is the distance between the waves’ source and the outer boundary. According to [74] (see also [53–55, 75]), the interval of stable computation that the NRBC can provide increases as that distance increases. At the same time, we recall that the whole idea of introducing the auxiliary sources in the context of lacunae-based time marching is to separate between the interior problem that may have a rather elaborate formulation and the auxiliary problem that is governed by the standard Maxwell’s equations and as such, is Huygens’ (see the discussion in the very beginning of the current section — Section 7.7). The auxiliary currents are compactly supported in space inside the computational domain. Their original construction would place them in a

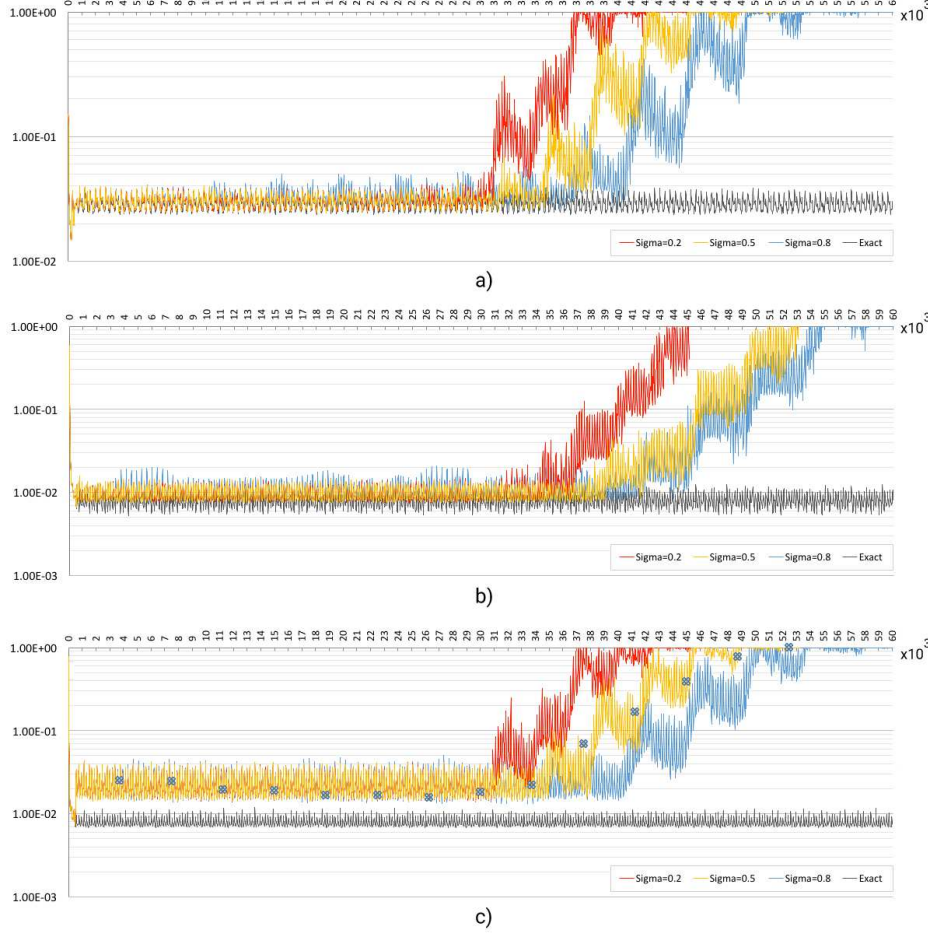


Figure 42: Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) for three different values of the smoothing parameter σ in (92), (93). Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile when the computational domain is terminated by the exact boundary conditions.

narrow region adjacent to the outer boundary from the interior side. Their later modification aimed at preventing the accumulation of charge in auxiliary sub-problems extends the support of auxiliary currents to the entire computational domain. However, either way the distance between the region where the auxiliary currents are defined and the boundary where the Hagstrom-Warburton NRBC is set appears small, which may lead to a more rapid onset of instability. Figure 43 shows the maximum relative error for the field components $E_y(x, y, z, t)$ and $H_x(x, y, z, t)$ as we place the artificial outer boundary further away from the region where the auxiliary currents are supported. The computational parameters are as follows: the size of computational domain for the interior problem is $6.5 \times 6.5 \times 6.5$; the spatial and temporal steps are 0.1083 and 0.0361, respectively; $T_0 = 40$; $\sigma = 0.2$; and $T_g = 1500$. As one can see, the overall time of stable computation increases dramatically as the distance between the auxiliary currents and the outer boundary of the auxiliary problem becomes larger.

We have also studied the performance of the combined methodology (Hagstrom-Warburton

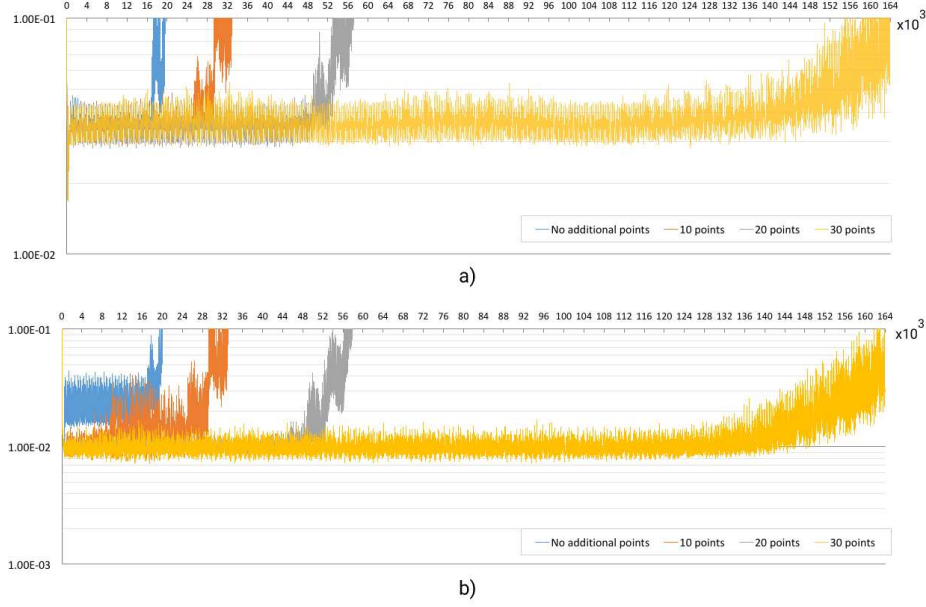


Figure 43: Maximum relative error for the field components $E_y(x, y, z, t)$ – (a) and $H_x(x, y, z, t)$ – (b) as we increase the auxiliary domain size by a given number of grid points in each coordinate direction. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.

NRBC and PML at the outer boundary, with lacunae-based time marching driven by auxiliary currents) as it depends on the grid size. Figure 44 shows the maximum relative error for the field components $E_y(x, y, z, t)$, $H_x(x, y, z, t)$, and $H_z(x, y, z, t)$ computed on two different grids with sizes $h = 0.108$ and $h = 0.054$. The size of the computational domain for the interior problem is $6.5 \times 6.5 \times 6.5$; the auxiliary problem is computed on the domain of the same size; $T_0 = 40$; $\sigma = 0.2$; and $T_g = 1500$. One can see the second order grid convergence for the field component $E_y(x, y, z, t)$. The other two components do not demonstrate the convergence. Most likely, this is an indication that the reflection error exceeds the discretization error. However, it may also be related to the fact that the Hagstrom-Warburton NRBC is set for $E_y(x, y, z, t)$. Altogether, the issue of grid convergence requires a further investigation. As for the interval of stable performance, Figure 44 shows that the grid refinement does not affect it in a substantial way.

In addition, we have conducted a grid convergence study for one individual auxiliary subproblem on three successively refined grids with sizes $h = 0.108$, $h = 0.054$, and $h = 0.027$. The results are presented in Figure 45. They basically indicate the same type of behavior as the results shown in Figure 44. For the field component $E_y(x, y, z, t)$, the residual solution inside the lacuna decreases with the design rate (second order) as the grid is refined. This is expected although on the finest grid the solution seems to be already affected by reflections. For the other two components though, the “depth” of the lacuna does not increase with grid refinement. We will further investigate this behavior in the future.

7.8 Intermediate conclusions on Hagstrom-Warburton NRBC

Altogether, the lacunae-based time marching has very substantially improved the long-time performance of the Hagstrom-Warburton NRBC (or, equivalently, the complete radiation boundary

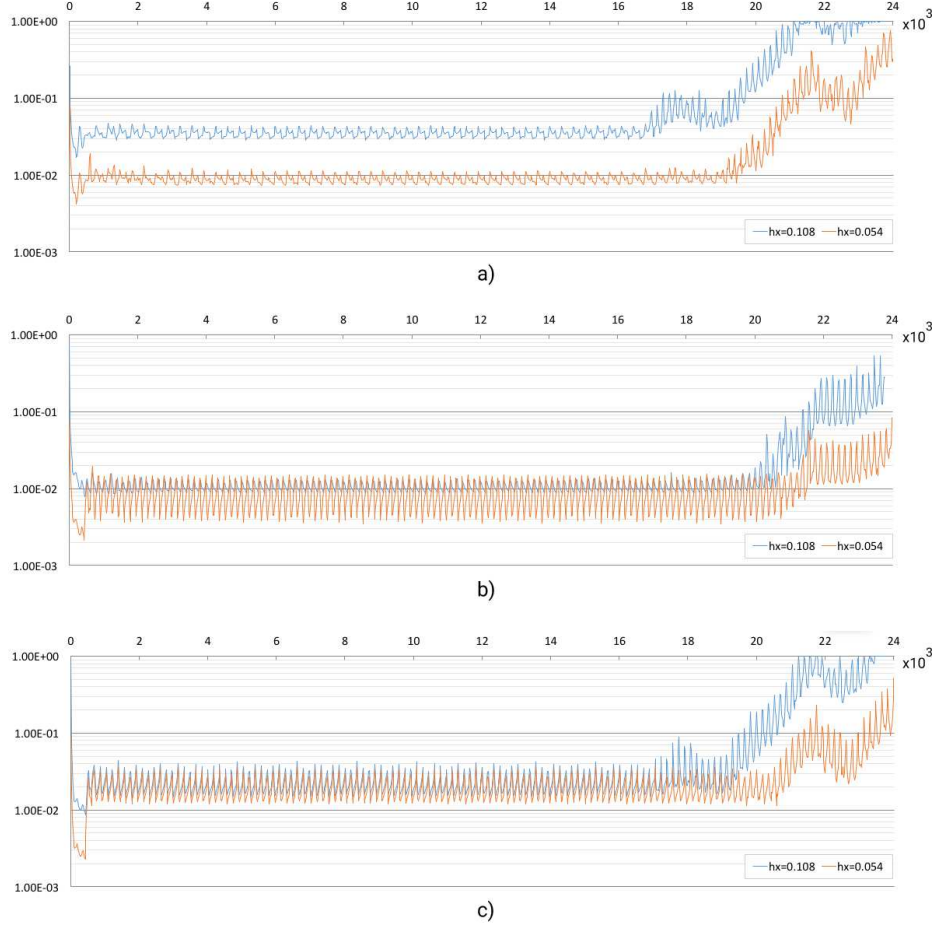


Figure 44: Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) as we vary the grid size. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.

conditions, see [53–55, 74, 75]). In the case of the original sources, we have achieved the design temporally uniform performance for as long as we have run the simulations, see Section 7.6. In the case of auxiliary sources, the interval of stable performance was increased by about an order of magnitude compared to what the genuine non-corrected NRBC could offer, see Section 7.7. The long-time deterioration in the case of auxiliary currents is apparently caused by a higher level of reflections produced by the Hagstrom-Warburton NRBC that translates into an insufficient “depth” (i.e., quality) of the lacunae found in our simulations, see, e.g., Figure 41. Yet the actual mechanism of how the “contamination” of lacunae migrates from one auxiliary sub-problem to the next one is not quite clear and will require additional attention.

As far as the level of reflections per se that we have observed for the Hagstrom-Warburton NRBC [53–55, 74, 75], which is moderate in its own right but still considerably higher than, say, that of the unsplit PML [72], there may be several factors that affect it.

- The reflections may be due to our overly sophisticated implementation that involves both a NRBC and a PML, see Figure 38. This type of implementation is fundamentally unnecessary,

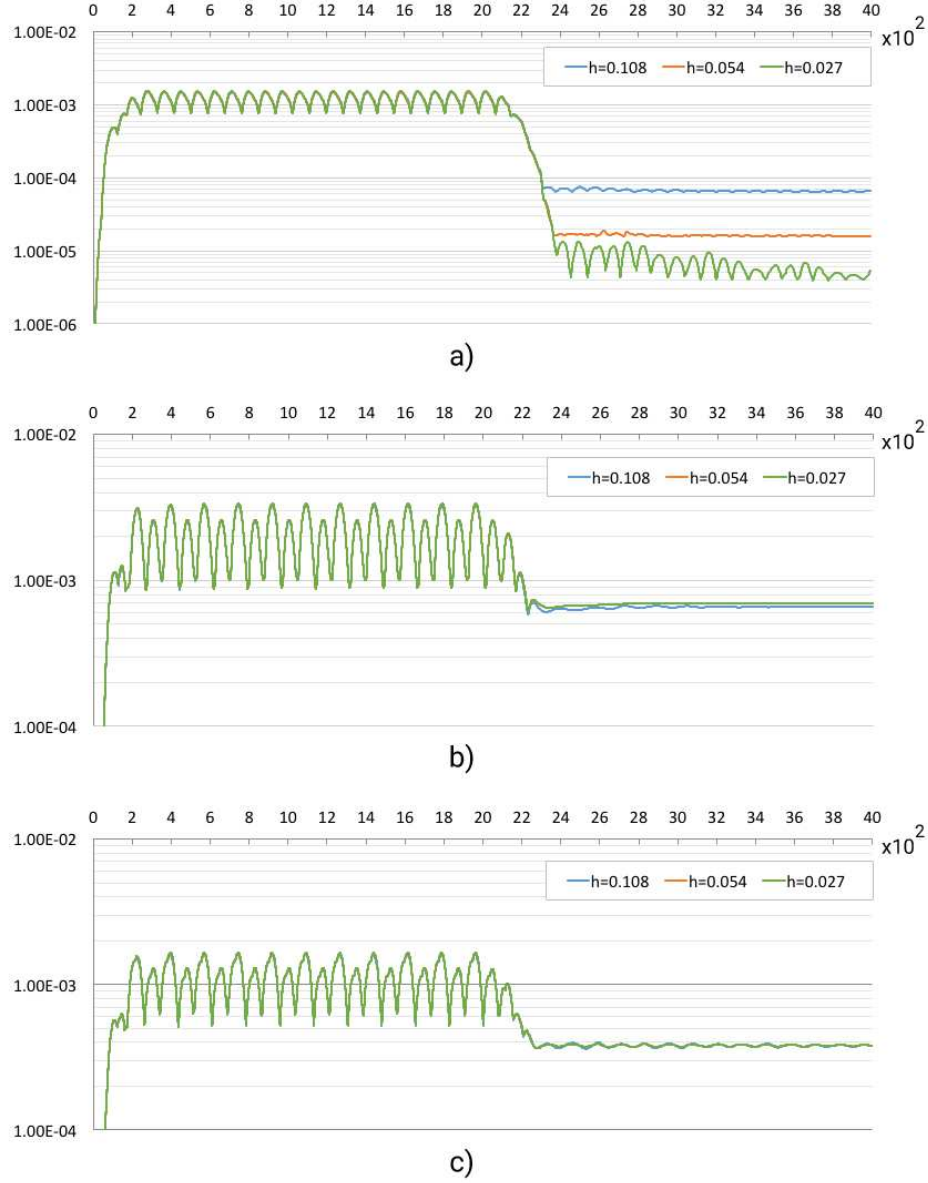


Figure 45: Maximum absolute value of the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) for the second auxiliary sub-problem. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.

yet in the meantime it has allowed us to avoid dealing with the well-known issue of corners for artificial boundary conditions.

- The reflections may also be due to the choices we made for the number of auxiliary functions that define the NRBC, as well as for the values of the Higdon angles.
- The performance of the NRBC clearly improves as the size of the auxiliary domain increases, see Figure 43. This is in accordance with the theoretical predictions, yet it remains to be

see whether the improved performance can also be obtained for smaller domains by adjusting other parameters that characterize the boundary conditions.

As such our next step will be to employ and test a more general and at the same time more straightforward implementation of the complete radiation boundary conditions via double absorbing boundaries, see [56–58]. It has recently been made available by the authors via www.rbcpack.org. This implementation includes the treatment of corners, i.e., edges and vertices of the computational domain, and thus allows one to set the NRBC on all of its six faces. The main goal is to see whether the previously observed deficiencies are due to our in-house implementation of the NRBC that appears sub-optimal or they fundamentally characterize the methodology [53–55, 74, 75].

The use of the double absorbing code provided by the authors of [58] will involve several stages:

- Combining the double absorbing boundaries from www.rbcpack.org with our own interior solver and making preliminary plain runs (i.e., runs with no lacunae) for the same source terms that drive the solution and for otherwise the same computational setting as we have employed previously, see Sections 6 and 7.1, 7.2.
- Optimizing the parameters of the boundary conditions, i.e., the number of auxiliary functions, the Higdon angles, and the distance between the sources and the outer boundary, in order to minimize the reflections. This stage is of central importance, as it will allow us to determine whether a better performance of the NRBC can in principle be achieved compared to what we have observed in our own in-house implementation.
- In case a better performance of the boundary conditions can indeed be achieved, then the double absorbing boundary should be combined with the lacunae-based time marching, and the resulting algorithm should be tested over long simulation times.

8 Computational complexity of ABCs

Our goal in this section is to compare the performance, i.e., the computational complexity, of the various artificial boundary conditions (ABCs) that we have used throughout the project. All boundary conditions are implemented in our in-house Fortran code, with the exception of the double absorbing boundary conditions (DAB), for which we are using the original C++ implementation by the authors of the methodology.

We are using two test solutions in our numerical experiments. Both represent the field driven by a continuously operating source (electric current), which is compactly supported in space (a simulated antenna with a given feed). The first one is a TM mode with a divergence free (i.e., solenoidal) source. The second one is a TE mode, for which the source is not solenoidal. Both cases appear considerably harder than those typically used in the literature for testing the boundary conditions. The latter, on the other hand, involve only the initial conditions that drive the field and no currents that operate continuously in time.

A very important observation we have made is that the case with a non-solenoidal source is more challenging for the ABCs. Namely, the performance of both the unsplit PML and the DAB method deteriorates very considerably already for moderate simulation times.

We compare the results against the benchmark solution computed with the help of the exact boundary conditions, i.e., the boundary conditions derived from the backward engineered exact solution. On a given grid, this benchmark numerical solution is unimprovable from the standpoint of both accuracy and complexity.

Our overall findings are as follows. Computational complexity of the low order local ABCs (such as Sommerfeld, Higdon, etc.) is low; it is basically the same as that of the exact boundary conditions. However, their accuracy is grossly insufficient as they generate substantial reflections. The unsplit PML that we have employed for computations is much more accurate; the reflections are below the discretization error on the grid and the overall accuracy is the same as that for the exact boundary conditions. The complexity of computations with the PML is higher than the benchmark, but only by about 20%. However, the PML can sustain the aforementioned good accuracy only for the divergence free source. For the non-solenoidal source, its accuracy rapidly deteriorates after an initial period of stable performance.

This is precisely where the lacunae-based time marching plays the central role. At the expense of an additional modest increase of computational complexity, it removes the long-time deterioration of the solution with the PML and enables a temporally uniform performance.

The lacunae-based time marching can also be used in the capacity of a standalone closure at the artificial outer boundary, without having it combined with the PML. This version of the algorithm is much more expensive. Its advantage though is that it provably removes any error associated with the domain truncation and maintains this performance over arbitrarily long simulation times. This means, in particular, that a solution obtained with the help of this procedure can serve as an accuracy benchmark even when no exact solution is available.

The DAB methodology can be thought of as the ultimate development of the local high order ABCs. The accuracy of this method is controlled by its order (or, equivalently, the number of recursions) that, in turn, determines the computational cost. For the divergence free source, the method performs well over long times and provides the accuracy comparable to that of the exact boundary conditions already with relatively few recursions. Yet for the non-solenoidal source the magnetic field quickly loses the accuracy. The accuracy can be partially restored only at the expense of substantially increasing the number of recursions (and thus the cost). Even in that case, the accuracy is not restored completely and the error continues to increase, albeit slowly, as the time

elapses. It is not known what causes the deterioration of the DAB solution for the magnetic field in the case of a non-solenoidal source. Moreover, the boundary condition is set only for the electric field and the magnetic field is computed off the electric field values. At the same time, the electric field itself does not deteriorate. Therefore, a modification of the boundary condition for the electric field is not likely to correct the behavior of the magnetic field. Altogether, it indicates that the DAB methodology requires further investigation.

8.1 Introduction

The large class of computational methods to handle wave problems in unbounded or very large domains is based on truncating the unbounded domain by introducing an artificial boundary. The aim of this paper is to compare computational complexity of various artificial boundary conditions for Maxwell's equations. Consider cubic domain with side l . We use the staggered second order accurate explicit Yee scheme. An electromagnetic wave in Cartesian coordinates consists of six components: $E_x, E_y, E_z, H_x, H_y, H_z$ of electric field E and magnetic field H . The computational domain consists of the interior and the border of both fields. We consider artificial boundary conditions for electric field E . Thus, all the points of magnetic field H on the border are surrounded by the points of E field and may be updated using standard Yee scheme algorithm. The update procedure of one time-step consists of the following steps:

- Update interior points of E components using Yee scheme;
- Update interior points of H components using Yee scheme;
- Compute boundary points of E components using artificial boundary condition;
- Update boundary points of H components using Yee scheme.

Due to the staggered grid, various components of E and H fields consist of the different number of points. For example, components of E field are described by $N \times N \times (N - 1)$ grids and components of H are described by $N \times (N - 1) \times (N - 1)$ grids. We would like to get an asymptotic estimation of computational complexity when N is sufficiently large that is why we neglect this difference in the grids and consider $N \times N \times N$ grid for each component of the electromagnetic wave. Let T be the total time of numerical simulation and Δt be the time step size, satisfying CFL criterion $\Delta t \leq h/\sqrt{3}$, where $h = l/(N - 1)$ is a spatial step that is the same in all directions. We require that the maximum level of reflections provided by artificial boundary condition is ε . Consider various artificial boundary conditions and provide the theoretical and numerical estimation of computational complexity.

8.2 Estimation of computational complexity

8.3 Test solutions and exact boundary condition

To study the performance of different artificial boundary conditions (ABC), we need to be able to compute the actual numerical error on the grid at different moments of time. We construct the solution by "backward engineering" that is driven by a point-wise oscillating dipole antenna located at the origin of the coordinate system. To suppress the singularities at the origin, the solution is regularized by means of a smoothly decaying function. Thus, we can solve Maxwell system with right-hand sides numerically and evaluate the error by comparing the solution computed on the grid with the exact solution. Note that the most of the authors in publications use solutions based on

the initial conditions only. Our setup is more general because we can study the long-time behavior of the boundary conditions using the source that continually generates electromagnetic waves.

Using the described approach we constructed two different solutions. The first solution is a propagating TM-mode of electromagnetic wave that is driven by divergence-free source. The second solution is more general. It is a propagating TE-mode of electromagnetic wave driven by non divergence-free source. Both sources are located at the center of the computational domain.

We can use the exact solution as a boundary condition for truncation of the unbounded domain. Such boundary condition can be used as a benchmark for estimation of the computational complexity because it doesn't require any additional computations. If we use exact solution as artificial boundary condition for E field then computational complexity of one time step is as follows:

$$C_0(N) = N^3 a, \quad (94)$$

where $a = \text{const}$ that depends on the implementation of the Yee scheme, the software and hardware of the computer. This algorithm has minimal computational complexity if we already know the solution everywhere at each time step. Note that in our numerical experiments we do not pre-compute the currents and exact solution. In this case, the computational time of this artificial boundary condition is not zero and it depends on the implementation of the test solution. The numerical results for divergence free source are presented in the Table 1.

Obviously, the exact solution is not known in real industrial problems and this artificial boundary condition is not applicable in practice. Thus, we have to use artificial boundary conditions, such as low-order ABC, PML or DAB.

8.4 Local low-order ABC

The simplest artificial boundary conditions that may be used are local low-order ABCs. There are several ABCs that were used in practice since 1970s. The first one is "zero-order" Sommerfeld-like boundary condition [69] that provides a very crude approximation. It has a very cheap implementation, but this ABC generates a high level of spurious reflections. In our numerical experiments, the maximum of relative error was more than 17% that is unacceptable in real experiments. We implemented and tested second-order Higdon ABC, Betz-Mittra ABC and Mur ABC [69]. These ABCs provide a lower level of spurious reflections than Sommerfeld ABC, but the accuracy of any low-order ABC is not satisfactory in most cases. Furthermore, the major disadvantage of the ABCs is that they require a special treatment at corners formed by the intersection of two flat segments of the artificial boundary.

Each local low-order boundary conditions is implemented directly at each point of the boundary, that is why the estimation of the computational complexity coincides with (94). The numerical results for low-order ABCs are also presented in Table 1. Note that the performance of low-order boundary condition can't be improved by using smaller spatial step. In practice, it is not possible to reach the desired low level of reflections by using these ABCs.

8.5 Unsplit uniaxial PML

One of the most common approaches for truncation of unbounded domains for wave equations is the perfectly matched layer (PML). The PML method is attractive because it is efficient and easy to implement. The treatment of the corners and edges of the computational domain when using the PML is fairly straightforward. Consider Unsplit uniaxial PML [76] that is described by the absorbing layer surrounding the computational domain. The PML itself is terminated by the perfect electric conductor conditions (PEC).

The PML provides the level of reflections ε_p that depends on the thickness p of the PML and parameter σ which describes absorbing properties of the perfectly matching layer. One may choose p and σ to achieve the desired level of reflections $\varepsilon_p \leq \varepsilon$. The PML is usually discretized with the same spatial step h . In this case, the PML is $D = p/h$ points thick. Consequently, the number of points in the PML depends on the number of points in the domain $D = \nu N$.

The estimation of computational complexity in the case takes following form:

$$C_{pml}(N) = (1 + 2\nu)^3 N^3 a, \quad (95)$$

where $a = \text{const}$ that depends on the implementation of PML, implementation of the Yee scheme, software and hardware of the computer. Furthermore, in the PML region the Maxwell system gets modified and requires more operations than standard Yee algorithm. We note that parameter ν may be small enough when the number of points in the domain is large. Table 1 shows the results of numerical simulations. In fact, the PML is only 1.2 times more expensive than the exact boundary condition that we use as a benchmark. By the way, if we compare computational time that is needed for updating the boundary points of the computational domain, we see that the 10-points PML is 200 times more expensive than Sommerfeld ABC and 90 times more expensive than second-order ABCs because we have to enlarge the computational domain. However, the PML provides very low level of spurious reflections.

Unfortunately, it is known that PML is prone to error growth in long-time simulations. We don't see this undesirable behavior if we use the first test solution with divergence-free source (see Fig. 1). The 10-point Unsplit PML provides a very low level of spurious reflections for a long time without any error growth in our numerical simulations. However, we see the error growth if the second test solution with non divergence-free source is used (see Fig. 2). We don't know the reasons for this behavior, but, obviously, the PML can't be used successfully everywhere. Furthermore, one can't predict the long-time error growth for particular parameters of the PML or the solution, because there is no theoretical foundation of this phenomena. That is why we have to use lacunae-based time marching algorithm to provide long-time simulations.

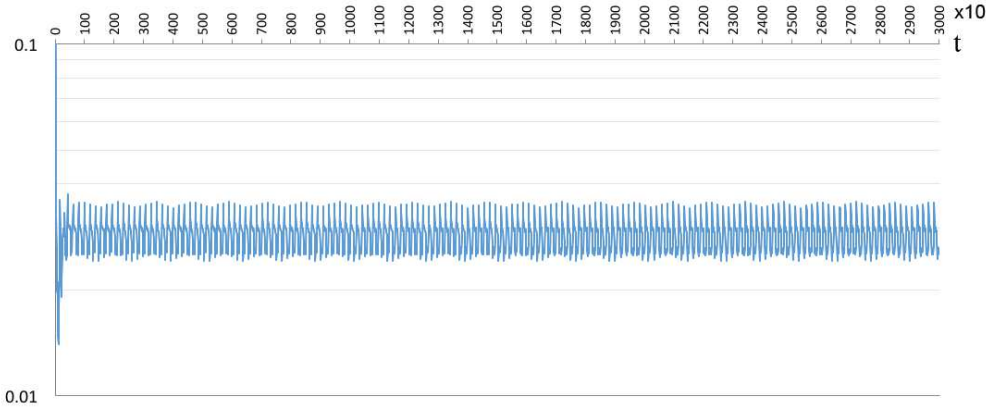


Figure 46: The maximum absolute relative error of E_y component. Unsplit PML is used as boundary condition in the test with divergence-free source.

8.6 Lacunae-based time marching algorithm with PML

We can use lacunae-based time-marching algorithm to avoid long time deterioration of the PML. Suppose we want to solve the Maxwell system with non-divergent sources j and j_M . The lacunae-

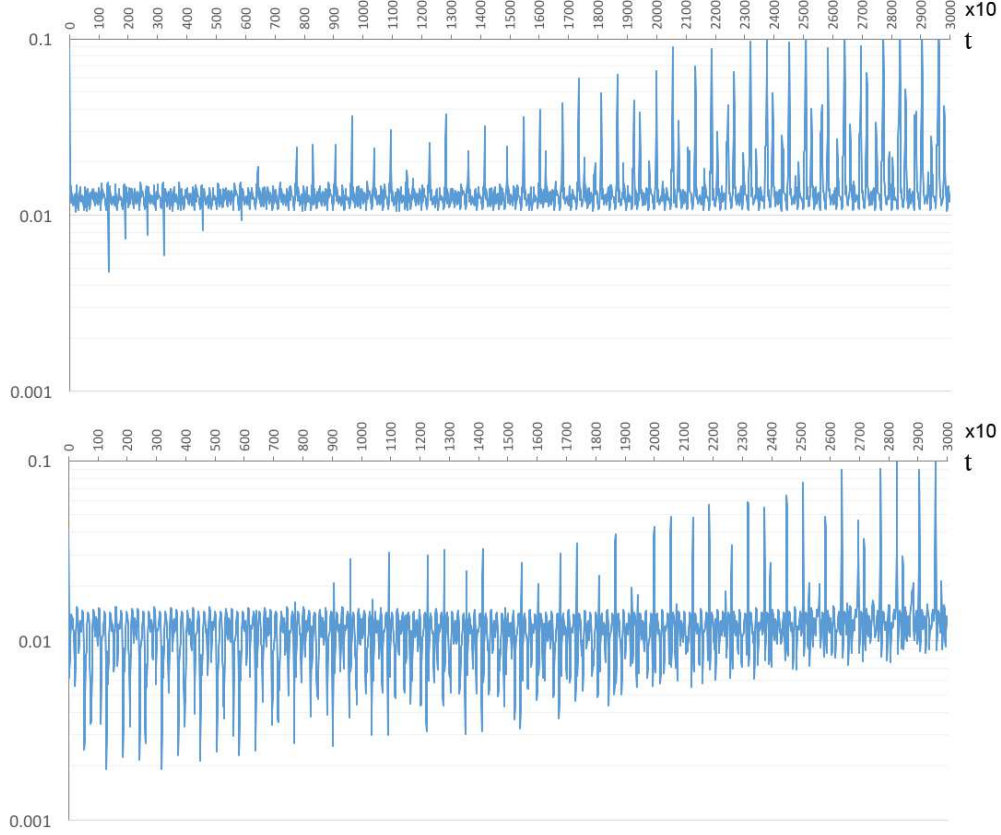


Figure 47: The maximum absolute relative error of E_y (upper) and E_z (lower) components. Unsplit PML is used as boundary condition in the test with non divergence-free source.

based time marching algorithm is based on the Huygens' principle. We use the following property of Maxwell's equations in vacuum: provided that the currents driving the pure Maxwell system are compactly supported in both space and time, the propagating electromagnetic waves have sharp fronts. Once the outgoing waves have left the bounded computational domain, there is no need to further update the fields. We decompose the original problem into a series of independent partial subproblems driven by compactly supported in space and time partial currents.

Consider one subproblem surrounded by PML (see Fig.3).

Let us introduce the time interval T_{int} defined as the sum of the time T_s needed for the waves to cross the domain plus the operational time of the sources T_0 .

$$T_{int} = T_0 + T_s. \quad (96)$$

The original computational domain will fall into the lacuna after T_{int} and will remain inside the lacuna for all $t > t_2$. Consequently, we can solve the partial problem on the finite interval T_{int} and then the solution of the problem will reach its steady state on the domain of interest. We consider cubic domain covered with $N \times N \times N$ grid that is why we should choose T_s as follows:

$$T_s = \frac{1}{c} \text{diam} S = \frac{\sqrt{3}}{c} N h. \quad (97)$$

The overall solution can subsequently be obtained by linear superposition of the partial subproblems. Consider the following partition of unity that is used for construction of partial currents (see

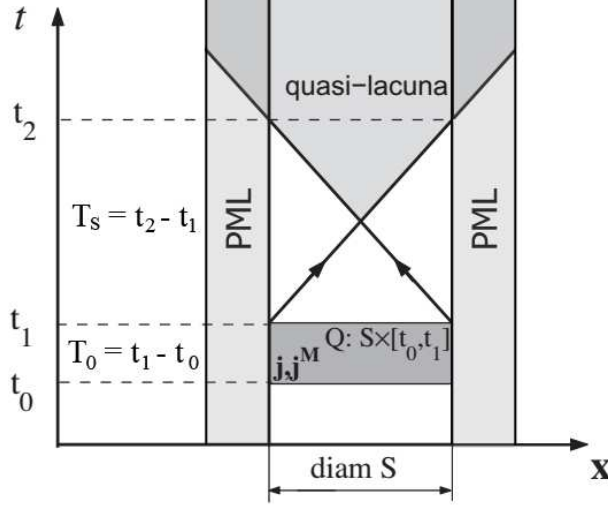


Figure 48: Partial subproblem surrounded by PML.

Fig. 4):

$$1 = \sum_{i=0}^{\infty} \zeta(t - (i-1)T_1), \quad \text{supp}\zeta(t) = [T_1 - \Delta T, 2T_1]. \quad (98)$$

Multiplying the original current densities j and j^M by the right-hand side of (98), we get an individual i -th partial source that operate only during a fixed finite time interval $T_0 = T_1 + \Delta T$.

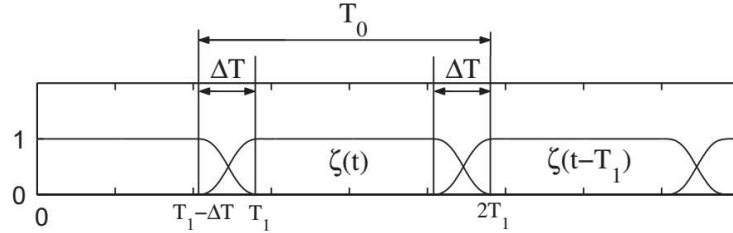


Figure 49: A smooth partition of unity.

If the total time T of the numerical simulation is sufficiently large than the total amount of partial subproblems:

$$K \approx T / (T_0 - \Delta T). \quad (99)$$

The intersection of the subproblems takes some part of T_0 that is why $\Delta T = \mu T_0$. Let T_p be the total time, when the level of reflections produced by PML is less than ε . Obviously, the total time of each auxiliary problem should be as big as possible to reduce the amount of partial subproblems. That is why we choose $T_{int} = T_p$. In this case we have

$$T_0 = T_p - \frac{\sqrt{3}h}{c}N. \quad (100)$$

Using the estimation of computational complexity of PML (95), we can say that computational complexity of one time step of the new algorithm takes the following form

$$C_{lacpml} = \frac{T_p}{(1 - \mu)(T_p - c^{-1}diamS)} (1 + 2\nu)^3 N^3 a.$$

Thus, if $T_p \gg c^{-1} \text{diam} S$ than

$$C_{lacpml} = \frac{(1 + 2\nu)^3}{(1 - \mu)} N^3 a, \quad (101)$$

where $a = \text{const}$ that depends on the implementation of decomposition of the original problem into the sequence of partial subproblems, implementation of the PML and Yee scheme, software and hardware of the computer. Note that $\mu > 0$ to provide smooth partition of unity.

We conclude that the computational complexity of this algorithm depends on the size of the region where two auxiliary problems intersect. If this region is small, then the computational complexity of this algorithm tends to the computational complexity of PML. Table 1 shows the results of numerical simulations with the first test solution that is driven by divergence-free source. We didn't see the error growth with this solution, but we use it to compare computational complexity of this algorithm with the others.

8.7 Lacunae-based time marching algorithm with PML and divergence-free auxiliary currents

The original lacunae approach is valid only for the Maxwell system in vacuum with divergent-free source. Nevertheless, we still can use lacunae time marching algorithm in more general case if we employ the divergence-free auxiliary currents. These currents should be formed in the near-boundary thin layer using the solution of the interior problem. We use the currents of the special form to avoid charge accumulation inside the lacunas. Particularly, the auxiliary currents generate divergence-free extension of the electromagnetic field. For this purpose, we solve Neumann problem for the Laplace equation by means of FFT algorithm.

It is known that FFT algorithm is of $O(N \log N)$ complexity. However, the implementation of the divergence-free auxiliary currents requires not only FFT algorithm, but also the number of steps including transmitting of the data between auxiliary problems and FFT solver. We use the same estimation of computational complexity (101) to compare the values of the constant a in the original lacunae-based time marching algorithm with PML and the modified lacunae-based algorithm with PML that requires calculation of auxiliary currents. The numerical experiments show that the calculation of modified auxiliary sources do not enlarge the computational time significantly.

Thus, the lacunae-based time marching algorithm is applicable for non-Huygens' problems that may have neither classical lacunae nor quasi-lacunae. Figure 5 shows the maximum absolute relative errors of E_y and E_z components in case of using Unsplit PML and in case of using lacunae-based time marching algorithm with PML and divergence-free auxiliary currents. We don't see any error growth in case of using latter absorbing boundary condition.

8.8 Lacunae-based time marching algorithm without PML

We would like to note that the lacunae-based time marching can be used with no PML, i.e., as a standalone closure at the artificial outer boundary. The idea is to expand the computational domain so as to allow using the PEC condition at the outer boundary for partial subproblems. If the computational domain is sufficiently large, then the reflected wave will not reach the original computational domain.

Consider one partial problem (see Fig. 6). If the PEC will be far enough, then can be sure that reflected wave will not reach the domain before T_{int} in this case. We can choose the operational time of partial current $T_0 \geq 2\Delta T$ using the following ideas. If T_0 is large than we should increase

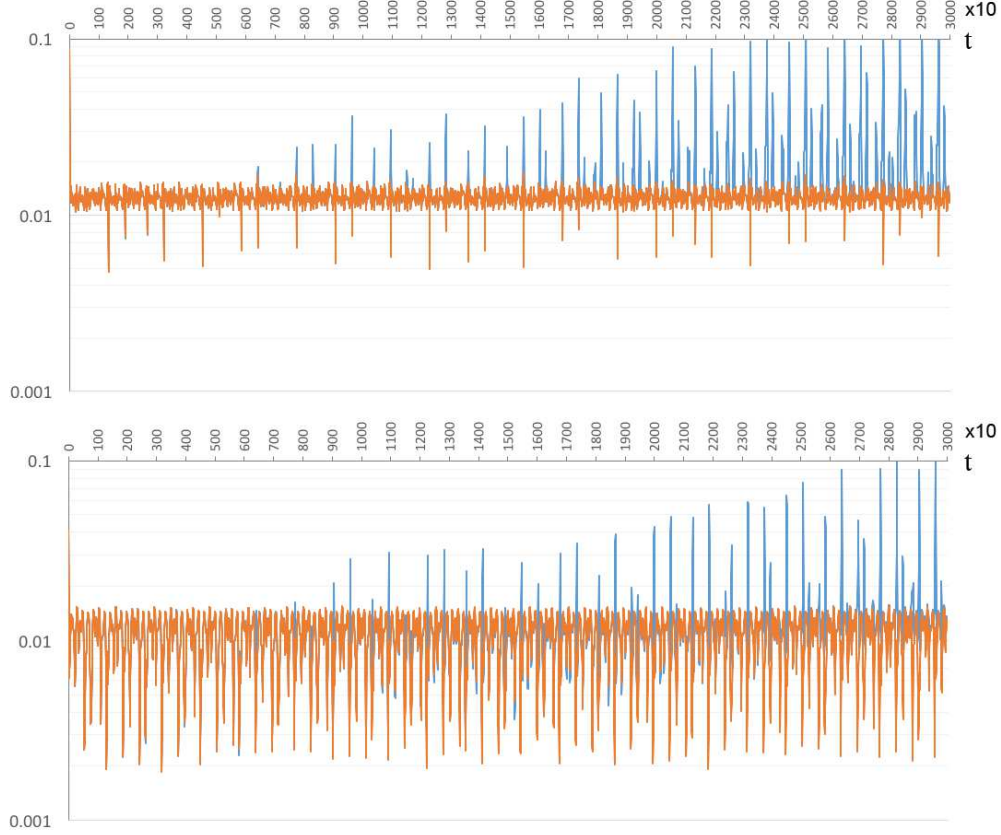


Figure 50: The maximum absolute relative error of E_y (upper) and E_z (lower) components. The blue curve represents the test with PML and the orange curve represents the test with lacunae-based time marching algorithm with PML and divergence-free auxiliary currents.

the size of the domain significantly. On the other hand, if T_0 is small than the number of partial problems that should be calculated simultaneously may be large. Let $T_0 = \kappa T_s$ and $\Delta T = \mu T_0$. In accordance to (100), we have

$$T_{int} = \frac{\sqrt{3}h}{c}N(1 + \kappa). \quad (102)$$

Thus, the size of original domain will be increased for $2R$ points in each direction,

$$R = \frac{c}{2h}T_{int} = \frac{\sqrt{3}}{2}N(1 + \kappa). \quad (103)$$

We conclude that the computational complexity of one time step for one partial problem is $(1 + \sqrt{3}(1 + \kappa))^3 N^3 a$. We use the same estimation of the amount of partial subproblems (99). Thus, we get the total computational complexity of the method of lacunas on one time step:

$$C_{lac} = \frac{1 + \kappa}{(1 - \mu)\kappa}(1 + \sqrt{3}(1 + \kappa))^3 N^3 a. \quad (104)$$

This function has minimum with

$$\kappa = \frac{\sqrt{4 + \sqrt{3}} - 1}{3\sqrt{3}} = 0.268 \quad (105)$$

Thus, the total estimation of computational complexity is

$$C_{lac} = \frac{154.42}{1 - \mu} N^3 a, \quad (106)$$

where $a = \text{const}$ that depends on the implementation of decomposition of the original problem into the sequence of partial subproblems, implementation of the Yee scheme, software and hardware of the computer. The total amount of subproblems that should be calculated simultaneously is

$$M(\mu) = \left\lceil \frac{T_s + \Delta T}{T_0 - \Delta T} \right\rceil + 1 = \left\lceil \frac{4.73}{1 - \mu} \right\rceil,$$

where $\lceil \cdot \rceil$ denotes rounding to a higher integer.

This algorithm is much more expensive than the lacunae-based algorithm with PML. However, this algorithm doesn't involve any artificial boundary condition for partial subproblems that is why it has theoretically proven estimation of the level of the error.

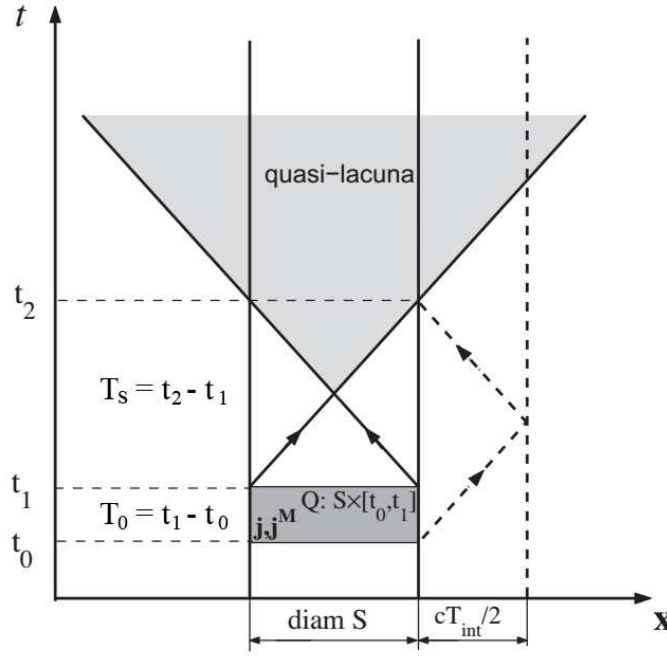


Figure 51: Lacuna of the solution generated by compactly supported sources.

8.9 Double absorbing boundary layer

The high-order ABCs form another large class of artificial boundary conditions for wave equations developed since 1990s. We studied the Givoli-Neta ABC [73] and Hagstrom-Warburton ABC [53]. In fact, there is no efficient implementation of these high-order ABCs in 3D, because it is hard to devise the special treatment for the corners and edges for the ABCs.

However, we consider local radiation boundary condition sequences for second order finite difference approximations to Maxwell's equations based on the idea of complete radiation boundary condition (CRBC) for wave equation. The double absorbing layer (DAB) is the implementation of this idea for staggered grid in time and space used in FDTD methods [58]. This boundary condition

is constructed by forming a thin non-reflecting layer on which we apply the CRBC on two parallel boundaries. The algorithm is based on sequential calculation of the auxiliary variables inside the DAB layer P times, where P is the number of recursions.

The authors of the DAB proposed an optimization scheme that can be employed to rapidly select the parameters of the DAB to minimize the maximum reflection coefficient ρ of the boundary condition. It is shown that ρ decreases exponentially with P . Precisely,

$$P \propto \ln \left(\frac{1}{\rho} \right) \ln \left(\frac{cT}{\delta} \right). \quad (107)$$

Thus, the implementation of the DAB requires four input parameters:

- δ - the minimum separation between the boundary and any scatterers, sources, or other inhomogeneities,
- T - the total simulation time when the DAB provides the proper maximum reflection coefficient ρ ,
- P - the number of auxiliary variables in the DAB called 'recursions';
- c - light speed that is equal to unity in the following experiments.

This theoretical estimation of the maximum reflection coefficient is of great interest, because it allows to choose parameters of the DAB according to the total simulation time and the distance between all the inhomogeneities inside the computational domain and the boundary. Furthermore, the implementation of the DAB includes special procedures for edges and corners. We use an original implementation of the DAB that is available online

<http://faculty.smu.edu/csc/projects/rbcpack/YeeCRBC/index.html>.

The DAB is prone to the long-time deterioration of the solution. The error grows very slowly in case of using divergence-free source. Figure 7 shows the maximum absolute relative errors of E_y , H_x and H_z components in case of using DAB. We don't see any sufficient error growth. However, the error grows rapidly in the H components in case of using non divergent-free source (see Fig.8). One can use more recursions in the DAB to suppress the fast deterioration of the solution. Numerical experiments show that the computational time depends on the number of recursions linearly. It is necessary to use many recursions to compute the solution with acceptable level of error even in short-time simulations (see Fig.9). This leads to significant increase in computational time. Finally, we don't know the reasons for this long-time unacceptable behavior of the solution.

We note that it is quite hard to get estimation of computational complexity for this type of boundary condition, because the update procedure consists of several steps including forward and backward updates of the auxiliary variables inside the boundary layer. Besides, the updating procedures differ for the faces, edges, and corners of the computational domain. We assume that the most rough estimation may be obtained in the form similar to (94).

8.10 Numerical simulations - serial implementation

We use our FORTRAN code to compare computational complexity of the boundary conditions numerically. Consider computational domain with the side $l = 10$. The $100 \times 100 \times 100$ grid is used to perform numerical solution of Maxwell equations. The total simulation time is 50 'seconds', the spatial step is $h = 0.1$ and the time step is $\Delta t = 0.03(3)$. We use computational server based on Linux with 16 cores Intel Xeon CPU E5-2698 v3 with 2.30 GHz. Table 1 shows comparison of an

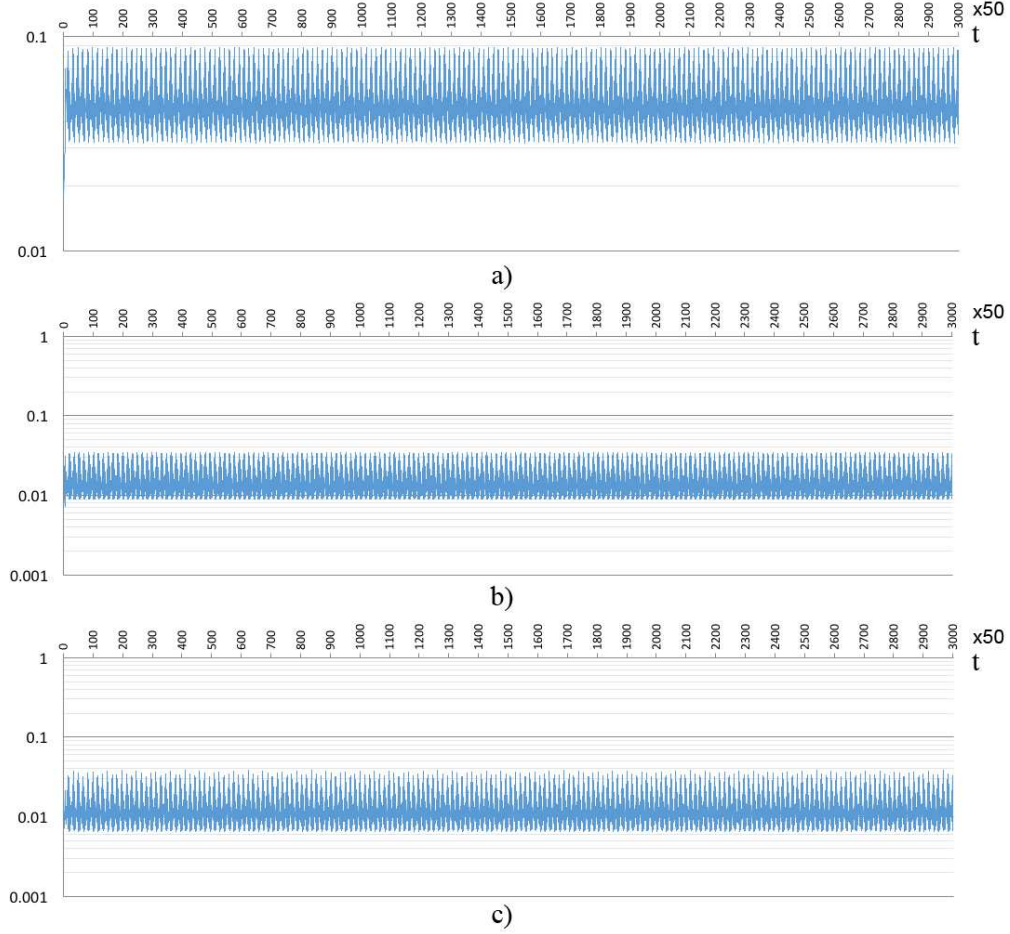


Figure 52: The maximum absolute relative error of E_y (a), H_x (b), H_z (c) components. DAB is used as boundary condition in the test with divergence-free source.

average time T needed for one time step update for different boundary condition. We calculate values of implementation coefficients a and absolute maximum relative error ε for each ABC. Note that calculation of the sources takes significant time. Thus, we calculate an average time needed for one time step update T_b of the boundary for E field components for the main problem. The value T_b is the time of all the steps needed for domain closure of electric field of the main problem including time for updating of the auxiliary problems, DAB or PML. We also calculate $\zeta = T_b/T$ to compare the portion of the boundary condition in the total simulation time.

The local low-order ABCs take less time than others, because these boundary conditions are very simple. However, the absolute maximum error level is high when Sommerfeld and Higdon ABCs are used. We implemented 10-point Unsplit PML, that is why $\nu = 0.1$ in (95). Obviously, Unsplit PML algorithm is more expensive than low-order ABCs, but it generates low level of spurious reflections. As we mentioned above, the PML may be prone to error growth in long-time simulation. The PML may be improved by using lacunae-based time-marching algorithm to suppress the error growth. The total operational time of each partial current should be as big as possible to reduce the computational complexity, but we use $T_0 = 10$ and $\mu = 0.2$ for the test. We see that using lacunae-based time marching algorithm doesn't increase the total computational time significantly. Then, we implemented modified lacunae-based method with auxiliary divergence-free sources. We

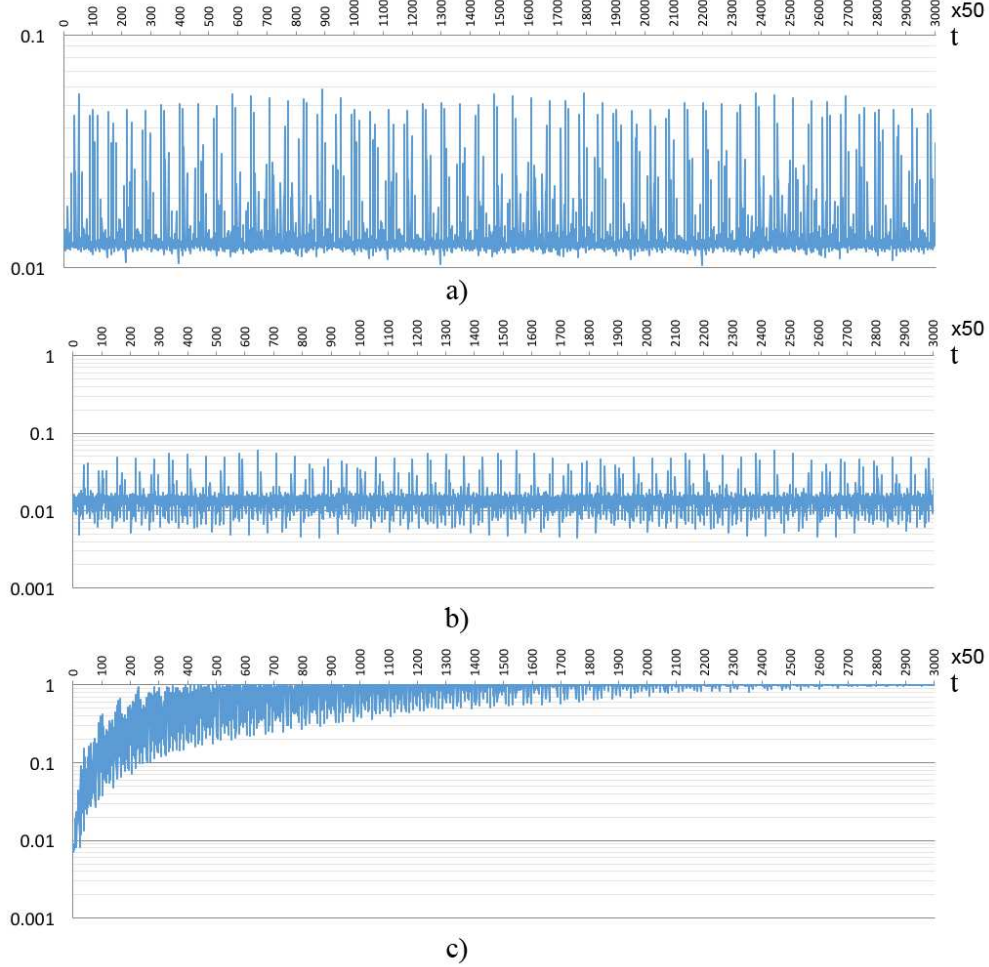


Figure 53: The maximum absolute relative error of E_x (a), E_z (b), H_y (c) components. DAB is used as boundary condition in the test with non divergence-free source.

see that this procedure doesn't increase the total computational time also, but it allows to use the PML in case of non divergent-free source.

It is interesting to get the estimation for lacunae-based algorithm that doesn't involve PML. We use an optimal value (105) and $\mu = 0.2$ in our numerical experiments. Note that this algorithm is very expensive even if the optimal parameters are used, because we need to enlarge the computational domain by 220 additional points in each direction, so the grid in auxiliary problems consists of $320 \times 320 \times 320$ points for each field component. We calculate six auxiliary problems simultaneously. The total simulation time in this case was more than 11 hours that is 34.6 times more than Unsplit PML.

We use the original C++ code for the DAB. It is known that the computational time depends on the language that is used in programs, that is why it is hard to compare DAB with other ABCs. However, we used 4 auxiliary variables in the DAB and calculated and average time needed for one time-step update. We can compare the computational complexity of the boundary conditions by comparing the portion of the updating of boundary in the time-step algorithm. We see that the updating of the boundary takes about 1.8% of the time-step in case of using exact solution as

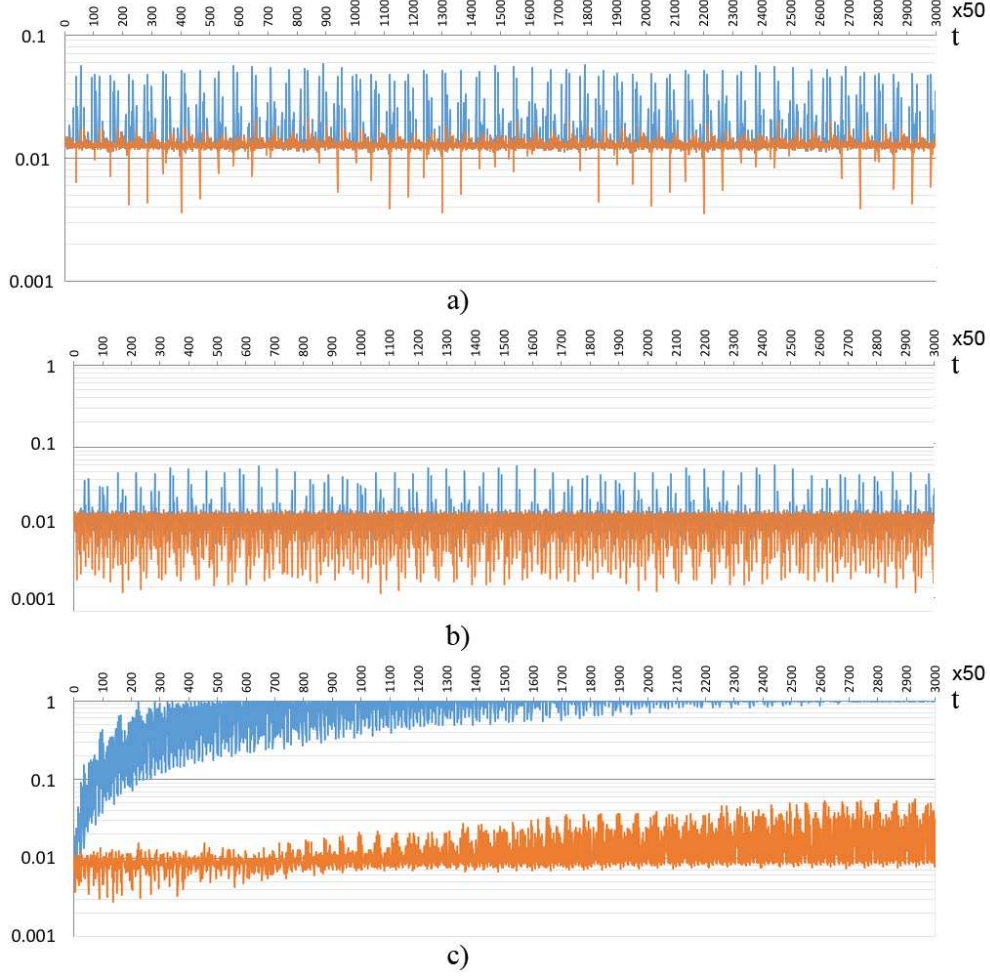


Figure 54: The maximum absolute relative error of E_x (a), E_z (b), H_y (c) components. DAB with 4 recursions (blue) and 10 recursions (orange) are used as boundary condition in the test with non divergence-free source.

boundary condition when FORTRAN program is used. The same portion takes the updating of the boundary in case of using exact solution as boundary condition in C++ program. Thus, we see that the portion of the DAB with four recursions is close to the portion of the 10-points PML.

8.11 Parallelization with OpenMP

Our next goal is to compare parallelization of different boundary conditions. We implemented parallelization based on OpenMP standard. OpenMP is the API that supports multi-platform shared memory multiprocessing in C, C++ and FORTRAN. The OpenMP provides 'fork-join' model. The program begins as a single process: the master thread. The master thread executes sequentially until the first parallel region construct is encountered. Next, the master thread creates a team of parallel threads (fork). The statements in the program that are enclosed by the parallel region construct are then executed in parallel among the various team threads. When the team threads complete the statements in the parallel region construct, they synchronize and terminate, leaving only the master thread (join). The number of parallel regions and the threads that comprise

ABC Type	T (sec)	a	ε	T_b (sec)	ζ
Exact solution	0.668	6.682×10^{-7}	3.62×10^{-2}	1.202×10^{-2}	0.018
Sommerfeld ABC	0.657	6.565×10^{-7}	17.08×10^{-2}	6.672×10^{-4}	0.001
Higdon ABC	0.667	6.665×10^{-7}	5.44×10^{-2}	1.649×10^{-3}	0.003
Betz-Mittra ABC	0.686	6.858×10^{-7}	4.76×10^{-2}	1.708×10^{-3}	0.003
Mur ABC	0.712	7.115×10^{-7}	8.98×10^{-2}	2.156×10^{-3}	0.003
Unsplit 10-points PML	0.805	4.656×10^{-7}	3.56×10^{-2}	0.145	0.181
Lacunas, PML	1.301	1.506×10^{-7}	3.56×10^{-2}	0.634	0.487
Lacunas, PML and aux. currents	1.614	1.869×10^{-7}	3.56×10^{-2}	0.838	0.519
Lacunas without PML	26.709	3.459×10^{-8}	3.56×10^{-2}	25.82	0.967
Exact solution (C++)	0.561	5.609×10^{-7}	3.62×10^{-2}	8.165×10^{-3}	0.015
DAB (C++), 4 recursions	0.669	6.688×10^{-7}	3.56×10^{-2}	0.113	0.168
DAB (C++), 10 recursions	0.855	8.551×10^{-7}	3.56×10^{-2}	0.302	0.353

Table 2: Comparison of the one time-step for different boundary conditions in serial implementation.

them are arbitrary.

The OpenMP may be used for parallelization of Yee scheme. This scheme is an explicit central difference scheme for the approximation of spatial derivatives coupled with the Leapfrog scheme for the approximation of temporal derivatives. There are different approaches on parallelization of Yee scheme on the distributed and shared memory machines (see for instance [77], [78]). We use the simplest approach that is not optimal but easy to implement. The parallelized version of the Yee algorithm is serial in time and use spatial decomposition by one axis when the field update is split among all processors. We add OpenMP directives before the for-loops to provide parallelization of the updating procedures for E and H components of the field.

The same approach provides parallelization of various boundary conditions. In the case of low-order ABCs each face of the cubic domain is separated into regions by one axis. Each region is calculated in one thread that provides parallelization. Unsplit PML may be parallelized in the same way. We separate the computational domain and PML into the regions by one axis and then calculate all the regions at the same time in different threads. When lacunae-based time-marching algorithm is used for domain truncation we have the sequence of auxiliary problems. The simplest way to use parallelization is to calculate each auxiliary problem independently and separate the domain of each problem into regions. The better performance may be achieved if we use nested parallelization and separate threads into groups. Then, we will be able to calculate each auxiliary problem with the group of threads and update several auxiliary problems simultaneously. In our implementation, we don't use this idea and the auxiliary problems are calculated sequentially but with parallelization of each auxiliary problem by separation of the computational domain.

The DAB is more complicated boundary condition. The computational domain is surrounded with two-points thin layer that should be updated several times. Furthermore, the original algorithm requires separate updating procedure for edges and corners of each face of the cubic domain. This leads to significant problems in parallelization of the algorithm. We use the original implementation of the DAB provided by the authors. There is some implementation of OpenMP in the DAB algorithm, but in fact, the program works much slower in case of using parallelization. We don't know the reasons for this issue, that is why we don't use any parallelization of DAB algorithm in our experiments. By the way, the parallelization is still used to update interior points of the domain in the case of using DAB.

We define the speed-up as $S = T/\hat{T}$ where T is the time of one time-step on the one-processor

machine and $\hat{T}$ is the time of one time-step on multiprocessor machine with shared memory that we use. We also define an efficiency of the parallelization as $\eta(p) = S/p$, where p is the number of processors. We also calculate $\hat{\zeta} = \hat{T}_b/\hat{T}$ to compare the portion of the boundary condition in the total simulation time. Table 2 shows the comparison of one time-step update for different boundary conditions with parallel implementation.

ABC Type	$\hat{T}$ (sec)	a	$\hat{T}_b$ (sec)	$\hat{\zeta}$	S	$\eta(16)$
Exact solution	0.112	1.122×10^{-7}	5.088×10^{-3}	0.045	5.96	0.37
Sommerfeld ABC	0.110	1.104×10^{-7}	1.132×10^{-3}	0.010	5.95	0.37
Higdon ABC	0.116	1.161×10^{-7}	9.432×10^{-4}	0.008	5.74	0.36
Betz-Mittra ABC	0.114	1.144×10^{-7}	1.368×10^{-3}	0.012	5.99	0.37
Mur ABC	0.148	1.479×10^{-7}	1.103×10^{-2}	0.075	4.81	0.30
Unsplit 10-points PML	0.138	7.965×10^{-8}	3.195×10^{-2}	0.232	5.85	0.37
Lacunas, PML	0.286	3.310×10^{-8}	0.173	0.606	4.55	0.28
Lacunas, PML and aux. currents	0.362	4.194×10^{-8}	0.199	0.550	4.46	0.28
Lacunas without PML	3.944	5.108×10^{-9}	3.561	0.903	6.77	0.42
Exact solution (C++)	0.070	6.998×10^{-8}	5.700×10^{-3}	0.081	8.01	0.50
DAB(C++), 4 recursions	0.217	2.166×10^{-7}	0.143	0.660	3.09	0.19
DAB (C++), 10 recursions	0.433	4.326×10^{-7}	0.359	0.830	1.98	0.12

Table 3: Comparison of the one time-step for different boundary conditions in parallel implementation.

9 Numerical experiments with OVERTURE

9.1 Background

OVERTURE is a toolkit for solving partial differential equations on structured, overlapping, and hybrid grids. It provides a high level C++ interface for rapid prototyping of PDE solvers built upon optimized C and Fortran kernels, a library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations, and support for block structured adaptive mesh refinement (AMR). OVERTURE has extensive grid generation capabilities, CAD interfaces, interactive graphics and data base support, command scripts with embedded Perl commands. PDE solvers built upon OVERTURE include CGMX — a time domain Maxwell’s equations solver, see [79], CGMP — a multi-physics solver, and others.

During the first project year, two NCSU participants (Dr. Petropavlovsky and Dr. Tsynkov), assisted by Dr. Fedoseyev (Computational Sciences, LLC) and Dr. Henshaw (RPI), have implemented a full-fledged lacunae-based time marching algorithm into OVERTURE. This involved the installation of the latest version of OVERTURE (version 26) on an NCSU computer platform (Mac OS X); verification and testing of the new installation via standard benchmarks provided with the OVERTURE distribution; extended testing that entailed changing the input parameters for the standard benchmarks, testing of the various standard far-field boundary conditions that are built into OVERTURE; design and verification of the special test case intended for the numerical experiments with lacunae-based time marching; building additional subroutines for OVERTURE that would include the generation of the auxiliary sources for each individual field component, the solution of the auxiliary problem with the help of lacunae, and feeding the auxiliary solution back to the main problem to provide the boundary closure; testing the accuracy and complexity of the combined nu-

merical procedure. Numerical experiments described in the first annual report (September 2015) fully corroborate the theoretical design properties of the lacunae implementation into **OVERTURE**.

Note that **OVERTURE** integrates Maxwell's equations in their equivalent second order form (independent wave equations for the individual Cartesian components of the electric and magnetic field). From the standpoint of implementing the lacunae-based time marching this carries an additional significant advantage. Namely, lacunae require solving a collection of auxiliary problems driven by specially constructed auxiliary currents. In the case of a first order system, particular care must be exercised to prevent those currents from accumulating the charge over time; this involves building an additional subtle component of the algorithm. Otherwise, the overall performance of the method may deteriorate. In the second order case though this issue does not manifest itself at all.

In the aforementioned implementation of the lacunae-based time marching with **OVERTURE**, we have used lacunae in the capacity of a standalone closure at the artificial outer boundary, i.e., without combining them with any other ABC or PML. This, in fact, was one of the early original ideas for using the lacunae for computations, see, e.g., [80], and this is the venue that we have taken for testing the lacunae with **OVERTURE** for the first time. The advantage of the pure lacuna-based closure is that it is provably exact. In other words, there is no error associated with the domain truncation because the outgoing waves are basically let out of the domain and then killed off in the special buffer region. The disadvantage is that this buffer region may appear rather large. More specifically, the size of the auxiliary domain should be chosen sufficiently large so as to prevent the reflected waves from reaching the boundaries of the interior domain by the time the latter completely falls into the lacuna.

The focus of our second year work with the code **OVERTURE** is on the combined implementation of the lacunae-based time marching along with a PML. The overall computational domain in this case is much smaller than in the case of pure lacunae-based treatment; it involves only the interior domain and the PML that surrounds it. The PML that is available in **OVERTURE** has been specifically designed to work with the second order form of Maxwell's equations adopted in the code.

The procedure of installation, testing and running the **OVERTURE** is presented in the Appendix A and B.

9.2 Governing equations

The original first order Maxwell system of equations can be reduced to decoupled second order equations for the electric and magnetic field. The equation for the electric field reads:

$$\varepsilon\mu\partial_t^2\mathbf{E} = \Delta\mathbf{E} + \nabla(\nabla\ln\varepsilon\cdot\mathbf{E}) + \nabla\ln\mu\times(\nabla\times\mathbf{E}) - \nabla\left(\frac{1}{\varepsilon}\rho\right) - \mu\partial_t\mathbf{j}, \quad (108)$$

where ε is the electric permittivity, μ is the magnetic permeability, $\mathbf{j}$ is the density of the extraneous electric current and ρ is the density of the electric charge. These quantities are related to one another via the continuity equation:

$$\frac{\partial\rho}{\partial t} + \operatorname{div}\mathbf{j} = 0.$$

For all our simulations, we use $\varepsilon = 1$ and $\mu = 1$, so equation (108) takes the form of the standard three-dimensional wave (d'Alembert) equation:

$$\partial_t^2\mathbf{E} = \Delta\mathbf{E} + \mathbf{F} \quad (109)$$

driven by the external source:

$$\mathbf{F} = -\nabla\rho - \partial_t\mathbf{j}. \quad (110)$$

In this report, we present two types of numerical experiments done on the basis of equation (109):

- (a) Type I experiment: Simulation of the electromagnetic field generated in vacuum by a given current $\mathbf{j}$ and its respective charge ρ (e.g., antenna with a given feed). The computational domain is terminated by a PML combined with the lacunae-based time marching.
- (b) Type II experiment: Simulation of the scattering of a given impinging plane electromagnetic wave about an object of a given composition and shape. The simplest scatterer is a perfectly conducting sphere. No extraneous currents are present. Again, the computational domain is terminated by a PML combined with the lacunae-based time marching.

9.3 Reference solution for type I experiments

To assess the performance of the lacunae-based algorithm in Type I experiments, we use a reference solution in the form of a propagating TM-mode of electromagnetic field. Its non-zero components in the spherical coordinates are:

$$E_r(r, \theta, t) = \psi(r) \frac{2d_0 \cos \theta}{r^3} \left(\chi + \frac{\dot{\chi}r}{c} \right) - \frac{d_0 \cos \theta}{r^2} \left(\chi + \frac{\dot{\chi}r}{c} + \frac{\ddot{\chi}r^2}{c^2} \right) \frac{\partial \psi}{\partial r}, \quad (111)$$

$$E_\theta = \psi(r) \frac{d_0 \sin \theta}{r^3} \left(\chi + \frac{\dot{\chi}r}{c} + \frac{\ddot{\chi}r^2}{c^2} \right), \quad (112)$$

$$H_\varphi = \psi(r) \frac{d_0 \sin \theta}{cr^3} \left(\dot{\chi}r + \frac{\ddot{\chi}r^2}{c} \right), \quad (113)$$

where

$$\psi(r) = 1 - e^{-\alpha r^6}, \quad (114)$$

$$\chi(t - r/c) = \sin^4[\omega_1(t - r/c)] + \sin^4[\omega_2(t - r/c)], \quad (115)$$

and the upper dot denotes differentiation with respect to t . The fields (111)–(113) are generated by the following non-zero components of the electric current:

$$j_r = \frac{d_0 \cos \theta}{cr^2} \left(\dot{\chi} + \frac{\ddot{\chi}r}{c} + \frac{\ddot{\chi}r^2}{c^2} \right) \frac{\partial \psi}{\partial r}, \quad (116)$$

$$j_\theta = -\frac{d_0 \sin \theta}{cr^3} \left(\dot{\chi}r + \frac{\ddot{\chi}r^2}{c} \right) \frac{\partial \psi}{\partial r}. \quad (117)$$

The forcing term $\mathbf{F}$ of (110) also involves the electric charge ρ which is obtained by integration of the currents:

$$\rho = -\frac{\cos \theta}{r^3} \left[\left(r \frac{\partial^2 \psi}{\partial r^2} - 2 \frac{\partial \psi}{\partial r} \right) (r \dot{\chi} + \chi) + \left(r^3 \frac{\partial^2 \psi}{\partial r^2} + \frac{\partial \psi}{\partial r} \right) \ddot{\chi} + r^3 \frac{\partial \psi}{\partial r} \ddot{\chi} \right]. \quad (118)$$

The factor $\psi(r)$ of (114) is $\mathcal{O}(r^6)$ as $r \rightarrow 0$ and suppresses all the singularities that may appear in the functions (111)–(113) and (116)–(118) at $r = 0$. On the other hand, the derivatives $\frac{\partial \psi}{\partial r}$ and $\frac{\partial^2 \psi}{\partial r^2}$ decay very rapidly for large r which effectively implies $\mathbf{F} = \mathbf{0}$ (to machine precision) sufficiently far away from the origin. By appropriately choosing the parameter α one can ensure that $\text{supp} \mathbf{F} \subseteq S_0$, where S_0 is a given bounded domain that contains the origin and $\text{supp} \mathbf{F}$ is the set where $\mathbf{F} \neq \mathbf{0}$. We also note that the solution (111)–(113) is a regularized (at $r = 0$) version of the field radiated by the elementary dipole $d(t) = d_0 \chi(t)$ located at the origin and aligned with the z -axis.

9.4 Computational setup

In this section, we briefly describe the implementation of the lacunae-based integration technique. We conduct the computations in the full 3D setting ignoring symmetries of the solution, if any.

9.4.1 Interior problem and computational domain. In the experiments of type I, the interior problem (IP) consists of solving the differential equation (109) driven by the source function $\mathbf{F}$ of (110) and subject to the homogeneous initial conditions. In the experiments of type II, the same differential equation (109) is solved for $\mathbf{F} \equiv \mathbf{0}$, but subject to the appropriate boundary conditions on the surface of the scatterer (which is given and fixed). For example, in the case of a perfectly electrically conducting (PEC) scatterer the tangential component of the total electric field must be equal to zero.

For either type I or type II experiments, the computational domain S for the IP is chosen in the form of a cube:

$$S = \{(x, y, z) : -a \leq x, y, z \leq a\}. \quad (119)$$

centered at the origin of the Cartesian coordinate system. For type I experiments, the forcing term (110) is assumed compactly supported inside S . For type II experiments, the scatterer is assumed to be fully contained inside S as well. Outside S , electromagnetic waves propagate in free space, which has to be modeled by imposing the artificial boundary conditions.

9.4.2 Auxiliary problems. The lacunae-based algorithm requires that of a number of specially chosen auxiliary problems (APs) were time marched concurrently with the IP. The APs are set on the auxiliary domain S_{aux} for the same second order wave equation (109) and are driven by the external sources $\mathbf{F}_i$:

$$\partial_t^2 \mathbf{E}_i = \Delta \mathbf{E}_i + \mathbf{F}_i, \quad i = 1 \dots N_0. \quad (120)$$

N_0 in formula (120) stands for the number of APs. The domain S_{aux} includes the domain S of (119) as a subset and typically is larger than the latter. In our simulations, S_{aux} is taken as a cube with side b centered at the origin ($b > a$):

$$S_{\text{aux}} = \{(x, y, z) : -b \leq x, y, z \leq b\}. \quad (121)$$

9.4.2.1 Sources for the auxiliary problem. The APs are driven by the sources $\mathbf{F}_i$ constructed at each time step with the help of the numerical solution $\mathbf{E}$ to the interior problem:

$$\mathbf{F}_i = \eta_i(t) [\partial_t^2 \mu \mathbf{E} - \Delta \mu \mathbf{E}]. \quad (122)$$

The functions $\eta_i(t)$ provide a partition of unity in time and have compact support of length T_0 . Using the notation in Figure 55, we have: $\eta_i(t) = \zeta(t - (i - 1)T_1)$.

The function $\mu = \mu(x, y, z)$ in (122) is a smooth “ramp” function growing from 0 to 1 in a narrow near-boundary domain S_ε adjacent to the boundaries of S from inside. We define the set S_ε in a straightforward way, as the union of six planar layers of width ε located next to the faces of S $x = \pm a$, $y = \pm a$ and $z = \pm a$ on the interior side:

$$S_\varepsilon = S_\varepsilon^{(x+)} \cup S_\varepsilon^{(x-)} \cup S_\varepsilon^{(y+)} \cup S_\varepsilon^{(y-)} \cup S_\varepsilon^{(z+)} \cup S_\varepsilon^{(z-)}, \quad (123)$$

where

$$S_\varepsilon^{(x+)} = \{a - \varepsilon \leq x \leq a, \quad -a \leq y \leq a, \quad -a \leq z \leq a\}, \quad (124)$$

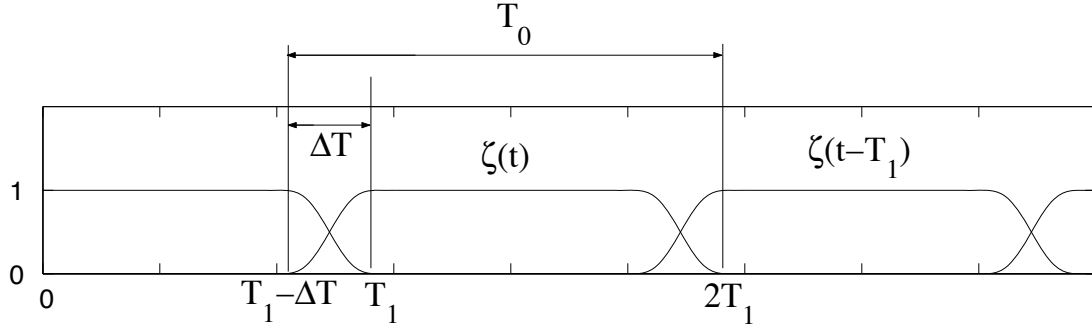


Figure 55: The partition of unity in time.

$$S_\varepsilon^{(x-)} = \{-a \leq x \leq -a + \varepsilon, \quad -a \leq y \leq a, \quad -a \leq z \leq a\}, \quad (125)$$

$$S_\varepsilon^{(y+)} = \{-a \leq x \leq a, \quad a - \varepsilon \leq y \leq a, \quad -a \leq z \leq a\}, \quad (126)$$

$$S_\varepsilon^{(y-)} = \{-a \leq x \leq a, \quad -a \leq y \leq -a + \varepsilon, \quad -a \leq z \leq a\}, \quad (127)$$

$$S_\varepsilon^{(z+)} = \{-a \leq x \leq a, \quad -a \leq y \leq a, \quad a - \varepsilon \leq z \leq a\}. \quad (128)$$

$$S_\varepsilon^{(z-)} = \{-a \leq x \leq a, \quad -a \leq y \leq a, \quad -a \leq z \leq -a + \varepsilon\}. \quad (129)$$

The layers (124)–(129) partially overlap near the edges and corners of S_ε , see Figure 56(a).

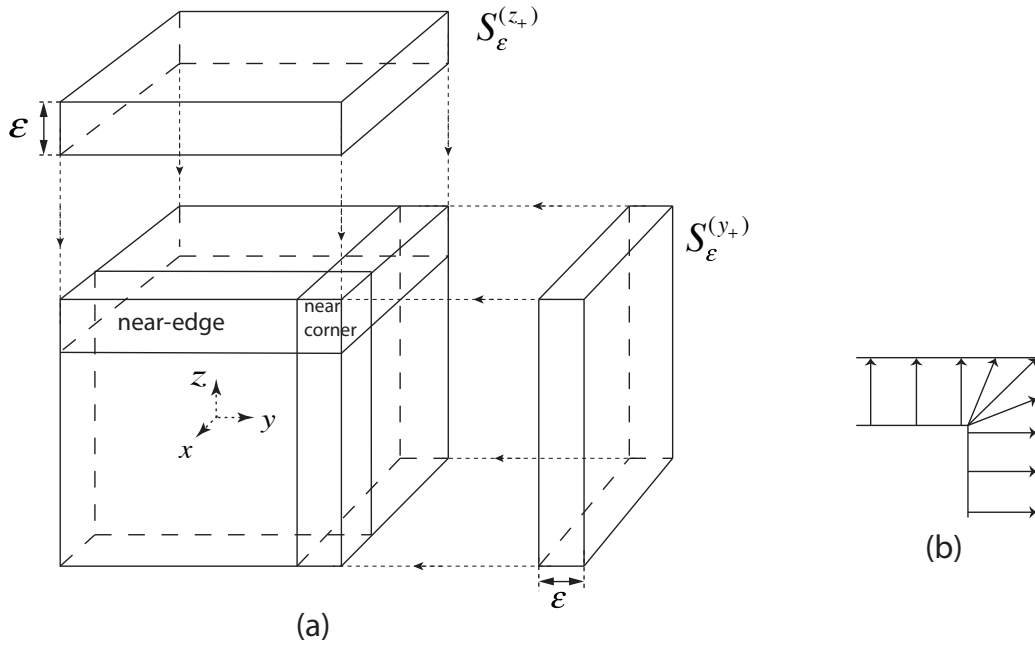


Figure 56: (a) The near-boundary layer S_ε (only two of the six planar layers are shown); (b) The direction of growth of μ in the near-edge domain.

The ramp function $\mu = \mu(x, y, z)$ is constructed with the help of a single-variable function:

$$P_4(x) = \begin{cases} 0, & x \leq 0, \\ 35x^4 - 84x^5 + 70x^6 - 20x^7, & 0 \leq x \leq 1, \\ 1, & x \geq 1, \end{cases} \quad (130)$$

that has three continuous derivatives for $-\infty < x < \infty$. On the non-overlapping parts of S_ε , the function μ smoothly grows from 0 to 1 along the outer normal to the respective layer. For example, for the non-overlapping part of $S_\varepsilon^{(x+)}$ we have:

$$\mu(x, y, z) = P_4\left(\frac{x - a + \varepsilon}{\varepsilon}\right), \quad (x, y, z) \in S_\varepsilon^{(x)} \setminus \{S_\varepsilon^{(y+)} \cup S_\varepsilon^{(z+)} \cup S_\varepsilon^{(y-)} \cup S_\varepsilon^{(z-)}\},$$

and similarly for remaining non-overlapping parts of $S_\varepsilon^{(x-)}$, $S_\varepsilon^{(y\pm)}$, and $S_\varepsilon^{(z\pm)}$.

In the near-edge regions where two layers overlap, the direction of growth makes a smooth 90° turn (see Figure 56(b)), and μ grows there as a function of the distance from the edge, for example:

$$\begin{aligned} \mu(x, y, z) &= P_4\left(\frac{r_{xz}}{\varepsilon}\right), \quad r_{xz} = \sqrt{(x - a + \varepsilon)^2 + (z - a + \varepsilon)^2}, \\ (x, y, z) &\in S_\varepsilon^{(x+)} \cap S_\varepsilon^{(z+)} \setminus \{S_\varepsilon^{(y+)} \cup S_\varepsilon^{(y-)}\}. \end{aligned}$$

Finally, in the near-corner regions, e.g., $S_\varepsilon^{(x+)} \cap S_\varepsilon^{(y+)} \cap S_\varepsilon^{(z+)}$, the function μ grows as a function of the distance from the vertex:

$$\mu(x, y, z) = P_4\left(\frac{r}{\varepsilon}\right), \quad r = \sqrt{(x - a + \varepsilon)^2 + (y - a + \varepsilon)^2 + (z - a + \varepsilon)^2}.$$

The construction of the function $\mu = \mu(x, y, z)$ allows one to compute the partial sources (122) using only the near-boundary values of the solution to the IP. This property will prove crucial in the experiments of type II due to the specifics of the OVERTURE design, see Section 9.6.

9.4.2.2 Boundary closure for the auxiliary problems. The lacunae-based algorithm relies on the concept of waves leaving the computational domain. According to the Huygens' principle, the region behind the aft front of the outgoing waves is free of the electromagnetic field. Therefore, the auxiliary problems set on S_{aux} must be supplemented with some transparent boundary conditions capable of absorbing the outgoing waves without spurious reflections. We employ the perfectly matched layer (PML) implemented in OVERTURE for that purpose. In general we note that various PMLs are used extensively by the computational electromagnetics community as they are known to be very efficient wave absorbers. Their disadvantage is that in some situations a PML may go unstable already for moderate computational times. *Lacunae-based time marching allows us to fix this undesirable behavior for both type I and type II numerical experiments.*

Two technical yet important remarks are in order:

- The PML in OVERTURE can be set only on a single grid. Since we are going to use the PML on multiple grids (specifically, on N_0 grids, see (120)) we modified the code accordingly.
- As OVERTURE solves Maxwell's equation in the second order form, the PML is applied to the 3D wave equation. The performance of the PML in this setting is not as thoroughly investigated as that for the original first order system. Hence, we may expect some differences in the wave absorption and stability properties compared to the first order setting. To guarantee that

this relatively unexplored second order PML still performs well and does not contaminate the lacunae, we will verify the results that it yields against genuine lacunae-based integration.

Let us briefly remind what the genuine (i.e., original) lacunae-based time marching is. Consider a source of waves located inside S and operating over the time interval T_0 . Once this source ceases to operate, all the waves it has generated will completely leave the domain S after the time $\text{diam}S/c$ elapses. In other words, the total “lifespan” of the solution on S before the onset of lacuna is $T_0 + \text{diam}S/c$. Therefore, provided that there is sufficient separation between the interior domain S and the auxiliary domain S_{aux} :

$$\text{dist}(\partial S, \partial S_{\text{aux}}) \geq T_0/2 + \text{diam}(S)/2c, \quad (131)$$

no wave that leaves the domain S , travels toward the outer boundary of S_{aux} , and gets reflected there, will be able to reenter S before the latter completely falls into the lacuna. On the other hand, once S is inside the lacuna, there is no need to advance the solution of the AP any further, and the computation can be stopped. Consequently, as long as the auxiliary domain S_{aux} is chosen according to (131), the solution on the interior domain S that is obtained with the help of lacunae appears completely insensitive to the treatment of the outer boundary ∂S_{aux} . One can also say that there is no error associated with the domain truncation, because all the outgoing waves leave S and there are no incoming waves until the solution on S reaches its zero steady state. Hence, the solution computed this way can provide a benchmark for any other approach, for example, the one that involves making the auxiliary domain S_{aux} smaller than prescribed by (131) and setting the PML at its outer boundary.

9.5 Type I experiments

In these experiments, we solve equation (109) driven by the external source (110), which is derived from (116)–(118).

9.5.1 Parameters of the computational setting. We use the following parameters for the experiments of type I:

- Half-size of the interior computational domain S : $a = 2.7$.
- Half-size of the auxiliary computational domain S_{aux} in the case where it is truncated by the PML: $b = 7.7$. The PML is applied from the inside of ∂S_{aux} , see Figure 57.
- Half-size of the auxiliary computational domain S_{aux} chosen according to (131) in the case where the genuine lacunae-based time marching is employed: $b = 17.7$.
- Width of the near-boundary layer S_ε : $\varepsilon = 1.05$.
- Operational time of each partial source $\eta_i(t)$: $T_0 = 16 \approx 1.7\text{diam}S$.
- Size of the overlap between consecutive $\eta_i(t)$: $\Delta T = 0.7 \times T_0$.
- Maximum number of auxiliary problems to be computed simultaneously: $N_0 = 2$.
- Parameter controlling the effective spatial support of the source: $\alpha = 1$.
- Frequencies of the source: $\omega_1 = 1$, $\omega_2 = \omega_1/\sqrt{2}$.

9.5.2 Grids. For the experiments of type I, we use two sets of grids that we call Grid 1 and Grid 2. The grids from the second set are twice as fine compared to those from the first set.

The set Grid 1 includes a three-dimensional Cartesian grid of $37 \times 37 \times 37$ nodes for the interior computational domain S . The grid size in each direction is the same: $h_x = h_y = h_z = 0.15$. The time step is $h_t = 0.05$. For the auxiliary domain S_{aux} in the case where it is terminated by the PML we use a grid of $97 \times 97 \times 97$ nodes. The width of the PML, which is adjacent to each face of S_{aux} from the inside, is 25 nodes. The PML region and the domain S are separated by a 5-node gap, see Figure 57. For the auxiliary domain S_{aux} in the case of the genuine lacunae-based time marching, the grid dimension in space is $237 \times 237 \times 237$ nodes. The width of the near-boundary layer S_ε , which is inside S next to its outer boundary ∂S , is 7 nodes (four nodes for the growing part of the ramp function μ and 3 nodes for its flat part). The boundary condition is derived from the solution to the AP and applied to the IP at the boundary point BP, see Figure 57.

The grids in the set Grid 2 are twice as fine as their respective counterparts in the set Grid 1. The resulting dimensions are obtained by setting the grid factor $n = 2$, see Figure 57, whereas for the set Grid 1 this factor is $n = 1$. The time step for Grid 2 is $h_t = 0.05/2 = 0.025$, and the spatial steps are $h_x = h_y = h_z = 0.15/2 = 0.075$.

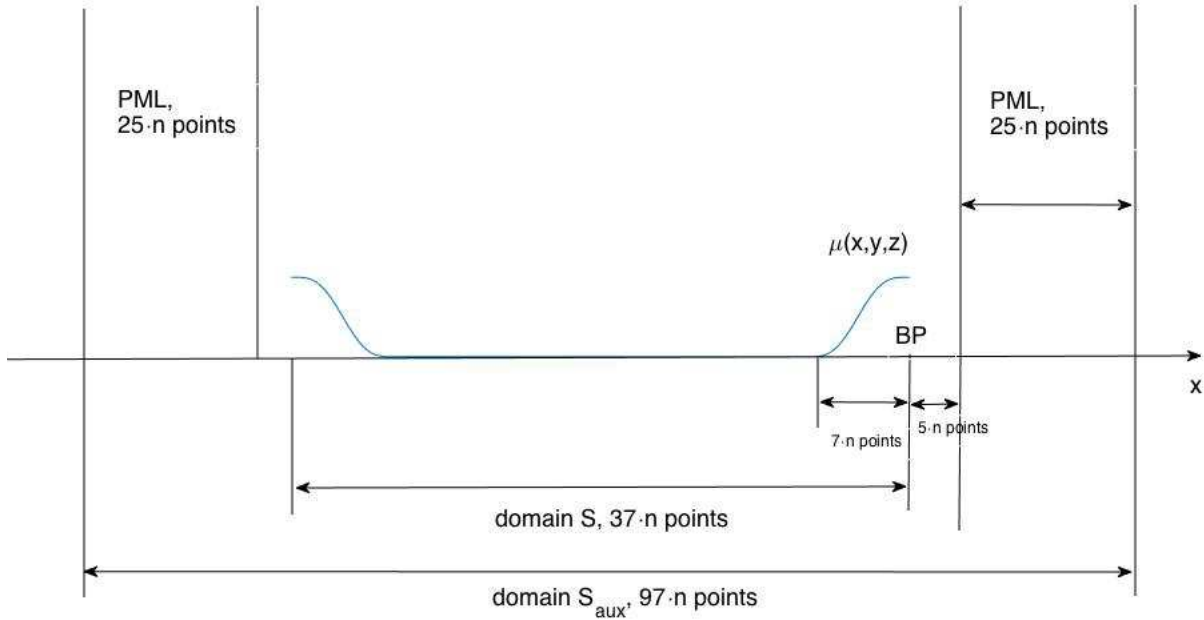


Figure 57: Grid structure. The cross-section of the entire computational domain along the x -axis.

9.5.3 PML design and parameters. We have conducted a thorough investigation into the optimal parameters of the PML implemented in *OVERTURE* (width of the layer, its strength, and shape of damping profile). By optimality we mean the minimal possible level of reflections combined with the stable behavior over a period of time required for the proper performance of the lacunae-based integration. As a result, we have chosen a fourth order profile for the damping coefficient:

$$A\xi^4,$$

where ξ varies from 0 to 1 in the PML. The optimal value of the PML strength A and the width of the layer (in grid nodes) appear closely related. After a series of runs, $A = 17$ along with a

25-node width were chosen as a decent compromise between the stability and absorption of the outgoing waves. Narrower PMLs (between 10 and 15 nodes) have demonstrated poorer absorption which could not be compensated for by increasing A due to a quickly developing instability. Of course, this choice of parameters may be sensitive to the specific problem formulation e.g., whether it involves experiments of type I or type II, what external sources are used, etc.

The domains S and S_{aux} are shown in Figure 58 using the built-in OVERTURE plotting tools.

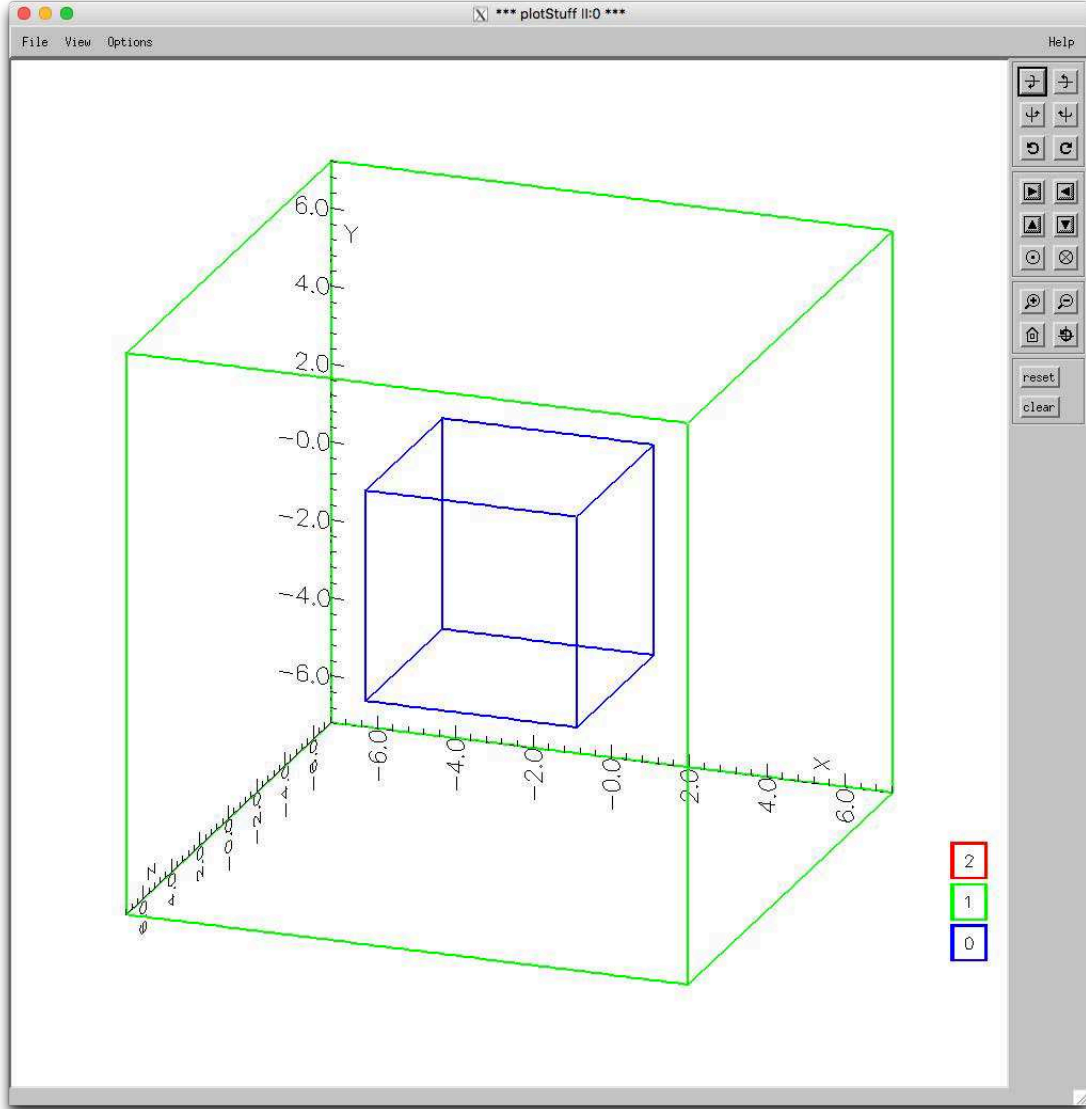


Figure 58: Domains S and S_{aux} in the case where the latter is terminated by the PML.

Figure 59 is a screenshot of the grid summary for the set Grid 1 provided by OVERTURE.. SmallBox corresponds to the domain S ; BigBox and BigBox2 correspond to the grids for two APs on S_{aux} . A “perfectElectricalConductor” boundary condition on the SmallBox grid is set only because OVERTURE requires to set some condition on every grid. The actual boundary condition for SmallBox was taken as the sum of the solutions to the two APs at the node BP, see Figure 57, in accordance with the design of the lacunae-based integration.

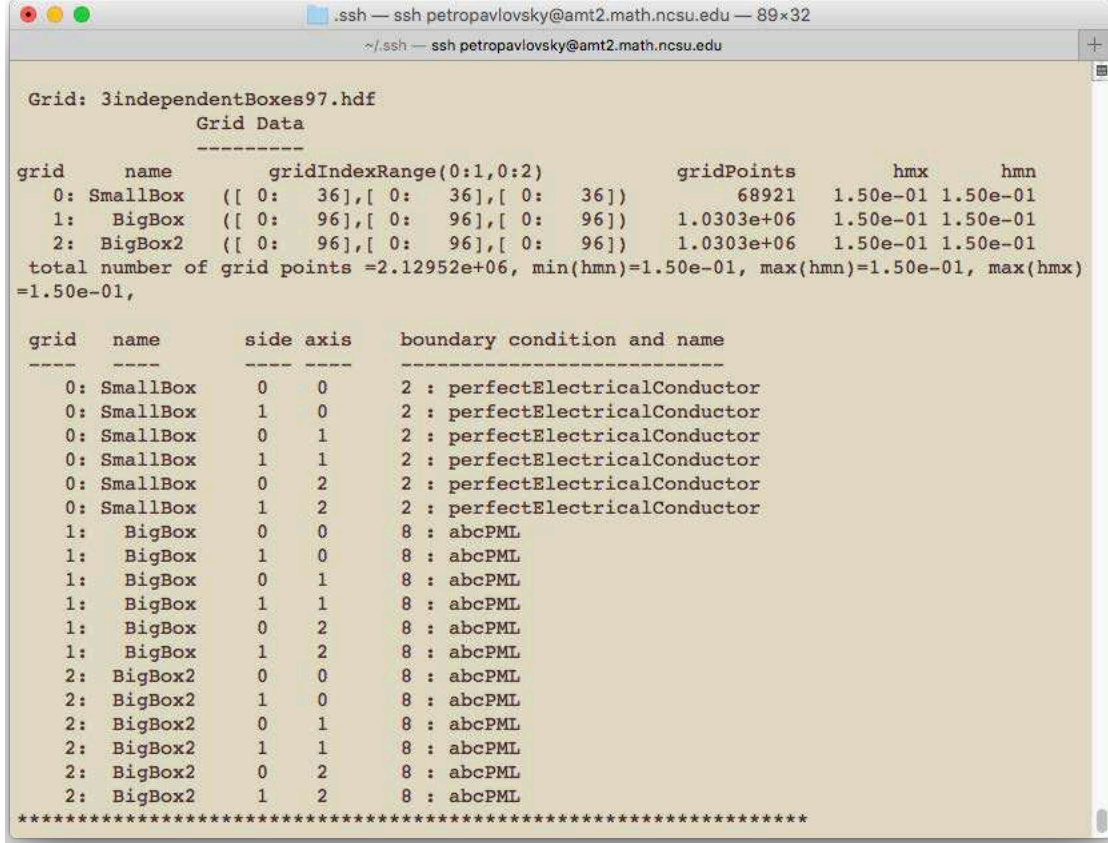


Figure 59: Grid summary for the set Grid 1.

9.5.4 Results of computations. In this section, we present the error profiles of the electric field components generated by the external sources (110), (116)–(118). We perform the full 3D simulations that do not rely on the axial symmetry of the test. As a measure of the error, we use the ∞ -norm:

$$\frac{\max_{i,j,k} |E_{i,j,k} - E_{i,j,k}^{\text{ref}}|}{\max_{i,j,k} |E_{i,j,k}|} \quad (132)$$

where $E_{i,j,k}$ is the field component E_y or E_z evaluated at the grid node (i, j, k) . The maxima in (132) are taken over the computational domain S for the IP. The error (132) is computed with respect to the reference solution $E_{i,j,k}^{\text{ref}}$ given by formulae (111)–(112) and recast into the Cartesian system. There is no need to compute the error of E_x separately because it is equal to the error of E_y at any moment of time. This can be easily understood from the symmetry considerations specific to the reference solution (111)–(112). Indeed, E_x and E_y are the projections of the planar component E_{xy}

$$E_{xy} = E_r \sin \theta + E_\theta \cos \theta$$

onto the x and y axes, respectively:

$$E_x = E_{xy} \cos \varphi, \quad E_y = E_{xy} \sin \varphi. \quad (133)$$

Since the test is axially symmetric and E_{xy} does not depend on the azimuthal angle φ , any norm of both quantities in (133) taken over the domain centered at the origin has the same value.

In Figure 60, we compare two types of boundary closure for the problem: (i) the pure PML approach when the problem is integrated on the domain S_{aux} which is truncated by means of the PML with no use of no lacunae and (ii) the lacunae-based algorithm combined with the same PML. The parameters of the PML are the same in both cases, see Section 9.5.3. Figure 60 clearly indicates the pure PML approach leads to an abrupt divergence after a relatively short period of stable behavior. The error in Figure 60 is bounded by 1 because we use the computed solution rather than the reference solution in the denominator of (132). The computed solution $E_{i,j,k}$ itself tends to infinity, i.e., blows up, in this situation. Yet the lacunae-based algorithm provides a perfect correction for the instability developing in the PML. The computational time T in these runs corresponds to about 120 diameters of S . Over this time, about 60 partial APs were “recycled.”

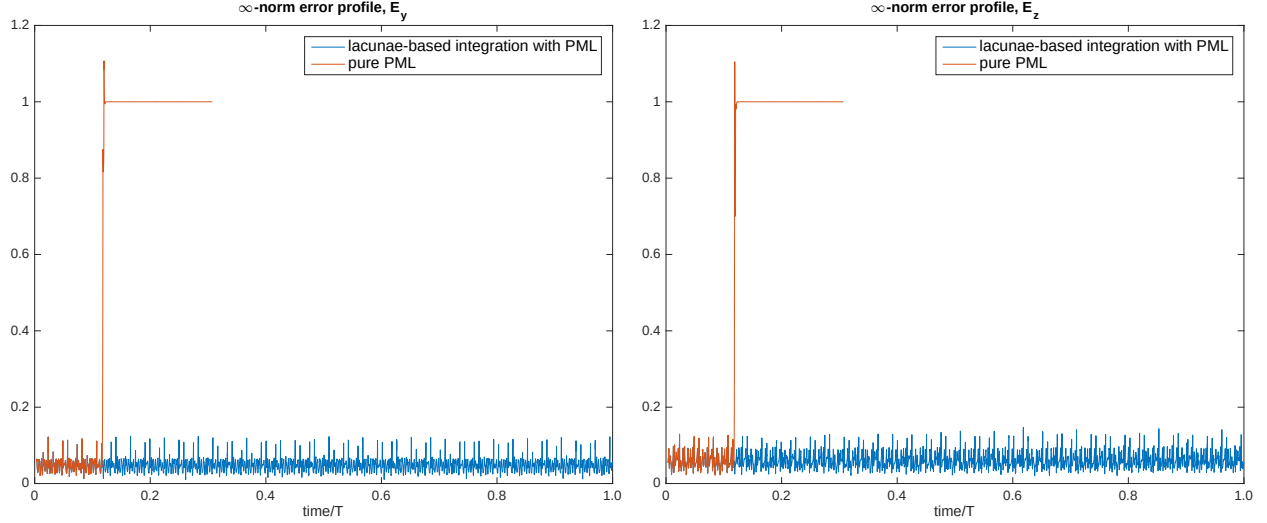


Figure 60: Left: error profile of E_y (∞ -norm) computed under the pure PML approach and the lacunae-based integration. Right: the same for E_z .

In Figure 61, we demonstrate the second order grid convergence of the algorithm that involves the lacunae-based integration combined with the PML.

In Figure 62, we investigate the quality of the PML we have employed by evaluating the level of the residual numerical noise in the lacunae. Actually, Figure 62 shows the evolution of the maximum of the solution to a given partial AP (specifically, the second AP) over its lifespan $T_0 + \text{diam}S/c$. The maximum is taken over the interior computational domain S . In accordance with the theory, the aforementioned maximum of the solution should become small after the source ceases to operate and the waves it has generated leave the domain S . Hence, the quantity of interest is precisely the magnitude of the residual solution inside the lacuna. It may depend, in particular, on the extent of spurious reflections due to the chosen treatment of the artificial outer boundary.

As a benchmark that allows us to assess the quality of the lacunae, we use the numerical solution obtained for the same setting but with the help of the genuine lacunae-based time marching (no PML), see Section 9.4.2.2. In this case, the auxiliary domain S_{aux} must be chosen sufficiently large, see formula (131). In practice, we take $b = 17.7$. Then, the waves reflected from the outer boundary ∂S_{aux} do not reach the domain S over the lifespan of the AP. The domain S_{aux} is truncated with the PEC boundary conditions because those boundary conditions do not matter in this setting.

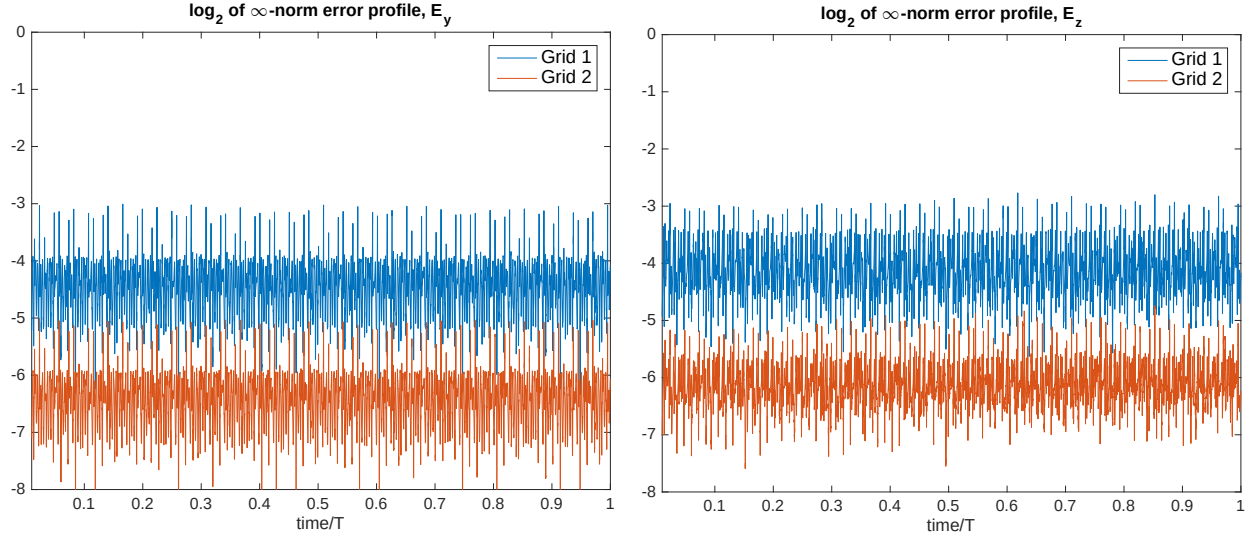


Figure 61: Left: error profiles for E_y (binary logarithm of ∞ -error norm) on two successive grids. Right: the same for E_z .

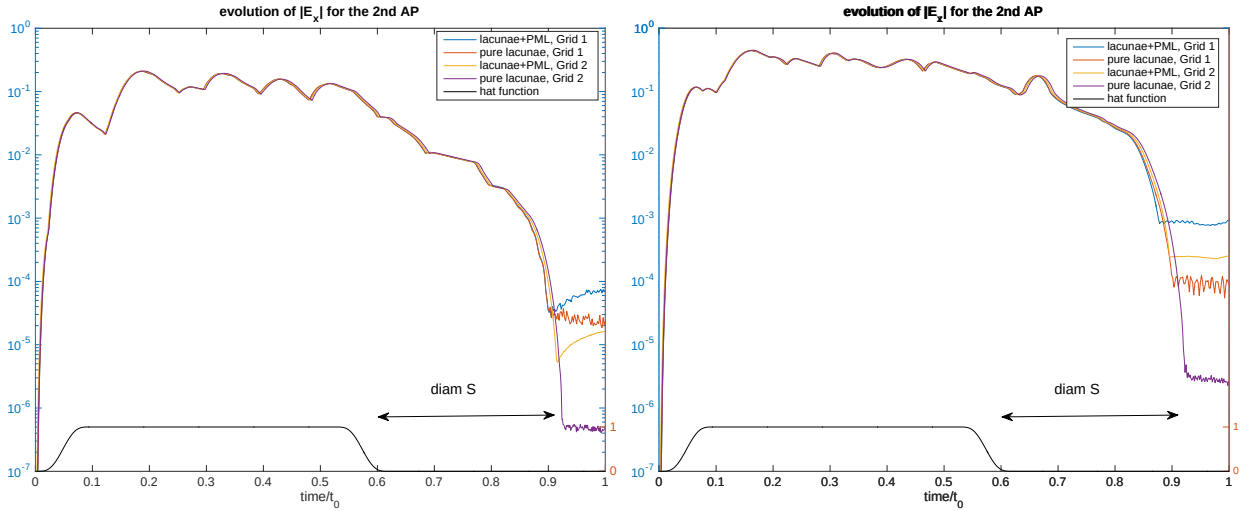


Figure 62: Left: evolution of $\max |E_y|$ in time for the 2nd AP computed using the combined PML+lacunae approach, as well as using the original lacunae-based integration, on two successive grids, Grid 1 and Grid 2. Right: the same for $|E_z|$.

Figure 62 suggests that the PML built into OVERTURE is not a perfect wave absorber. Compared to the pure lacunae runs on the same grids, it leads to about one order of magnitude higher residual solution inside the lacuna for Grid 1 and to even worse result for Grid 2 which is twice as fine. The super convergence inside the lacunae is basically absent in the simulations that involve lacunae+PML whereas it is clearly present in the case of the pure lacunae-based integration.

9.6 Type II experiments

In this section, we consider external scattering by perfectly electrically conducting (PEC) bodies. For our simulations we choose two scattering shapes. The first one is a sphere, and the second one is a geometrically more complex body of revolution that schematically represents a reentry vehicle.

The excitation in the scattering problems is provided by a given impinging wave. It is taken in the form of a linearly polarized (along the y axis) monochromatic transverse plane wave of unit amplitude:

$$\mathbf{E} = \mathbf{e}_y \cos(\omega t - kx). \quad (134)$$

For $k > 0$, the wave (134) propagates in the positive x direction. The wavelength in (134) is assumed equal to one. The characteristic size of both scattering shapes that we are going to consider, the sphere and the reentry vehicle, will also be of order one. This rules out the use of any asymptotic solutions (either the geometrical optics for short waves or the diffraction of long waves on a small obstacle) and requires a direct numerical simulation of the scattered field.

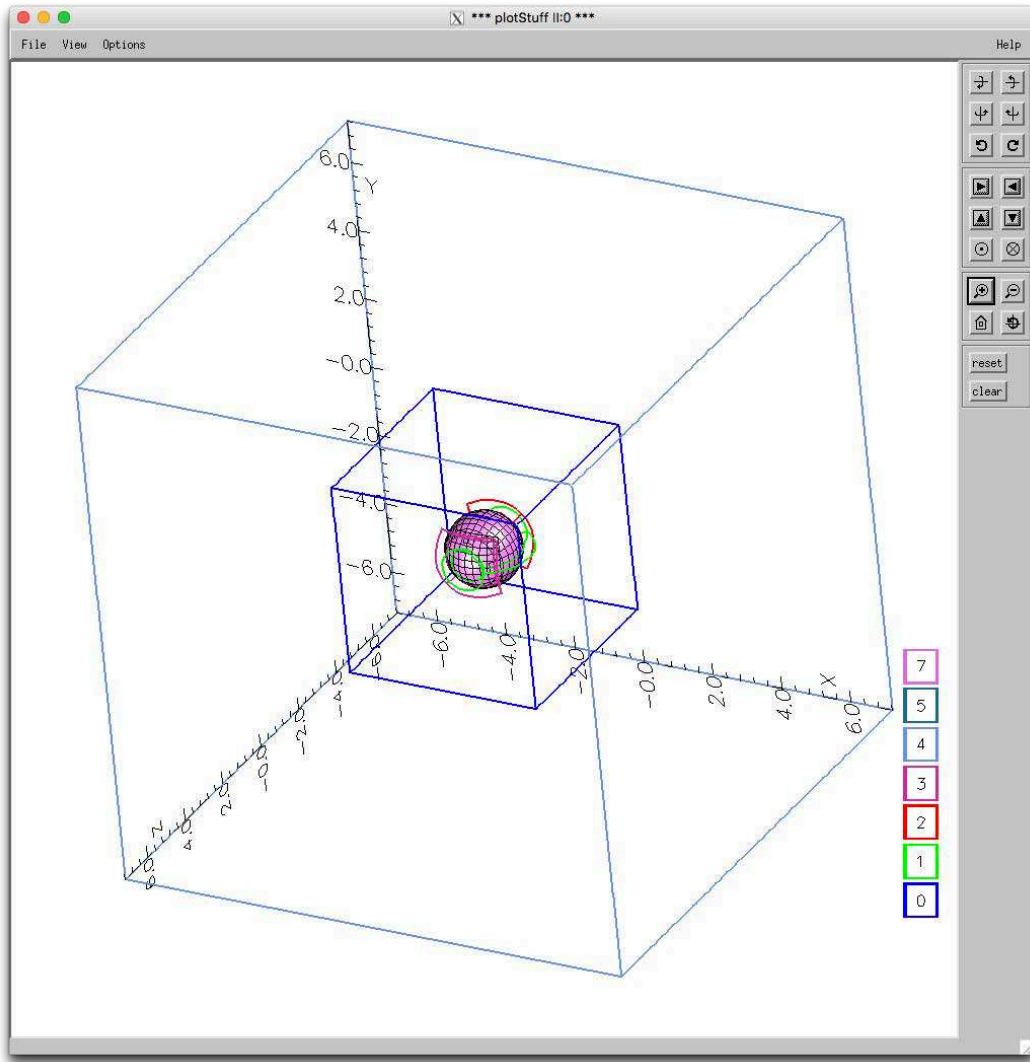
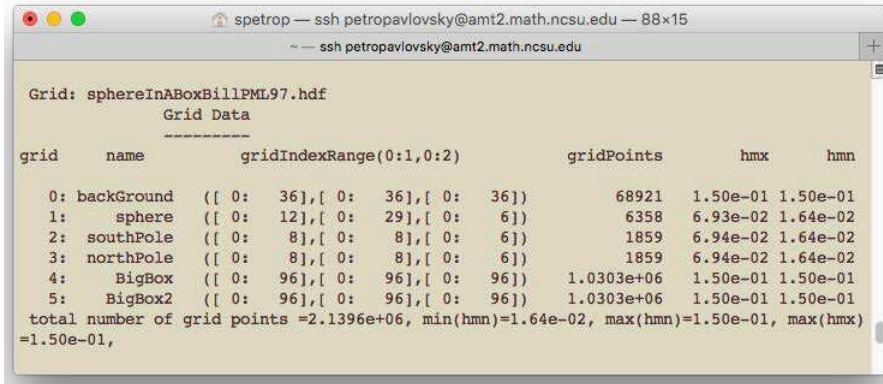


Figure 63: Computational domains for the sphere.

The sphere of radius one is embedded into the domain S and centered at the origin, see Figure 63. The scattering process is modeled by imposing the PEC boundary conditions on the sphere, which is done by a special built-in subroutine in **OVERTURE**. The PEC boundary conditions require that the tangential component of the electric field be equal to zero. This is equivalent to having the tangential component of the scattered field equal to minus the tangential component of the impinging field (134). The initial conditions are taken homogeneous for simplicity. This is inconsistent with the boundary data. Hence, we expect to see a transition period in the simulations, after which the solution will go into a harmonic oscillation mode.

The interior and auxiliary domains S and S_{aux} for the experiments of type II are chosen exactly the same as those for the experiments of type I, see section 9.5.1. An important new construct for the experiments of type II is the sphere itself. It is represented in **OVERTURE** by means of two special mappings called patches. The latter are discretized by overlapping grids.

To investigate the grid convergence for scattering problems, we use three sets of grids, Grid 1, Grid 2, and Grid 3. The grid sizes for S and S_{aux} in Grid 1 are the same as for type I experiments, see Section 9.5.2. The structure of the overlapping grids that define the sphere is chosen as suggested by the command file **scattering.cmd** in **OVERTURE**. The sets Grid 2 and Grid 3 are clones of Grid 1 refined by a factor of 2 and 4, respectively. The grid summary for each of these grid sets is presented in Figures 64-66.

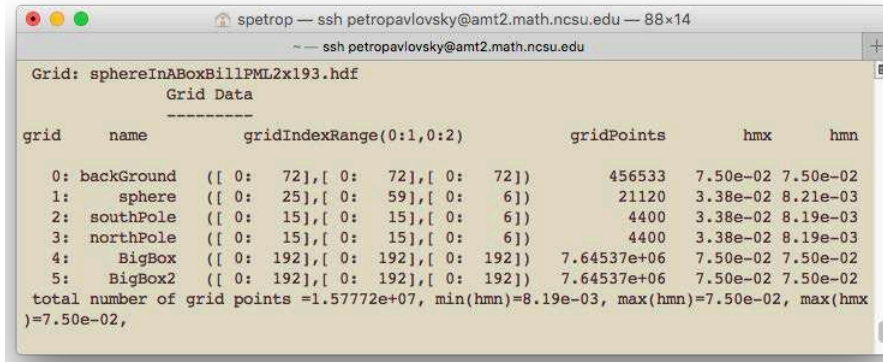


```

Grid: sphereInABoxBillPML97.hdf
Grid Data
-----
grid      name      gridIndexRange(0:1,0:2)      gridPoints      hmx      hmn
0: backGround ([ 0: 36],[ 0: 36],[ 0: 36])      68921      1.50e-01 1.50e-01
1: sphere ([ 0: 12],[ 0: 29],[ 0: 6])      6358      6.93e-02 1.64e-02
2: southPole ([ 0: 8],[ 0: 8],[ 0: 6])      1859      6.94e-02 1.64e-02
3: northPole ([ 0: 8],[ 0: 8],[ 0: 6])      1859      6.94e-02 1.64e-02
4: BigBox ([ 0: 96],[ 0: 96],[ 0: 96])      1.0303e+06      1.50e-01 1.50e-01
5: BigBox2 ([ 0: 96],[ 0: 96],[ 0: 96])      1.0303e+06      1.50e-01 1.50e-01
total number of grid points =2.1396e+06, min(hmn)=1.64e-02, max(hmn)=1.50e-01, max(hmx)=1.50e-01,

```

Figure 64: Grid summary for the set Grid 1.



```

Grid: sphereInABoxBillPML2x193.hdf
Grid Data
-----
grid      name      gridIndexRange(0:1,0:2)      gridPoints      hmx      hmn
0: backGround ([ 0: 72],[ 0: 72],[ 0: 72])      456533      7.50e-02 7.50e-02
1: sphere ([ 0: 25],[ 0: 59],[ 0: 6])      21120      3.38e-02 8.21e-03
2: southPole ([ 0: 15],[ 0: 15],[ 0: 6])      4400      3.38e-02 8.19e-03
3: northPole ([ 0: 15],[ 0: 15],[ 0: 6])      4400      3.38e-02 8.19e-03
4: BigBox ([ 0: 192],[ 0: 192],[ 0: 192])      7.64537e+06      7.50e-02 7.50e-02
5: BigBox2 ([ 0: 192],[ 0: 192],[ 0: 192])      7.64537e+06      7.50e-02 7.50e-02
total number of grid points =1.57772e+07, min(hmn)=8.19e-03, max(hmn)=7.50e-02, max(hmx)=7.50e-02,

```

Figure 65: Grid summary for the set Grid 2.

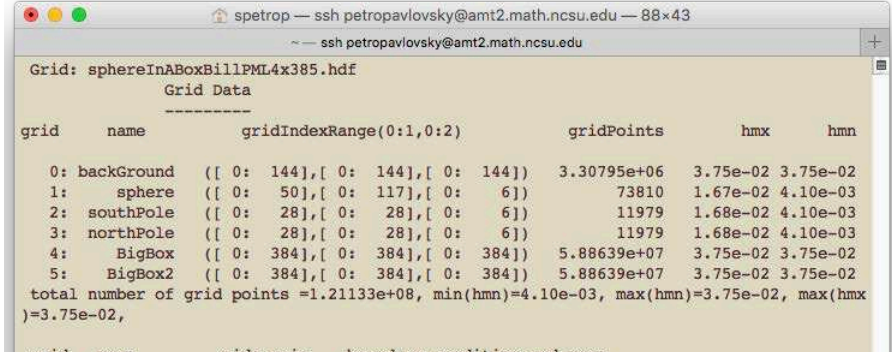


Figure 66: Grid summary for the set Grid 3.

It is important to note that the overlapping grids that we employ to discretize the sphere may lead to an unstable behavior of the algorithm even in its original setting, i.e., before we add any lacunae-based routines. To suppress the instability, **OVERTURE** introduces artificial dissipation, which is generally small and can be controlled by the user. In Figure 67, we demonstrate the unstable behavior at zero artificial dissipation with or without the lacunae-based integration. Specifically, we show the reflected field at the spatial location $(-2.5, 0, 0)$ in front of the sphere. After a short period of stable performance, it begins to grow rapidly, which is a clear indication of instability.

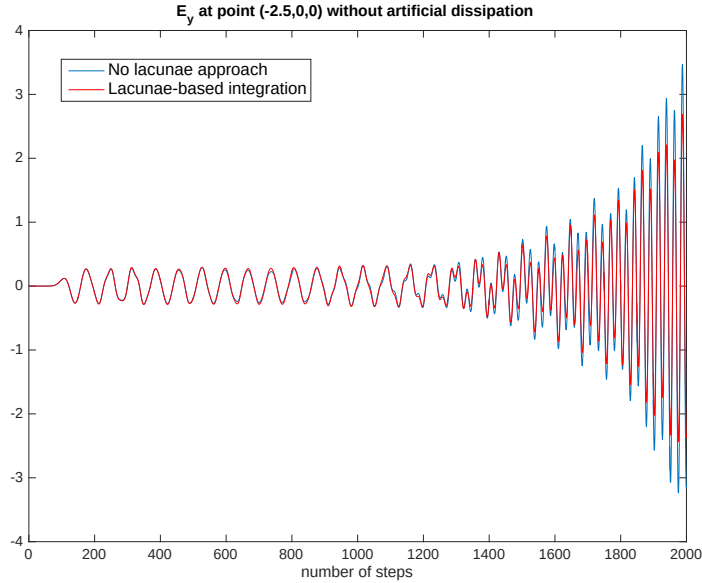


Figure 67: The reflected field E_y at the point $(-2.5, 0, 0)$ in the absence of artificial dissipation.

To suppress the rapid growth observed in Figure 67, one needs to switch on the artificial dissipation. The non-zero dissipation terms in the governing equations make them different from the pure 3D wave equation. Yet we are still using the pure 3D d'Alembert operator to create the partial sources (122). This mismatch will hamper the performance of the lacunae-based algorithm if the dissipation is set everywhere on S including the near-boundary region S_ϵ where the partial sources are localized. Of course, we could have changed the definition of the partial sources (122) and the AP by introducing the dissipative terms, but then the AP would not have the true lacunae

because the dissipation leads to the diffusion of waves. Instead, we make the dissipation non-zero not everywhere on S but only in a small vicinity of the scatterer. Outside this small region, i.e., relatively far away from the sphere, we use no artificial dissipation. As such, the IP takes the form appropriate for application of the lacunae-based algorithm: some complex equations inside S that turn into the 3D wave equation near the boundaries. In Figure 68 (left panel), we show what happens if the artificial dissipation is non-zero everywhere. Specifically, we plot the reflected field E_y at the point $(-2.5, 0, 0)$. The result is obviously incorrect, there is some modulation which is inconsistent with the expected purely oscillatory behavior of the field at any spatial location. On the right panel of Figure 68, we show the computation where the dissipation is set only next to the sphere. One can clearly see the expected pure oscillations. Notice also the difference in amplitudes. Both plots correspond to the lacunae-based approach combined with PML at the outer boundary of the AP. The profile of the 25-node PML is defined in Section 9.5.3 with the strength $A = 17$.

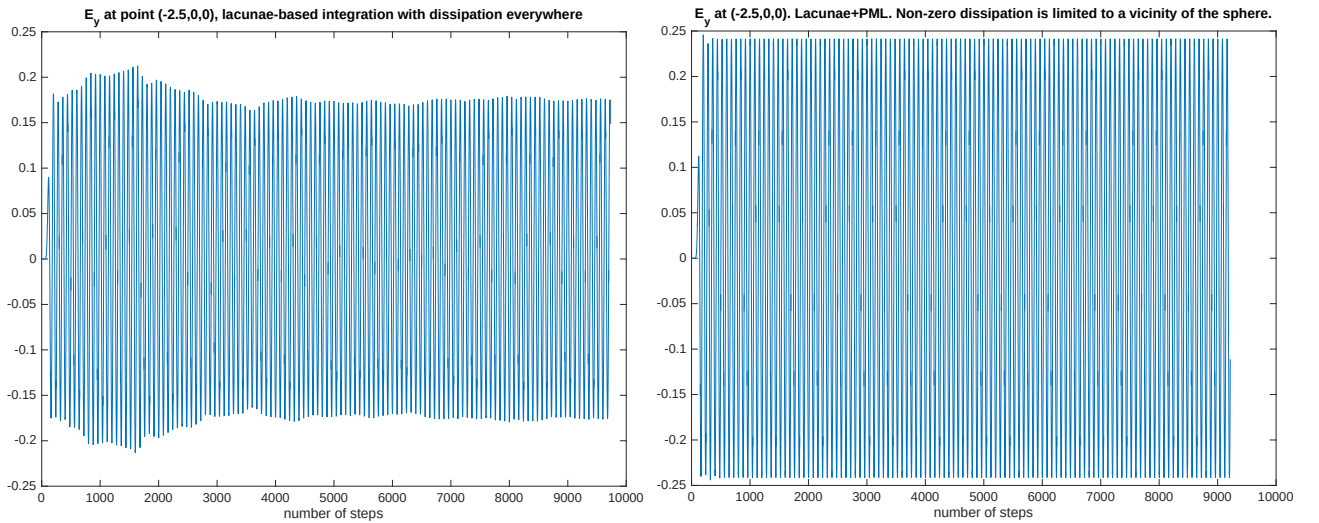


Figure 68: Left: the reflected field E_y at the location $(-2.5, 0, 0)$; lacunae-based integration+PML at the outer boundary of the APs. Artificial dissipation is used everywhere on S . Right: the same but with non-zero dissipation only near the sphere.

In Figure 69, we demonstrate the instability that develops in this scattering problem when the PML is used as the only means for truncating the computational domain, i.e., without the lacunae-based time marching. The strength of the PL is taken as $A = 83$. In the same figure, we present the results for the combined lacunae+PML approach, which clearly removes the blow-up behavior. To corroborate that the performance remains stable over long times, we run the same lacunae+PML computation ($A = 83$) for the total time T that corresponds to about 120 diameters of S . The results are presented in Figure 70. We observe stable oscillations with no signs of divergence.

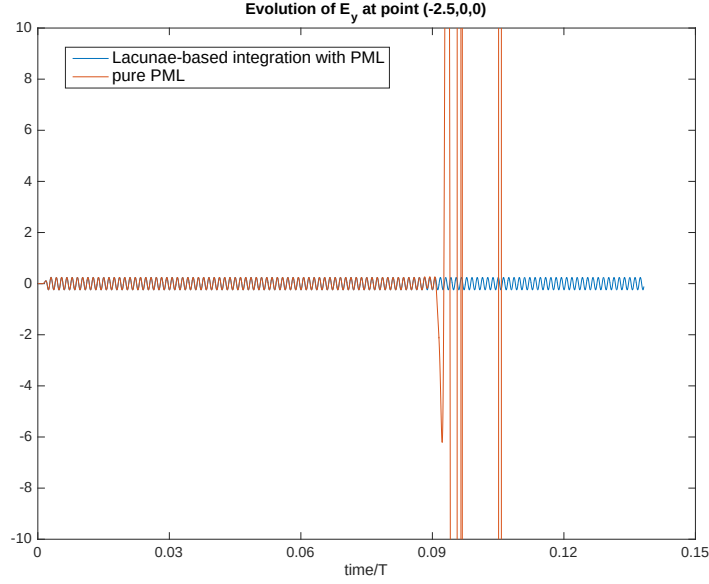


Figure 69: The reflected field E_y at the location $(-2.5, 0, 0)$. Pure PML and the combined PML+lacunae approach.

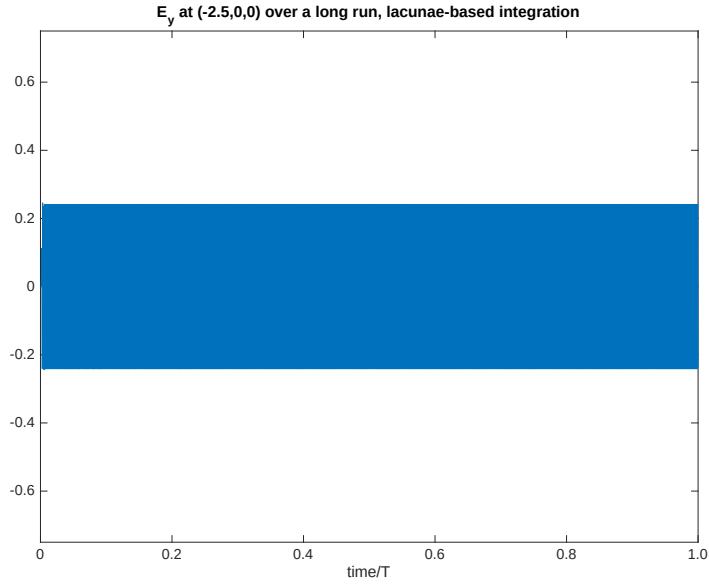


Figure 70: The reflected field E_y at the location $(-2.5, 0, 0)$ over a long run. Lacunae-based integration combined with the PML.

Next, we demonstrate the grid convergence of the algorithm. The exact solution for the scattering about a sphere is known but it is given by an infinite series that is not easy to use as a reference for numerical simulations. That is why we study the convergence in the sense of fundamental sequences (i.e., in the sense of Cauchy). Namely, we use the following quantity:

$$\max_{\mathbf{x}} \frac{|E^{m+1}(\mathbf{x}) - E^m(\mathbf{x})|}{|E^{m+2}(\mathbf{x}) - E^{m+1}(\mathbf{x})|} \quad (135)$$

as a measure of grid convergence. The superscript “ m ” in formula (135) indicates the number of the consecutive grid (Grid 1, Grid 2, or Grid3). The grid functions E^m , E^{m+1} , E^{m+2} that correspond to different grids must be taken at the same spatial location $\mathbf{x}$, and the maximum in (135) is then taken over the domain S less the sphere.

Let us first visualize the process of convergence. In Figure 71, we plot the reflected field E_y at the spatial location $(-2.5, 0, 0)$ on three consecutive grids. The overall computational time for Figure 71 is limited by the large dimension of Grid 3, which makes the simulations on a single processor almost prohibitively expensive.⁸ Yet we have still been able to compute a sufficient number of oscillations. Figure 71 shows that the difference between the solutions on the third and second grids is much smaller than the difference between the first and the second grids, which is a qualitative indication of convergence. The number of time steps shown in Figure 71 corresponds to the coarsest grid, which is the set Grid 1. For the sets Grid 2 and Grid 3, this would correspond to every other step and every fourth step, respectively.

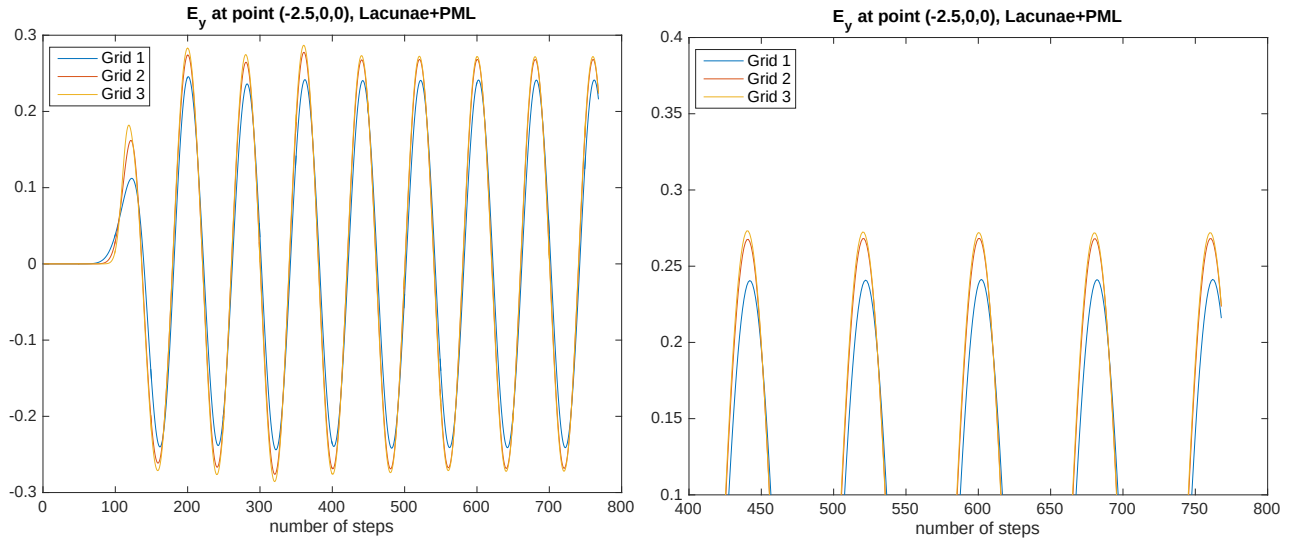


Figure 71: Left: the reflected field E_y at point $(-2.5, 0, 0)$ on three successive grids. Lacunae-based integration+PML. Right: an enlarged fragment of the left panel.

Next, we use the Cauchy-type measure (135) to actually quantify the grid convergence. In that regard we note that ideally the value of (135) for a second order accurate scheme should be equal to four. In general, the higher this value, the more rapid the convergence.

In Figure 72, we plot the quantity (135) for several hundred time steps. We see that it does not remain flat, it rather oscillates. Initially, one can observe a few slowdowns of convergence, when the curve in Figure 72 is below the target value of 4. However, as the time elapses those drops in the convergence rate all but disappear. The deterioration of convergence at an early stage of computation may be attributed to the inconsistency between the non-zero boundary data and zero initial data of the problem. This is basically a discontinuity at $t = 0$ that may noticeably affect the solution in the beginning but should “wear off” later on. As an argument in favor of this explanation, notice that the time when the amplitude in left panel of Figure 71 gets stabilized (about 400 steps) is approximately the same as the time in Figure 72 by which the convergence

⁸In the future, we hope to be able to use a multi-processor implementation of OVERTURE. It is generally available, but has not yet been ported to NCSU servers.

drops disappear. On the other hand, in Figure 72 we also observe frequent convergence peaks that are well above the design rate of 4. This behavior may due to a very coarse grid in the set Grid 1.

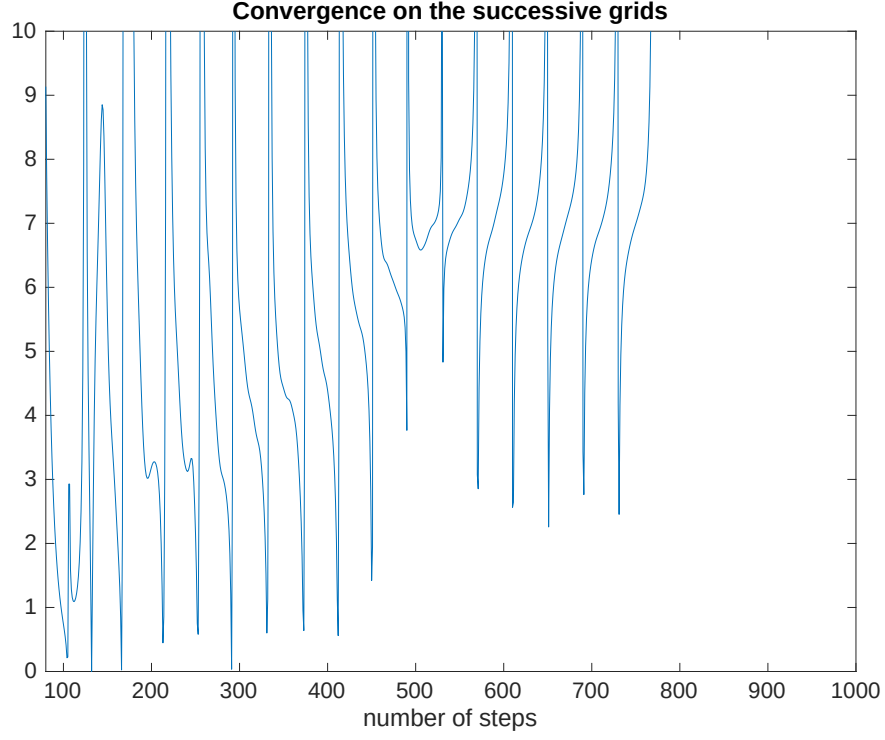


Figure 72: Convergence of E_y on three consecutive grids.

To check whether the absorption capabilities of the PML built into **OVERTURE** depend on the specific formulation, we present the evolution of the solution to the second AP over its lifespan t_0 , see Figure 73. Unlike in the same tests for type I experiments (Figure 62), we observe a superconvergence inside the lacuna between Grid 1 and Grid 2. Yet when going to Grid 3, the convergence inside the lacuna still slows down.

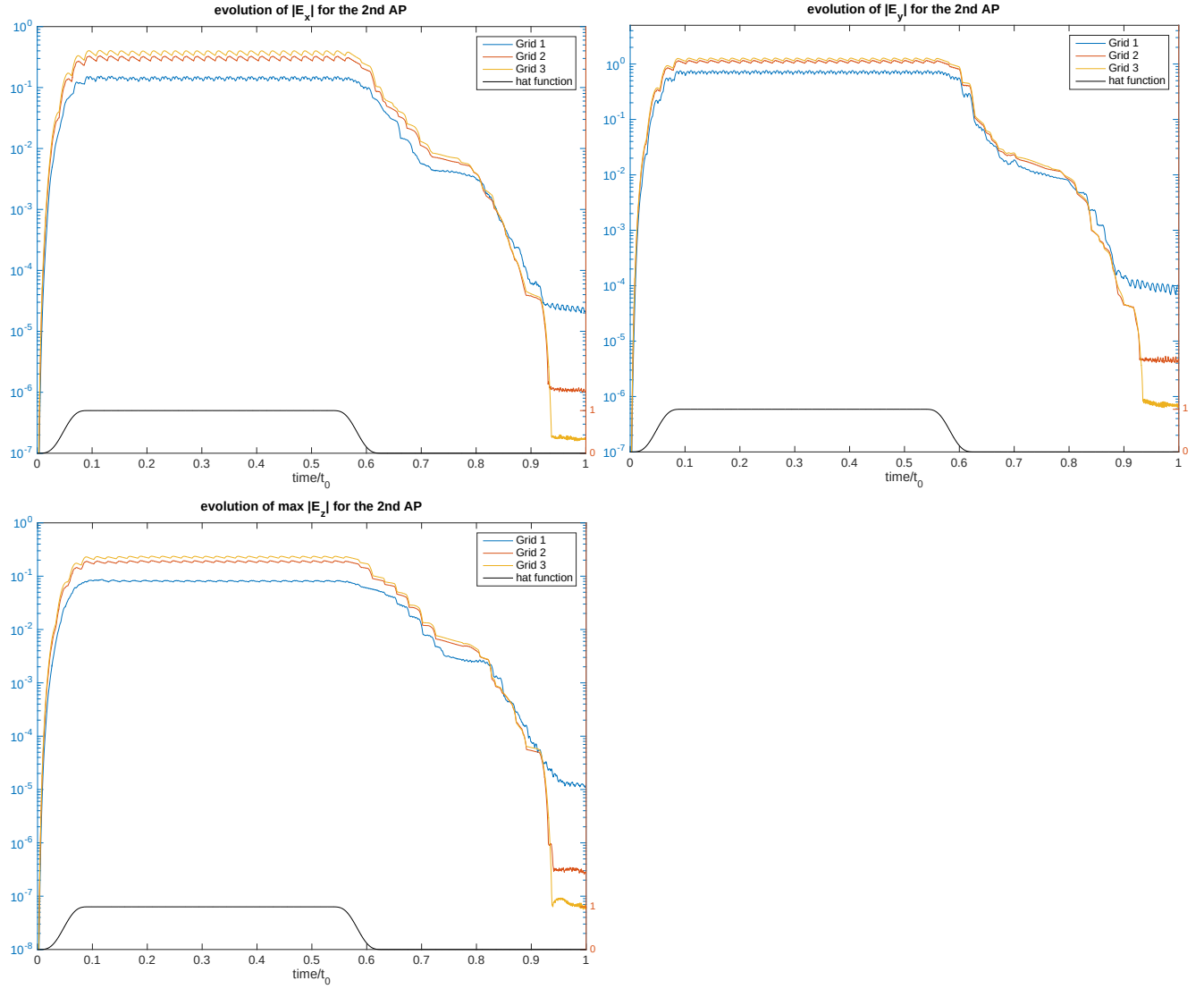


Figure 73: Maximum values of $|E_x|$, $|E_y|$, and $|E_z|$ for the 2nd AP on three consecutive grids. Lacunae-based integration+PML.

In Figure 74, we are showing a 3D plot of the scattered field at the moment of time $t = 0.03T$, where T is large, about 120 diameters of S . This time is beyond the initial period of transient behavior. The plane wave impinges on the sphere from the right along the x axis. The actual plot shown in Figure 74 was done in MATLAB using the numerical data saved in OVERTURE.

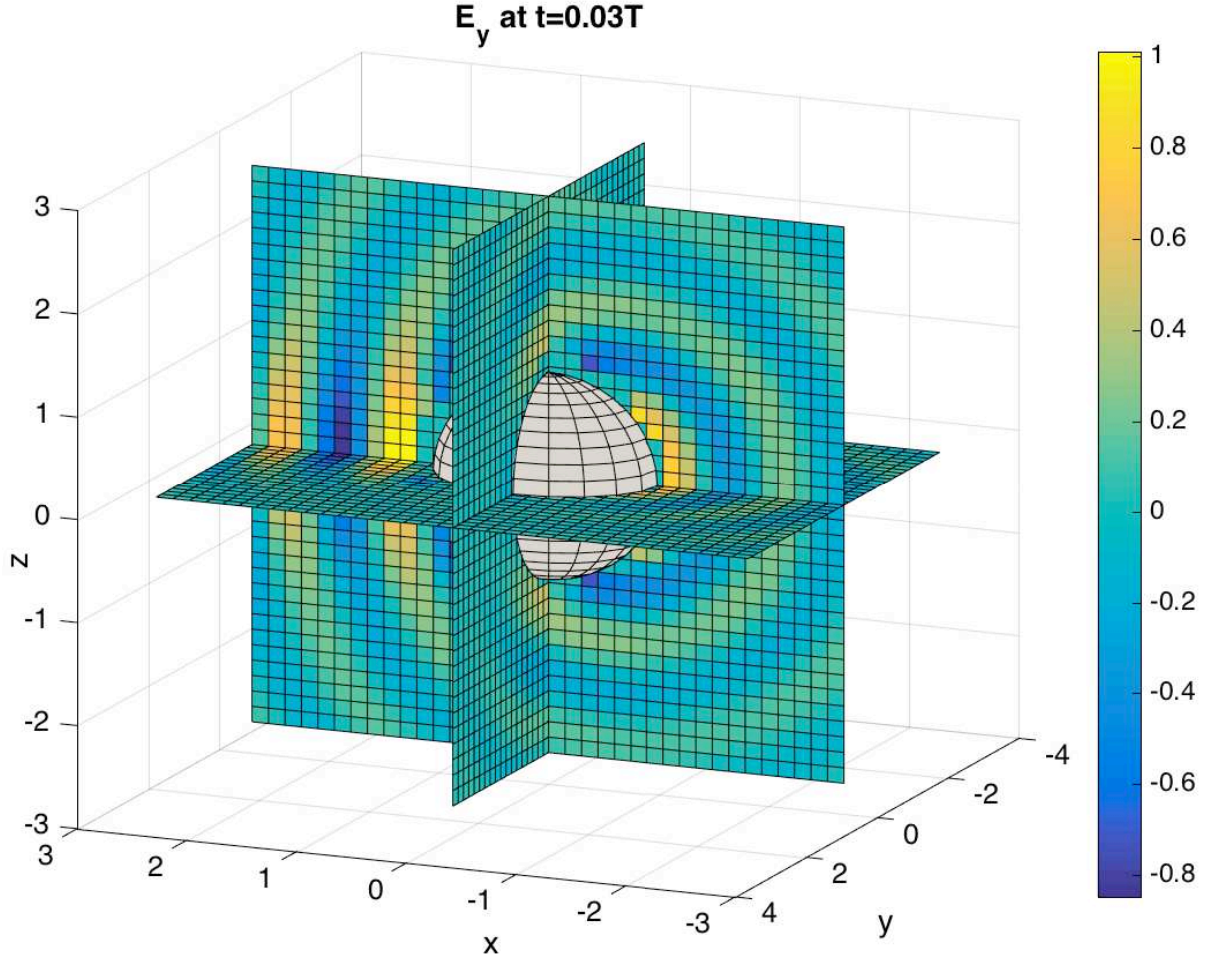


Figure 74: Scattered field E_y on S at the time $t = 0.03T$. The set Grid 1 is used.

Next, we describe numerical simulation of the scattering of a plane impinging wave about a PEC body of revolution that represents a model reentry vehicle, see Figure 75. The computational setup is schematically shown in Figure 76. The domain S is defined as a box with dimensions $[-4, 5] \times [-4, 5] \times [-4, 5]$; it is discretized by a $61 \times 61 \times 61$ grid. The auxiliary domain S_{aux} is a box of dimension $[-8, 10] \times [-8, 10] \times [-8, 10]$ discretized by a $121 \times 121 \times 121$ grid. Inside S , there is a body of revolution built with the help of the appropriate **OVERTURE** tools and discretized by means of the overlapping grids, see Figures 76–77. The domain S is not centered at the origin because the scatterer is displaced from zero along the x -axis (under the current implementation of this object in **OVERTURE**). Figure 78 provides a summary for all the grids used. A monochromatic plane wave of unit wavelength impinges on the body from the right, i.e., $k = -2\pi$ in formula (134) so that the wave travels in the negative x direction. This plane wave is polarized along the y axis. As before, the artificial dissipation is introduced only near the scatterer rather than on the entire do-

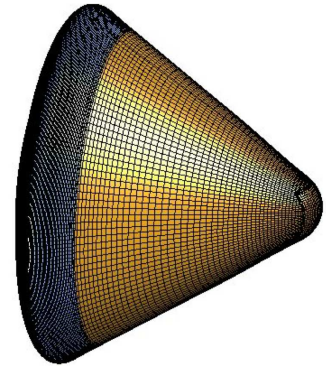


Figure 75: Reentry vehicle.

main S . A 25-node PML is set at the artificial outer boundary in accordance with Section 9.5.3; the strength of the PML is $A = 58$.

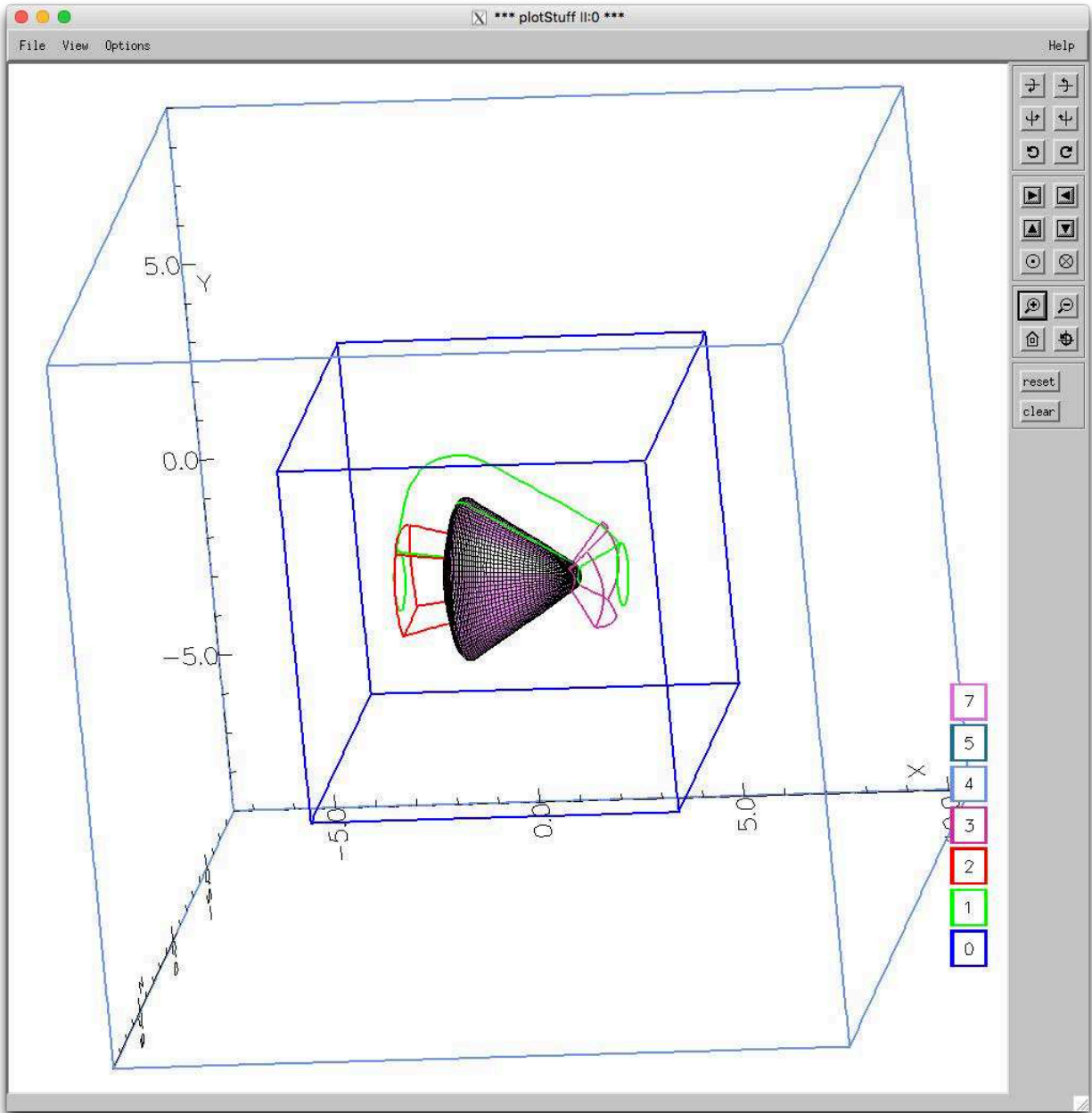


Figure 76: Computational domain for the reentry vehicle.

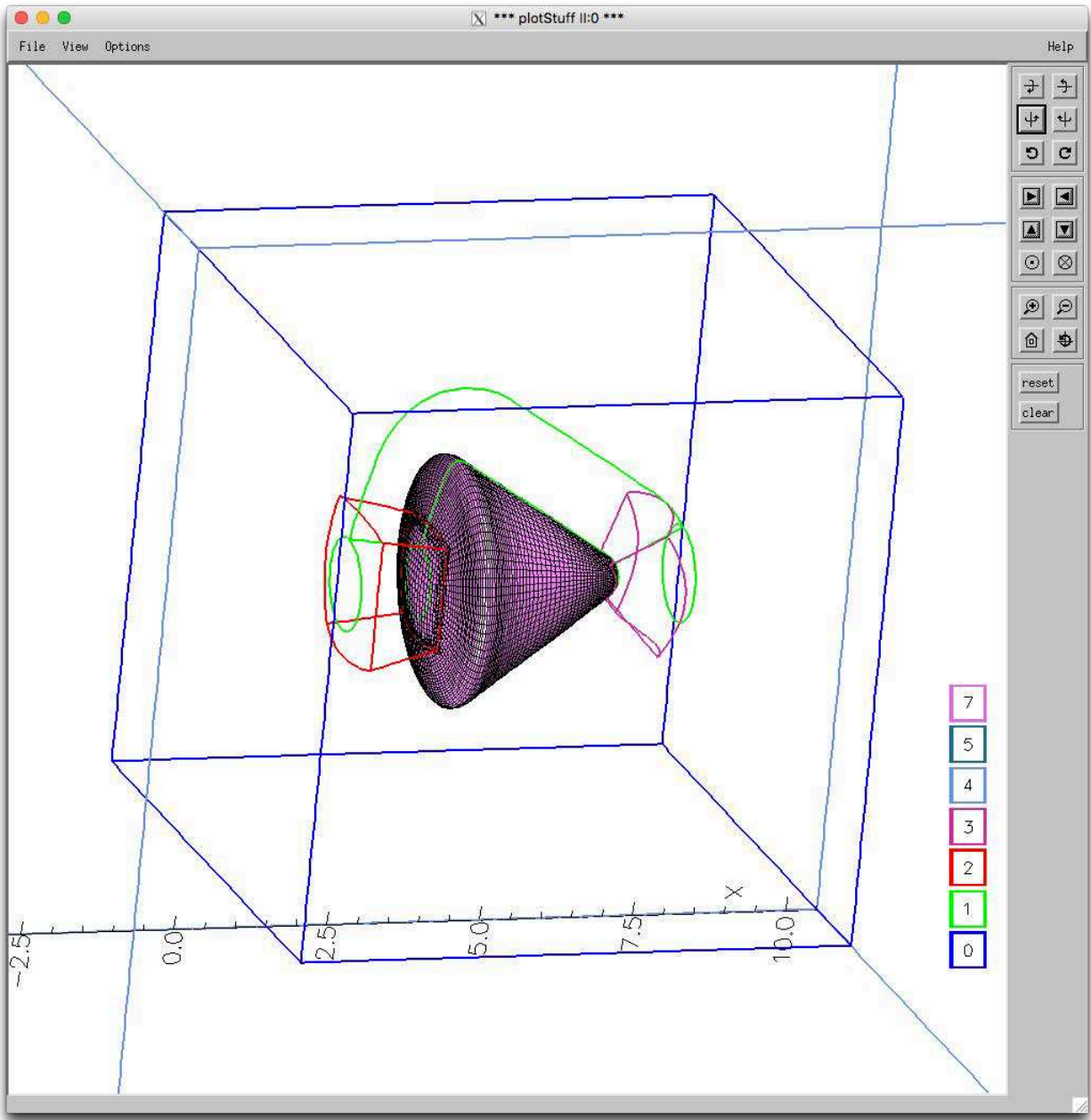


Figure 77: A zoomed in view for the reentry vehicle.

```

spetrop — cs@precise64: ~/over2/Overture.v25d/sampleGrids — ssh -p 2222 cs...
~ — cs@precise64: ~/over2/Overture.v25d/sampleGrids — ssh -p 2222 cs@localhost -X -C

Grid Statistics for grid 0, name=backGround. (rectangular) (mapping=Box
Mapping)
grid lines : [0:60,0:60,0:60], total points = 226981
ghost points: [2:2,2:2,2:2]
cell volume : 3.37e-03
grid spacing: [1.50e-01,1.50e-01,1.50e-01] [dx,dy,dz]
bc = [1,2, 3,4, 5,6] share=[0,0, 0,0, 0,0] isPeriodic=[0,0,0] dw=3

Grid Statistics for grid 1, name=hull-nurbs. (curvilinear) (mapping=Nur
bsMapping)
grid lines : [0:72,0:84,0:20], total points = 120960
ghost points: [2:2,2:2,2:2]
cell volumes: [1.17e-05,6.67e-04,8.70e-03] [min,ave,max]
grid spacing: [2.69e-02,8.17e-02,3.83e-01] [min,ave,max] (r1)
: [1.42e-02,1.14e-01,2.36e-01] [min,ave,max] (r2)
: [1.41e-02,6.00e-02,1.04e-01] [min,ave,max] (r3)
number of negative volumes = 0
bc = [0,0, -1,-1, 7,0] share=[0,0, 0,0, 7,0] isPeriodic=[0,2,0] dw=3

Grid Statistics for grid 2, name=hull-bow-cap-nurbs. (curvilinear) (map
ping=NurbsMapping)
grid lines : [0:20,0:20,0:20], total points = 8000
ghost points: [2:2,2:2,2:2]
cell volumes: [6.10e-05,6.35e-04,2.16e-03] [min,ave,max]
grid spacing: [6.84e-02,9.92e-02,1.46e-01] [min,ave,max] (r1)
: [6.84e-02,9.92e-02,1.46e-01] [min,ave,max] (r2)
: [1.41e-02,6.00e-02,1.04e-01] [min,ave,max] (r3)
number of negative volumes = 0
bc = [0,0, 0,0, 7,0] share=[0,0, 0,0, 7,0] isPeriodic=[0,0,0] dw=3

Grid Statistics for grid 3, name=hull-stern-cap-nurbs. (curvilinear) (m
apping=NurbsMapping)
grid lines : [0:16,0:16,0:20], total points = 5120
ghost points: [2:2,2:2,2:2]
cell volumes: [2.03e-05,6.01e-04,4.17e-03] [min,ave,max]
grid spacing: [3.57e-02,7.90e-02,2.14e-01] [min,ave,max] (r1)
: [3.57e-02,7.90e-02,2.14e-01] [min,ave,max] (r2)
: [1.41e-02,6.00e-02,1.04e-01] [min,ave,max] (r3)
number of negative volumes = 0
bc = [0,0, 0,0, 7,0] share=[0,0, 0,0, 7,0] isPeriodic=[0,0,0] dw=3

Grid Statistics for grid 4, name=BigBox. (rectangular) (mapping=BoxMap
ping)
grid lines : [0:120,0:120,0:120], total points = 1771561
ghost points: [2:2,2:2,2:2]
cell volume : 3.37e-03
grid spacing: [1.50e-01,1.50e-01,1.50e-01] [dx,dy,dz]
bc = [1,2, 3,4, 5,6] share=[0,0, 0,0, 0,0] isPeriodic=[0,0,0] dw=3

Grid Statistics for grid 5, name=BigBox2. (rectangular) (mapping=BoxMap
ping)
grid lines : [0:120,0:120,0:120], total points = 1771561
ghost points: [2:2,2:2,2:2]
cell volume : 3.37e-03
grid spacing: [1.50e-01,1.50e-01,1.50e-01] [dx,dy,dz]
bc = [1,2, 3,4, 5,6] share=[0,0, 0,0, 0,0] isPeriodic=[0,0,0] dw=3

*****Grid Statistics Summary*****
number of grids =6
Total number of grid points 3915738 (gridIndexRange), 4378850 (dimension) <<<
Total number of interpolation points 21172
cell volumes: [1.17e-05,3.28e-03,8.70e-03] [min,ave,max]
*****

```

Figure 78: Summary of the grids used for computing the scattering about a reentry vehicle.

In Figures 79–82, we are showing 3D plots for the total field (incident+scattered). Specifically, Figures 79–80 correspond to a relatively early moment of time, $t = 0.1T$, where T is equal to 120 diameters of S . Figures 81–82 correspond to a later moment of time, $t = 0.3T$. In Figures 81–82, one can notice the accumulation of numerical artifacts around the scatterer, which is not expected for a solution that is basically periodic in time. Most likely, the artifacts should be attributed to a

growing instability due to the use of overlapping grids. This is indeed an instability, as indicated by the temporal profile of the scattered component E_y shown in the left panel of Figure 83. This profile sharply blows up after $t = 0.35T$, which is typical for a developing numerical instability. Of course, to be able to attribute the instability unambiguously to a given cause, e.g., to the overlapping grids, one needs to rule out other possible causes, in particular, to eliminate the influence of the artificial outer boundary. To do so, i.e., to make sure that it is not the lacunae-based time marching that is causing the instability, we performed the same computation in the genuine OVERTURE framework using the Engquist-Magda artificial boundary conditions at ∂S . The artificial dissipation was applied on the entire domain S . In these computations, we observed exactly the same behavior as that of the blue curve in the left panel of Figure 83 — a blowup of the solution shortly after $t = 0.35T$. This suggests that the instability is, in fact, induced by the overlapping grids that discretize the scatterer rather than by the treatment of the artificial outer boundary. Of course, it is entirely possible that this instability could be reduced or eliminated by adjusting the artificial dissipation. This, however, would require an additional thorough numerical investigation that we have not performed. Instead, we compare the performance of the pure PML closure at the artificial outer boundary against that of the lacunae+PML approach over the interval of time for which the foregoing interior instability does not manifest itself yet. Figure 83 clearly demonstrates that the solution terminated by the PML itself blows up fairly early on, whereas the lacunae-based time marching combined with the PML eliminates this blowup and maintains stable performance, as expected by design. This stable performance is later disrupted by the instability which develops within the IP due to the overlapping grids and thus cannot be cured by employing an alternative treatment of the artificial outer boundary.

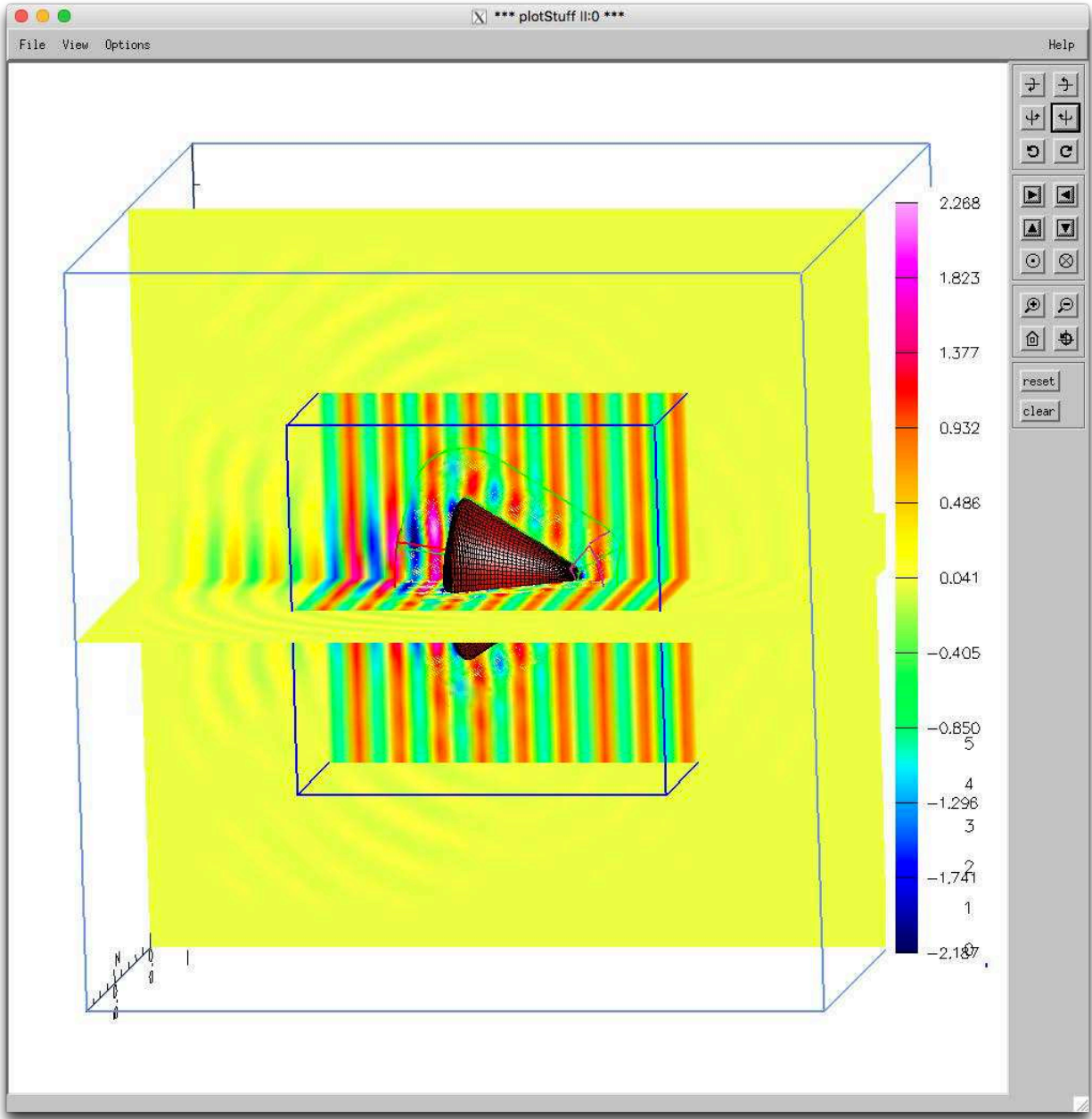


Figure 79: E_y at $t = 0.1T$. Total field (incident+scattered) in the small box (domain S), reflected field only in the big box (domain S_{aux}). Lacunae+PML approach.

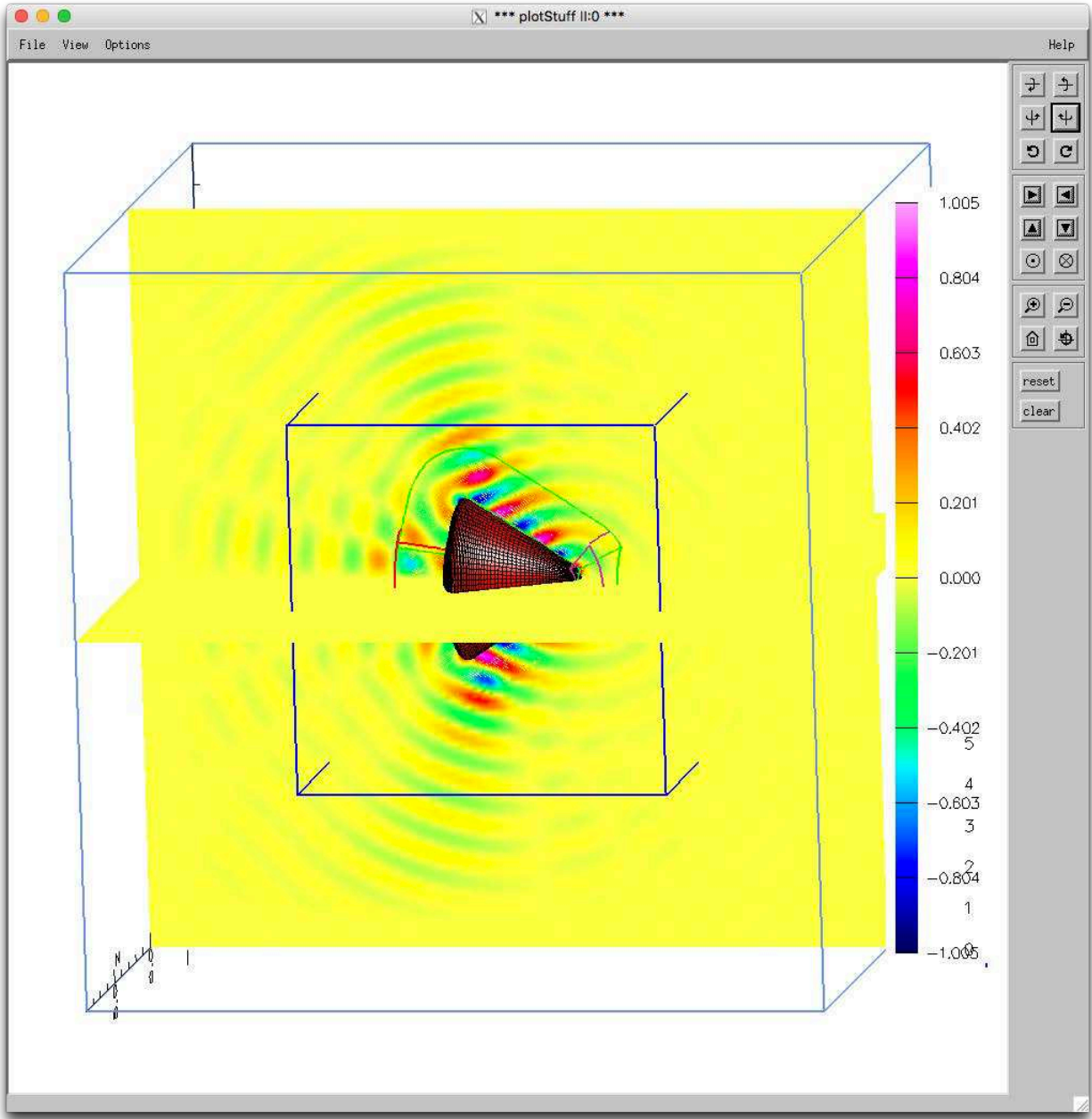


Figure 80: E_x at $t = 0.1T$. Total field (incident+scattered) in the big box (domain S_{aux}). Lacunae+PML approach.

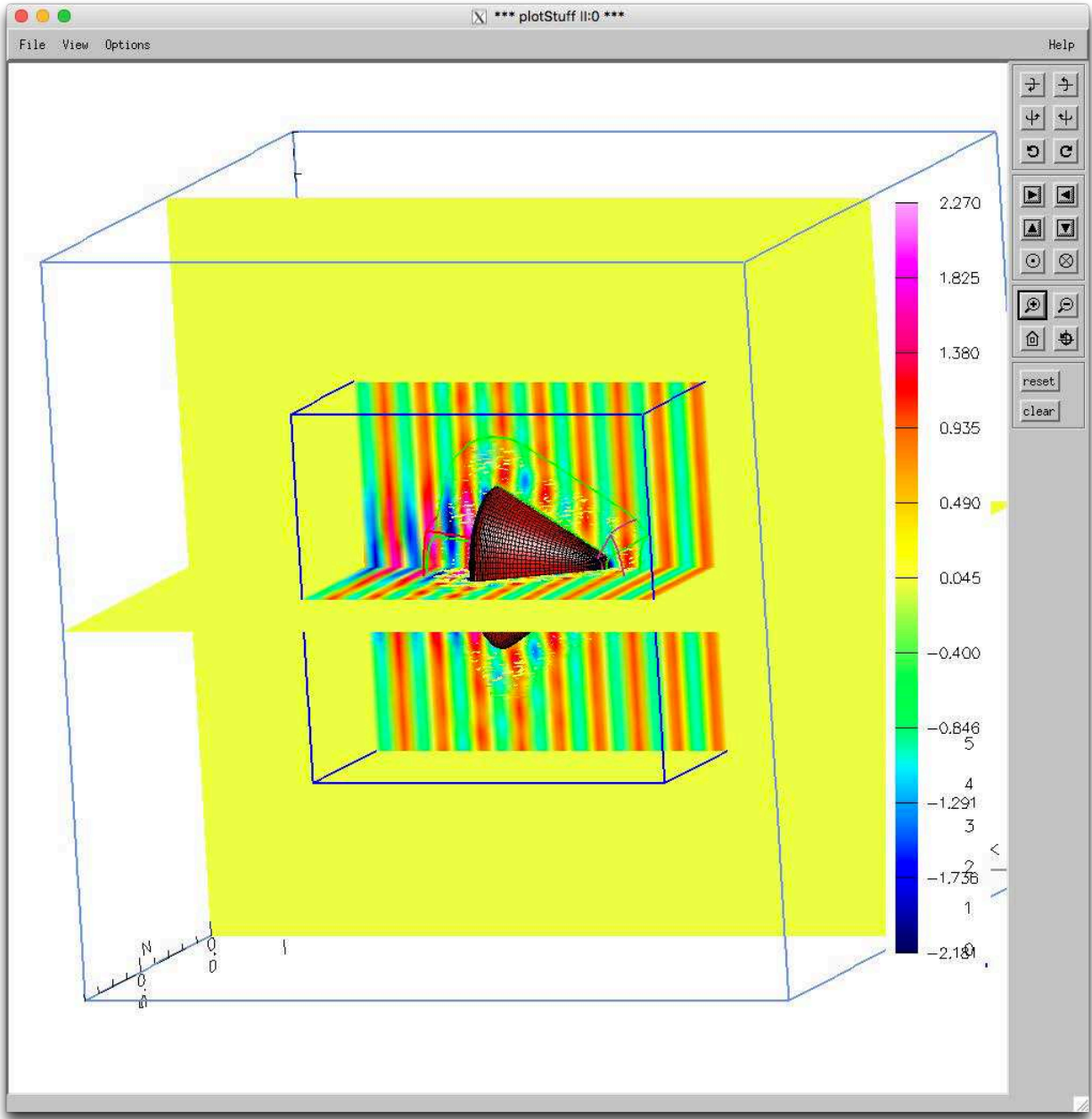


Figure 81: E_y at $t = 0.3T$. Total field (incident+scattered) in the small box (domain S). Lacunae+PML approach.

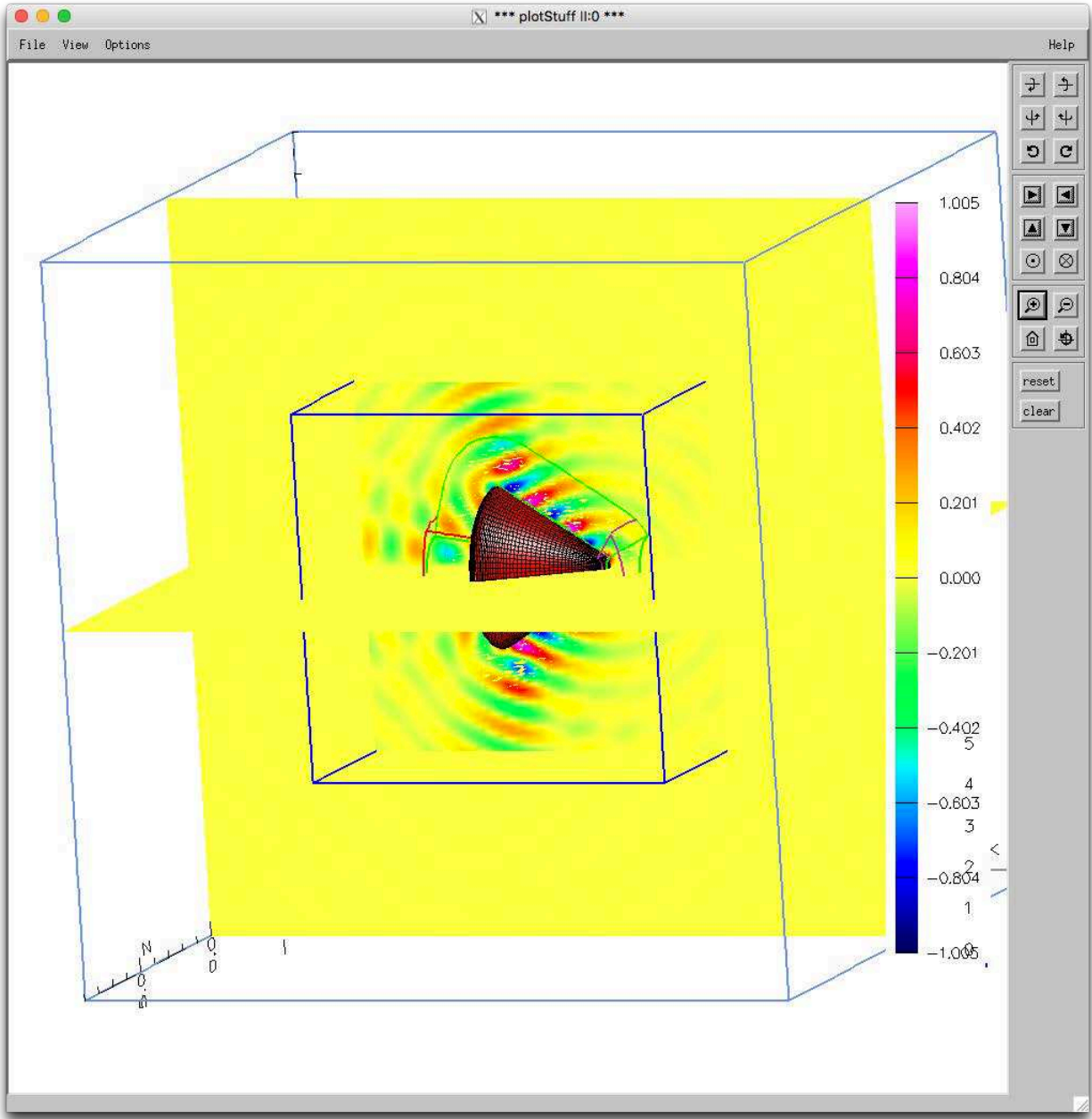


Figure 82: E_x at $t = 0.3T$. Total field (incident+scattered) in the small box (domain S). Lacunae+PML approach.

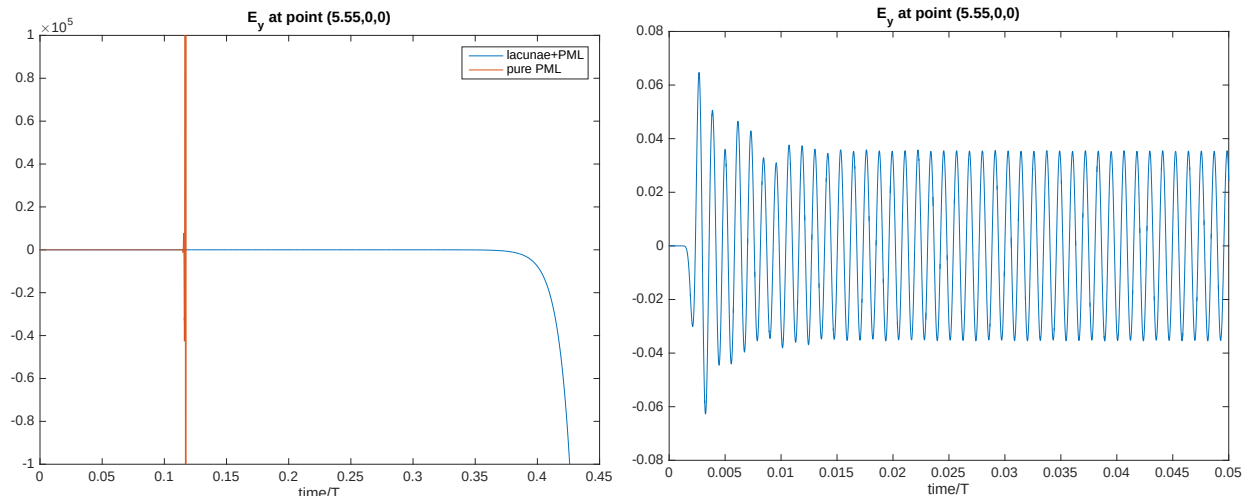


Figure 83: Left: Scattered field E_y vs time. Pure PML and lacunae+PML approaches are presented. Right: Zoom in for an early stage of the evolution of E_y .

10 Application of lacunae to other boundaries

10.1 General

Our long-term goal is to apply the lacunae-based approach to a broad class of frequently encountered wave propagation problems that are predominantly smooth: geometrically large regions with smooth material parameters separated by several interface boundaries, e.g., scatterers in large volumes of free space. We would like to achieve high efficiency and flexibility with respect to the geometry and the boundary conditions required for addressing realistic applications.

The main thrust of the current project is still on artificial outer boundaries for Maxwell's equations. The application to other types of boundaries is a promising future direction that we investigate preliminarily. As such, we focus on the simplest possible formulation — an interior Dirichlet initial boundary value problem (IBVP) for the scalar d'Alembert equation:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \Delta u = f(t, \mathbf{x}). \quad (136)$$

Equation (136) is integrated by the standard central difference second order accurate scheme, which is most inexpensive; high order schemes will be introduced later. Of course, it is well known that the main disadvantage of finite difference methods is in dealing with complicated geometries (and also solutions with low regularity). This is precisely where and why the difference potentials [59–62] that allow one to handle non-conforming boundaries on regular grids become very instrumental.

For problems with complex geometry, one often uses the finite element method (FEM) and its extensions [81–85], as well as the discontinuous Galerkin (DG) method [86]. Besides dealing with complex geometries, these methods can efficiently approximate the solutions with low regularity. However, unlike finite differences, they become increasingly sophisticated and expensive when high order accuracy is needed, which is a must for wave propagation problems that are susceptible to numerical pollution (the dispersion error), see [87, 88] and [89, Section 4.6.1].

Traditional boundary element methods (BEM) apply to steady-state or time-harmonic problems (elliptic PDEs). In these methods, linear boundary value problems are reduced to integral equations with respect to equivalent boundary sources. BEM impose no limitations on the shape of the

boundary and automatically account for the correct far field behavior of the solution. However, these methods rely on the explicit knowledge of the fundamental solution, and the treatment of the boundary conditions requires care in choosing the boundary sources so that to maintain the well-posedness of the boundary representation. Moreover, standard BEM cannot be used directly for unsteady wave propagation problems (hyperbolic PDEs). Their time-dependent applications are limited to combined problems with a clearly identifiable elliptic component, such as slow speed flows of viscous fluid [90, 91] or water waves [92].

A special class of the so-called retarded potential boundary integral equations (RPBIE), see [93, 94], provide a venue toward extending the BEM from elliptic to hyperbolic PDEs. However, the corresponding time domain numerical methods that were explored by both mathematics [95, 96] and engineering [97, 98] research communities are not nearly as popular as their frequency domain counterparts. One difficulty is that many time domain discretizations of RPBIEs appear prone to instabilities, even if the well-posedness of the RPBIE per se can be guaranteed in the first place (second kind BIE vs. first kind BIE). For the most part, however, the reason is that as the time elapses the boundary extends. Hence, the computation of convolutions involved in the formulation of RPBIEs becomes increasingly more expensive, given that it typically relies on Laplace transform methods [99, 100] and does not take into account any special properties of the formulation that may be advantageous, such as the Huygens' principle. On the contrary, in our work the Huygens' principle and lacunae play the central role, see Section 10.3.

10.2 The method of difference potentials

In our prior work we have considered time-harmonic waves in two dimensions [59–62]. This research employed Ryaben'kii's method of difference potentials [101–104], which can be viewed as a discrete analog of Calderon's potentials and Calderon's boundary equations with projections in functional analysis [105, 106]. It is a very general approach that enabled us to handle general boundaries not aligned with the grid and still maintain high order accuracy. In particular, we considered interior, exterior, and transmission/reflection problems [59, 60]. We have also analyzed problems with singularities due to jumps in the boundary data [61, 62]. Our intention is to extend this approach to time-dependent problems (i.e., broadband applications). The key advantages of the method of difference potentials are as follows:

- *Maximum generality of handling the boundary and/or interface conditions.* The entire class of functions satisfying the underlying differential equation is characterized by an equivalent boundary equation (Calderon's boundary equation with projection). Only later are specific boundary conditions imposed. Any type of boundary conditions can be handled with equal ease, see [61, 62], including mixed, nonlocal, discontinuous, interfaces, etc.
- The problem can be discretized on a regular structured grid, yet *the boundary can have an arbitrary shape* and need not necessarily conform to the grid. This causes no loss of accuracy and there are no adverse effects due to staircasing [107, 108].
- Boundary representations inherit the accuracy of the core discretization. Hence, *high order approximations for problems with non-conforming boundaries can be easily obtained.*
- Unlike BEM, the methodology does not require numerical approximation of singular integrals.

10.3 Lacunae and the Huygens' principle

Thus far, the method of difference potentials has mostly been used for steady-state and time-harmonic problems (elliptic PDEs). We will extend it to three-dimensional time-dependent hyperbolic problems using lacunae and the Huygens' principle, which are in the center of the current

STTR project. According to this principle, the waves governed by certain hyperbolic PDEs and systems in three space dimensions have sharp aft fronts. In the regions of space-time behind the aft fronts the solution is zero. Those regions are called secondary lacunae in the sense of Petrowsky [109]. For Maxwell’s equations, in some cases lacunae are replaced by quasi-lacunae, for which the solution behind the aft fronts becomes electrostatic, i.e., time-independent, but not necessarily zero [64]. The presence of lacunae enables an equivalent representation of the solution as the superposition of a finite non-increasing number of wave packets that each has a finite non-increasing “lifespan” on the domain of interest [110].

Previously, we have used lacunae for obtaining far-field artificial boundary conditions (ABCs) for the wave (d’Alembert) equation [80], acoustics (linearized Euler’s) equations [111], and Maxwell’s equations [63]. As lacunae effectively replace the integration over a long time (from the initial to the current moment) by integration over a bounded interval (rather, a finite set of such intervals), the resulting ABCs, while being exact, have only a limited and non-increasing extent of temporal non-locality, which is a major advantage. In the framework of the current STTR project, lacunae have proven very efficient not only as a standalone closure at the artificial outer boundary, but also as a way of stabilizing other ABCs or PMLs (see also [65, 112]), because the same consideration — boundedness of the integration interval — yields a temporally uniform estimate for any error associated with the domain truncation.

A formal extension of Calderon’s boundary operators to the time domain would require computation from the initial to the current moment of time. The use of lacunae, however, allows us to replace it by computation over a bounded and non-increasing time interval.

10.4 Computing Calderon’s operators with the help of lacunae

The method of difference potentials can be applied to an unsteady problem the same way it is applied to steady-state or time-harmonic problems [59–62]. The overall dimension of the problem increases by one (space+time), while the formal constructs of the grid sets and operators remain the same. In Figure 84, we are showing the structure of the grid boundary γ for the time-dependent case. The grid boundary is a key element in the constructs of difference potentials; the densities of the potentials are defined at the grid boundary. In the unsteady case, it is constructed from the initial to the current time. As the time elapses it needs to be extended.

Computing the Calderon operators on the entire grid boundary γ starting from time zero may be expensive though, especially when a representation in the basis is needed that would involve multiple solves of the time-dependent auxiliary problem. However, there is a way of limiting the temporal non-locality of the boundary operators. It exploits the presence of lacunae in the solutions to the d’Alembert equation (136).

Assume the initial conditions are zero and the source term f of equation (136) is compactly supported on a bounded domain $Q \subset \mathbb{R}^3 \times [0, +\infty)$. Then, the solution $u = u(\mathbf{x}, t)$ is known to have a secondary lacuna, or simply lacuna, in the sense of Petrowsky [109]:

$$u(\mathbf{x}, t) \equiv 0 \quad \forall (\mathbf{x}, t) \in \bigcap_{(\boldsymbol{\xi}, \tau) \in Q} \{(\tilde{\mathbf{x}}, \tilde{t}) \mid \|\tilde{\mathbf{x}} - \boldsymbol{\xi}\| < c(\tilde{t} - \tau), \tilde{t} > \tau\} \stackrel{\text{def}}{=} \Lambda. \quad (137)$$

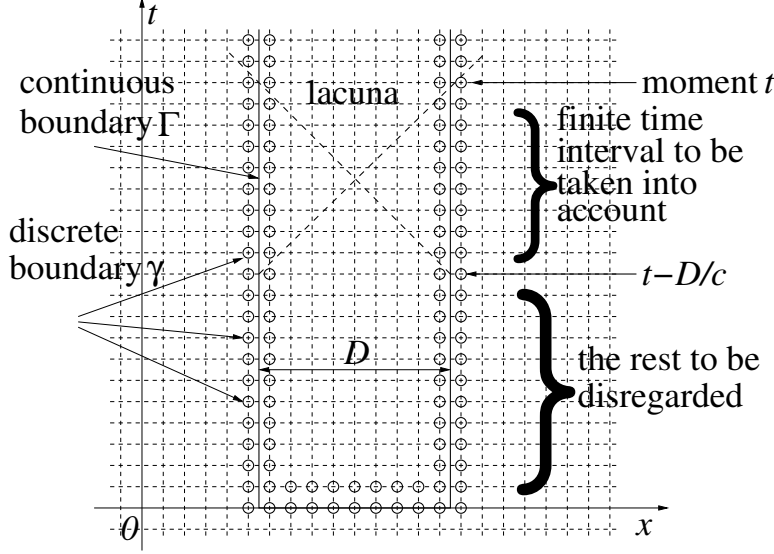


Figure 84: Schematic for computing the discrete Calderon's operators in an unsteady framework.

Formula (137) implies that the lacuna Λ is the intersection of all characteristic cones (i.e., forward light cones) of the wave equation (136) once the vertex of the cone sweeps the support Q of the right-hand side $f(\mathbf{x}, t)$. Physically, Λ is the part of space-time, on which the waves generated by a compactly supported source have already passed, and the solution has become zero, see Figure 85.

The phenomenon of lacunae is inherently three-dimensional. The surface of the lacuna represents the trajectory of aft (trailing) fronts of the waves. The existence of sharp aft fronts in odd-dimension spaces is known as the Huygens' principle, as opposed to the so-called wave diffusion, which takes place in the spaces of even dimension, see, e.g., [113].

For computing the unsteady potentials and projections, the source terms of the discrete wave equation will be located on or near the grid boundary γ , see Figure 84. Then, it is clear that for any given time t , the solution anywhere on the computational domain, including the boundary, will not depend on the sources that are further behind in time than $t - D/c$, where D is the diameter of the domain and c is the propagation speed. Indeed, for those "retarded" sources the computational domain will completely fall into the lacuna, see Figure 84.

Lacunae do not lead to a full localization of Calderon's boundary operators in time. Nevertheless, they yield a fixed and non-increasing bound, D/c , for the extent of the temporal non-locality. This, in turn, enables an efficient computation of the boundary operators in the unsteady case.

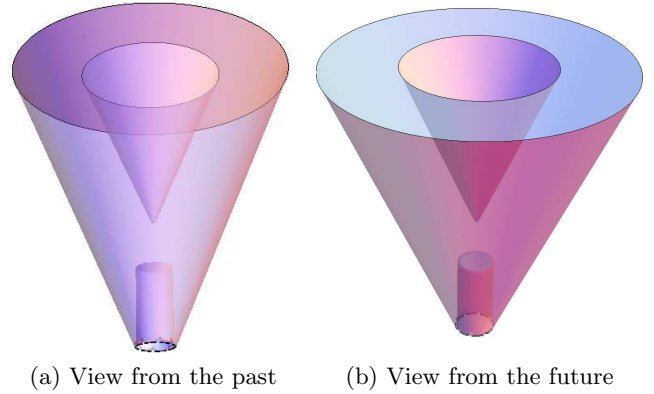


Figure 85: Secondary lacuna in the sense of Petrowsky [109] in the solution to the wave equation driven by a compactly supported source. The vertical axis represents time, the hyperplane normal to it represents space. The lacuna is a region of space-time behind the aft fronts of the propagating waves, on which the solution is zero (the interior cone).

In the next section, we describe the setting and present the results of the initial proof-of-concept computations.

11 Numerical tests for lacunae and the Dirichlet problem

11.1 Formulation of the problem

We seek the solution to the homogeneous three dimensional wave equation in the ball of radius R centered at the origin of the Cartesian system, see Figure 86. On the boundary ∂S_0 , the Dirichlet boundary conditions are imposed:

$$\begin{cases} \partial_t^2 u - \Delta u = 0 \text{ on } S_0, \\ \text{Dirichlet b.c. on } \partial S_0. \end{cases} \quad (138)$$

For simplicity, the propagation speed in equation (138) is chosen as $c = 1$.

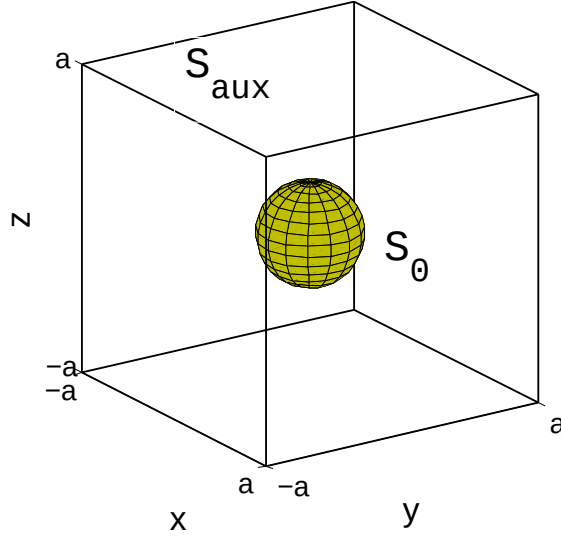


Figure 86: Computational domain S_0 and auxiliary domains S_{aux} .

11.2 Reference solution

As a reference solution, we use the plane wave:

$$u^{\text{ref}}(\mathbf{x}, t) = \cos(\mathbf{k}\mathbf{x} - \omega t), \quad (139)$$

that obviously solves the 3D wave equation (138). The frequency $\omega = 1$ and the wave vector $\mathbf{k} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$, so that the dispersion relation $\omega = k$, $k = |\mathbf{k}|$, holds (recall, $c = 1$). The Dirichlet boundary data in (138) are simply the trace of (139) on the boundary ∂S_0 .

11.3 Application of difference potentials

We introduce the densities $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$:

$$\boldsymbol{\xi}^{(1)} = \sum C_i^{(1,0)} \begin{pmatrix} \zeta_i(s) \\ 0 \end{pmatrix} + C_i^{(1,1)} \begin{pmatrix} 0 \\ \zeta_i(s) \end{pmatrix}, \quad (140)$$

$$\boldsymbol{\xi}^{(2)} = \sum C_i^{(2,0)} \begin{pmatrix} \zeta_i(s) \\ 0 \end{pmatrix} + C_i^{(2,1)} \begin{pmatrix} 0 \\ \zeta_i(s) \end{pmatrix}, \quad (141)$$

defined on the space-time domain boundaries:

$$\Gamma^{(1)} = [t_0, t_1^{\text{end}}] \times \partial S_0 \quad (142)$$

and

$$\Gamma^{(2)} = [t_2^{\text{start}}, t_F] \times \partial S_0, \quad (143)$$

respectively, see Figure 87. The densities are expanded into series using an appropriate basis $\zeta_i(s)$.

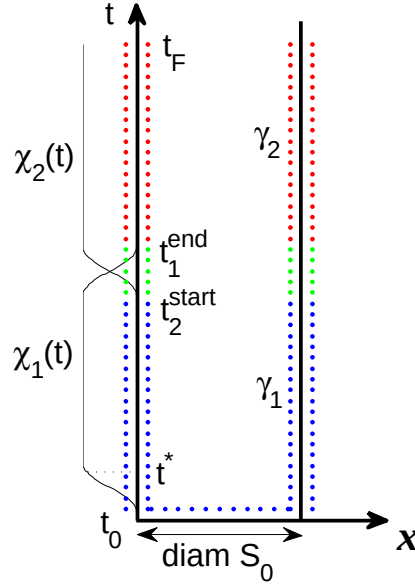


Figure 87: Large scale space-time stencil of the algorithm.

The density $\boldsymbol{\xi}^{(2)}$ is the unknown function, whereas $\boldsymbol{\xi}^{(1)}$ is assumed known on every step of the large scale time marching algorithm.

In the framework of the difference potentials, the IBVP (138) is equivalently recast as a problem for $\boldsymbol{\xi}^{(2)}$ formulated only at the boundary:

$$\begin{cases} \partial_t^2 u - \Delta u = 0 \\ \text{Dirichlet b.c. on } \partial S_0 \end{cases} \xLeftrightarrow{\text{boundary reformulation}} \begin{cases} P_{\gamma_2} \boldsymbol{\xi}^{(2)} = \boldsymbol{\xi}^{(2)} \\ l_{\gamma_2} (P_{S_0} \boldsymbol{\xi}^{(2)}) = g \end{cases} \quad (144)$$

Furthermore, spectral representation (141) of the density $\boldsymbol{\xi}^{(2)}$ allows one to reduce the boundary problem (144) to the set of linear equations solved on γ_2 in the sense of least squares:

$$\mathbf{Q}^{(2,1)} \mathbf{c}^{(2,1)} = -\mathbf{Q}^{(2,0)} \mathbf{c}^{(2,0)} - \mathbf{Q}^{(1,1)} \mathbf{c}^{(1,1)} - \mathbf{Q}^{(1,0)} \mathbf{c}^{(1,0)}, \quad (145)$$

where $\mathbf{c}^{(i,j)} = \{C_1^{(i,j)}, C_2^{(i,j)}, \dots\}$. Equation (145) is specific for the Dirichlet problem: by solving (145) we obtain the coefficients $\mathbf{c}^{(2,1)}$, i.e., reconstruct the normal derivative of the unknown solution. The columns of matrices $\mathbf{Q}^{(i,j)}$ are solutions to the partial problems generated by the basis

elements $\begin{pmatrix} \zeta_i(s) \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ \zeta_i(s) \end{pmatrix}$ and considered on the discrete counterpart γ_2 of the set Γ_2 . These matrices are pre-computed only once and are used afterwards for updating $\mathbf{c}^{(2,1)}$.

11.4 Partial auxiliary problems

Partial subproblems are the inhomogeneous 3D wave equations with the right-hand sides constructed with the help of the basis elements. These problems are solved on the cube

$$S_{\text{aux}} = [-a, a] \times [-a, a] \times [-a, a]$$

chosen sufficiently large to prevent the influence of the boundary conditions set on the faces of S_{aux} .

11.5 Parameters of the experiment

The parameters we have used for the preliminary testing of the foregoing algorithm are as follows:

- (a) Radius of the computational domain S_0 : $R = 1$.
- (b) The final computational time $T_{\text{final}} = 5000R = 2500 \times \text{diam}S_0$.
- (c) Cartesian discretization grid for the computational domain S_0 : spatial steps $h_x = h_y = h_z = 2R/15$, time step $h_t = h_x/3$.
- (d) Size of the auxiliary computational domain S_{aux} : $a = 4R$.
- (e) The length of the hat functions $\chi_{1,2}(t)$ used to make the right-hand sides smooth in time: $T_{\text{hat}} = 5$ (see Figure 87).
- (f) The fraction of the flat part in the total length T_{hat} : $\sigma = 0.7$.
- (g) The basis functions for the expansion of the density: we use spherical harmonics $Y_{l,m}(\theta, \varphi)$ in space and Chebyshev polynomials $T_n(t)$ in time, i.e., $\zeta_i(s) = T_n(t)Y_{l,m}(\theta, \varphi)$.

11.6 Parallelization

For the most part, the CPU time in the algorithm is spent on computing the matrices $\mathbf{Q}^{(i,j)}$ (4 items). However, each column of these matrices can be computed completely independently of the others. This provides an excellent opportunity for employing parallelization in the code. The code was written using OpenMP interface and run on a 32-core shared memory server. The observed speed-up is about 20-25 times compared to a single-core implementation.

11.7 Results of the experiments

In Figure 88, we present the error profiles for different dimensions of the basis. We can see that as the number of basis functions increases the error decreases, which shows the convergence of our numerical solution with respect to the dimension of the basis. In Figure 88, N stands for the maximum degree of the Chebyshev polynomial and L denotes the maximum orbital index of the spherical function $Y_{l,m}(\theta, \varphi)$ (while $m = -L, -L+1, \dots, L-1, L$).

In Figure 89, we are showing the profiles of the error for one particular basis, $L = 8$ and $N = 9$, but on two different discretization grids. The simulation time is equivalent to 2500 times the time needed for the waves to cross the computational domain. Figure 89 corroborates the design second order of grid convergence of the proposed algorithm.

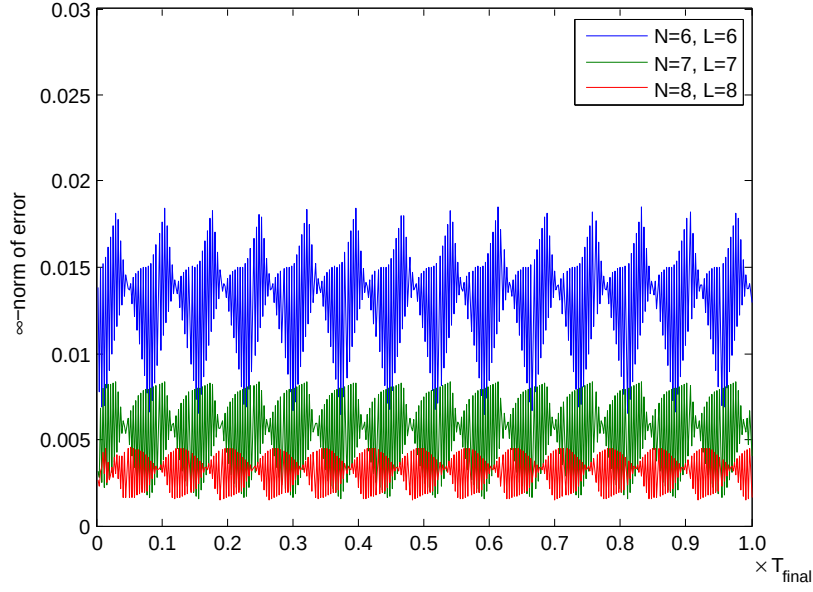


Figure 88: Error of the numerical solution as compared against the reference solution (139) for three sets of basis functions of increasing dimension. Demonstrates convergence of the numerical solution with respect to the dimension of the basis on the continuous boundary.

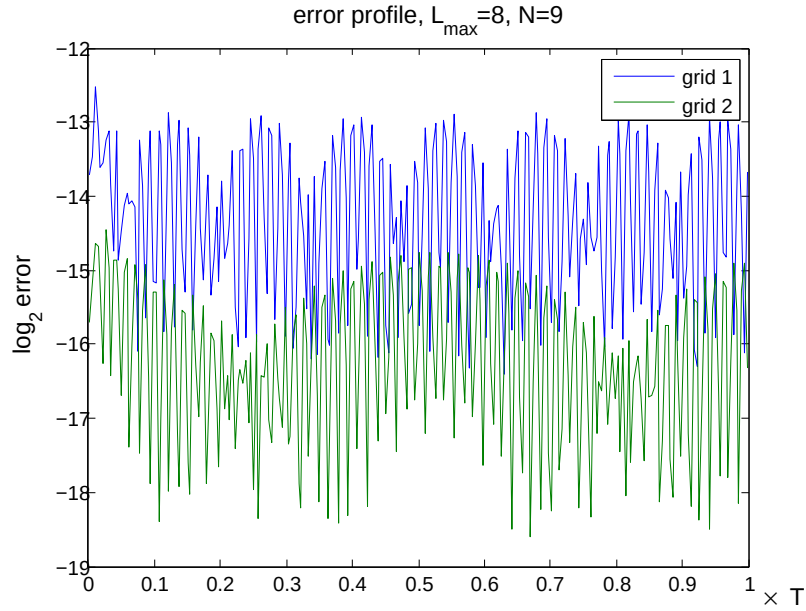


Figure 89: Error of the numerical solution as compared against the reference solution (139) for two different discretization grids, with the size of the second grid being one half of the size of the first grid. Demonstrates the second order of grid convergence.

12 Computations for lacunae and the Neumann boundary conditions

Instead of problem (138), we consider

$$\begin{cases} \partial_t^2 u - \Delta u = 0 \text{ on } S_0, \\ \text{Neumann b.c. on } \partial S_0, \end{cases} \quad (146)$$

where the Neumann boundary data on ∂S_0 in problem (146) are derived from the exact solution (139). Following the same approach based on difference potentials as employed in Section 11, we obtain the boundary equation for the Neumann problem [cf. equation (145)]:

$$Q^{(2,0)} c^{(2,0)} = -Q^{(2,1)} c^{(2,1)} - Q^{(1,1)} c^{(1,1)} - Q^{(1,0)} c^{(1,0)}. \quad (147)$$

Equation (147) allows one to compute the coefficients $c^{(2,0)}$, i.e., reconstruct the trace of the unknown solution (the Dirichlet data) on the boundary $\Gamma^{(2)}$, see (143), assuming that the trace of its normal derivative (the Neumann data) is given.

12.1 Notation and terminology

Terminology: By the “no-lacunae technique” we mean an approach equivalent to the method of difference potentials for a time-dependent non-homogeneous equation. The inhomogeneity is due to the initial data at the bottom of the cylinder, and it requires solution of one additional auxiliary problem. That is, we are “lifting the bottom” of the cylinder over the entire computational interval. The resulting boundary equation with projection also becomes inhomogeneous. This technique does not use lacunae and thus is valid, at least theoretically, for any evolution problem. Its advantage over the conventional finite differences is that it allows for non-conforming boundaries on regular structured grids. By the “lacunae-based approach” we mean the version of time marching that involves only the lateral boundary of the cylinder. It is valid for homogeneous Huygens’ equations and has an additional advantage of an asymptotically lower complexity for long runs.

Some parameters:

- (a) Radius of the computational domain S_0 : $R = 1$.
- (b) The computational time $T = 1500R = 1500$.
- (c) Grid for the computational domain S_0 : spatial steps $h_x = h_y = h_z = 2R/15$, time step $h_t = h_x/3$.
- (d) The basis functions for the expansion of the density: we use spherical harmonics $Y_{l,m}(\theta, \varphi)$ in space and Chebyshev polynomials $T_n(t)$ in time, i.e.,

$$\zeta_i(s) = T_n(t) Y_{l,m}(\theta, \varphi). \quad (148)$$

Our general observations regarding numerical solution of the Neumann problem are as follows:

- In the beginning of the simulation, the algorithm demonstrates the design rate of grid convergence (second order for the current implementation).
- Over long simulation intervals, the error shows a slow yet steady linear growth.

- If, instead of the general Chebyshev basis in time we use only the trace of the exact solution and its normal derivative, the error growth is removed. In this particular case, of course, the solution is sought for in a space of functions that is artificially narrowed down.

In general, the linear growth of the error in time is consistent with the standard well-posedness estimates for initial boundary value problems for the wave (d'Alembert) equation, see, e.g., [114]. If the equation is homogeneous, the well-posedness constants are allowed to increase linearly in time, while for the inhomogeneous equation the rate of increase may even become quadratic.

As of yet, we do not have a definite answer though as to why the error growth manifests itself for the Neumann problem but does not manifest itself for the Dirichlet problem, given that the overall well-posedness estimates for both problems are essentially the same. A notable difference between the two formulations is that the homogeneous wave equation subject to a zero Neumann boundary condition allows perturbations of the form $A + Bt$, where A and B are constants (i.e., a uniform in space solution that grows linearly in time). It is not immediately clear, however, whether the error growth that we are observing can be attributed to the perturbations of this form.

In the rest of this section, we report our initial numerical findings for the Neumann IBVP (146).

12.2 The Klein-Gordon equation

To investigate any potential differences in numerical performance, we also consider a Neumann IBVP for the Klein-Gordon equation:

$$\begin{cases} \partial_t^2 u - \Delta u + m^2 u = 0 \text{ on } S_0, \\ \text{Neumann b.c. on } \partial S_0. \end{cases} \quad (149)$$

The Klein-Gordon equation differs from the d'Alembert equation of (146):

$$\begin{cases} \partial_t^2 u - \Delta u = 0 \text{ on } S_0, \\ \text{Neumann b.c. on } \partial S_0, \end{cases}$$

by the additional term $m^2 u$, where m is a constant. The Klein-Gordon equation is also known as the dispersive wave equation.

The benefits from analyzing the Klein-Gordon equation (149) are:

- Equation (149) does not admit solutions in the form $A + Bt$. Therefore, the two suspect basis elements: $T_0(t)Y_{0,0}$ (constant) and $T_1(t)Y_{0,0}$ (linear in time) that support the error growth in the case of the conventional wave equation, are no longer in the set of solutions to (149).

- The plane wave (139):

$$u^{\text{ref}}(\mathbf{x}, t) = \cos(\mathbf{k}\mathbf{x} - \omega t)$$

can still be used as a reference solution to (149), except that the dispersion relation now reads: $\omega^2 = |\mathbf{k}|^2 + m^2$.

- By setting $m = 0$ we return to the standard d'Alembert equation.

12.2.1 Results of the Klein-Gordon simulations. We solve the Neumann problem (149) with $m = 1$ on the two successive grids using (139) as a reference solution. We use the full basis (148) without any correction procedures like eliminating the suspect elements. We use the “no-lacunae technique” as the Klein-Gordon equation is not Huygens’. The resulting error profiles are shown in Figure 90. They are flat, which provides additional evidence that the basis elements $T_0(t)Y_{0,0}$ (constant) and $T_1(t)Y_{0,0}$ (linear in time) may catalyze the error growth specifically for the wave equation (146). In doing so we note that the general behavior of the well-posedness constants for both equations, Klein-Gordon and d’Alembert, is the same: linear growth in time is admissible. We also compute numerically the gradient of the solution (right panel) for the reason that will become clear later.

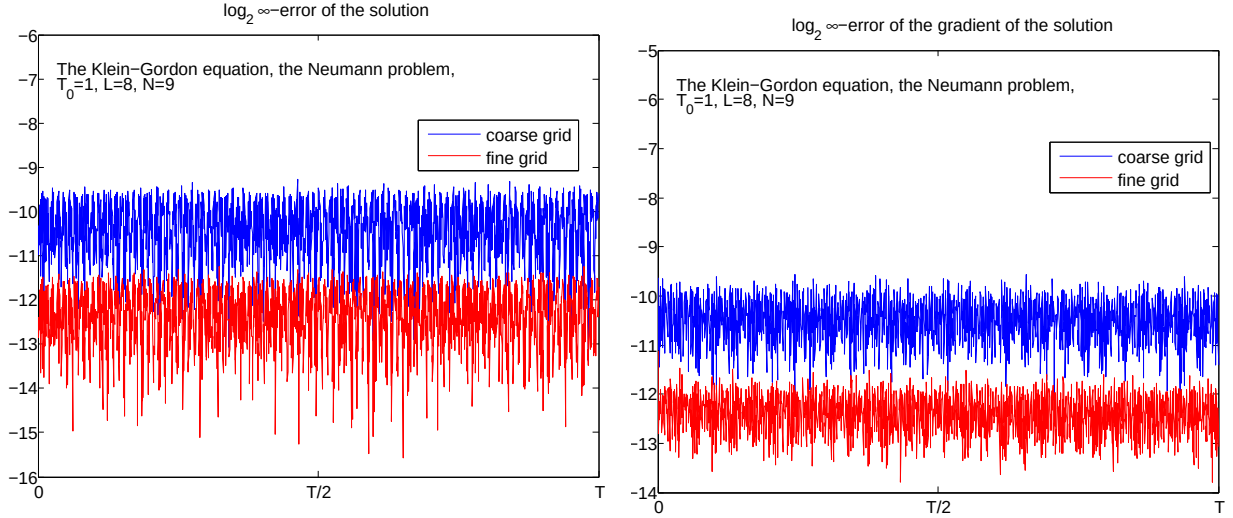


Figure 90: $\log_2$ - L_∞ -error profiles of the Neumann IBVP for the Klein-Gordon equation (149). On the left: error of the genuine solution u . On the right: $\log_2$ -error of $|\nabla u|$. Both plots correspond to the “no-lacunae” technique.

12.3 Choosing ∇u instead of u for the wave equation

We turn back to the IBVP (146) for the wave equation by setting $m = 0$ in the Klein-Gordon equation. Now, under the full basis implementation, we additionally compute the gradient of the solution. In this section we are still using the “no-lacunae technique” even though the wave equation does have lacunae.

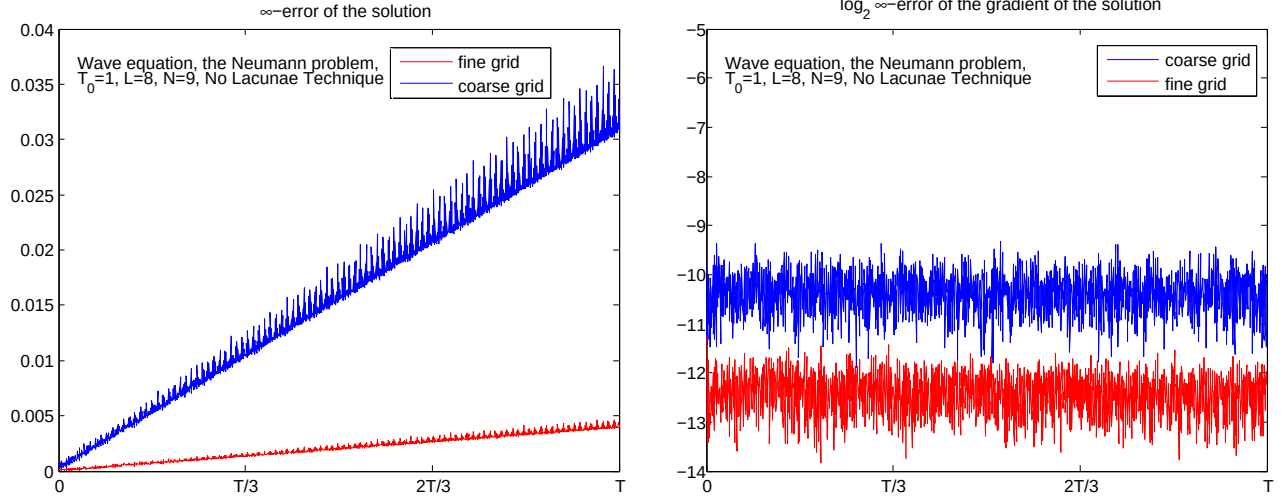


Figure 91: L_∞ -error profiles of the Neumann IBVP (146). On the left: error of the genuine solution u . On the right: $\log_2$ -error of $|\nabla u|$. Both plots correspond to “no-lacunae” technique.

12.3.1 Results of simulations. The results are presented in Figure 91. On the left, we observe a linear error growth which is typical for our simulations of the plain wave equation with Neumann boundary conditions. However, on the right **the error profile for the gradient is flat**. In other words, by a simple spatial differentiation we eliminate the linear growth of the kind $A + Bt$, which is intuitively understandable.

The right panel of Figure 91 clearly demonstrates the second order grid convergence, because the fine grid is twice as fine as the coarse grid, and the corresponding error is four times smaller. Moreover, the error profiles in the left panel of Figure 91 can also be understood as showing the grid convergence. Indeed, even though the error on each grid increases linearly as the time elapses, at every given moment of time the error on the fine grid is at least four times smaller than the error on the coarse grid.

12.3.2 Caveat. There is one caveat, however. As the height of the cylinder T_0 increases the results become worse, see an example in Figure 92. The error on the fine grid starts accumulating. We attribute this effect to a low dimension of the basis $\{T_n(t)\}$ for a taller cylinder, i.e., insufficiently accurate spectral representation of the boundary data on the lateral boundary of the tall cylinder. In practice, we monitor the error of representation of the boundary data w.r.t. the analytical results. We observe that at $T_0 = 5$ the precision of the representation drops by 2-3 orders of magnitude compared to the case of $T_0 = 1$. Reducing the height to, say, $T_0 = 3$ with the same dimension of the basis makes the error profile flat again, similar to Figure 91 (not shown). Alternatively, for a taller cylinder it is possible to increase the number of elements $T_n(t)$ in the basis, which restores the precision required for a stable time marching. The latter statement will be confirmed in practice when running the lacunae-based version. It is important to realize that under the lacunae-based approach we cannot reduce the height of the cylinder arbitrarily because of the given finite length of the temporal prehistory of the solution. In practice, we cannot use a cylinder shorter than $T_0 = 5$. So, in the next section we increase the dimension of the basis $\{T_n(t)\}$ from $N = 9$ to $N = 14$.

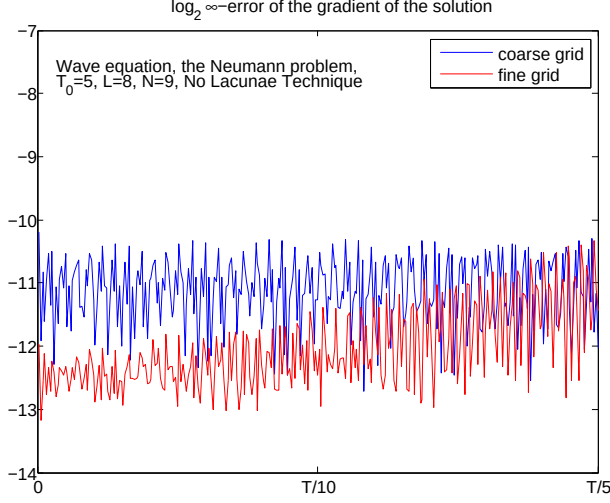


Figure 92: $\log_2 L_\infty$ -error profile of $|\nabla u|$ for the Neumann problem (146). The only distinction from Figure 91 (right panel) is the increased height of the cylinder: $T_0 = 5$ instead of $T_0 = 1$. Again, “no-lacunae” technique. Also, notice a shorter computational time compared to Figure 91 as the error on the fine grid starts accumulating so that it makes no sense to advance any further.

12.4 Lacunae-based implementation

In this section, we present the error of the numerical solution to the Neumann IBVP for the d’Alembert equation under the lacunae-based time marching for both the solution itself and its gradient, see Figure 93. The error behavior in the left panel that corresponds to the solution itself still needs to be better understood. However, the error profiles in the right panel that correspond to the gradient of the solution are non-growing, same as in Figure 91. We also note that the dimension of the basis $\{T_n(t)\}$ has been increased, as discussed previously.

12.5 Intermediate conclusions

- (a) For an equation other than the pure d’Alembert equation (the Klein-Gordon equation in our example), one can solve the Neumann problem and obtain non-increasing error profiles.
- (b) The d’Alembert equation: the error for the solution increases gradually, while the error for its gradient does not grow and shows the grid convergence with the design rate. The solution itself can also be interpreted as converging, despite the growth of the error in time. It is to be noted that in many physically relevant cases the wave equation per se governs a potential, e.g., the velocity potential in acoustics or the scalar or vector field potential in electromagnetism. Getting from the potential to a physically observable quantity requires precisely the differentiation, which also happens to remove the error growth in the simulations.
- (c) The simplified “no-lacunae” algorithm proves to be a viable technique. It is general and applies to any evolution equation. Its important advantage is the ability to adjust the height of the cylinder and thus decrease the number of basis elements, which translates into much lower memory requirements compared to the lacunae-based version. Indeed, the matrix $\mathbf{Q}$ on the fine grid for the “no-lacunae” algorithm is about 2-4 Gb, whereas under the lacunae-based approach it could be up to 20 Gb. This also slows down the I/O operations of the code,

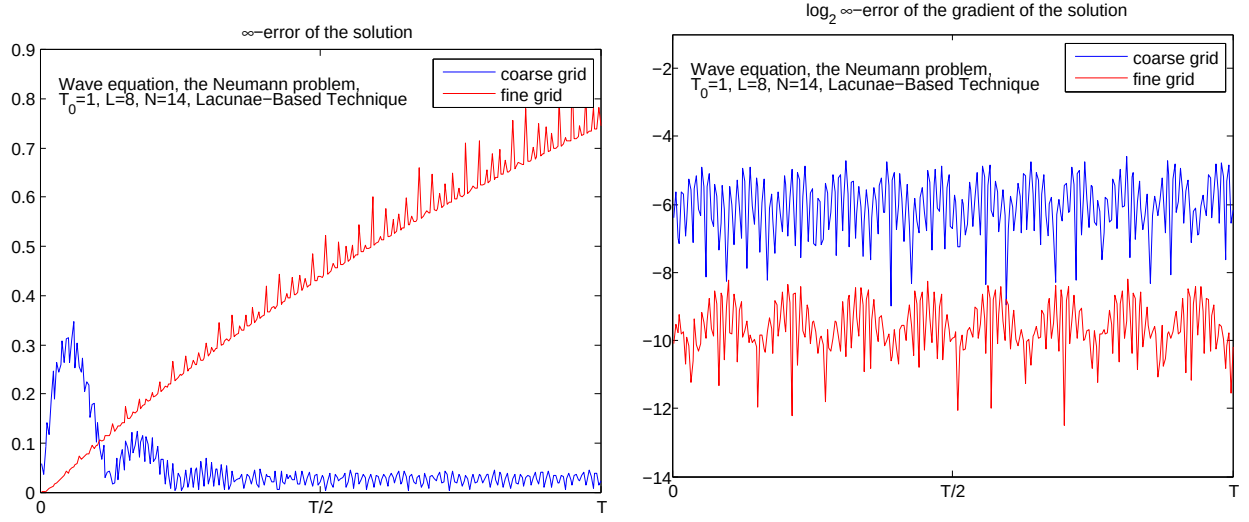


Figure 93: L_∞ -error profiles of the Neumann problem (146). On the left: error of the solution u itself. On the right: $\log_2$ -error of $|\nabla u|$. Both plots use a lacunae-based approach. A typo in the legends: $T_0 = 5$ rather than $T_0 = 1$.

although the QR decomposition and overall time marching appear quite reasonable for both procedures. Finally, the time needed to compute $\mathbf{Q}$ is currently much higher for the lacunae-based version since it requires a large auxiliary computational domain to accommodate the outgoing waves, while the “no-lacunae” version can use a much smaller one.

13 Lacunae implementation in the AFRL ICEPIC code

During Phase II, our innovative algorithms are being implemented as software tools, coupled with the ICEPIC, as well as other HPC CEM codes such as LLNL OVERTURE code and WARPX multi-fluid plasma/Maxwell solver. The approach and algorithm are extended to problems of interest of the US Army, US Air Force, and DoD Prime Contractors. In this Phase II, we also are addressing other major issues: the efficient use of multi-core OpenMP and high-performance MPI algorithms. A set of documentation on theoretical results, software design, and implementation is being developed.

This section describes the work on the lacuna implementation into ICEPIC code that has been planned for Year 2 of the Phase II project.

13.1 Introduction to the AFRL ICEPIC code

ICEPIC is the Improved Concurrent Electromagnetic Particle-In-Cell code used to model high power microwave (HPM) devices: high-power microwave tubes, including magnetrons, klystrons, gyrotrons, and Magnetically Insulated Line Oscillators (MILOs) Fig. 94. ICEPIC has been developed by numerous scientists and engineers since 1994 at the AFRL at Kirtland AFB, Albuquerque, NM, and is considered as a critical AFRL technology. Dr. A. Greenwood, is one of the key developers of ICEPIC and now he is TPOC for ICEPIC. He holds the patent for the “All Cavity Magnetron Axial Extractor”, which he designed using ICEPIC simulation. Dr. Greenwood is very supportive for the proposed effort.

Coupling of lacunae-based integration with ICEPIC will allow us to demonstrate its universality, i.e., the applicability and efficiency for a wide range of discretizations and solvers. Moreover, the coupling to be designed for parallel implementation and demonstrate efficient scaling of CPU time vs. the number of nodes as the original ICEPIC.

The Particle-in-cell (PIC) method consists of finite-difference time-domain (FDTD) electromagnetics with the addition of the relativistic Lorentz force law and charged particles, Figure 1. The code is massively parallel with scalability tested to over 16000 CPUs [129]. The relevant physics to model HPM devices is included in the code. A number of improvements have been done in the ICEPIC, compared to the basic algorithm used. For example employed Yee finite-difference time-domain (FDTD) technique has been updated to improve the dispersion relation, and to avoid a non-physical Cherenkov radiation. Many of ICEPIC applications are related to continuous-wave microwave power generation, which provide a significant challenge to the lacuna implementation in ICEPIC. Therefore a detailed knowledge of ICEPIC numerical techniques is of great importance. Several of the key elements of ICEPIC implementation are provided in the next subsections.

13.2 Users of ICEPIC

ICEPIC has about 20 users at AFRL, many of them are experimentalists. The code is extensively used for virtual prototyping of high-power microwave devices. Besides AFRL, ICEPIC is used at the Air Force Institute of Technology and at Arizona State University.

13.3 Algorithms implemented in ICEPIC

The technique employs Yee finite-difference time-domain (FDTD) [71] electromagnetics with the addition of charged particles. While the FDTD electromagnetic fields are defined at discrete lattice points, the charged particles are free to move to arbitrary points in space. Similar to the staggering of the FDTD electromagnetic fields in time, each particle position is known at integer time steps

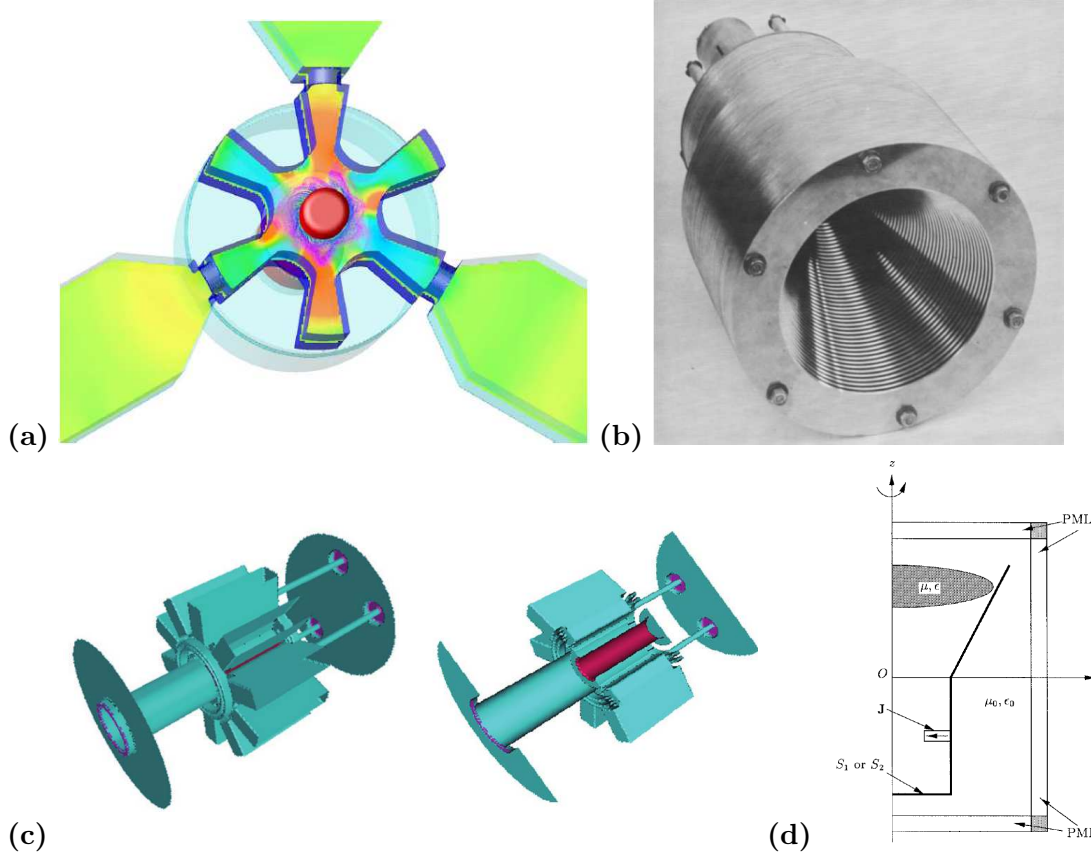


Figure 94: (a) Simulated AFRL A6-3 magnetron geometry used in performance testing by ICEPIC [129], simulates the electrodynamics and charged-particle dynamics of microwave RF-systems (Maxwell’s equations and the Lorentz force law). (b) Photo of the corrugated horn antenna (from [127], reproduced from [134]). (c) Modern CW industrial heating magnetrons are capable of producing as high as 300 kW of continuous-wave microwave power at frequencies around 900 MHz and are sold commercially [132]. (d) Diagram of a typical computational domain for ICEPIC.

while the velocity is known at half integer time-steps. At each time-step, the electromagnetic fields are weighted to the particle position, and the particle is “pushed” according to the relativistic Lorentz force law.

13.3.1 Improved Yee finite-difference time-domain (FDTD) technique. The Yee FDTD method employs a cubic lattice of cells to approximate Maxwell’s *curl* equations on a staggered grid (Fig. 95), given by:

$$\begin{aligned}\frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \mathbf{E}, \\ \frac{\partial \mathbf{E}}{\partial t} &= c^2 \nabla \times \mathbf{B} - \eta \mathbf{c} \mathbf{J},\end{aligned}\tag{150}$$

where c is the speed of light in the medium and η is the wave impedance. It achieves second order accuracy by staggering the placement of $\mathbf{E}$ and $\mathbf{B}$ in both space and time and approximating all derivative operators by centered finite differences. The electric fields ($\mathbf{E}$) are placed on cell edges

and the magnetic flux densities (B) are placed on cell faces, as shown in Fig. 95, and the electric fields are known at timesteps $n\Delta t$ while the magnetic flux densities are known at timesteps $(n + 1/2)\Delta t$, Fig. 13.3.1.

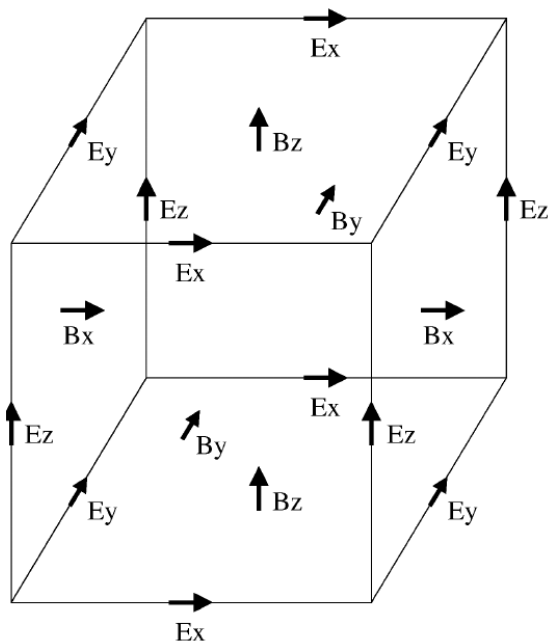


Figure 95: Staggered Yee grid for FDTD. Placement of the field components in the Yee FDTD method.

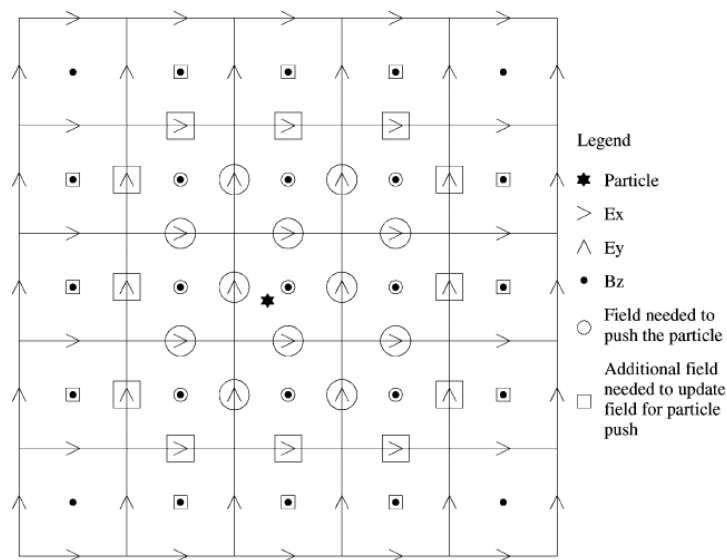


Figure 96: Fields required to push the particle (shown as a star) are shown circled. Additional fields needed to update the push fields one time-step are shown boxed.

In the Yee FDTD grid, electromagnetic waves propagate according to the dispersion relation

$$\left[\frac{1}{c\Delta_t} \sin\left(\frac{\omega k_x \Delta_t}{2}\right)\right]^2 = \left[\frac{1}{\Delta_x} \sin\left(\frac{k_x \Delta_x}{2}\right)\right]^2 + \left[\frac{1}{\Delta_y} \sin\left(\frac{k_y \Delta_y}{2}\right)\right]^2 + \left[\frac{1}{\Delta_z} \sin\left(\frac{k_z \Delta_z}{2}\right)\right]^2, \quad (151)$$

where ω is the radian time frequency, and k_x , k_y , and k_z are the wavenumbers of the propagation parallel to the associated coordinate directions. In contrast, electromagnetic waves in real, physical space propagate according to

$$\frac{\omega^2}{c^2} = k_x^2 + k_y^2 + k_z^2 = k^2, \quad (152)$$

The Yee FDTD dispersion relation is seen to approximate the physical dispersion relation (152) at low frequency as shown in Fig. 97. Note that the numerical dispersion relation approximates the physical dispersion relation exactly up to the highest frequency the grid can support when the wave propagation is along a three dimensional (3D) diagonal of the grid and $\Delta^t = \Delta^{t,max}$. For all other directions of propagation and timesteps, the propagation speed becomes slower than the physical speed of light (c) at high frequency. If charged particles travel at $0.8c$ in the simulation, note that the wave dispersion curves cross the particle speed line at high frequency, *indicating the presence of numerical Cerenkov radiation*. Note that the axes in Fig. 97 are normalized to Δ , indicating that the problem cannot be alleviated by refining the grid.

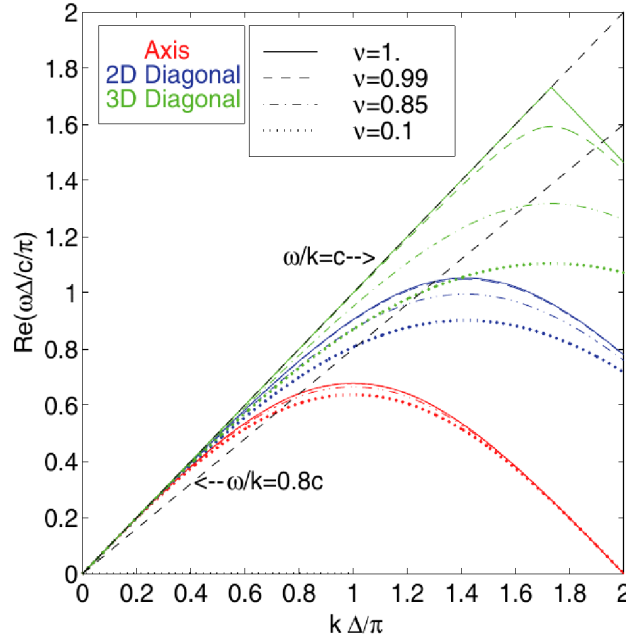


Figure 97: Yee FDTD numerical dispersion relation for propagation along a grid axis, a 2D grid diagonal, and a 3D grid diagonal compared to the physical dispersion relation. The FDTD parameters are $\Delta x = \Delta y = \Delta z = \Delta$ and $\Delta t = \nu \Delta_{t,max} = \nu \Delta / \sqrt{3}$.

13.3.2 Altering the computational stencil in ICEPIC. Altering the computational stencil involves changing the approximation to the spatial derivatives for the FDTD method. The standard Yee FDTD method uses second order centered finite differences to approximate all spatial

derivatives. Thus **curl** $\mathbf{E}$ to update the Bz value at the center of Fig. 98 is computed using the Ex and Ey values shown in green in Fig. 98 as

$$\hat{z} \cdot (\nabla \times \mathbf{E})_{I+1/2, J+1/2, K} = \frac{E_{y, I+1, J+1/2, K} - E_{x, I+1/2, J+1, K} - E_{y, I, J+1/2, K} + E_{x, I+1/2, J, K}}{\Delta} + O(\Delta^2), \quad (153)$$

where the subscripts I, J, K indicate the quantity evaluated at $(I\Delta x, J\Delta y, K\Delta z)$ and $\Delta x = \Delta y = \Delta z = \Delta$ is assumed. The **curl** can also be approximated using the values shown in red as

$$\hat{z} \cdot (\nabla \times \mathbf{E})_{I+1/2, J+1/2, K} = \frac{E_{y, I+2, J+1/2, K} - E_{x, I+1/2, J+2, K} - E_{y, I-1, J+1/2, K} + E_{x, I+1/2, J-1, K}}{\Delta} + O(\Delta^2), \quad (154)$$

or using the values shown in blue as

$$\hat{z} \cdot (\nabla \times \mathbf{E})_{I+1/2, J+1/2, K} = \frac{1}{6\Delta} (E_{y, I+2, J+1/2, K} + E_{y, I+2, J+3/2, K} - E_{x, I+3/2, J+2, K} - E_{x, I-1/2, J+2, K} - E_{y, I-1, J+3, K} - E_{y, I-1, J-1/2, K} + E_{x, I-1/2, J-1, K} + E_{x, I+3/2, J-1, K}) + O(\Delta^2), \quad (155)$$

Each of the above curl approximations is individually second order accurate. Thus, a linear combination is also second order accurate, and the curl is approximated by K_1 times Eq. (5) plus K_2 times Eq. (6) plus $(1 - K_1 - K_2)$ times Eq. (4) [130]. Note that useful values of K_1 and K_2 are $K_1 \leq 0$ and $K_2 = 0$ or $K_2 = 2K_1$. The case $K_1 = K_2 = 0$ recovers the standard Yee FDTD scheme, and the case $K_1 = -1/8, K_2 = 0$ results in a fourth order accurate approximation to the **curl** operator.

The dispersion relation for the extended stencil FDTD method is given by

$$\left[\frac{1}{c\Delta_t} \sin\left(\frac{\omega\Delta_t}{2}\right) \right]^2 = f_{x,y}f_{x,z} + f_{y,z}f_{y,x} + f_{z,x}f_{z,y}, \quad (156)$$

where

$$f_{\alpha,\beta} = \frac{1}{\Delta_\alpha} \{ [1 - K_1 - K_2] \sin\left(\frac{k_\alpha \Delta_\alpha}{2}\right) + \frac{1}{3} [K_1 + K_2 \cos(k_\beta \Delta_\beta)] \sin\left(\frac{3K_\alpha \Delta_\alpha}{2}\right) \} \quad (157)$$

More cases of different choices of K_1 and K_2 are considered in [129]. Here we can remark that for the extended stencil the timestep can be significantly increased. When $K_2 = 0$, the maximum stable timestep for the extended stencil method is $1/(1 - (4/3)K_1)$ times the maximum stable timestep for the standard Yee method. When $K_2 = 2K_1$, the maximum stable timestep is $1/(1 - (8/3)K_1)$ times the maximum stable timestep for the standard Yee method. The dispersion relation is plotted for the case $K_1 = -1/8, K_2 = 0$ and $\Delta x = \Delta y = \Delta z = \Delta$ in Fig. 6. In comparing Figs. 6 and 4, note that the situation in Fig. 4 is much improved. In Fig. 4, the 0.8c particles do not exceed the numerical propagation speed for most timesteps until just before the grid cuts off the propagation. Note that the grid cut off point in any direction corresponds to the point where the dispersion curves reach their maximum and start to decrease. As K_1 becomes more negative, the situation improves even more, as demonstrated by Fig. 5. Note, however, that as K_1 becomes more negative, the maximum stable timestep also decreases.

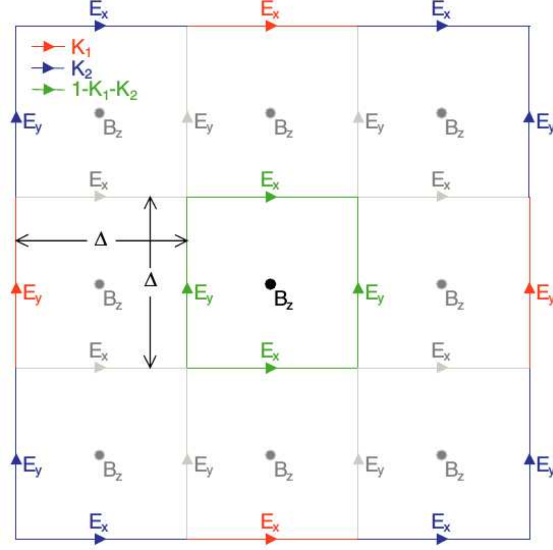


Figure 98: E_x and E_y values used to compute $\text{curl } \mathbf{E}$ to update B_z . The standard Yee FDTD scheme uses the values shown in green. The values shown in red and blue can be used in addition to the values shown in green.

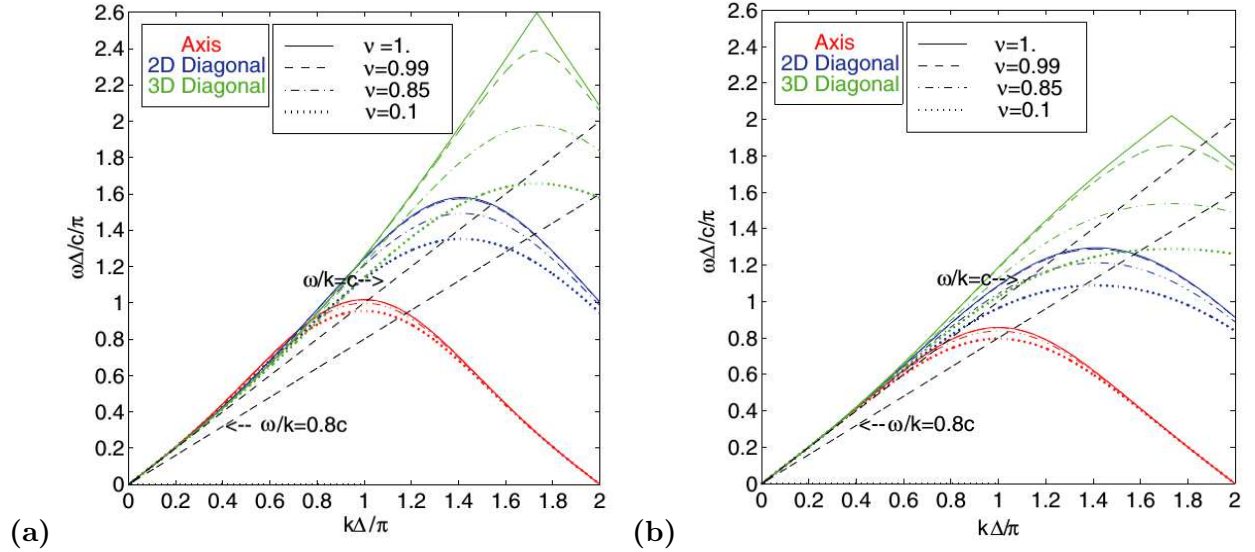


Figure 99: (a) Extended stencil numerical dispersion relation for $K_1 = -3/8$; $K_2 = 0$; $\Delta x = \Delta y = \Delta z = \Delta$; and $\Delta t = \nu \Delta_{t,max} = (2/3)\nu \Delta/c/\sqrt{3}$. (b) Extended stencil numerical dispersion relation for $K_1 = -1/16$, $K_2 = 2K_1$, $\Delta x = \Delta y = \Delta z = \Delta$; and $\Delta t = \nu \Delta_{t,max} = (6/7)\nu \Delta/c/\sqrt{3}$.

13.3.3 Filtering. The Fourier transform is used to eliminate numerical Cerenkov radiation. Several filters are employed: (i) Godfrey filter, and (ii) Friedman filter. When using filtering schemes, care must be taken not to alter the physics of interest in the numerical simulation. In examining the effect of filters on the wave dispersion relation, it is found that the ideal filter is characterized by an all pass magnitude response with zero phase in the frequency band of physical interest and increasing phase as the slowly traveling high frequencies near the grid cut off point are

approached. Note that the ideal filter is not a low-pass filter [128].

13.3.4 ICEPIC parallel performance. The coupling of the lacuna method to the ICEPIC to be designed for parallel implementation and demonstrate efficient scaling of CPU time vs. the number of nodes as the original ICEPIC.

The finite-difference time-domain technique for ICEPIC simulation of electromagnetic and low-density plasma phenomena is computationally expensive and can require tens of thousands of computer hours to produce one solution. Substantial gains can be made through memory streamlining (factors of 2.3 faster), efficient cache usage (factors of 3 improvement), and through better parallel design (improving scalability to four times the number of CPUs). These improvements are documented and tested across five different supercomputing hardware platforms for idealized problems designed to highlight the effect of the changes. Then, the cumulative effect of these changes are tested across the five different systems for a typical problem of interest, a relativistic magnetron, on 48 CPUs which shows a factor of two to seven reduction in run-time, or best case, from 21 h to only 3 h.

Traditional ICEPIC loop which attempts to overlap computation and communication but has multiple synchronization points each time-step is shown in Fig. 100. The overall performance of any parallel code is determined by two factors: the parallel efficiency (determined by Amdahl’s law) and the serial performance. FDTD codes tend to be limited exclusively by memory access as can be seen by examining the standard Yee update. Therefore, memory bandwidth is the limiting factor, as the CPU can perform updates at four times the memory speed. Consequently, optimizations in memory access provide a much higher return in real performance than optimizations in the number of floating-point operations.

The analysis for particle updates is quite similar, albeit more complicated. There are three steps to the particle update; first, interpolate the fields to the particles. Second, update the particle velocity through the Boris push [135], and finally, update the position and weight the currents. The memory bandwidth is the key limiting factor in EM-PIC code performance. Therefore, any reduction of load on the memory bus results in an improvement in update performance. One way to reduce memory requirements is to assume a spatially constant grid or physical properties. This yields a factor of two fewer memory references than required in the general update; however, code generality is lost if these assumptions are made. ICEPIC attempts to strike a balance between reduced memory and generality by storing cell properties such as cell size and electromagnetic permittivity, permeability, and conductivity in a fast look-up table implemented by a red black balanced tree [136].

Improving the performance of single CPU code after memory streamlining resulted in 2.22X code speed-up (theoretical value is 2.33X) [129]. Employing the related cache locality for update performance by particle sorting resulted in additional 3X speedup of the code. The redesigned ICEPIC look is shown in Fig. 102. The parallel performance of the code has been tested on a simulation of AFRL A6-3 relativistic magnetron geometry, Fig. 101. In realistic performance tests, the problem size was fixed at roughly 8 M cells and 700 K particles. This problem is scaled from 4 CPUs (the lowest number of CPUs which have sufficient memory) to 128 CPUs while maintaining the size of the problem. Performance results of these runs are shown in Fig. 103.

When scaled the problem size with the number of CPUs one can see much larger gains [Fig. 104(b)] due to the improved algorithm. This problem is scaled using a typical mesh spacing proportional to the quad root of the number of CPUs, maintaining the final simulated time (number of time-steps proportional to number of CPUs to the 1/4th power). With this scaling there are more particles in the simulation, and therefore, more likelihood for the simulation to be out of balance

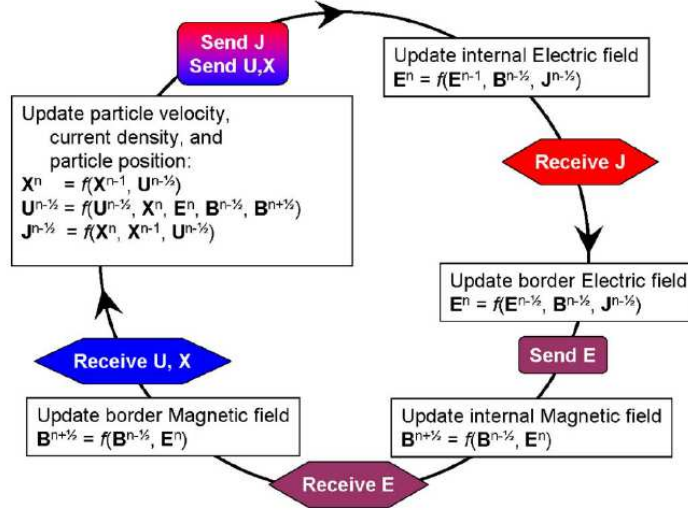


Figure 100: ICEPIC loop which attempts to overlap computation and communication but has multiple synchronization points each time-step.

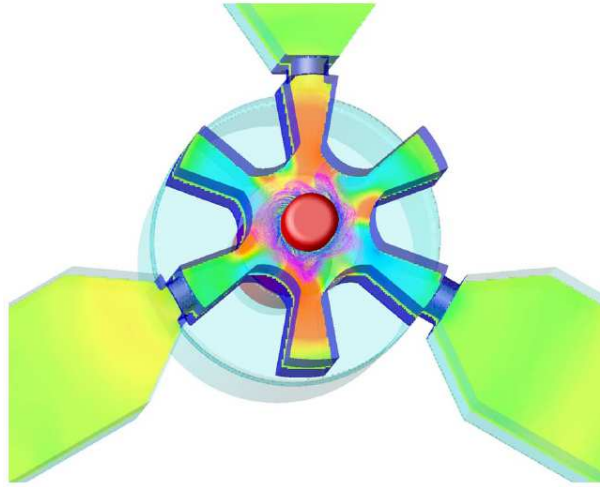


Figure 101: Simulated AFRL A6-3 magnetron geometry used in performance testing.

since the particles tend to be clustered in a small area and each CPU owns a smaller portion of the total area.

The full effect of all these changes is tested on a standard magnetron at 48 CPUs, mimicking the tests for the improvements in parallelization listed above. The results in Table 4 show that there is roughly a factor of two to seven improvement with the combined effects of all these changes. This has a huge impact on daily operations [129].

13.3.5 Summary on parallel performance. Great care must be taken in the development of EM-PIC codes to achieve high levels of performance. These codes are typically dominated by memory access patterns, and not by CPU performance. Reduction in the amount of memory being passed through the CPU results in a direct reduction of run-time. Furthermore, reusing memory already loaded results in improved performance. Gains from two to four can be achieved by a slight

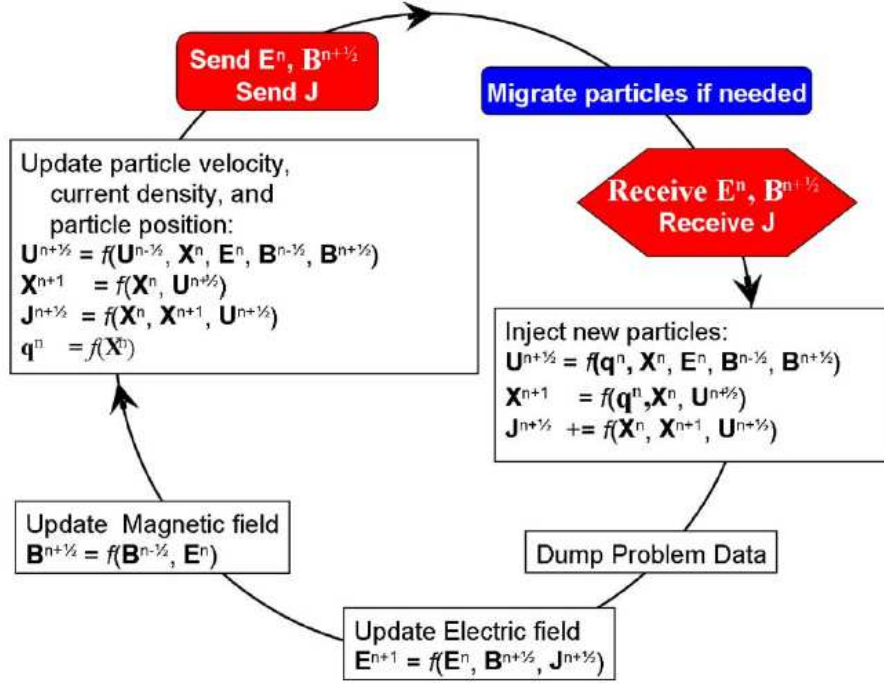


Figure 102: Redesigned ICEPIC loop with a single communication point each timestep.

restructuring of the memory access patterns.

Machine	A	B	C	D
CPU	Xeon EM64T 3.6 GHz	2-Core Opteron Y255 2.6 GHz	Itanium 2 1.6 GHz	Power5+ 1.9 GHz
Memory	2 GB	2 GB	2 GB	2 GB
Cache	2 MB	1 MB	8 MB L3	2 MB L2/8 MB L3
Memory Bandwidth	10.6 GB/s	6.4 GB/s	6.4 GB/s	21.2 GB/s
OS	Linux 2.6.5	Cray Linux 1.5.39	Linux 2.6.5	IBM AIX 5.3L
Compiler	32 Bit Intel C 9.1	64 Bit Pathscale C 3.0	64 Bit Intel C 9.1	64 Bit XLC 8.0
Interconnect	Myrinet	Cray SeaStar	SGI NUMA	HPS

Table 4: Specification of hardware tested

Machine (see Table I)	A	B	C	D
Run-time Before (s)	75171	19979	102489	20377
Run-time After (s)	10868	8226	16048	10042
Factor of Improvement	692%	243%	638%	203%

Table 5: Performance improvements due to all changes referenced in this report for the AFRL A6-3 magnetron performed with 48 CPUs.

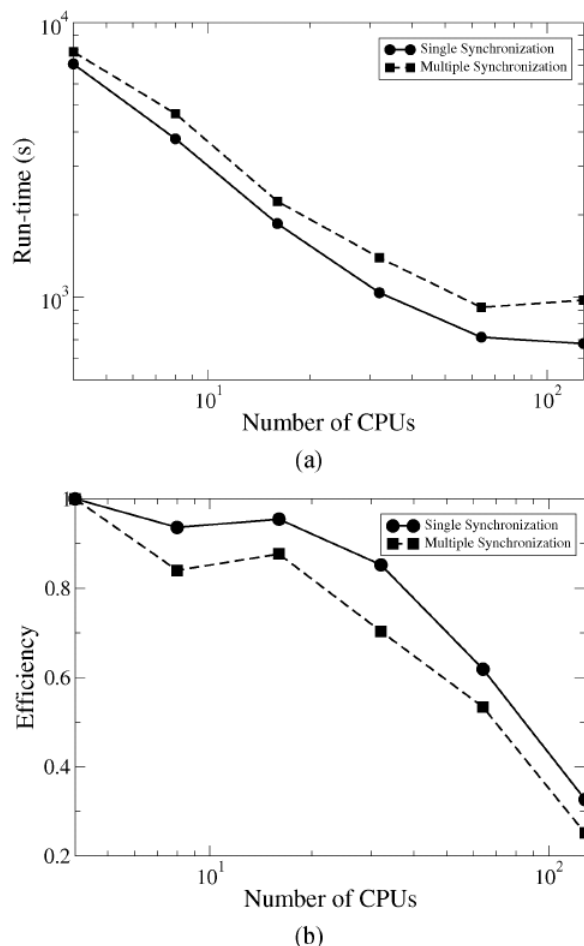


Figure 103: Run-time and efficiency for the AFRL A6-3 magnetron for both the single and multiple synchronization point algorithms. The runs are performed on machine B for a fixed problem size. (a) Run-time. (b) Efficiency.

13.4 ICEPIC installation and testing

ICEPIC is a large C++ code (350,000 source lines) developed by numerous scientists and engineers since 1994 at the AFRL at Kirtland AFB, Albuquerque, NM. It includes a number of additional related packages, and has been installed in a number of AFRL/Army sites.

The main option available is a generic Linux/GNU installation and compiling, which is based on cmake utility for C++. We have been able to install ICEPIC on the recent Linux OpenSUSE 13.2 system and Ubuntu 12.04. Cmake was unable to find gcc/g++ compilers, which are available. In case of OpenSUSE13.2, the following are additional commands to install packages needed and to modify the environmental variables:

```
# sudo zypper install mpich-devel
# export PATH=$PATH:/usr/lib64/mpich/bin
```

Also the C++ source correction was needed to fix H5FDstdio.c to define missing types: `_off64_t` and `off64_t`:

```
# diff hdf5/src/H5FDstdio.c hdf5/src/H5FDstdio.c.org
```

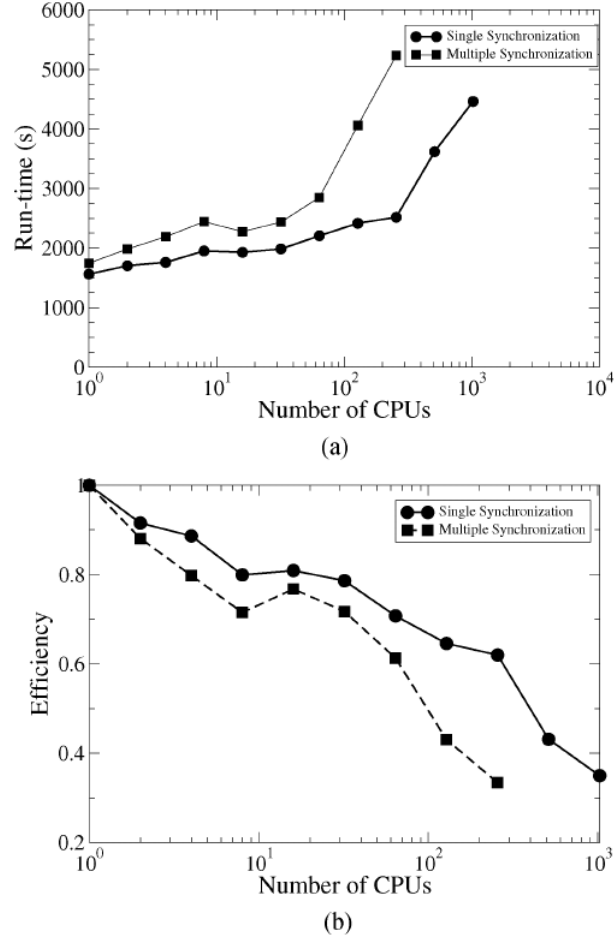


Figure 104: Run-time and efficiency for the AFRL A6-3 magnetron for both the single and multiple synchronization point algorithms. The runs are performed on machine with a number of processors maintaining constant $(NX*NY*NZ)/\Delta t \#CPU$. (a) Run-time versus # of processors (b) Efficiency.

```
66,68d65
< typedef __off64_t off_t;
< #define off64_t __off64_t
<
#
```

This was done for gcc compiler version 4.8.3 20140627: gcc-4_8-branch revision 212064 SUSE Linux, and cmake version is cmake-3.0.2-1.1.x86_64.

While working on the lacuna implementation in the ICEPIC, we have been informed by AFRL *Dr. Peter Mardahl*, that a newer version of the ICEPIC is available. We have installed the most recent version of ICEPIC, which has required a newer version of cmake for the OpenSUSE 13.2, cmake-3.0.2-1.1.x86_64 was OK for that. A number of ICEPIC tests (ten+) has been successfully completed, see e.g. Section 14, and Section 15.

Now we are completing the implementation and testing the lacuna method in the ICEPIC code.

13.5 PML module in ICEPIC

The analysis conducted on the ICEPIC source code resulted in a selection of a candidate module as a template for lacuna implementation in the ICEPIC. This is the PML module of the ICEPIC.

In this case we have the needed auxiliary layers structures ready for the solution of the auxiliary problem [64]. Therefore this module serve as a perfect template for the lacuna implementation. Below we also provide a description of PML in ICEPIC.

When simulating electromagnetic waves, it is often desirable to simulate an open region of space. Unfortunately, this requires enforcement of the Sommerfeld radiation condition, which is a boundary condition at infinity. Exact enforcement of this condition at a finite distance is non-local in space and in time. Thus, approximate boundary conditions, known as absorbing boundary conditions, are used. Qualitatively, when a wave strikes a good absorbing boundary condition, it should leave the computational domain and never return, independent of its frequency or the angle at which it strikes the boundary. The introduction of perfectly matched layer (PML) in the mid 1990's by J. P. Berenger [115] represents a major advance in absorbing boundary conditions. PML remained the best known absorbing boundary condition technology till the invention of the Lacuna method. A well designed PML can absorb waves such that any reflected waves are below numerical precision. It can, however, be a challenge to design the PML. Fig. 105 shows how PML works. The source of radiation is at the left edge of the picture, and the PML occupies the space to the right of $z = 0$. Notice how the fields are quickly attenuated to zero inside the PML. After the PML absorbs the waves, the space can be truncated with any boundary condition, and little to no reflection results.

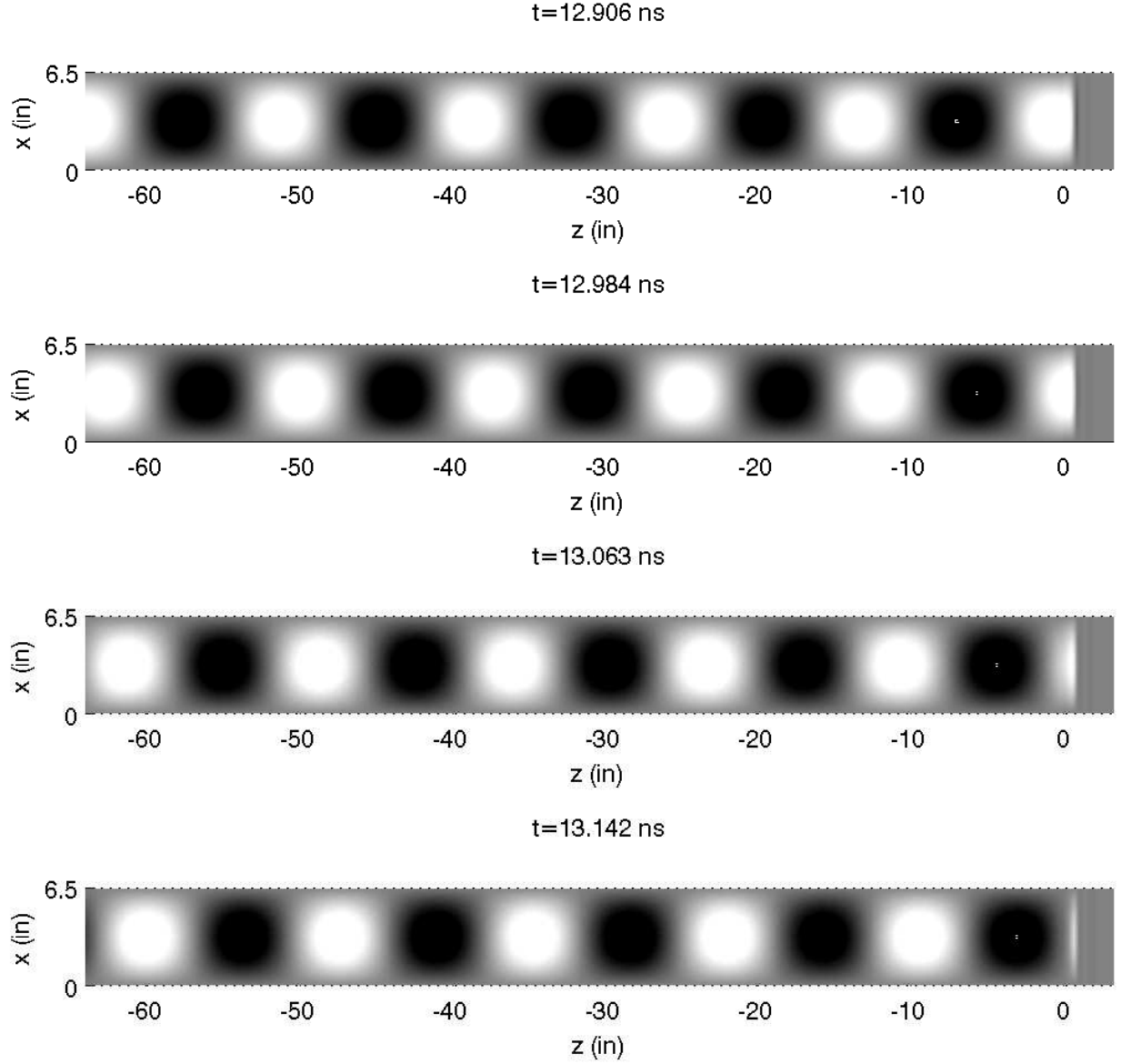


Figure 105: Picture of PML in action. The source of radiation is at the left end, and the PML begins at $z = 0$.

13.6 Formulation

To understand how to design a good PML, it is first necessary to understand some background about the formulation. Berenger originally proposed a split-field formulation, which can suffer from numerical difficulty due to being weakly well-posed [116]. The stretched coordinate and uniaxial material formulations alleviate this problem [117]. The coordinate stretching formulation is summarized here.

Let

$$\begin{aligned}\tilde{x} &= s_x x \\ \tilde{y} &= s_y y \\ \tilde{z} &= s_z z\end{aligned}\tag{158}$$

and replace x in Maxwell's equations with $s_x x$, y with $s_y y$, and z with $s_z z$. The quantities s_x , s_y , and s_z are known as coordinate stretching factors. The normal Maxwell's equations are recovered if $s_x = s_y = s_z = 1$. Further, it can be shown that no reflections occur [117] if s_x is a function of x only, s_y is a function of y only, and s_z is a function of z only. Thus, the coordinates can be “stretched” near the edges of the domain to absorb outgoing waves.

In order to absorb outgoing waves, the PML should mimic a lossy material. Thus, the regular PML is characterized in the frequency domain ($e^{j\omega t}$ convention) by

$$s_\beta = 1 + \frac{\sigma_1(\beta)}{j\omega\epsilon}\tag{159}$$

where β represents x , y , or z , and ϵ is the permittivity of the background material, usually the permittivity of free space. Note that σ_1 and several other PML parameters in the expressions below are typically tapered functions of β ; this is discussed further in Sec. 13.7. The regular PML works well to absorb propagating waves, but it fails to absorb evanescent waves [118–120]. Thus, the real part of the stretching factor can be made different from unity [118]

$$s_\beta = \kappa_1(\beta) + \frac{\sigma_1(\beta)}{j\omega\epsilon}\tag{160}$$

Complex frequency shifted PML (CFS-PML) is introduced to better absorb evanescent waves [120–122]

$$s_\beta = \kappa_1(\beta) + \frac{\sigma_1(\beta)}{\alpha_1(\beta) + j\omega\epsilon}\tag{161}$$

However, the CFS-PML fails to absorb low frequency propagating waves; thus, the second order PML is introduced [120]

$$s_\beta = \left(\kappa_1(\beta) + \frac{\sigma_1(\beta)}{\alpha_1(\beta) + j\omega\epsilon} \right) \left(\kappa_2(\beta) + \frac{\sigma_2(\beta)}{j\omega\epsilon} \right)\tag{162}$$

The regular, CFS, and second order PMLs are all available within ICEPIC. Which PML should be chosen depends on what types of waves must be absorbed. The regular PML is perfectly adequate to absorb waves in free space. The CFS PML should be used to absorb waves in hollow waveguides, where evanescent modes may be present. The second order PML should be used if an inner conductor is added to the waveguide (as it is in coax), where both evanescent modes and low frequency propagating modes may be present. Note that unabsorbed evanescent modes can cause problems in a simulation, including numerical instability. Thus, in a waveguide, it is important to select a PML that absorbs any spurious evanescent modes, even if no such modes are expected to be present. Selection of the PML and its parameters in ICEPIC will be discussed in more detail in the next report.

13.7 Tapering PML Parameters

Although no reflections occur in continuous space if s_β is a function of β only (β represents x , y , or z), large jumps in s_β do result in reflections on a discrete mesh. Thus, the PML parameters are tapered from their free space values to large values needed to absorb electromagnetic radiation. Referring to Fig. 106,

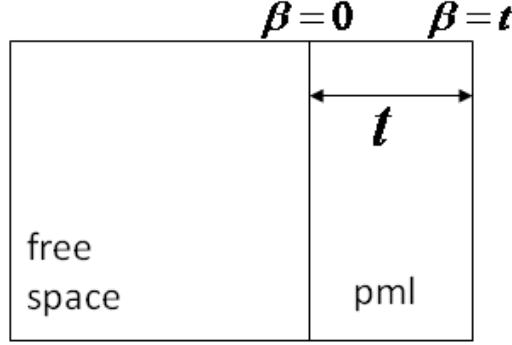


Figure 106: Geometry for tapering of PML parameters.

the PML parameters are given by

$$\begin{aligned}
 \sigma_1(\beta) &= \sigma_{1,\max} \left(\frac{\beta}{t} \right)^{n_{\sigma_1}} \\
 \kappa_1(\beta) &= 1 + (\kappa_{1,\max} - 1) \left(\frac{\beta}{t} \right)^{n_{\kappa_1}} \\
 \alpha_1(\beta) &= \alpha_{1,\text{start}} + (\alpha_{1,\text{end}} - \alpha_{1,\text{start}}) \left(\frac{\beta}{t} \right)^{n_{\alpha_1}} \\
 \sigma_2(\beta) &= \sigma_{2,\max} \left(\frac{\beta}{t} \right)^{n_{\sigma_2}} \\
 \kappa_2(\beta) &= 1 + (\kappa_{2,\max} - 1) \left(\frac{\beta}{t} \right)^{n_{\kappa_2}}
 \end{aligned} \tag{163}$$

Note that, with the exception of α_1 , the PML parameters revert to free space values (coordinate stretching factors become 1) at $\beta = 0$, which is the location of the interface with free space. A jump in α_1 does not result in a large spurious reflection, so there is no need for it to revert to a free space value. In fact, it is common to choose $\alpha_{1,\text{start}} = \alpha_{1,\text{end}}$, thereby eliminating tapering.

13.8 Specification of PML to ICEPIC

A PML is a boundary condition and is specified in ICEPIC in a [Bound#] section. [Bound#].method is required to be set to PML. [Bound#].dir is also required and is set to the direction normal to the boundary. [Bound#].shape is required, and as with other boundary conditions, it should encompass the faces over which the boundary condition applies. Note that, as with other boundary conditions, [Bound#].shape is a fraction of a cell size thick; the thickness of the PML is specified by either [Bound#].rdepth or [Bound#].ndepth. As the names suggest, use [Bound#].rdepth to specify the PML depth in real units (the actual unit depends on [Cartesian].in.units), or use [Bound#].ndepth to specify the depth as a number of cells. The thickness of

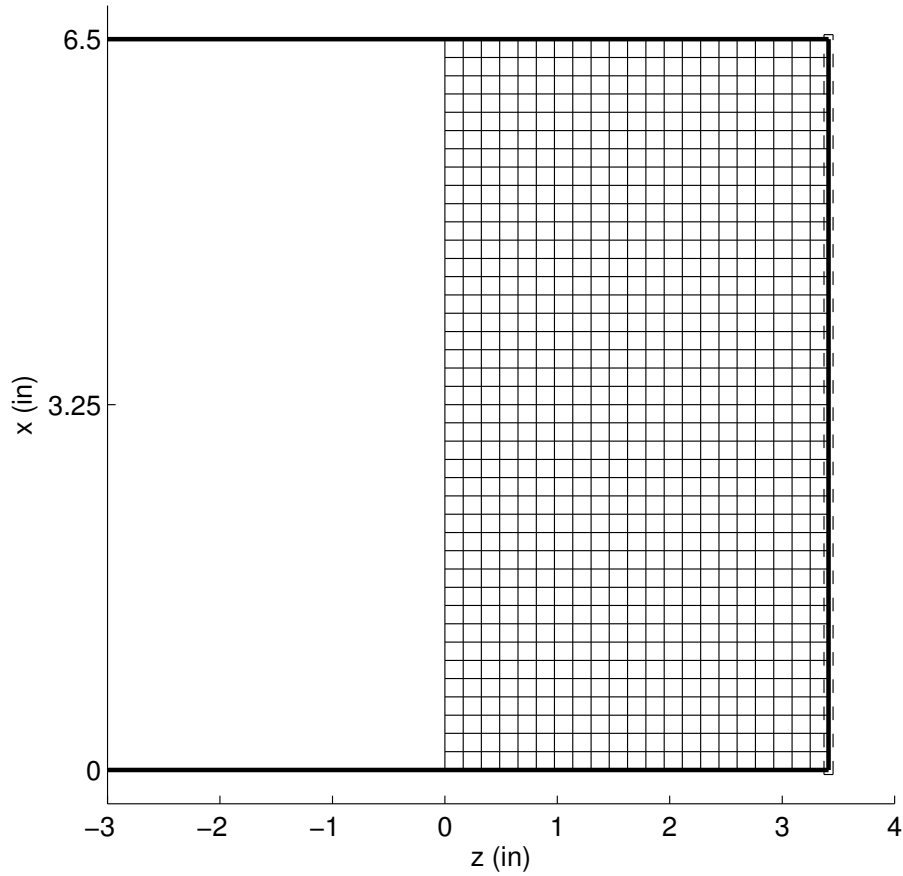


Figure 107: Slice of the geometry of a PML in a WR-650 waveguide. The PML occupies the gridded region. The waveguide walls are shown as darkened lines. The dotted line shows the region encompassed by `[Bound#].shape`. In this case, `[Bound#].ndepth` is 21, and `[Bound#].dir` is Z. All of the space inside the waveguide walls is marked as vacuum by the `[Shape#]` commands, and the PML grows inward (i.e., in the minus Z direction in this example) into the vacuum region.

the PML grows inward into the computational domain from the faces marked by `[Bound#].shape`. Thus, it is important to allow for the space to be occupied by the PML when designing the computational domain, and it is important that the `[Shape#]` sections specify this space as vacuum or as a simple dielectric region.

Fig. 107 shows which grid faces `[Bound#].shape` should enclose and where the resulting PML grows. As discussed in Sec. 13.6, it is important that the PML stretching parameters be a function only of the direction normal to the boundary. If a stretching parameter is a function of a transverse value, numerical instability can result. Thus, the shape specified by `[Bound#].shape` must extend to metal on all sides of the PML through its entire thickness. The code attempts to check the PML cells to ensure this condition is met and exits with an error if the condition fails. The error can be the result of a problem with `[Bound#].shape` or with one of the `[Shape#]` sections in the input file. Note that no problems are encountered when PMLs of different directions overlap at corner regions.

Table 6: Input file labels for PML parameters

Input file label	PML parameter	Default Value
[Bound#].sig1	$\sigma_{1,\max}$	
[Bound#].nsig1	n_{σ_1}	3
[Bound#].k1	$\kappa_{1,\max}$	1.0
[Bound#].nk1	n_{κ_1}	2
[Bound#].alstart	$\alpha_{1,\text{start}}$	
[Bound#].alend	$\alpha_{1,\text{end}}$	
[Bound#].na1	n_{α_1}	3
[Bound#].sig2	$\sigma_{2,\max}$	0.0
[Bound#].nsig2	n_{σ_2}	3
[Bound#].k2	$\kappa_{2,\max}$	1.0
[Bound#].nk2	n_{κ_2}	2

PML parameters are further controlled according to Table 6. PML is very flexible for use in different scenarios, but because of that flexibility, it is very difficult to select proper parameters. In the remainder of this chapter, first some guidance on selection of parameters is given, and then some sets of defaults that work for many scenarios are discussed.

For guidance on selecting PML parameters, first, n_{σ_1} , n_{κ_1} , n_{α_1} , n_{σ_2} , and n_{κ_2} do not usually need to be changed from their default values. Then, begin with a thickness of about one quarter wavelength at the frequency band of interest. If you are placing a PML to absorb waves propagating in free space, a regular PML is sufficient. TPML Input Examples he parameters, except the depth and σ_1 , can be left at their default values. The code has a convenient input specified by [Bound#].R; when [Bound#].R is specified, the code computes $\sigma_{1,\max}$ so that the maximum spurious reflection coefficient is [Bound#].R. The equation to compute $\sigma_{1,\max}$ is given by

$$\sigma_{1,\max} = -\frac{(n_{\sigma_1} + 1)\ln(R)}{2t} \quad (164)$$

where R is [Bound#].R and t is the thickness (i.e., depth) of your PML in real units. For a regular PML of about one quarter wavelength at the frequency band of interest, [Bound#].R=10⁻⁴ usually results in good absorption. However, note that there are two mechanisms for spurious reflection from a PML, and Eq. 164 considers only one of them. Eq. 164 computes the reflection based on the two way attenuation when a wave propagates through the PML. If the spurious reflection due to the discrete nature of the FDTD mesh dominates, you may observe higher reflection than R . In this case, the solution is probably a thicker PML to reduce the cell-to-cell jump in the parameters. Note that [Bound#].R should not be used for CFS or second order PMLs.

Next, if one is placing a PML in a hollow waveguide, one needs a CFS-PML. This requires specification of σ_1 , κ_1 , and α_1 . Default parameters that work well for standard rectangular waveguide sizes are precomputed, and setting [Bound#].mode=WR-xx where WR-xx is an entry in a "Table of PML defaults corresponding to standard rectangular waveguide sizes" (Table PML, see ICEPIC tutorial), causes the code to use the corresponding default values for $\sigma_{1,\max}$ and α (tapering of α is disabled by setting $\alpha_{1,\text{start}} = \alpha_{1,\text{end}}$). For all of the cases in Table PML, $n_{\sigma_1} = n_{\alpha_1} = n_{\sigma_2} = 3$, and $n_{\kappa_1} = n_{\kappa_2} = 2$, $\kappa_{1,\max} = 100.0$, $\alpha_{1,\text{start}} = \alpha_{1,\text{end}} = \alpha$, $\sigma_{2,\max} = 0.0$, $\kappa_{1,\max} = 1.0$ and [Bound#].ndepth=20. These defaults are derived so that the spurious reflection is less than -40 dB from 0.8 times the lowest cutoff frequency to 1.2 times the frequency at the top of the usable band when the cell depth is one twentieth of a wavelength at the highest frequency of the usable band.

If the cell size in your simulation is different from one twentieth of a wavelength at the highest frequency of the specified usable band, you should override the default value by specifying either [Bound#].rdepth or [Bound#].ndepth. Note that these default values also work well for other waveguide shapes such as circular cylindrical waveguide. Simply compute the lowest cut-off frequency for your waveguide and choose the entry in Table of PML defaults whose lowest cutoff frequency is closest to your waveguide.

Finally, if you are placing PML in a waveguide with a center conductor such as a coax, you need a second order PML. In this case, compute the lowest non-zero cut-off frequency of your waveguide and set [Bound#].mode to the entry in Table PML whose cut-off frequency is closest to the one you computed. Then, override the value for $\sigma_{2,\max}$ by specifying [Bound#].sig2. A good value is given by

$$\sigma_{2,\max} = -\frac{(n_{\sigma_2} + 1)\ln(R)}{2t} \quad (165)$$

where R is the maximum reflection coefficient you can tolerate (usually something on the order of 10^{-4}) and t is the thickness (i.e., depth) of your PML in real units. The comments about Eq. 164 considering only one of two PML reflection mechanisms also apply to Eq. 165. As you increase $\sigma_{2,\max}$, it may be necessary to thicken your PML.

13.9 Advanced Topics

There are a few advanced ideas to discuss relative to PML. First, it is possible to taper the PML parameters over a partial PML depth and leave them at their maximum values over the remainder of the depth. To do this, specify either [Bound#].rpdepth or [Bound#].npdepth. As usual, [Bound#].rpdepth is in real units while [Bound#].npdepth is a number of cells. The depth specified by one of these inputs is the depth over which the parameters are tapered. It should be less than the total depth of the PML specified by [Bound#].rdepth or [Bound#].ndepth.

Next, by default the back of the PML is simply metallic boundary condition. By specifying [Bound#].vp to be the phase velocity of maximum absorption, the metallic backing is replaced by a first order Mur absorbing boundary condition. PML absorption is usually sufficient so that this is not necessary.

Finally, there is a matlab script called “pml.m” in the ICEPIC tools directory. This script implements some very complicated equations derived by Berenger [123] to find the reflection from a CFS-PML, and it is used to find the defaults in Table PML. In contrast to Eq. 164 and Eq. 165, Berenger’s equations consider all reflection mechanisms from a CFS-PML. The equations can be used for regular PML as well as CFS-PML by setting the values for α to zero and the values for κ to one. For a second order PML, the two pieces (i.e., the subscript 1 and the subscript 2 parameters) can be adjusted independently with two applications of the script.

There are some descriptive comments at the top of the file “pml.m”, and the script inputs are a matlab vector giving the frequencies of interest in hertz, the depth of the PML in number of cells, the cell size in meters, the Courant value (which is the simulation time step divided by the maximum stable 3-D simulation time step), $\sigma_{1,\max}$ in Siemens/meter, n_{σ_1} , $\alpha_{1,\text{start}}$ in Siemens/meter, $\alpha_{1,\text{end}}$ in Siemens/meter, n_{α_1} , κ_1 , and n_{κ_1} . The next two inputs are called theta and X, and these need not be specified when designing PML in a waveguide. To avoid specifying a parameter in matlab, you can enter [] (i.e., a opening square bracket followed by a closing square bracket with nothing between them). The final input is the eigenvalue of the mode you want to absorb in inverse meters. The script plots the reflection from the PML in decibels as a function of frequency. It runs quickly, so it can be used to adjust the PML parameters until a desired level of performance is reached.

Thus, if the values in Table PML do not work for your situation, you can use this script to find parameters that work for you. For a second order PML, σ_2 and κ_2 are adjusted independently of the subscript 1 parameters. The script can be used to adjust σ_2 and κ_2 by setting $\alpha_{1,\text{start}}$ and $\alpha_{1,\text{end}}$ to zero and replacing the subscript 1 with the subscript 2 on the other parameters. These values are normally set the same as a regular PML for absorbing any low frequency propagating waves of interest. One final note on using the script to adjust CFS-PML parameters is that a good starting point for setting α_1 is to set $\alpha_{1,\text{start}} = \alpha_{1,\text{end}}$ a little less than $2\pi\epsilon f_c$ where ϵ is the permittivity of the material in the waveguide and f_c is the lowest non-zero cut-off frequency.

14 Practical ICEPIC applications: magnetron characterization and design

ICEPIC is an efficient tools to model high power microwave (HPM) devices: high-power microwave tubes, including magnetrons, klystrons, gyrotrons, and Magnetically Insulated Line Oscillators (MILOs), Fig. 108.

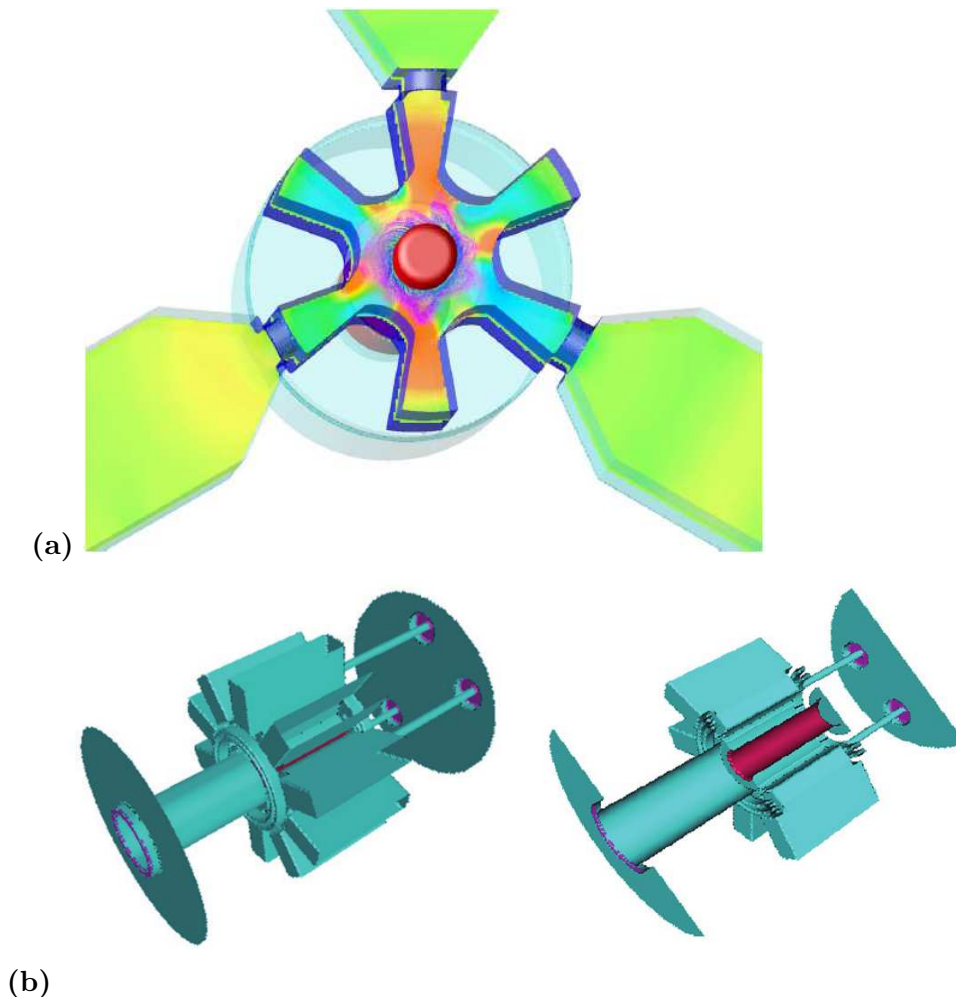


Figure 108: (a) Simulated AFRL A6-3 magnetron geometry used in performance testing by ICEPIC [129], simulates the electrodynamics and charged-particle dynamics of microwave RF-systems (Maxwell's equations and the Lorenz force law). (b) Modern continuous wave (CW) industrial heating magnetrons are capable for producing as high as 300 kW of continuous-wave microwave power at frequencies around 900 MHz and are sold commercially [132].

Modern CW industrial heating magnetrons are capable for producing as high as 300 kW of continuous-wave microwave power at frequencies around 900 MHz and are sold commercially [133]. However, to utilize these magnetrons in some specific research and scientific applications being of interest for the Air Force, the necessary adaptation and redesign are required. It means that the detailed knowledge of principles of their operation and full understanding of how the changes of the design parameters affect their operational characteristics are necessary.

Simulation is an efficient way of the estimation of the strapped magnetron operation and per-

formance. In this way the operational characteristics can be obtained : frequency and mode of magnetron oscillations, output microwave power and efficiency of magnetron operation, anode current and anode-cathode voltage dynamics, as demonstrated by AFRM team, Andreev and Hendricks [132], Fig. 108(b), and UNM team [131], Fig. 109.

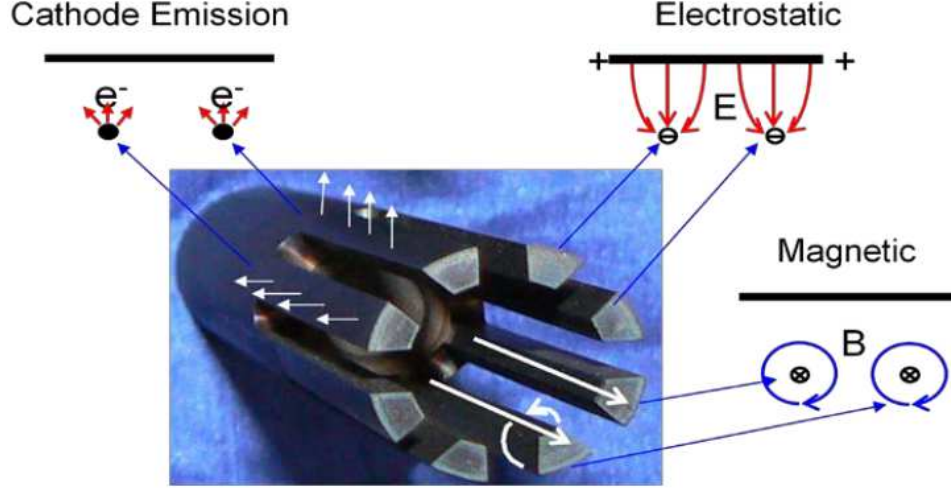


Figure 109: ICEPIC Simulations of a Pulsed Relativistic Magnetron with Transparent Cathode at UNM. Photograph of transparent cathode with field lines showing cathode priming, electrostatic priming, and magnetic priming [131].

Successful design and 3-D particle-in-cell simulation of a 140 GHz spatial-harmonic magnetron has been demonstrated by a Technische Universitat Hamburg group, Germany [138]. Their model consisted of an anode block with 28 side resonators, an impedance matching section (a $\lambda_g/4$ transformer), an output waveguide (WR6), and a cold secondary-emission cathode with two nonmagnetic end caps, which provide uniform magnetic field, and confine the electron cloud in the interaction region (Fig. 110).

14.1 AFRL A6-3 magnetron testcase for Lacuna method demonstration.

Due to the importance of the application area described above, we have selected another problem for the Lacuna method testing and comparison: the AFRM A-6 magnetron, Fig. 108(a) and Fig.111. Test case for this model has been completed successfully.

Total mesh contains 465,357 cells. PML boundary condition is set for each external face of the domain.

The simulation time is about 9mins on the 4 cores Intel Xeon 2.6GHz processor, and the ICEPIC results of A6-magnetron are presented in Fig. 112. Stability of simulations and accuracy dependence on the PML parameters are being investigated. We conduct a search of PML parameters range, where the solution becomes unstable. In that case, the Lacuna method will be applied to provide the numerical solution stabilization with a guaranteed accuracy at a large time.

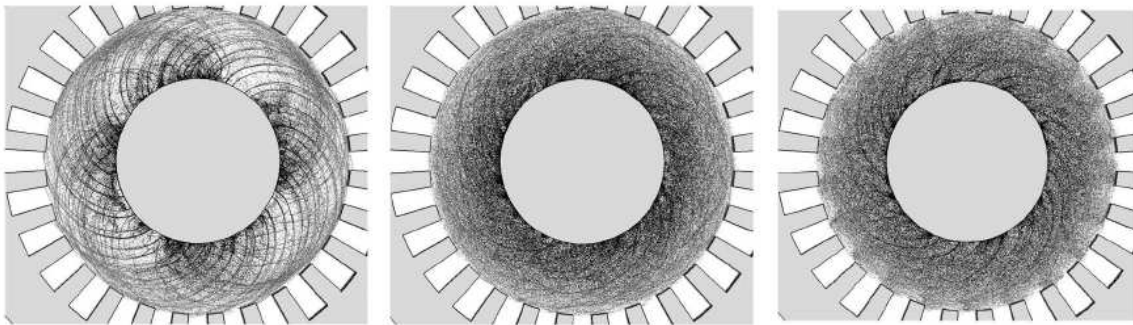
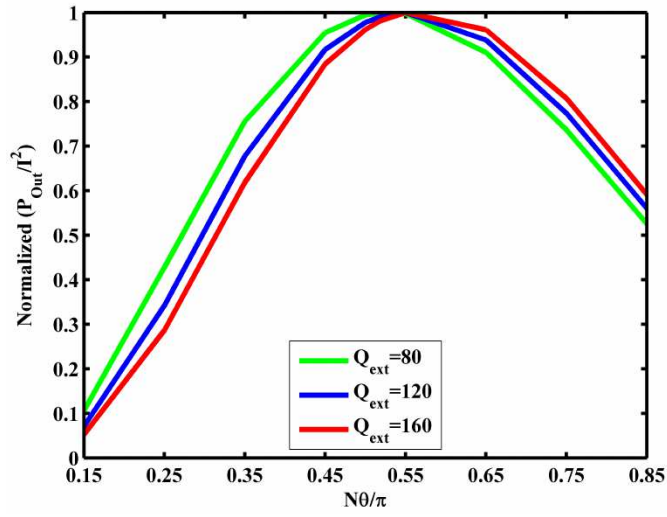
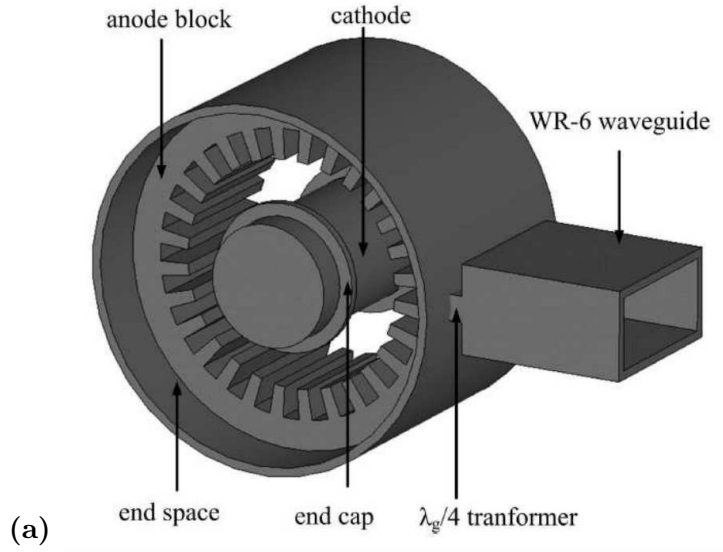


Figure 110: (a) Simulation model for a 140GHz spatial-harmonic magnetron, [138]. (b) Normalized values of P_{out}/I^2 versus $N\theta/\pi$ for Q_{ext} varying from 80 to 160. (c) Time-evolved spoke formation plotted in the $r\theta$ plane at (a) 0.4 ns, (b) 1.4 ns, and (c) 5 ns. The results are for $V_{DC} = 11.3\text{kV}$, $B = 0.79T$, and $Q_{ext} = 160$.

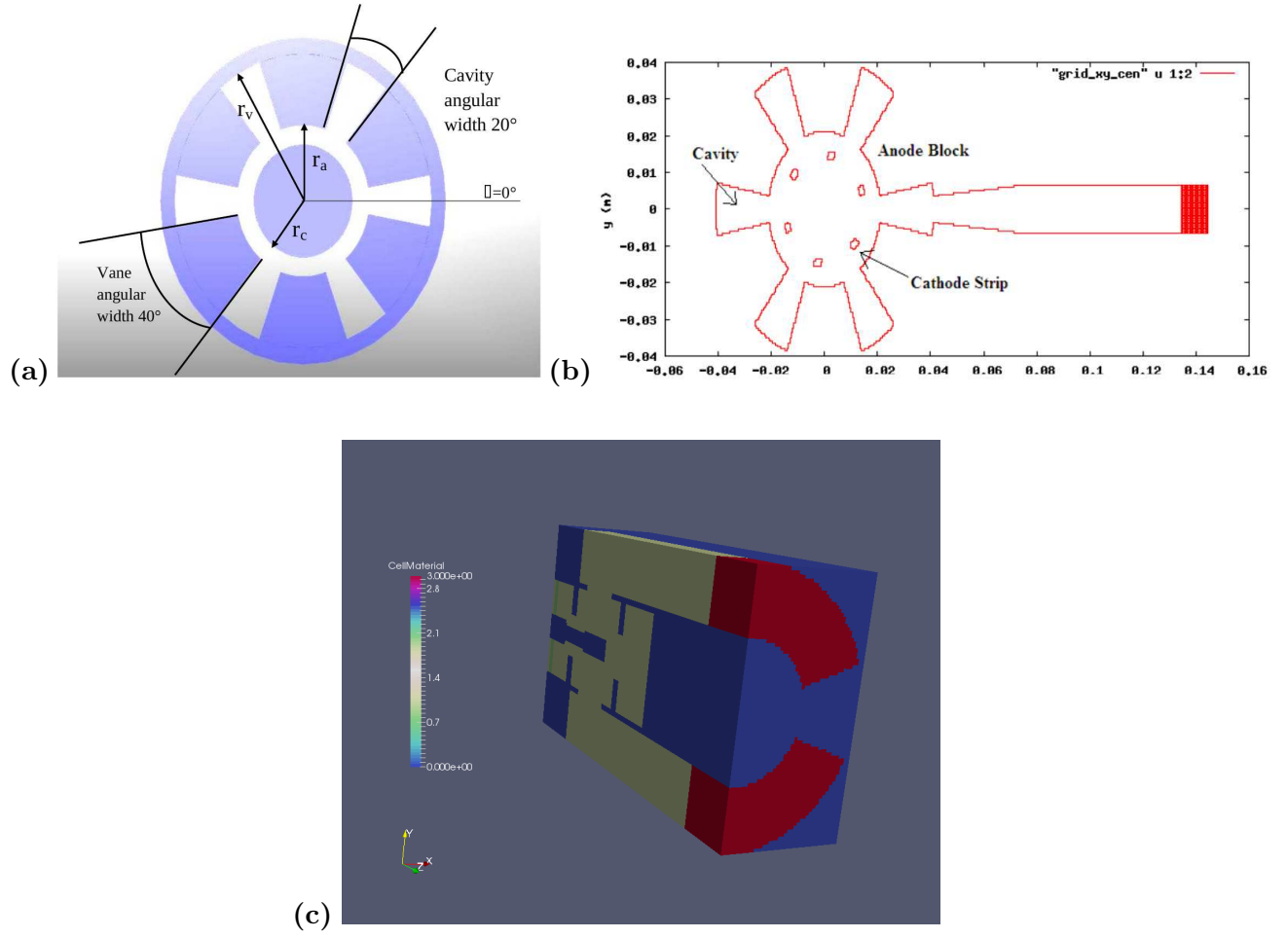


Figure 111: (a) Axial view of the A6 relativistic magnetron with a solid cathode. (b) Grid plot of A6 magnetron in ICEPIC. (c) Computational domain material composition plot, with a slice.

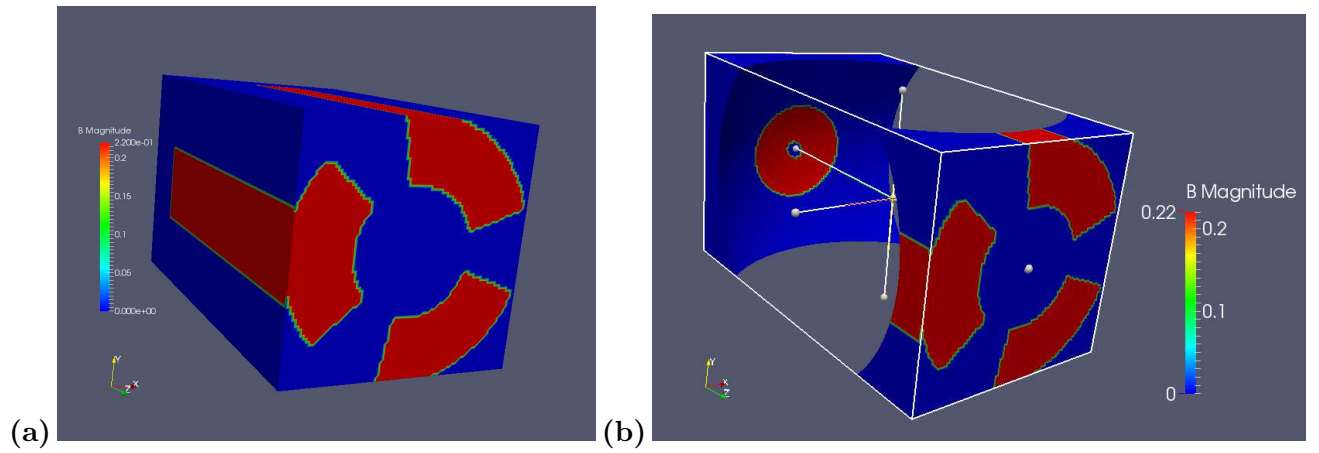


Figure 112: (a) Results of the ICEPIC simulations for the A6 relativistic magnetron, B-field magnitude. (b) Solution plot on the domain surface.

15 Practical ICEPIC applications: Vlasov antenna design

High power microwave (HPM) sources, such as the backward-wave oscillator (BWO), the gyrotron, and the vircator (virtual cathode oscillator), generate power in cylindrically symmetric transverse electric or transverse magnetic modes. The sidelobe generation, gain reduction, and inefficient power loading on the antenna aperture make these modes unsuitable for driving conventional antennas. This gave the idea of using mode converters at the output of these sources to convert these modes into a plane-parallel linearly polarized beam.

15.1 Vlasov antenna optimization

A Vlasov antenna [146], [147], [148] which is one of the most known mode converters used, is composed of a cylindrical waveguide with a shaped end, which can directly radiate energy from cylindrically symmetric modes in circular waveguides, without the need for an additional mode converter. The two well-known Vlasov antenna types come with a step cut and with a beveled cut [149], Fig. 113.

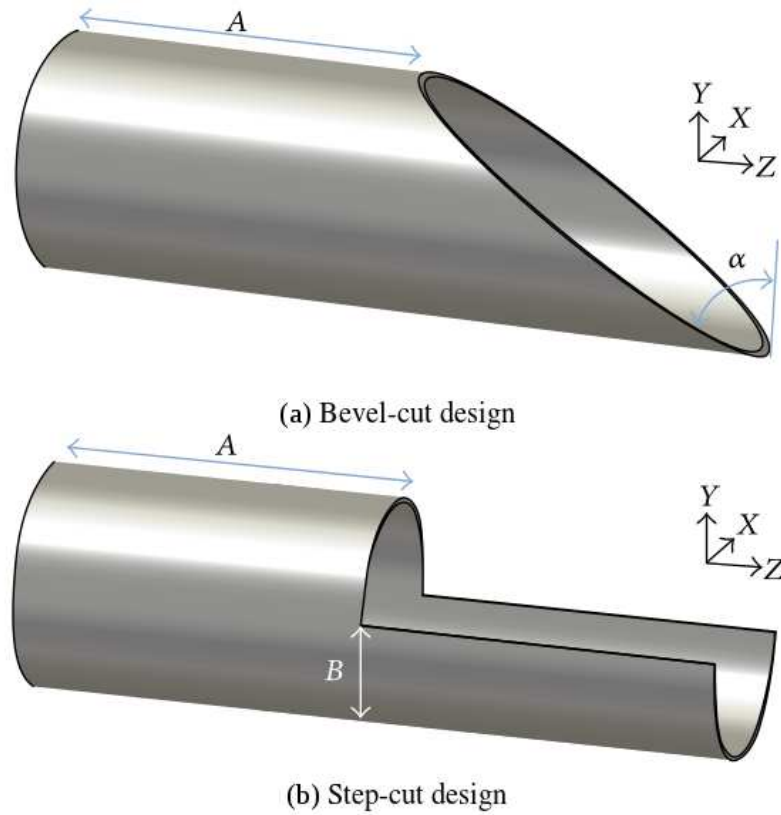


Figure 113: Configurations and parameters of bevel- and step-cut Vlasov antennas.

The first type, a waveguide aperture with a step cut, originally suggested by Vlasov, has sharp edges and therefore may suffer from electrical breakdown when radiating HPM [150]. The beveled cut was suggested by Wada and Nakajima [151] to avoid the sharp points of the step cut, leading to a more suitable shape for usage in HPM applications [152], Fig. 114.

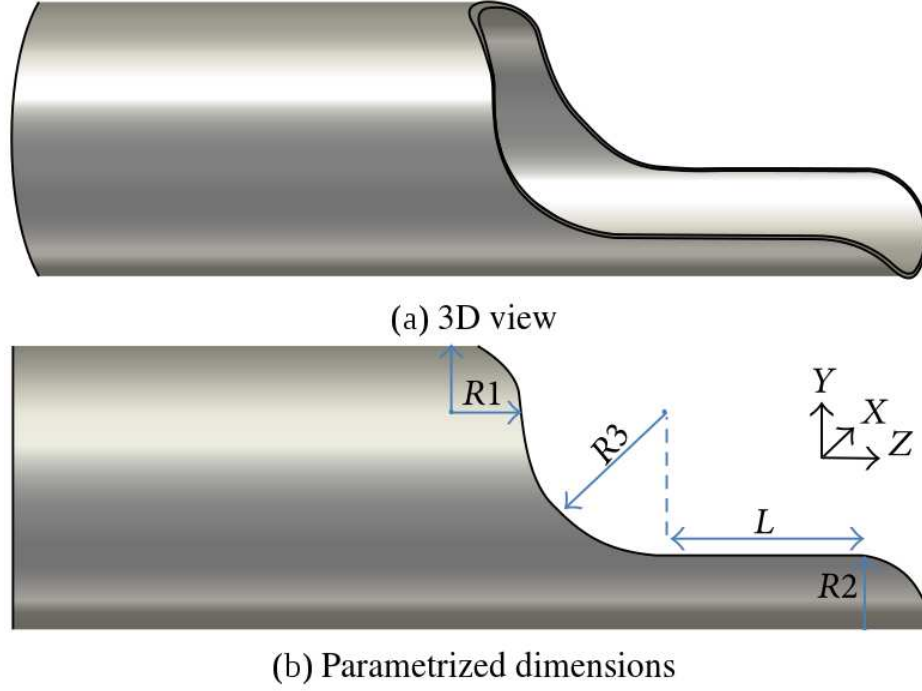


Figure 114: Vlasov antenna with proposed in [148] cut.

Using the bevel-cut antenna design in Fig. 114, the maximum radiation is obtained in the shifted direction corresponding to $\theta = 20$ deg and $\phi = 90$ deg, computed using HFSS code [148]. The added reflector, Fig. 115, can result in the maximum radiation is back along the axis of the waveguide; that is $\theta = 20$ deg, and $\phi = 90$ deg, when the length and the angle of reflector are optimized, Fig. 116.

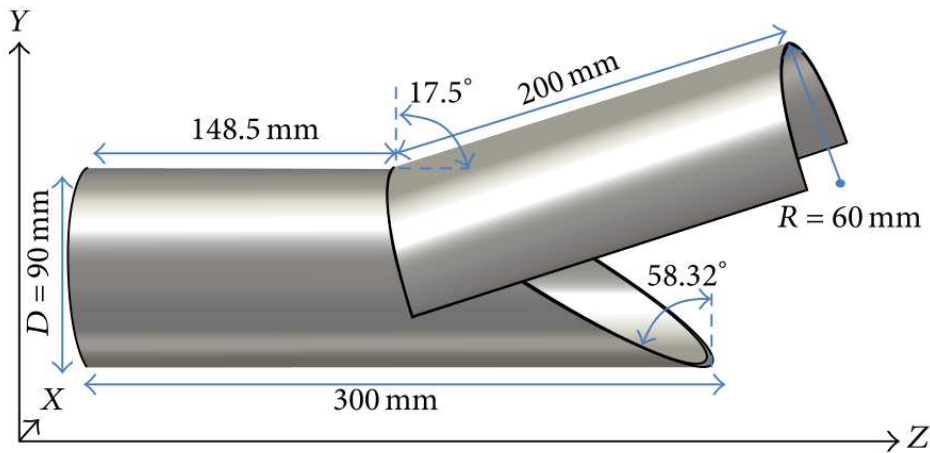


Figure 115: Bevel-cut Vlasov antenna with reflector.

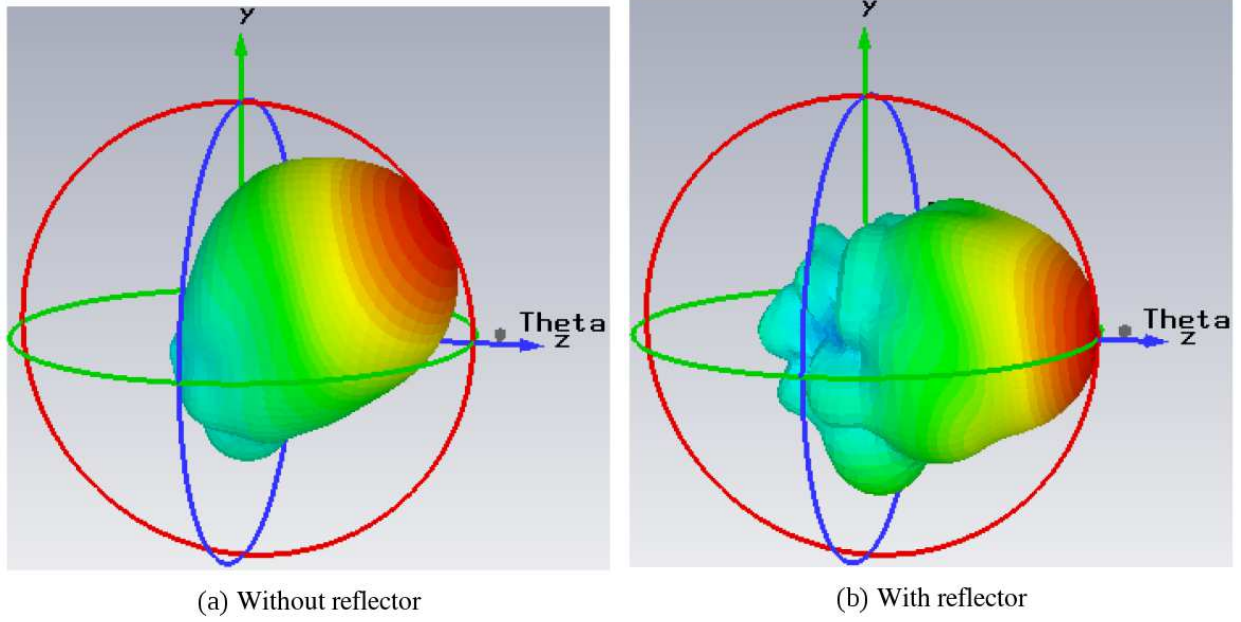


Figure 116: Bevel-cut 3D gain patterns: (a) without reflector and (b) with reflector.

15.2 Vlasov antenna simulation for Lacuna method demonstration

The beveled cut Vlasov antenna simulation with ICEPIC is a good example where lacuna method implementation can be tested. The geometry of the problem is shown in Fig. 117. Total mesh contains 257,000 cells and 271,000 points. PML boundary condition is set for each external face of the domain.

The simulation time is about 4mins, and the ICEPIC results are presented in Fig. 118 and Fig. 119. Stability of simulations and accuracy dependence on the PML parameters have been investigated.

We found out that the ICEPIC PML has been able to produce very stable results for a wide range of PML parameters. This PML implementation has many sophisticated features, that help to maintain the stability in the most cases.

Still we have found that a small value of PML depth can result in the exponentially growing instability for large simulation times ($t > 400$), in case of Vlasov antenna example, B-field in Figure 120, and E-field in Figure 121 (red curve). Applying the lacuna method results in a stable numerical solution for any large time (pink curve).

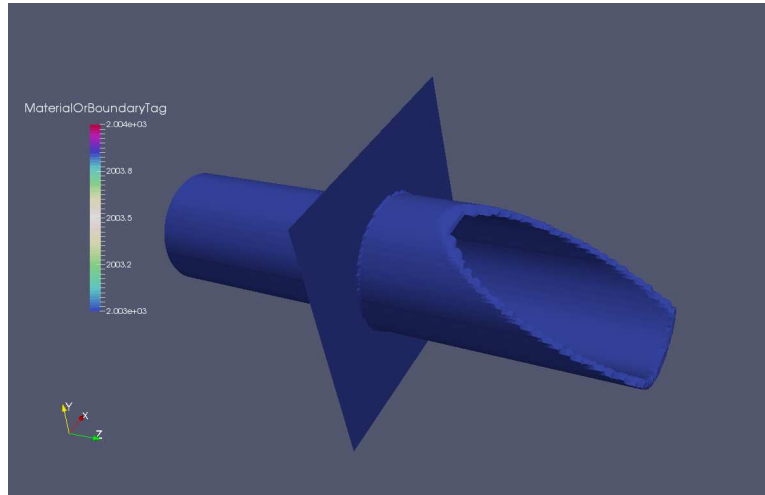


Figure 117: Bevel-cut geometry for simulation of Vlasov antenna with ICEPIC.

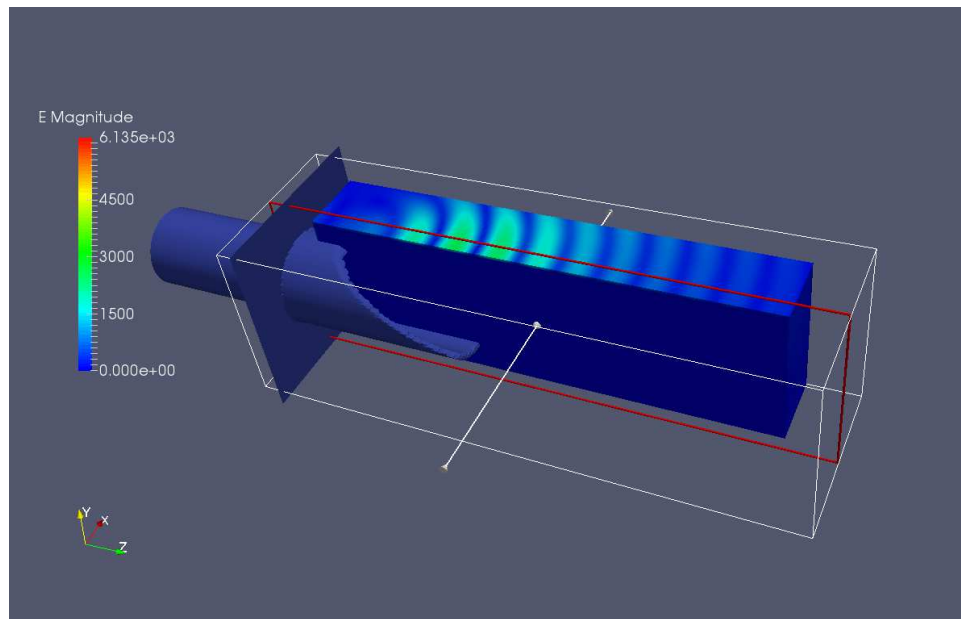


Figure 118: Bevel-cut 3D simulation of Vlasov antenna with ICEPIC. The E-field is shown on the top part of the computational domain.

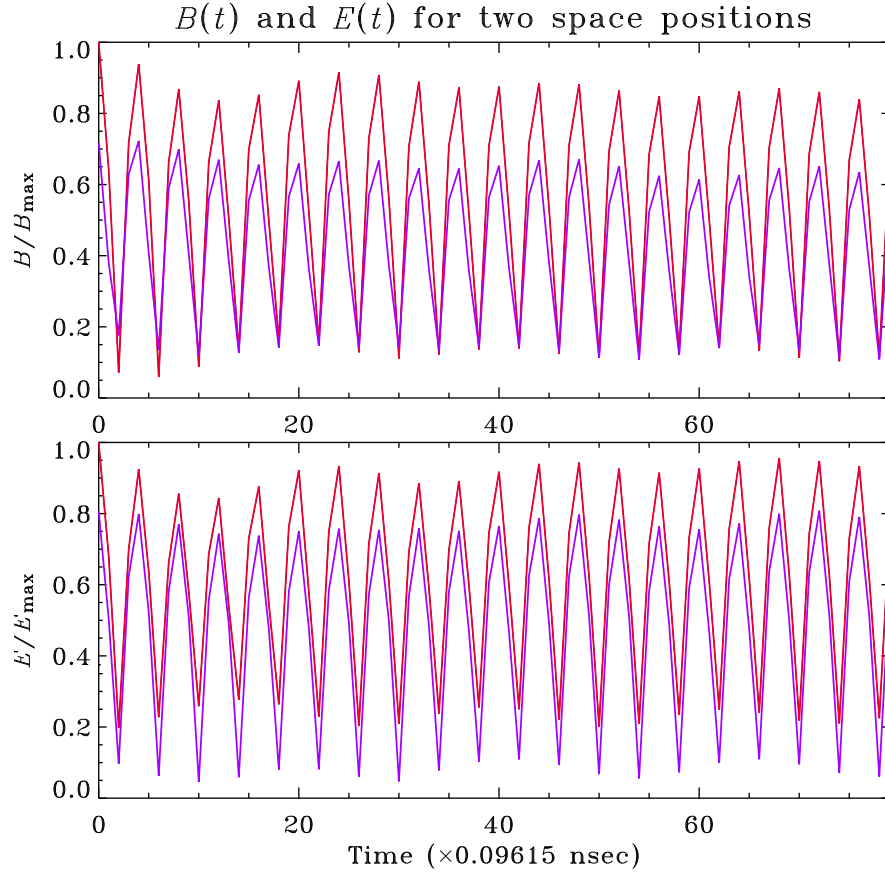


Figure 119: Vlasov antenna simulation with ICEPIC. The E- and B-field variations in time at points $[0,1,-0.03]$ (red curve) and $[0,1,0.06]$ (blue curve).

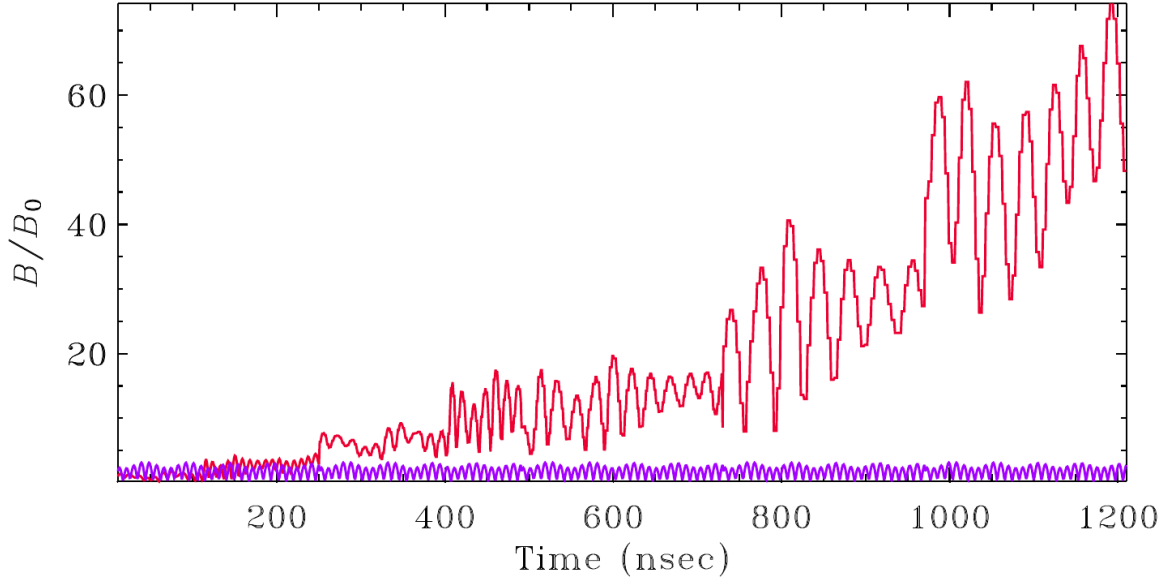


Figure 120: Vlasov antenna simulation with ICEPIC, PML depth=3 (red curve). The errors in B-field start to grow at time $t=100$, and exceed one order in magnitude at time $t=300$, continuing to grow exponentially. The lacuna method solution (pink curve) remains stable for any large time.

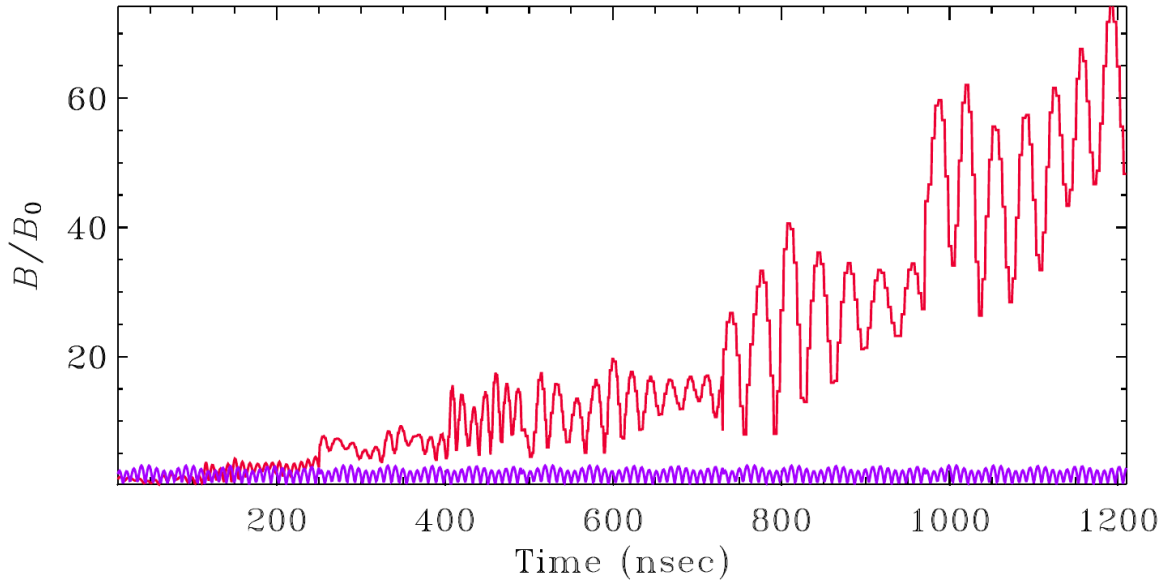


Figure 121: Vlasov antenna simulation with ICEPIC, PML depth=3 (red curve). The errors in B-field start to grow at time $t=100$, and exceed one order in magnitude at time $t=300$, continuing to grow exponentially. The lacuna method solution (pink curve) remains stable for any large time.

16 Lacuna BC Implementation in WARPX multi-fluid plasma code

The Lacuna BC Implementation in WARPX multi-fluid plasma code has been successfully completed. This Section describes recent development and details of the Lacuna BC implementation in the WARPX code.

Recent progress was focused on analyzing the quasi-spherical electromagnetic pulse and on developing benchmark simulation cases to test the lacunae-based boundary condition implementation in WARPX. Simulating open boundary conditions in three dimensions offers the greatest potential computational cost savings. Initializations that exercise this capability must be derived to satisfy the physical constraints described by the governing Maxwell's equations. Simulation results must be compared against a benchmark case that serves as the ideal standard.

16.1 Quasi-spherical Electromagnetic Pulse

As described in the July 2015 Progress Report, expressions have been derived for a quasi-spherical electromagnetic pulse that propagates in all three-dimensions. The divergence conditions are guaranteed by starting with the mixed potential formulation of Maxwell's equations. The electric and magnetic fields are defined using time-dependent potential functions

$$\mathbf{E} = -\nabla\phi - \frac{\partial\mathbf{A}}{\partial t} \quad (166)$$

and

$$\mathbf{B} = \nabla \times \mathbf{A}, \quad (167)$$

where ϕ and $\mathbf{A}$ are the retarded scalar and vector potentials. The Lorenz gauge allows Maxwell's equations to be written in a simplified form

$$\square\phi = \frac{\rho}{\varepsilon_0} \quad (168)$$

and

$$\square\mathbf{A} = \mu_0\mathbf{j}, \quad (169)$$

where $\square = \frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \nabla^2$ is the d'Alembert operator, which separates into component directions for Cartesian coordinate systems. A localized initial electromagnetic pulse is derived by assuming an isotropic vector potential that is peaked at the origin of the spherical pulse and decreases exponentially to zero away from the origin.

$$A_x = A_y = A_z = A_0 e^{-\frac{r}{R}}, \quad (170)$$

where A_0 is a scaling constant and R is the nominal radius of the initial pulse. The Cartesian spatial position is given by the radius,

$$r^2 = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2.$$

The pulse originates at coordinates (x_c, y_c, z_c) .

The magnetic field can be computed from Eq. (170) using the definition given by Eq. (167) to give

$$B_x = \frac{A_0 e^{-\frac{r}{R}}}{rR} (y - y_c - z + z_c), \quad (171)$$

$$B_y = \frac{A_0 e^{-\frac{r}{R}}}{rR} (z - z_c - x + x_c), \quad (172)$$

$$B_z = \frac{A_0 e^{-\frac{r}{R}}}{rR} (x - x_c - y + y_c). \quad (173)$$

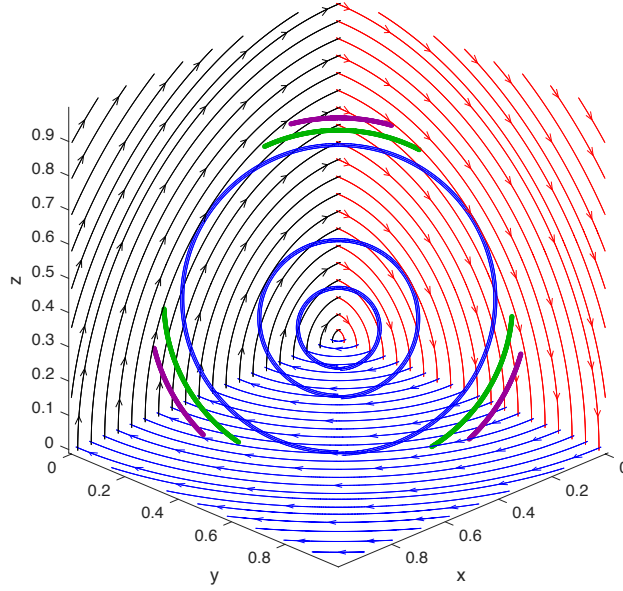


Figure 122: Streamlines of the initial magnetic field showing the vector projections of the magnetic field direction onto the principle planes (red, black, blue). The magnitude decreases from the origin. Also shown are sample 3D streamlines tracing the magnetic field. Blue streamlines are contained within the domain, but green and purple streamlines leave the domain at an oblique angle.

This initial condition for the magnetic field produces a quasi-spherical configuration that satisfies the zero-divergence condition. A visualization of the magnetic field is shown in Fig. 122.

The magnetic field given by Eqs. (171,172,173) were initialize in `WARPX` with the assumption that the problem had reflection symmetry in three directions and only one eighth of the problem domain needed to be simulated. This assumption has been determined to be incorrect. The magnetic field lines of the initialization are traced as streamlines in Fig. 122. Some magnetic field lines (shown in blue) are contained within the domain. However, some magnetic field lines (e.g. those shown in green and purple) leave the domain at an oblique angle, and do not satisfy the assumed reflection symmetry. The magnetic topology has the desired localization and physical consistency with Maxwell's equations, but it has a principal axis along the (1,1,1) vector. This problem requires a complete 3D Cartesian domain for a proper simulation.

The magnetic topology can be rotated to produce the desired alignment, for example to align the principal axis along the x axis. The rotation is accomplished by two elemental rotations: a rotation of $-\pi/4$ about the x axis followed by a rotation of $-\sin^{-1}(\sqrt{2}/3)$ about the z axis. The

resulting magnetic field is given as

$$B_x = 0, \quad (174)$$

$$B_y = \frac{A_1 e^{-\frac{r}{R}}}{rR} (z - z_c), \quad (175)$$

$$B_z = \frac{A_1 e^{-\frac{r}{R}}}{rR} (-y + y_c), \quad (176)$$

where A_1 is a scaling constant equal to $\sqrt{3}A_0$. This form of the magnetic field has the desired reflection symmetry in three directions aligned with the Cartesian coordinates, which has been verified by differentiating the magnetic field across the reflection planes.

Previous simulation results initialized with the magnetic field given by Eqs. (171,172,173) are unreliable since the boundary condition along the symmetry planes produced magnetic field values that were inconsistent with the initial conditions. Simulations need to be performed using the rotated magnetic field given by Eqs. (174,175,176).

16.2 Benchmark Case

Benchmark cases provide solutions against which simulation results can be compared to give a quantitative measure of effectiveness and accuracy. Simulations will be performed on truncated domains to test the implementation of the lacunae-based boundary condition in **WARPX**. A benchmark case has been developed that simulates the quasi-spherical electromagnetic pulse on a large domain. The simulation results are presented in Fig. 123.

The evolution of a mostly spherical wavefront can be seen in Fig. 123. The reflection symmetry across the three planes is evident.

The domain size is $3 \times 3 \times 3$, while previous calculations used an interior domain size of $1 \times 1 \times 1$. The behavior of the solution until the wavefront leaves the $1 \times 1 \times 1$ portion of the domain provides the comparison data for future simulations to test the implementation of open boundary conditions.

Future work will include testing the lacunae-based boundary condition implementations in **WARPX**. The simulations will be initialized using the rotated magnetic field given by Eqs. (174,175,176).

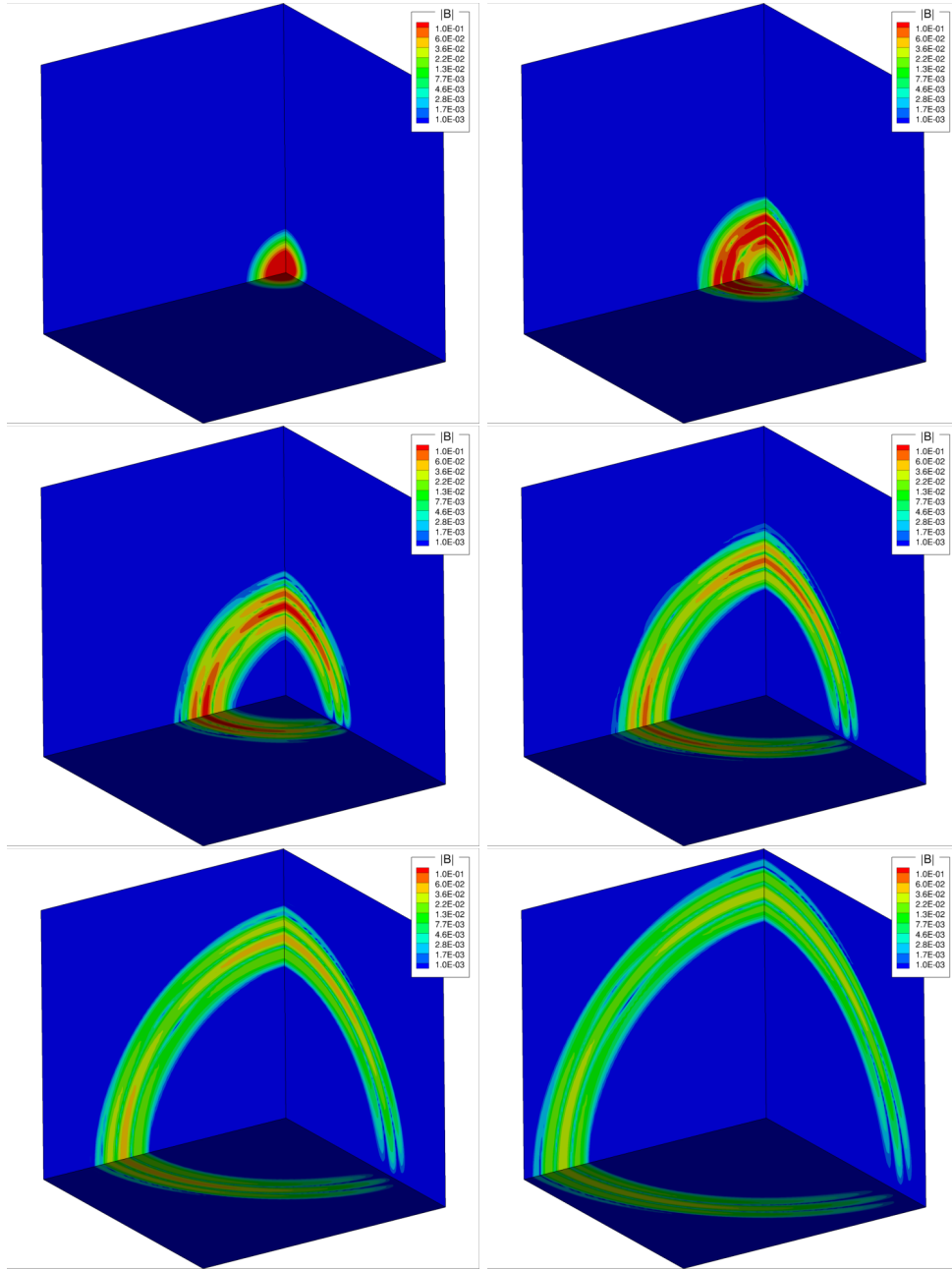


Figure 123: Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$.

Recent progress has focused on developing simulations to evaluate the performance of the lacunae-based boundary condition implementation in **WARPX** using the rotated quasi-spherical electromagnetic pulse that was reported in the December 2015 Progress Report. Benchmark data has been generated for the simulation on an extended domain that effectively removes any boundary condition interaction with the propagating pulse. Several simulations have been performed using the same initial conditions on a truncated domain surrounded by lacunae-based auxiliary boundary domains.

16.3 Rotated Quasi-spherical Electromagnetic Pulse

The derivation of a rotated quasi-spherical electromagnetic pulse was presented in the December 2015 Progress Report. The initial conditions for a pulse originating at coordinates (x_c, y_c, z_c) and aligned with the x axis are given by the magnetic field

$$B_x = 0, \quad (177)$$

$$B_y = \frac{A_1 e^{-\frac{r}{R}}}{rR} (z - z_c), \quad (178)$$

$$B_z = \frac{A_1 e^{-\frac{r}{R}}}{rR} (-y + y_c), \quad (179)$$

where A_1 is a scaling constant and the Cartesian spatial position is given by the radius,

$$r^2 = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2.$$

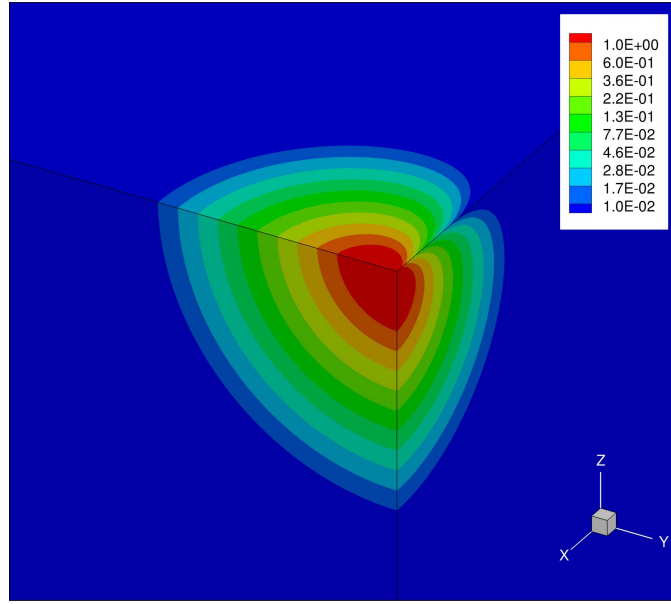
This initial condition for the magnetic field produces a quasi-spherical configuration that satisfies the zero-divergence condition. Furthermore, the magnetic field topology possess reflection symmetry in the three directions aligned with the Cartesian coordinates, which reduces the required computational domain to an octant. The initial conditions are shown in Fig. 124, representing the standard and compact spatial extent of the pulse.

16.4 New Benchmark Case

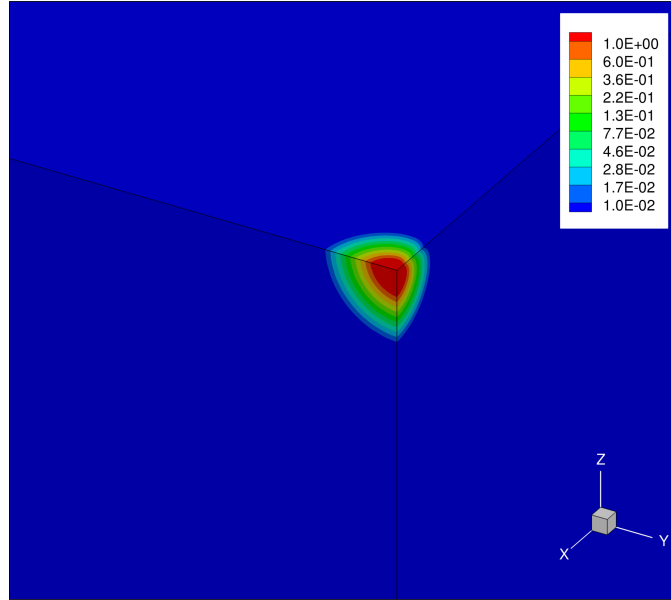
New benchmark cases have been generated that provide solutions against which simulation results can be compared to give a quantitative measure of effectiveness and accuracy. Simulations presented in Sec. 16.5 are performed on truncated domains to test the implementation of the lacunae-based boundary condition in **WARPX**. The benchmark cases simulate the evolution of the initial conditions for the x -aligned quasi-spherical electromagnetic pulse, shown in Fig. 124. However, the domain is extended to $3 \times 3 \times 3$, such that the pulse does not substantially interact with the boundary during the simulation period. The truncated boundary simulations have a domain of $1 \times 1 \times 1$. The simulation results are presented in Fig. 125 for the standard spatial extent of the pulse, which corresponds to Fig. 124(a), and in Fig. 126 for the compact spatial extent of the pulse, which corresponds to Fig. 124(b).

16.5 Simulations with Lacunae-based Boundary Conditions

As described in previous Progress Reports, the implementation of the lacunae-based boundary condition appends auxiliary domains to a truncated interior domain. The solution in the auxiliary domain is periodically re-integrated to damp the leading edge of the solution before it reflects from the exterior boundary of the auxiliary domain and re-enters the interior domain. This section



(a)



(b)

Figure 124: WARPX initialization of x -aligned quasi-spherical electromagnetic pulse showing logarithmic contours of the magnetic field magnitude. The magnetic field has B_y and B_z components and $B_x = 0$ as given by Eq. (177). Shown are the two cases representing the (a) standard and (b) compact spatial extents of the pulse. The computational domain sizes are identical.

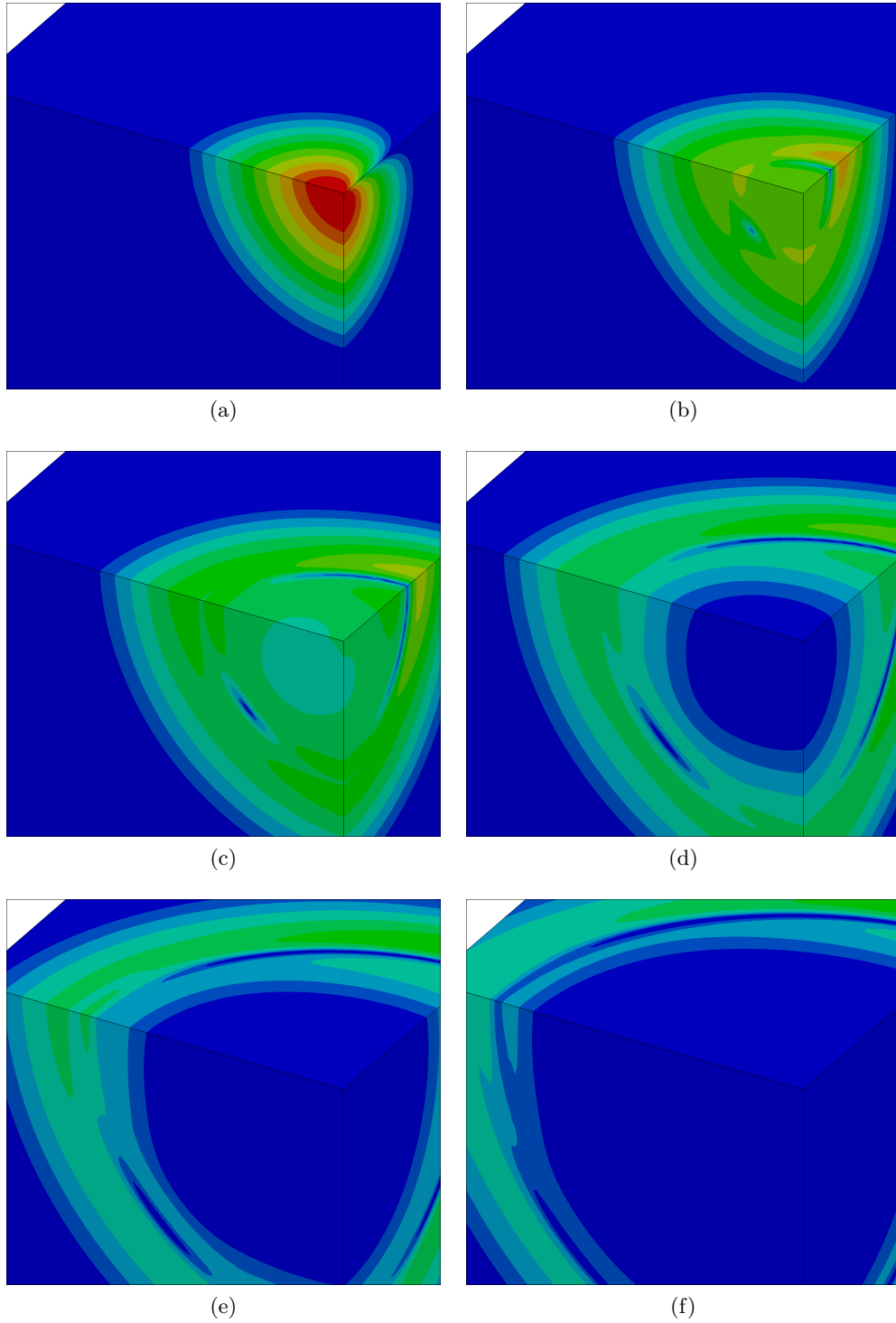


Figure 125: Simulation results from `WARPX` for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$.

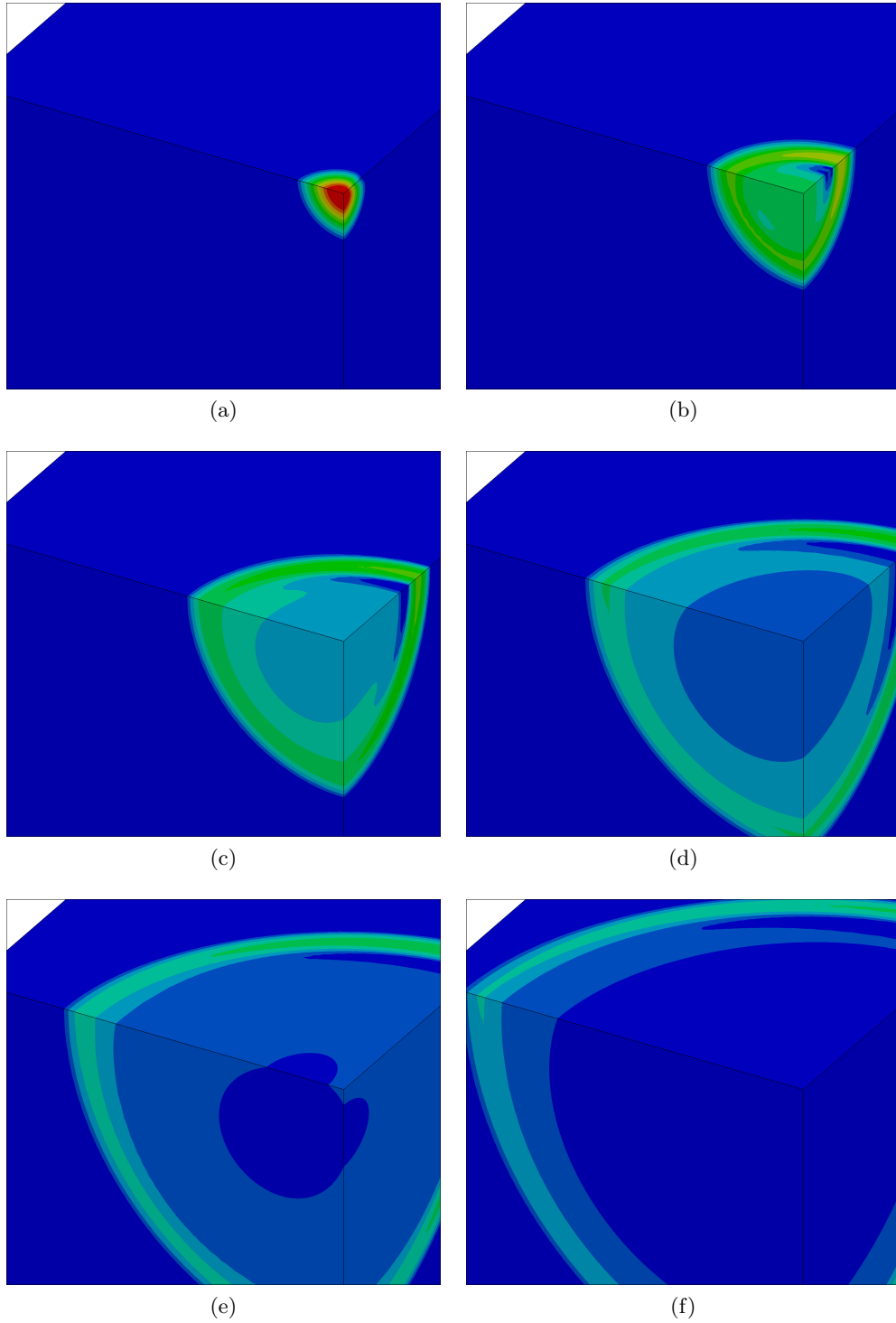


Figure 126: Simulation results from `WARPX` for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$.

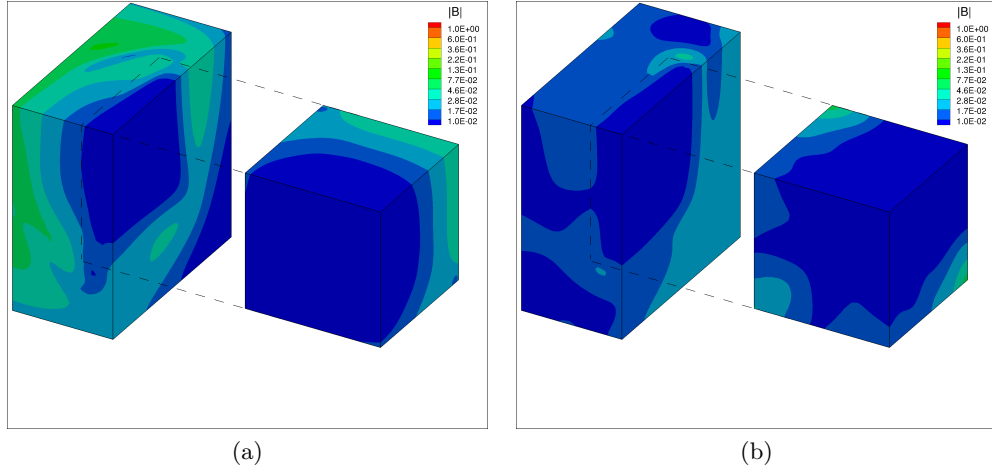


Figure 127: Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. No re-integration is performed for these simulations. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 125(d) and 125(f), which are plotted at the same scale.

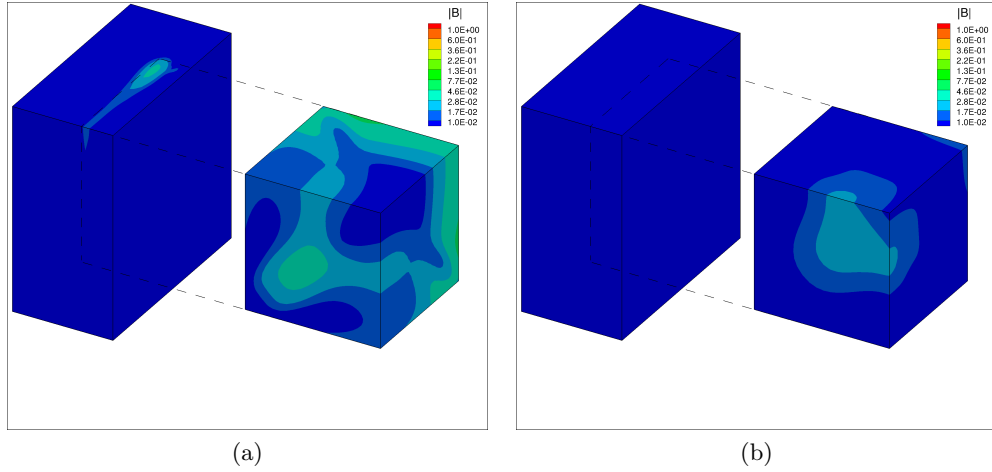


Figure 128: Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. Frequent re-integration is performed for these simulations every 0.05 time. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 125(d) and 125(f), which are plotted at the same scale.

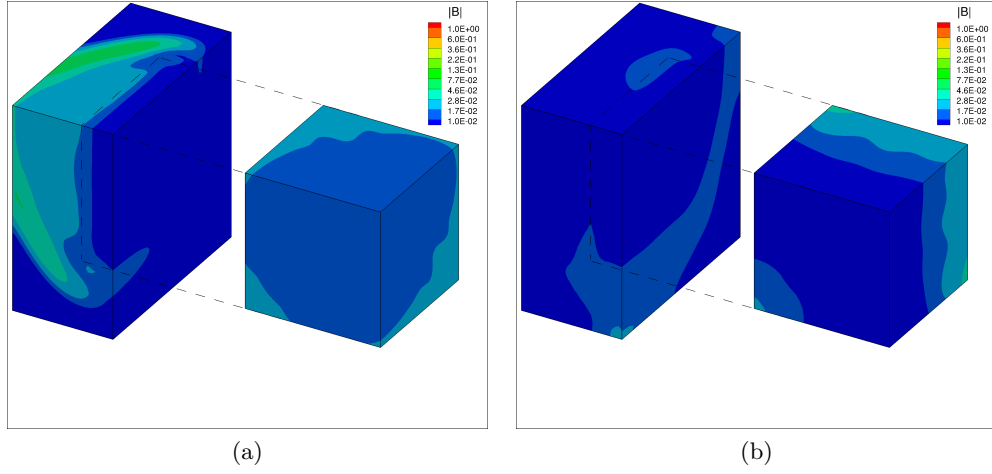


Figure 129: Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. No re-integration is performed for these simulations. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 126(d) and 126(f), which are plotted at the same scale.

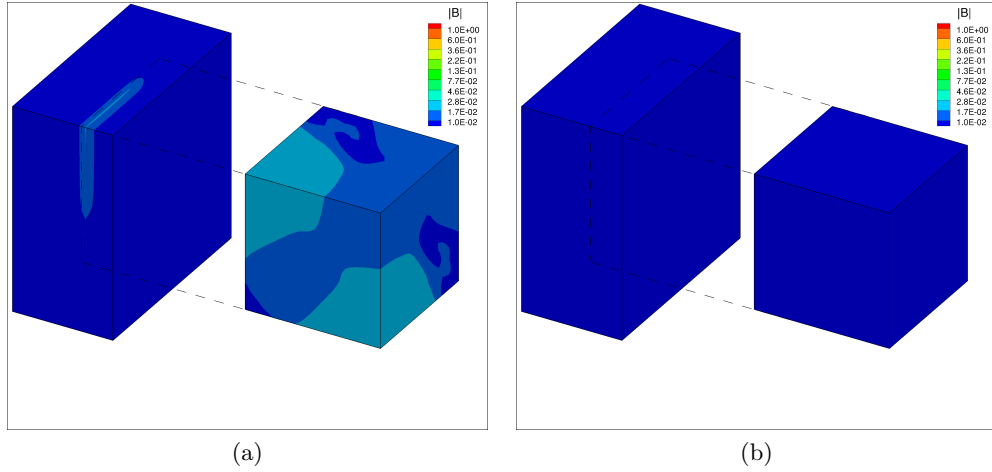


Figure 130: Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. Frequent re-integration is performed for these simulations every 0.05 time. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 126(d) and 126(f), which are plotted at the same scale.

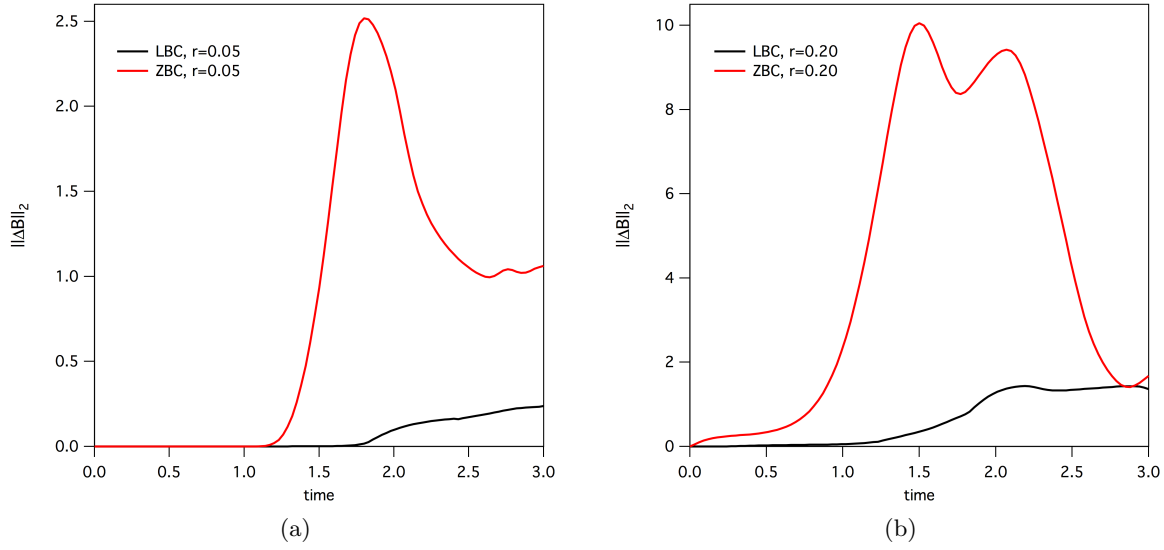


Figure 131: Temporal evolution of the error caused by the boundary condition implementation. Error is computed as the 2-norm in reference to a simulation with an extended domain such that the solution does not reach the boundaries during the simulation time. The errors are shown for the lacuna-based boundary condition implementation (LBC) and a zero gradient boundary condition implementation (ZBC). The lacuna-based boundary condition clearly performs better for both pulse initializations, the standard spatial extent Fig. 131(b) and the compact spatial extent Fig. 131(a).

presents simulation results using the truncated interior domain for the standard and compact initial conditions illustrating the effects of re-integration.

The compact initial condition demonstrates the largest benefit from the lacunae-based boundary conditions. With re-integration activated, the absence of a solution at $t = 2.5$ appears to match the benchmark case – comparing Fig. 130(b) to Fig. 126(f). All wavelength components produced from the compact initial conditions are able to fit into the auxiliary domain and are effectively damped. The standard initial condition produces wavelengths that are larger than the auxiliary domain.

A quantitative analysis is presented in Figures 131(a), 131(b), shows the evolution of the error caused by the boundary condition implementation.

Error is computed as the 2-norm in reference to a simulation with an extended domain such that the solution does not reach the boundaries during the simulation time. The errors are shown for the lacuna-based boundary condition implementation (LBC) and a zero gradient boundary condition implementation (ZBC). The lacuna-based boundary condition clearly performs better for both pulse initializations, the standard spatial extent, Figure 131(b) and the compact spatial extent Figure 131(a). Figure 126 from Section 16.5 illustrates the two spatial extents.

16.6 Porting and Running WARPX on ComScis Computer

Instructions for requesting access to the WARPX Git repository and for downloading and compiling required software are provided on the Wiki webpage maintained by the University of Washington at <http://sites.google.com/site/warpxwiki/>. The detailed instructions are not repeated here (see Appendix C for that), but a brief outline is provided with specific instructions to demonstrate the

lacunae-based boundary condition feature.

If not already installed on the target computer system, the OpenMPI and HDF5 (serial and parallel builds) libraries must be downloaded, built, and installed. In addition, the Scons utility must be installed. The WARPX should then be downloaded from the Git repository using the command

```
git clone git@warpx.sourcerepo.com:warpx/WARPX.git warpx
```

Building WARPX is accomplished within the src directory using the commands

```
cd warpx/src
scons parallel=no debug=no
```

The finite element discontinuous Galerkin version of the code that contains the lacunae-based boundary condition implementation is housed in the mixedDgFem branch. Checkout and switch to this branch using the command

```
git checkout mixedDgFem
```

Rebuild the code using the same scons command.

To run WARPX, a run directory is created, and the input file is created or copied into this directory. Before the WARPX executable is invoked, the input file must be preprocessed using a python script from the scripts directory included in the WARPX Git repository.

```
warpx/scripts/wxinpparse.py -i maxwellPulse abc3D 95.pin
```

This procedure generates a WARPX-readable input file that is then run using the command

```
warpx/src/build/xwarpx/xwarpx -i maxwellPulse abc3D 95.inp
```

Diagnostic information is printed to the terminal as the code runs. The amount of diagnostic information is controlled by the Verbosity variable in the input file. Output HDF5 files are generated as the code progresses. Python script files provide foundational utilities to read, post-process, and display the output files.

This particular input file maxwellPulse abc3D 95.pin simulates the rotated quasi-spherical electromagnetic pulse similar to the problem described in the Section [16.3](#).

The initialization uses the compact spatial extent of the pulse. The simulation uses a high-resolution mesh with a re-integration period of 0.5 time.

17 Applications of the developed tools: Modeling of electromagnetic scattering from an underground embedded body (IED in soil)

This important problem has been proposed for consideration to the project team by Traian Dogaru, Adelphi, MD U.S. Army Research Laboratory (traian.v.dogaru.civ@mail.mil). His team has conducted advanced research on Sensing-Through-the-Wall (STTW) radar technology for imaging complex building environments, static and moving human signature, see e.g. [158], [159], [160].

We greatly appreciate this idea, and the following sections present our initial results on formulation the problem and modeling of electromagnetic scattering from an underground embedded body (to identify IED in soil).

IED detection is an important application area for microwave radars

Terrorist groups, acting either in isolation or sponsored by one or more nation-states - and backed by a substantial infrastructure and support system - have become skilled practitioners of asymmetric warfare - and their weapon of choice remains the improvised explosive device (IED) [139].

"This year, nearly 1,900 US casualties have been caused by IEDs," Lt Gen Michael Barbero, head of the Joint IED Defeat Organisation (JIEDDO), told a US Senate subcommittee in December 2012. "I believe the IED will continue to be the weapon of choice against our forces and we must remain vigilant."

Desperate to stay one step ahead of the roadside bombers, US Department of Defence (DoD) experts are now experimenting with ultra-wideband (UWB) microwave radar, a leading-edge medical diagnostic technology developed to locate malignant tumours.

Ultra-wideband (UWB) microwave technology

Research into landmine detection using UWB radar took place in the UK during the Falklands conflict in the early 1980s and in the US in the aftermath of the Gulf War a decade later. Military research programmes in Germany, France and Russia followed, but it is only now, in the wake of pioneering research into the feasibility of UWB microwave imaging for early-stage breast cancer detection, that the subject has started to gain serious traction on both sides of the Atlantic.

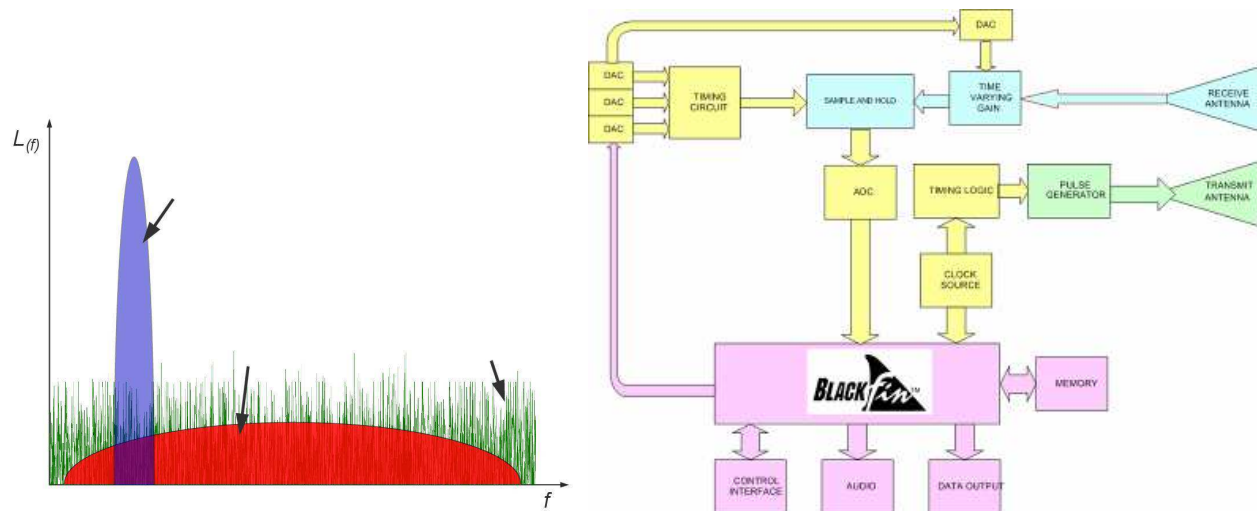


Figure 132: (a) The UWB-radar transmits in a wide spectrum with very low power across a much wider frequency than conventional radar systems and are usually very difficult to detect; (b) Minehound radar module system diagram [140].

Ultra Wideband (UWB) radar systems transmit signals across a much wider frequency than conventional radar systems and are usually very difficult to detect. The transmitted signal is significant for its very light power spectrum, which is lower than the allowed unintentional radiated emissions for electronics. The most common technique for generating a UWB signal is to transmit pulses with very short durations (less than 1 nanosecond).

The spectrum of a very narrow-width pulse has a very large frequency spectrum approaching that of white noise as the pulse becomes narrower and narrower. These very short pulses need a wider receiver bandwidth as conventional radar systems.

The amount of spectrum occupied by a signal transmitted by a UWB-radar (i.e. the bandwidth of the UWB signal) is at least 25% of the center frequency. Thus, a UWB signal centered at 2 GHz would have a minimum bandwidth of 500 MHz and the minimum bandwidth of a UWB signal centered at 4 GHz would be 1 GHz. Often the absolute bandwidth is larger than 1 GHz.

An example of a typical UWB radar is the pulsed noise radar. This radar is transmitting at the center frequency of 24 GHz single pulses with a duration of a few nanoseconds, and a pulse power of 4 mW. Its bandwidth is therefore 8 GHz. Whether this radar is transmitting or not can not be determined by measurement from the outside of the radar as its transmit pulses do not differ from environmental noise [141].

UWB microwave imaging uses the measurement of the transmission of microwave energy through an object to define its dielectric (insulating) properties. Put simply, by subjecting tissue to signals from across the radio spectrum at a much wider range than traditional X-rays, scientists at the University of Wisconsin-Madison in the US have been able to contrast normal fatty breast tissue with malignant tissue. This not only makes it possible to identify extremely small malignant tumours, but also to differentiate between benign and malignant growths [142].

Military chiefs hope to adapt the technology to detect IEDs in the field. A recent declassified Nato report on research into hand-held, stand-off vehicle-based and airborne detection systems outlined some of the current and future engineering challenges affecting IED detection, recognition and identification using UWB microwaves [143].

"UWB radar has emerged as the technology of choice for improved landmine detection,"

stated the report by UK specialist technical consultancy ERA Technology. "Recent developments using hand-held dual sensor technology combining electromagnetic induction EMI and ground-penetrating radar (GPR) have enabled improved discrimination against metal fragments to be demonstrated in live minefields [140].



Figure 133: (a) Remote-control "Panama" Land Rover with ground-penetrating radar to detect IEDs followed by Mastiff with Choker mine rollers [144]; (b) Although a number of vehicle based GPR systems have been trialled and reported on over the last 5 years the most extensively reported system has been the NIITEK radar shown [140]

Vehicle-based systems have been developed that use arrays of antennas and generate 3D data, which is then processed to provide a rolling map of detections.

However, the report also noted that impediments such as rocks and other debris significantly affected the performance of UWB systems and that handheld UWB detectors, such as the US Army's Minehound scanner, were limited thanks to a single pair of antennae.

"The ground contains significant levels of radar scatterers such as stones, animal burrows as well as man-made detritus and, consequently, radar encounters extremely high levels of clutter at short ranges," the report noted. "Overcoming this as well as achieving an adequate signal / noise ratio is a major technical challenge."

Countermeasures and the future of IED detection

Since its inception in 2006, JIEDDO has spent in excess of \$6bn developing IED countermeasures, while the US military has experimented with everything from lasers and mine-detection robots to chemical scent detectors and man-portable radio-frequency jammers, all aimed at mitigating the evolving IED threat to US warfighters in-theatre. UWB radar is the latest weapon in the fight against an ever-resourceful enemy.

In light of this assessment, research into countermeasures such as UWB radar remains a key weapon in the fight to protect warfighters from the very real threat posed by IEDs and hidden ordnance.

The proposed in this project innovative modeling and enhanced accuracy simulation tools for electromagnetic waves in complicated environment will help the the Army to:

1. Assess the technologies and devices for IED detection in the ground;
2. Better evaluate the performance at the early design stages;

3. Set the requirements for testing;
4. Reduce the cost and time of testing.

The modeling and design tools developed will provide a reduction in the cost and time-to-market through reduced experimental R&D, design cycle, and streamlined laboratory testing.

This Section describes the computational model, and analysis of the results from simulating an electromagnetic wave scattering from a body embedded in a dielectric using Overture's `Cgmx` time-domain solver for Maxwell's equations [79].

17.1 Problem specification

The following material is are based on notes from Prof. Semyon Tsynkov.

We consider a *plane wave* in the form of a *linear chirp* that impinges upon a body embedded in a horizontal slab of a dielectric. If the material surface of the dielectric is the plane $z = 0$ and the angle of incidence from the z -axis is θ then the incident field is of the form

$$\begin{aligned}\mathbf{E}_0(x, t) &= \hat{\mathbf{a}} P(t + \frac{x}{c} \sin(\theta) + \frac{z}{c} \cos(\theta)) = \mathbf{a} P(t - \frac{\hat{\mathbf{k}}}{c} \cdot \mathbf{x}), \\ \hat{\mathbf{a}} &= [-\cos(\theta), 0, \sin(\theta)], \\ \hat{\mathbf{k}} &= [-\sin(\theta), 0, -\cos(\theta)]\end{aligned}$$

where c is the speed of light in a vacuum and note that $\hat{\mathbf{a}} \cdot \hat{\mathbf{k}} = (\nabla \cdot \mathbf{E}_0 = 0)$.

A linear *chirp* of unit amplitude and duration τ is the waveform

$$\begin{aligned}P(t) &= \chi(t, \tau) e^{-i\phi(t)}, \\ \phi(t) &= \omega_0 t + \alpha t^2,\end{aligned}$$

where the *indicator* function $\chi(t, \tau)$ is

$$\chi(t) = \begin{cases} 1 & t \in [-\tau/2, \tau/2] \\ 0 & \text{otherwise } t \in [-\tau/2, \tau/2] \end{cases}$$

and where

$$\begin{aligned}\omega_0 &: \text{carrier frequency,} \\ \alpha = \frac{B}{2\tau} &: \text{chirp rate,} \\ B &: \text{band width.}\end{aligned}$$

The instantaneous frequency is

$$\phi'(t) = \omega_0 + 2\alpha t = \omega_0 + \frac{B}{\tau} t \quad \text{for } t \in [-\tau/2, \tau/2],$$

and varies linearly along the chirp (hence a em linear chirp).

Many radars use a narrow band chirp, e.g. $B/2\pi = 10\text{MHz}$. We will consider the parameters

$$\begin{aligned}\frac{\omega_0}{2\pi} &= 500\text{MHz} - 3\text{GHz}, \\ \frac{B}{2\pi} &= 10\text{MHz}, \\ \tau &= 5 \times 10^{-5}\text{s}.\end{aligned}$$

17.2 Simulation Time

The simulation time required to resolve the chirp wave may be prohibitive. The ranges for the carrier frequency is quoted to be, $\frac{\omega_0}{2\pi} = 500\text{MHz} - 3\text{GHz}$. Assume a fixed frequency and take the max value $\frac{\omega_0}{2\pi} = 3\text{GHz}$ and chirp duration $\tau = 5 \times 10^{-5}\text{s}$, the number of cycles computed is $\omega\tau = 1.5 \times 10^5$ cycles. Taking a reasonable value of 10 points per wavelength, then we need about 10^6 time steps. Based on CFL constraints, it is reasonable to assume $\Delta t \approx \Delta x$. For a refined grid $\Delta x \approx 10^{-2}$ so that the final integration time is $T = N\Delta t \approx 10^4$. Currently, wall-clock time for a simulation with final time $T = 20$ is about 6 minutes. Then a full simulation would take approximately 2 days on a single processor. Note that this is an upper estimate of the time to resolve the chirp wave.

17.3 One-way Wave

This problem poses a few challenges. Firstly, to minimize the effect of boundary-scattered waves from imperfect far-field boundary conditions, we want to implement a one-way planewave source that incorporates a linear chirped wave frequency. To accomplish this, we stagger two source terms in the scalar wave equation which are spatially and temporally offset so that waves travelling in the desired direction interfere constructively and waves travelling the opposite direction interfere destructively. Figure 135 shows the results for this idea in 1D. The sources used for this example represent point charges with a regularized Dirac delta function. To accomplish the same idea in 2D with an arbitrary angle of incidence will require a source term that does not rely on being located at a particular grid point since an arbitrary line will not in general pass through a grid point. For this case, a better spatial distribution will likely be a sharp Gaussian.

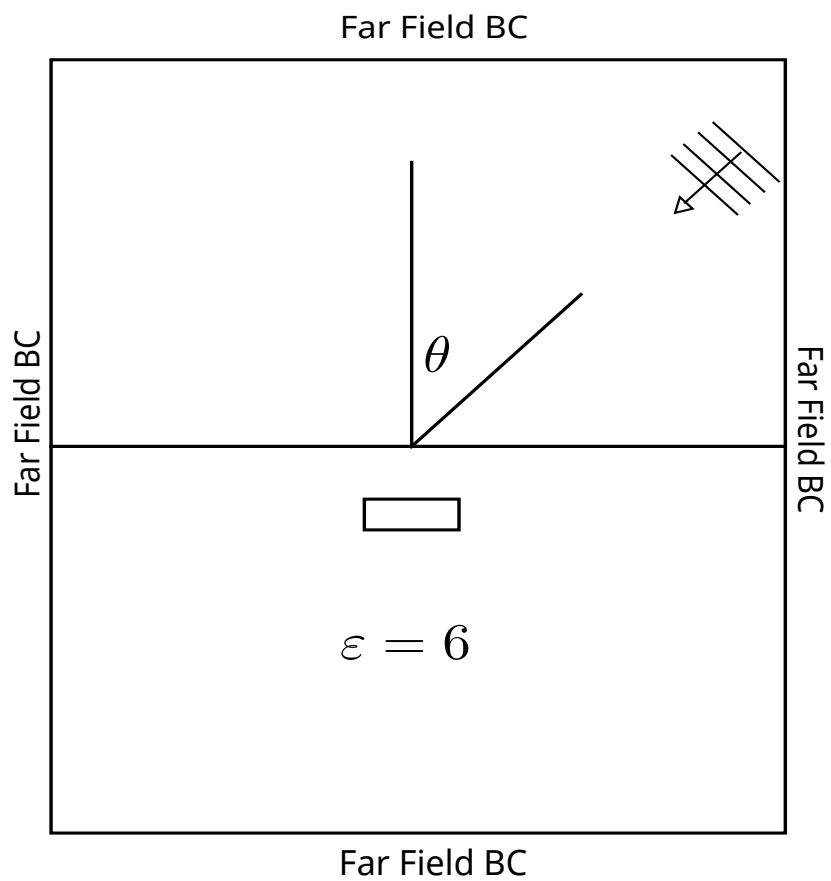


Figure 134: Planewave incident on interface at angle θ at $z = 0$.

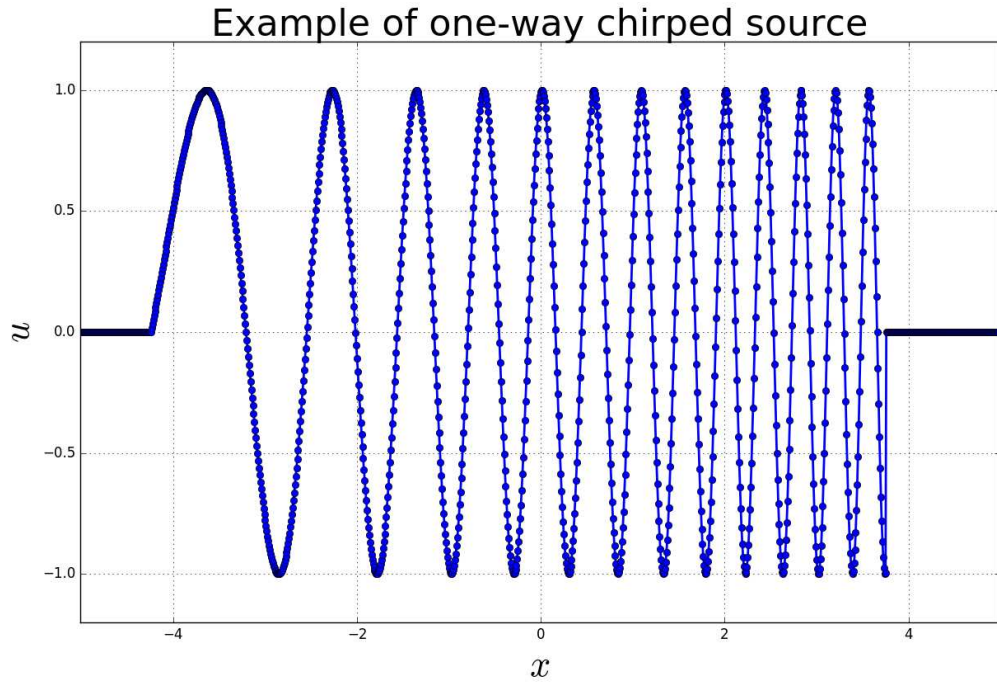


Figure 135: Result of numerically solving 1D scalar wave-equation $u_{tt} = c^2 u_{xx} + F(x, t)$ where $F(x, t)$ is chosen to produce one-way chirped planewaves. One point source is located at $x = 3.75$ and a second point source is placed one grid cell in front at $x = 3.75 + \Delta x$.

18 Results for electromagnetic scattering from a buried body

This section will contain progress and results for the 2D case. In two space dimensions we consider a rectangular body of width $W = 1$ and height $H = 0.5$ with top surface a distance $D = 1/3$ from the surface $y = 0$ as shown in Fig. 136.

A grid for this domain is shown in Figure 136 and was generated with Ogen using the command file `embeddedBodyGrid.cmd`. Finer grids are placed near the embedded body to more fully resolve the solution there while coarser grids are used in the far field to keep down artificial reflections from the far field boundaries.

Let $\mathcal{G}^{(j)}$ denote the grid with resolution factor j which has a grid spacing of approximately $\Delta s = 1/(10j)$ near the body.

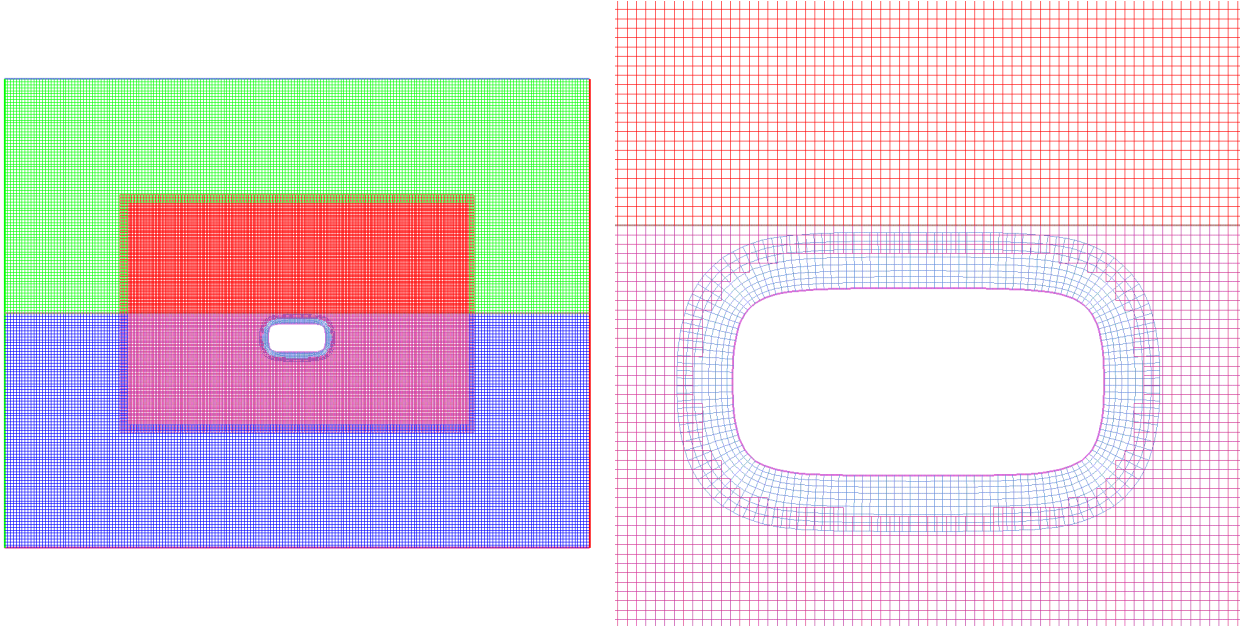


Figure 136: Composite $\mathcal{G}^{(4)}$ for the two-dimensional rectangular body embedded in a dielectric. The red and green grids cover the top material ($\epsilon = 1$) while the other grids cover the bottom material $\epsilon = 6$.

18.1 Plane Wave Source and Initial Conditions

18.1.1 Periodic in x . Fig. 137 shows preliminary results for a plane wave (no chirp). The top material has $\epsilon = 1$ (the speed of light is 1) while the bottom dielectric was chosen to have $\epsilon = 6$. The domain was periodic in x and the plane wave wave solution was specified as a time-dependent condition on the top boundary. The bottom boundary is an second-order Enquist-Majada absorbing boundary condition.

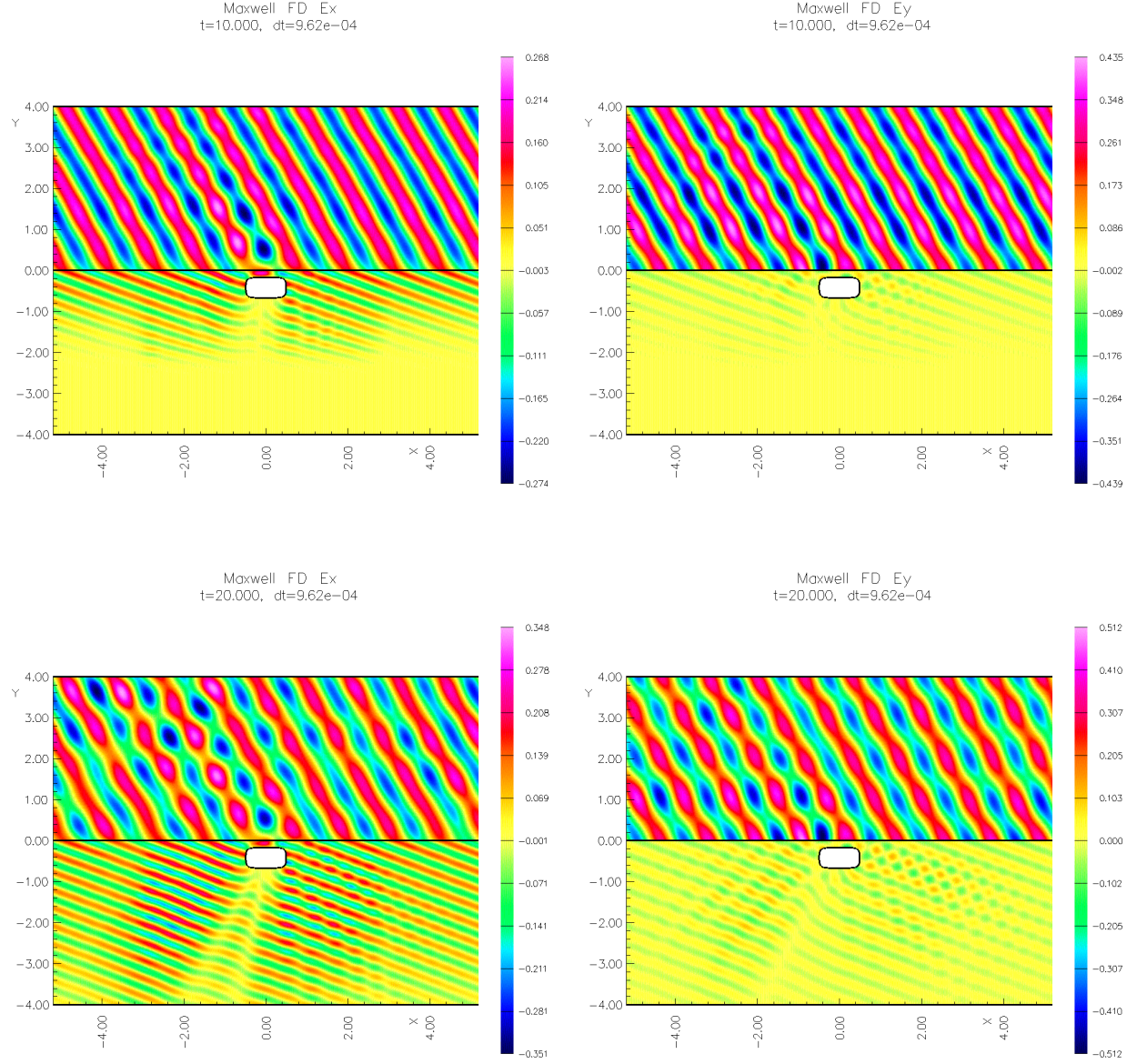


Figure 137: Preliminary results for a plane wave (no chirp). E_x and E_y fields at time $t = 10$ (top) and $t = 20$ bottom. Results for $\mathcal{G}^{(8)}$ (periodic BC's in x).

18.1.2 Radiation Boundary Conditions. The correct radiation boundary conditions across material interfaces have been derived and implemented so that stable solutions can be computed across the interface.

The scattered field can be computed by first computing the solution for no embedded body and again with the embedded body present. The difference of the solutions shows only the scattered field.

Figure 139 shows the computed solution using a planewave initial condition with frequency $\omega = 3\text{GHz}$.

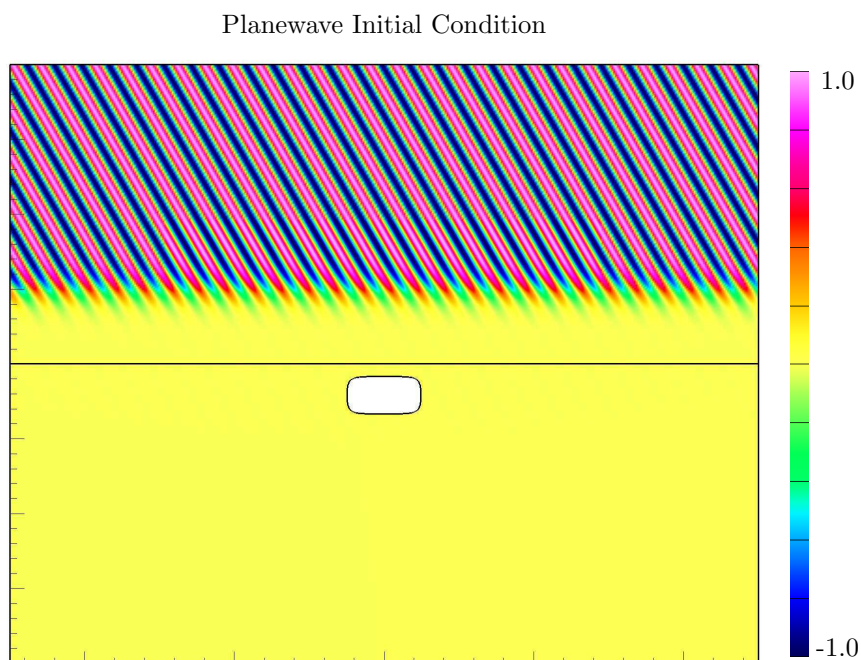


Figure 138: Figure above shows the planwave initial condition used to compute the scattered field in Figure 139.

E_x of scattered field $\omega = 3\text{GHz}$

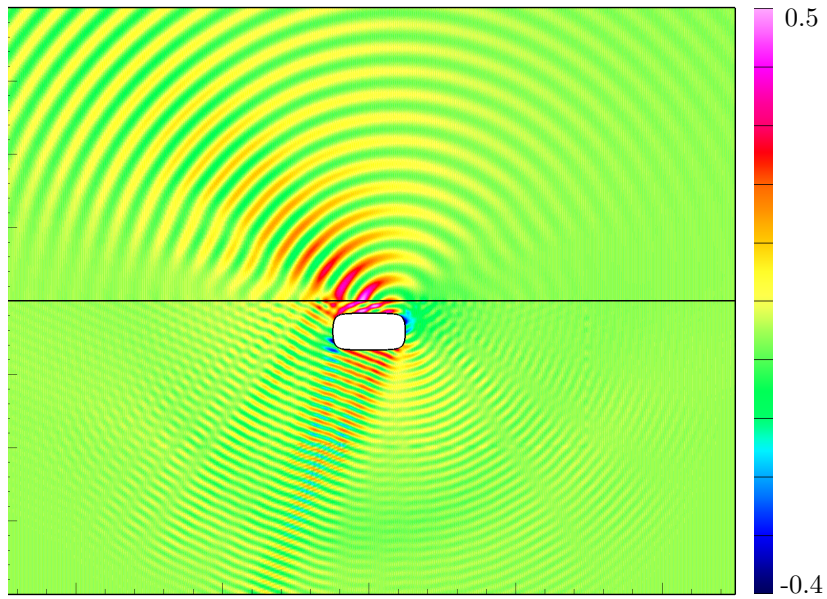


Figure 139: Above is the x -component of the scattered electric field. The boundary conditions for the domain edges are second-order Enquist-Majda absorbing boundary conditions. The body is treated as a perfect electrical conductor.

18.2 Antenna Source

This section will contain results and progress for scattering by an embedded body from an antenna source of radiation.

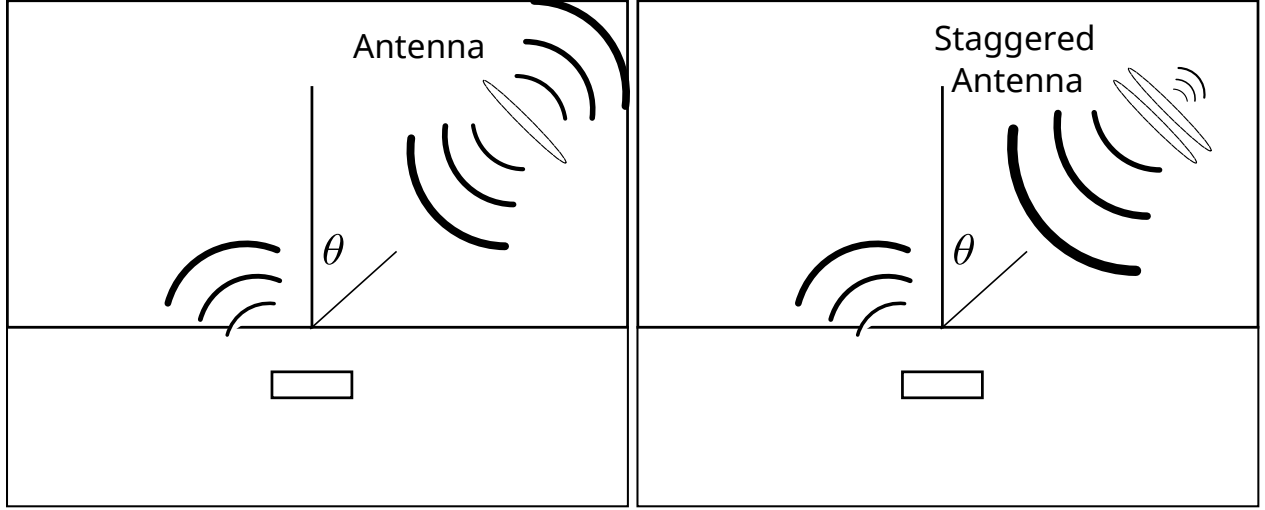


Figure 140: An illustration to demonstrate the idea of using two offset sources in order to minimize back propagation of waves that may contaminate the scattered field.

18.3 Solving for Scattered Field Directly

Here we consider an approach of solving for the scattered field that will not require adding a source to produce the incoming wave. We do this by using the analytic form of the transmitted wave assuming no embedded body to derive a Dirichlet boundary condition for the boundary of the embedded body.

Recall the chirped wave is defined as,

$$P(t) = \chi_\tau(t) e^{iat^2} e^{-i\omega_0 t} \quad (180)$$

Then transmitted wave in the soil is,

$$u^{(T)} = T \cdot P\left(t + \frac{x}{c'} \sin \theta' + \frac{z}{c'} \cos \theta'\right) \quad (181)$$

where c' is the wave speed in the soil, T is the transmission coefficient given by the well known formula

$$T = \frac{2\eta_1 \cos \theta}{\eta_2 \cos \theta + \eta_1 \cos \theta_2} = \frac{2 \cos \theta}{\sqrt{\epsilon} \cos \theta' + \cos \theta} \quad (182)$$

and the refracted wave angle θ' is given by Snell's Law

$$\frac{\sin \theta}{\sin \theta'} = \sqrt{\frac{\epsilon_2 \mu_2}{\epsilon_1 \mu_1}} = \sqrt{\epsilon} \quad (183)$$

where $\eta_i = \sqrt{\frac{\mu_i}{\epsilon_i}}$ and $\mu_1 = \mu_2 = 1$ and $\epsilon_1 = 1$, $\epsilon_2 = \epsilon$.

The total field in the soil is $E = u^{(T)} + u^{(Sc)}$. Using the PEC boundary condition on the embedded body, the tangential component must satisfy

$$u^{(Sc)} \Big|_{\partial\Omega}(t) = -u^{(T)} \Big|_{\partial\Omega}(t) \quad (184)$$

where $\partial\Omega$ is the boundary of the embedded body. This is a time-dependent Dirichlet condition. By assuming the initial scattered field is zero and using the boundary condition 184, we only need to solve an initial-boundary value problem for the scattered field.

The following sections contain computational results using this inhomogeneous PEC boundary condition implemented in the Overture cmgx solver. A convergence study is performed on a square with a PEC wall with no embedded body. Next, there is a figure of computed solution for a PEC cylinder with absorbing boundary conditions at the walls. Finally, a figure showing the results for the embedded body problem with full interface conditions and absorbing boundaries.

Note the indicator function $\chi_\tau(t)$ is implemented with hyperbolic tangents to create a smooth ‘tophat’ function.

18.4 Square

This section contains a convergence study using the inhomogeneous PEC boundary conditions described above. The grid is square with periodic boundary conditions on the top and bottom, absorbing boundary condition on the right and PEC boundary on the left.

N (points)	E_y	r	H_z	r
64	6.8e-5		7.0e-5	
128	4.1e-6	16.6	4.2e-6	16.7
256	2.5e-7	16.4	2.5e-7	16.8

Table 7: Convergence study for inhomogeneous PEC boundary condition. Columns labeled r contain the ratio of errors. They confirm the expected fourth order convergence.

18.5 Embedded Cylinder

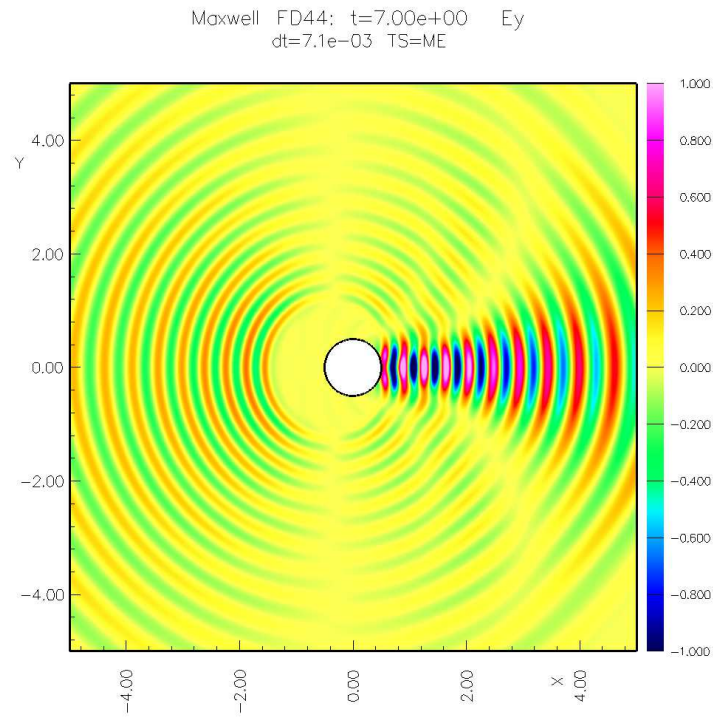


Figure 141: E_y component of scattered field from PEC cylinder at $t = 7$. Indicator function χ_τ turned off at $t = 6$. Bandwidth = 2

Figure 141 shows the results for a PEC cylinder embedded in a dielectric.

18.6 Embedded Body

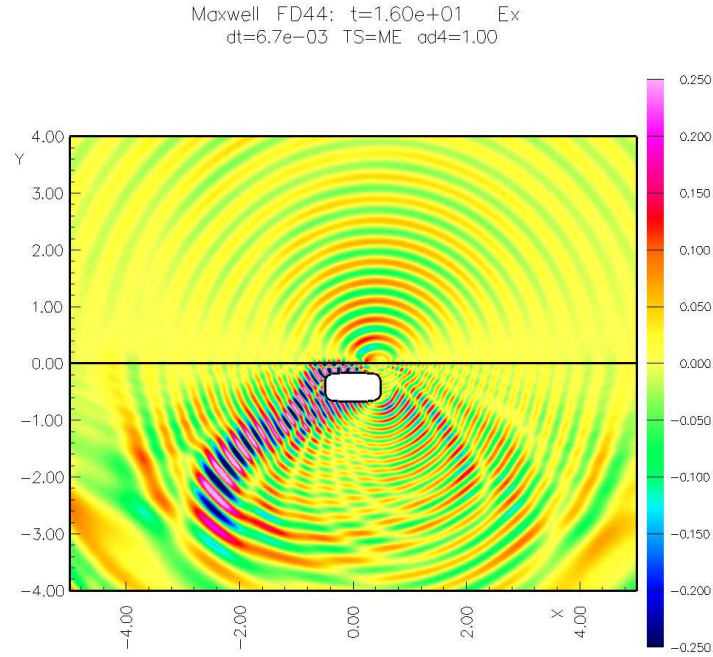


Figure 142: E_x component of scattered field from embedded body at $t = 16$ with full interface conditions and absorbing boundaries. Bandwidth = 4. Incident angle $\theta = 60$.

18.7 Sharpness Study

The following are results of a numerical study of the dependence of the scattered field on the sharpness of the corners of the embedded body. The command file `embeddedBodyGrid.cmd` accepts an argument `sharp` that has default value 20. This parameter determines how quickly the slope of the body's boundary will change at the corners. Larger values of sharpness produce corners closer to right angles.

Below are examples of the corners of the embedded body grid along with several more plots of the E_x component of the scattered field. Finally, we include line plots of the same field taken horizontally across the upper region of the domain with endpoints $(x_a, y_a) = (-5, 3)$ to $(x_b, y_b) = (5, 3)$. All results obtained for bandwidth = 2.

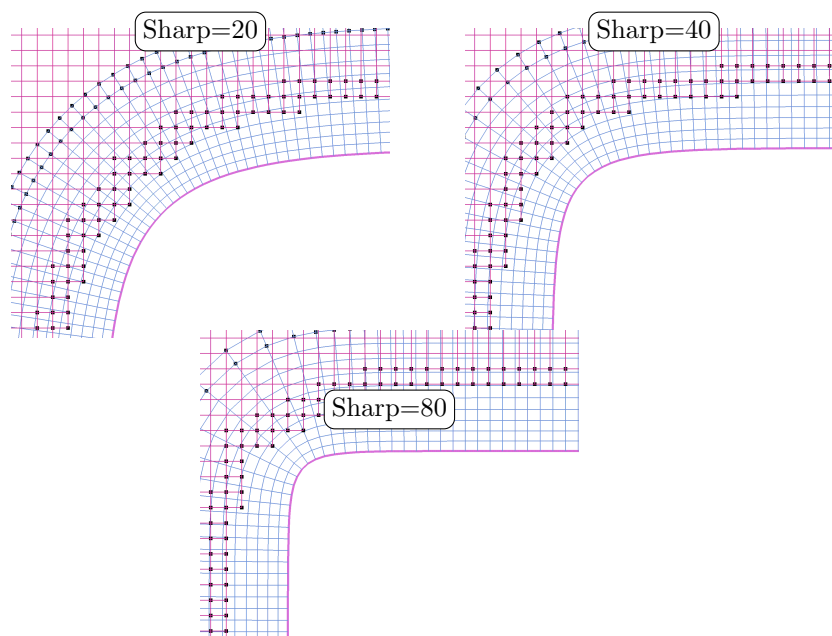


Figure 143: Above are zoomed in views of the upper-left corner of the embedded body grid. As the `sharp` parameter is increased, the corner approaches a right-angle.

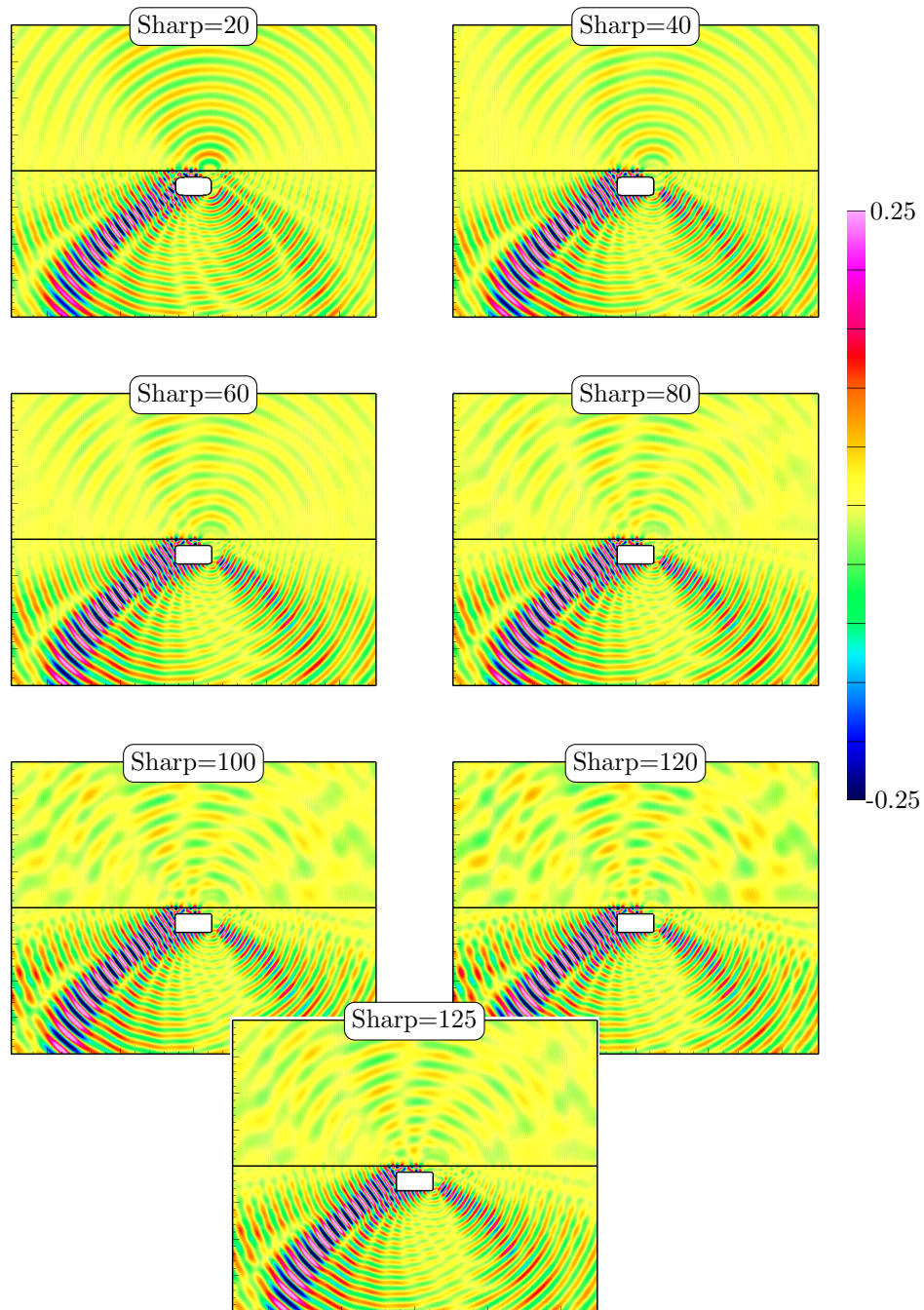


Figure 144: Above are plots of the E_x component of the scattered field for various values of **sharp**.

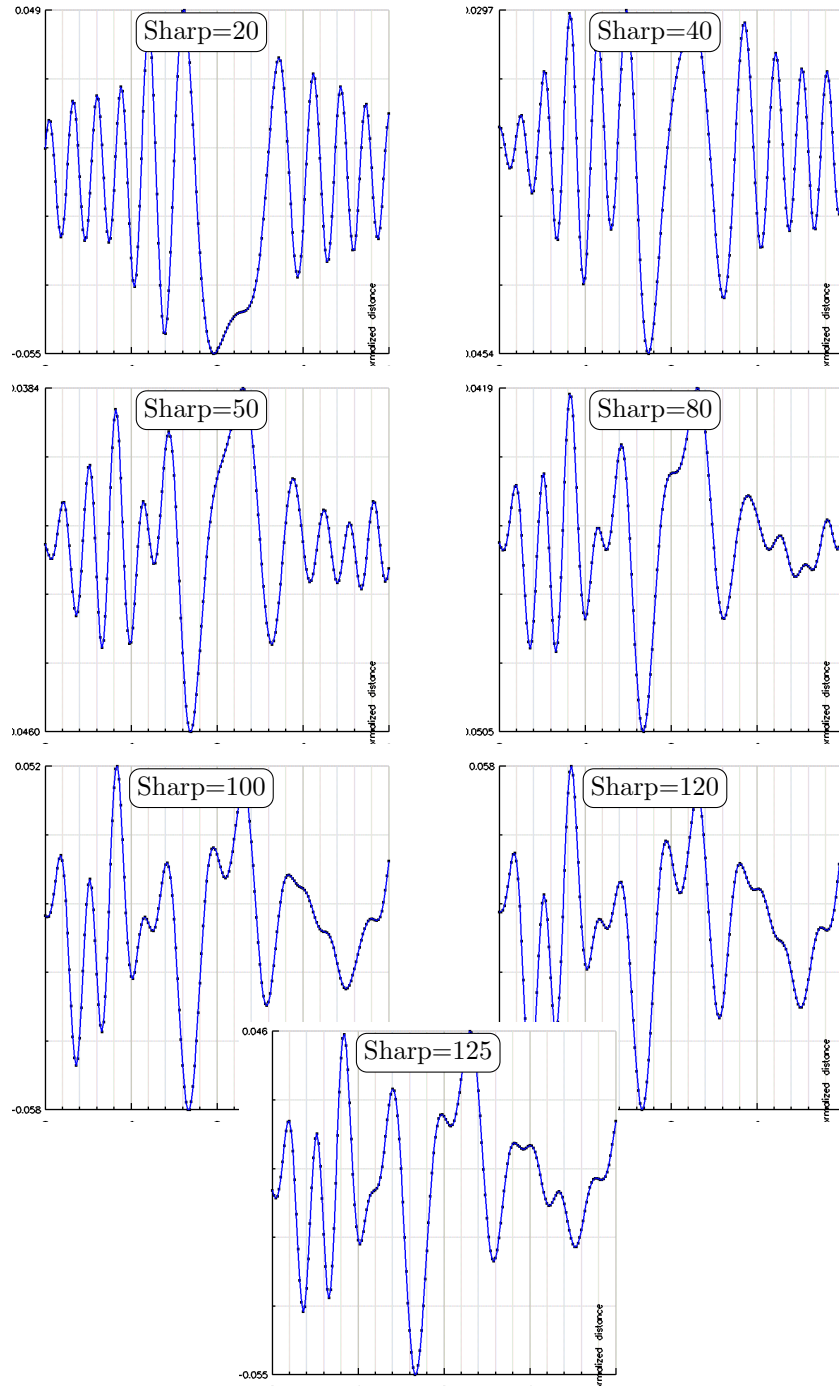


Figure 145: Above shows the E_x field as a line plot across the straight line from $x_a, y_a = (-5, 3)$ to $(x_b, y_b) = (5, 3)$. The field appears to be converging but it remains to define a quantitative metric to measure convergence.

The line plots seem to indicate the field is converging as the corners approach right angles. It remains to define a quantifiable metric so measure this convergence.

18.8 Convergence

Below are results of two grid refinement studies demonstrating self-convergence. Asymptotically, the method should be fourth-order accurate. Even in a complicated geometry with difficult boundary conditions, the convergence rate is approaching the theoretical prediction.

grid	e_{Ex}	ratio	e_{Ey}	ratio	e_{Hz}	ratio
G8	6.3e-3		6.6e-3		1.8e-4	
G16	7.1e-4	8.8	7.6e-4	8.7	2.7e-5	6.7
G32	8.1e-5	8.8	8.8e-5	8.7	4.1e-6	6.7
	3.14		3.12		2.74	

grid	e_{Ex}	ratio	e_{Ey}	ratio	e_{Hz}	ratio
G16	7.0e-4		7.4e-4		3.0e-5	
G32	7.0e-5	10.0	7.9e-5	9.4	7.3e-6	4.1
G64	7.0e-6	10.0	8.4e-6	9.4	1.8e-6	4.1
	3.32		3.23		2.03	

Table 8: L1-norm self convergence results. Left column has grid names indicating the refinement factor of the grid, three columns are the error in each component E_x , E_y , and H_z . Next to each component column is the ratio of the errors. Finally, the final row is an estimate of the order of accuracy.

18.9 Annular Domain

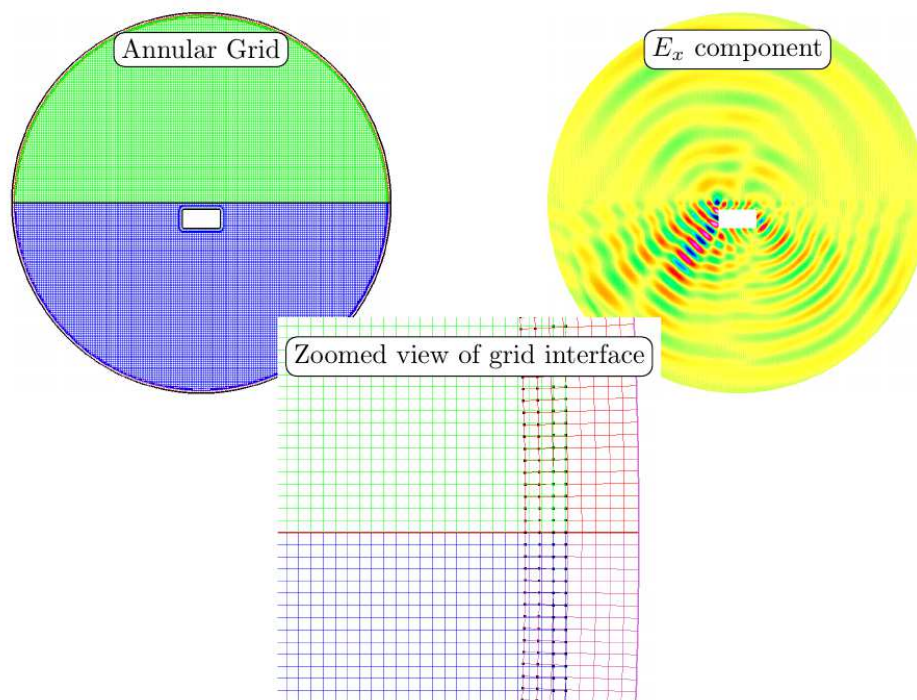


Figure 146: Figures showing computational setup and results for annular grid. Above left: Overlapping grid with annular boundary. Above right: E_x at $t = 12$. Bottom: Zoomed view of grid interface at right boundary.

This section demonstrates the use of overlapping grids to create a geometry more suited for the radiation boundary condition. The radiation boundary conditions are most effective when the angle of incidence is normal to the boundary. Numerical experiments suggest that the waves reflected by the boundary are reduced by approximately a factor of two when using the annular boundary.

18.10 Discussion of the results for Buried Body

We have continued the simulations for this problem, and discuss key elements of the computational model, and ways to improve the accuracy of results.

Parameters outlined in the problem description, such as radar wavelength, may have to be increased to make it consistent with the data provided in [140], that also makes quick computational results feasible.

Radiation boundary conditions across material interfaces have been derived and implemented so that solutions for a fixed frequency may be computed. General interface conditions are described in [79].

A source for the the linear chirp wave remains to be implemented and will be handled by an antenna source or a boundary condition on the embedded body.

It may be advantageous to implement the Kirchoff integral method to compute far-field solutions described in [145].

By implementing a Dirichlet boundary condition on the boundary of the embedded body, we can now solve for the scattered field directly. We demonstrate the accuracy and convergence for this boundary condition on simple geometry and conduct simulation using the more complicated embedded body geometry.

We have given rough evidence of convergence of the scattered field for a boundary approaching a boundary with right angle and will conduct more careful studies to demonstrate this further.

We will also conduct careful studies for on the radiation boundary conditions to verify they do not pollute the scattered field with errors.

The modeling and design tools developed in this project will provide a reduction in the radar cost cost and time-to-market through reduced experimental R&D, design cycle, and streamlined laboratory testing.

Further research will include: (a) the dielectric body (withous metal shell); (b) radar scatterers included, such as stones, animal burrows as well as man-made detritus; (c) analysis of radar images with the Artificial Neural Network (ANN) software, that have the ability of training/learning. The Computaional Sciences LLC team has a great prior experience with ANN for successful DoD applications, [153], [154], [155], [156], [157].

Incorporation of ANN algorithms into the radar IED recognition loop would greatly improve the mitigation of the evolving IED threat to US warfighters in-theatre.

List of Figures

1	Computational setup. The phenomena of interest are inside S , outside is the free space, which is truncated off and replaced with an ABC or a PML. The auxiliary sources are in a narrow region of width ε right next to ∂S on the inside.	13
2	Temporal partition of unity.	15
3	(a) Allocation of field components; (b) Temporal grid.	31
4	Grid for the electric potential.	32
5	Schematic 2D cross-section of the grid for the electric field. Red cells are the nodes in the interior of the computational domain and white cells are the nodes that lie at the boundary.	37
6	Block diagram of the method <code>DoStep(t)</code> that updates all nodes of the grid.	37
7	Schematic 2D cross-section of the grid for the electric field. Red cells are the nodes in the interior of the computational domain, yellow cells lie in the PML, and white cells are the nodes that lie at the outer boundary of the PML.	38
8	Block diagram of the method <code>DoStep(t)</code> in the case where the PML is used in the capacity of a boundary closure. $\mathbf{E}$ denotes the electric field and $\mathbf{H}$ denotes the magnetic field.	39
9	Schematic 2D cross-section of the grid for the electric field. Purple cells represent the domain of the Neumann problem the prevents the accumulation of charge. Red and green cells are the interior nodes of the main problem; green area contains the cells where the auxiliary currents are generated. Blue cells represent the interior nodes of the auxiliary problem. Yellow cells lie in the PML. White cells are at the outer boundary of the PML. Blue line is the outer boundary of the interior (i.e., main) problem; red line is the boundary for the auxiliary problem.	40
10	Block diagram for the implementation of the lacunae-based time marching algorithm. The colors are the same as in Figure 9; they represent different parts of the computational domain.	41
11	Relative error vs. time for the field components E_y (a) and H_x (b). The size of the computational domain in each direction is 5, and the grid dimension is $30 \times 30 \times 30$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].	44
12	Relative error vs. time for the field components E_y (a) and H_x (b). The size of the computational domain in each direction is 10, and the grid dimension is $60 \times 60 \times 60$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].	45
13	The grid convergence on the domain of size 10 for the field component E_y . The coarsest grid dimension is $60 \times 60 \times 60$; subsequent grids are refined by a factor of k , where $k = 2, 4$. Blue curve — Sommerfeld ABC (80), red curve — Higdon ABC (81), yellow curve — Mur ABC (83), grey curve — Betz-Mittra ABC (82), green curve — 15 point unsplit PML of [72].	45
14	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the number of Givoli-Neta angles.	47

15	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the size of the computational domain.	48
16	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles.	49
17	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of six Givoli-Neta angles.	50
18	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the grid size.	51
19	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the number of Givoli-Neta angles. First order accurate approximation of the ABCs in time.	52
20	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the size of the computational domain. First order accurate approximation of the ABCs in time.	53
21	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles. First order accurate approximation of the ABCs in time.	54
22	Maximum relative error of the field component E_y over the entire computational domain. Dependence of the interval of stable performance on the values of three Givoli-Neta angles. First order accurate approximation of the ABCs in time.	54
23	The functions $\Theta(t)$ that help partition the sources.	55
24	Relative error for the E_y component of the electromagnetic field.	56
25	Relative error for the H_x component of the electromagnetic field.	56
26	Relative error for the H_z component of the electromagnetic field.	57
27	Maximum relative error of the field component $E_y(x, y, z, t)$. Sommerfeld ABC (80) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E)	60
28	Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E)	61
29	Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order $(5, 0)$. The solution in the left panel is computed on the grid with $60 \times 60 \times 60$ and the solution in the right panel is computed on the grid with $120 \times 120 \times 120$ nodes.	62
30	Long-time growth of maximum relative error for the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order (P, E) . The blue curve corresponds to the solution computed on the grid with $60 \times 60 \times 60$ and the orange curve corresponds to the solution computed on the grid with $120 \times 120 \times 120$ nodes.	62
31	Maximum relative error of the field component $E_y(x, y, z, t)$. Second-order Higdon ABC (91) is used for auxiliary functions for the Hagstrom-Warburton NRBC of order $(5, 0)$. Long-time stability is enabled by the method of lacunae.	63
32	Schematic for two problems that allow us to combine the Hagstrom-Warburton NRBC and a PML on neighboring faces.	63

33	The performance of Hagstrom-Warburton NRBC on one face of the cube combined with the unsplit PML of [72] on the other five faces as it depends on the size of the computational domain. Curves of different color represent maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c).	64
34	Maximum relative error for $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) components. Orange curve – exact boundary condition, blue curve – unsplit PML [72].	65
35	Maximum relative error for $E_y(x, y, z, t)$ in long-time simulation.	66
36	Implementation of the Hagstrom-Warburton NRBC with lacunae and the original divergence-free source. The main problem (green) is partitioned into sub-problems. Each of the latter consists of the first problem (red) with Hagstrom-Warburton NRBC on the one face of the domain and the second problem (blue) terminated by the PML.	66
37	Maximum relative error for the electric field component $E_y(x, y, z, t)$: grey curve — computation with lacunae and partition of the original currents; orange curve — computation with no use of lacunae. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.	67
38	Schematic for the implementation of Hagstrom-Warburton NRBC with lacunae and auxiliary currents.	68
39	Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c): blue curve — lacunae applied to auxiliary currents; red curve — lacunae applied to original currents. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile for the case where the computational domain is terminated by the exact boundary conditions.	69
40	Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c), when the additional time of each auxiliary problem is T_g steps. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile for the case where the computational domain is terminated by the exact boundary conditions.	70
41	Maximum absolute value of the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c). Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. The first column represents the results when an additional time of each auxiliary problem is zero and the second column represents the results when $T_g = 1500$ steps.	71
42	Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) for three different values of the smoothing parameter σ in (92), (93). Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere. Black curve is the error profile when the computational domain is terminated by the exact boundary conditions.	72

43	Maximum relative error for the field components $E_y(x, y, z, t)$ – (a) and $H_x(x, y, z, t)$ – (b) as we increase the auxiliary domain size by a given number of grid points in each coordinate direction. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.	73
44	Maximum relative error for the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) as we vary the grid size. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.	74
45	Maximum absolute value of the field components $E_y(x, y, z, t)$ – (a), $H_x(x, y, z, t)$ – (b), and $H_z(x, y, z, t)$ – (c) for the second auxiliary sub-problem. Hagstrom-Warburton NRBC of order (5,0) is used as the boundary condition for $E_y(x, y, z, t)$ at the face $x = x_E$ of a cubic domain. Unsplit PML of [72] is used elsewhere.	75
46	The maximum absolute relative error of E_y component. Unsplit PML is used as boundary condition in the test with divergence-free source.	80
47	The maximum absolute relative error of E_y (upper) and E_z (lower) components. Unsplit PML is used as boundary condition in the test with non divergence-free source.	81
48	Partial subproblem surrounded by PML.	82
49	A smooth partition of unity.	82
50	The maximum absolute relative error of E_y (upper) and E_z (lower) components. The blue curve represents the test with PML and the orange curve represents the test with lacunae-based time marching algorithm with PML and divergence-free auxiliary currents.	84
51	Lacuna of the solution generated by compactly supported sources.	85
52	The maximum absolute relative error of E_y (a), H_x (b), H_z (c) components. DAB is used as boundary condition in the test with divergence-free source.	87
53	The maximum absolute relative error of E_x (a), E_z (b), H_y (c) components. DAB is used as boundary condition in the test with non divergence-free source.	88
54	The maximum absolute relative error of E_x (a), E_z (b), H_y (c) components. DAB with 4 recursions (blue) and 10 recursions (orange) are used as boundary condition in the test with non divergence-free source.	89
55	The partition of unity in time.	95
56	(a) The near-boundary layer S_ε (only two of the six planar layers are shown); (b) The direction of growth of μ in the near-edge domain.	95
57	Grid structure. The cross-section of the entire computational domain along the x -axis.	98
58	Domains S and S_{aux} in the case where the latter is terminated by the PML.	99
59	Grid summary for the set Grid 1.	100
60	Left: error profile of E_y (∞ -norm) computed under the pure PML approach and the lacunae-based integration. Right: the same for E_z	101
61	Left: error profiles for E_y (binary logarithm of ∞ -error norm) on two successive grids. Right: the same for E_z	102
62	Left: evolution of $\max E_y $ in time for the 2nd AP computed using the combined PML+lacunae approach, as well as using the original lacunae-based integration, on two successive grids, Grid 1 and Grid 2. Right: the same for $ E_z $	102
63	Computational domains for the sphere.	103

64	Grid summary for the set Grid 1.	104
65	Grid summary for the set Grid 2.	104
66	Grid summary for the set Grid 3.	105
67	The reflected field E_y at the point $(-2.5, 0, 0)$ in the absence of artificial dissipation.	105
68	Left: the reflected field E_y at the location $(-2.5, 0, 0)$; lacunae-based integration+PML at the outer boundary of the APs. Artificial dissipation is used everywhere on S . Right: the same but with non-zero dissipation only near the sphere.	106
69	The reflected field E_y at the location $(-2.5, 0, 0)$. Pure PML and the combined PML+lacunae approach.	107
70	The reflected field E_y at the location $(-2.5, 0, 0)$ over a long run. Lacunae-based integration combined with the PML.	107
71	Left: the reflected field E_y at point $(-2.5, 0, 0)$ on three successive grids. Lacunae-based integration+PML. Right: an enlarged fragment of the left panel.	108
72	Convergence of E_y on three consecutive grids.	109
73	Maximum values of $ E_x $, $ E_y $, and $ E_z $ for the 2nd AP on three consecutive grids. Lacunae-based integration+PML.	110
74	Scattered field E_y on S at the time $t = 0.03T$. The set Grid 1 is used.	111
75	Reentry vehicle.	111
76	Computational domain for the reentry vehicle.	112
77	A zoomed in view for the reentry vehicle.	113
78	Summary of the grids used for computing the scattering about a reentry vehicle.	114
79	E_y at $t = 0.1T$. Total field (incident+scattered) in the small box (domain S), reflected field only in the big box (domain S_{aux}). Lacunae+PML approach.	116
80	E_x at $t = 0.1T$. Total field (incident+scattered) in the big box (domain S_{aux}). Lacunae+PML approach.	117
81	E_y at $t = 0.3T$. Total field (incident+scattered) in the small box (domain S). Lacunae+PML approach.	118
82	E_x at $t = 0.3T$. Total field (incident+scattered) in the small box (domain S). Lacunae+PML approach.	119
83	Left: Scattered field E_y vs time. Pure PML and lacunae+PML approaches are presented. Right: Zoom in for an early stage of the evolution of E_y	120
84	Schematic for computing the discrete Calderon's operators in an unsteady framework.	123
85	Secondary lacuna in the sense of Petrowsky [109] in the solution to the wave equation driven by a compactly supported source. The vertical axis represents time, the hyperplane normal to it represents space. The lacuna is a region of space-time behind the aft fronts of the propagating waves, on which the solution is zero (the interior cone).	123
86	Computational domain S_0 and auxiliary domains S_{aux}	124
87	Large scale space-time stencil of the algorithm.	125
88	Error of the numerical solution as compared against the reference solution (139) for three sets of basis functions of increasing dimension. Demonstrates convergence of the numerical solution with respect to the dimension of the basis on the continuous boundary.	127
89	Error of the numerical solution as compared against the reference solution (139) for two different discretization grids, with the size of the second grid being one half of the size of the first grid. Demonstrates the second order of grid convergence.	127

90	$\log_2$ - L_∞ -error profiles of the Neumann IBVP for the Klein-Gordon equation (149). On the left: error of the genuine solution u . On the right: $\log_2$ -error of $ \nabla u $. Both plots correspond to the “no-lacunae” technique.	130
91	L_∞ -error profiles of the Neumann IBVP (146). On the left: error of the genuine solution u . On the right: $\log_2$ -error of $ \nabla u $. Both plots correspond to “no-lacunae” technique.	131
92	$\log_2$ - L_∞ -error profile of $ \nabla u $ for the Neumann problem (146). The only distinction from Figure 91 (right panel) is the increased height of the cylinder: $T_0 = 5$ instead of $T_0 = 1$. Again, “no-lacunae” technique. Also, notice a shorter computational time compared to Figure 91 as the error on the fine grid starts accumulating so that it makes no sense to advance any further.	132
93	L_∞ -error profiles of the Neumann problem (146). On the left: error of the solution u itself. On the right: $\log_2$ -error of $ \nabla u $. Both plots use a lacunae-based approach. A typo in the legends: $T_0 = 5$ rather than $T_0 = 1$	133
94	(a) Simulated AFRL A6-3 magnetron geometry used in performance testing by ICEPIC [129], simulates the electrodynamics and charged-particle dynamics of microwave RF-systems (Maxwell’s equations and the Lorenz force law). (b) Photo of the corrugated horn antenna (from [127], reproduced from [134]). (c) Modern CW industrial heating magnetrons are capable for producing as high as 300 kW of continuous-wave microwave power at frequencies around 900 MHz and are sold commercially [132]. (d) Diagram of a typical computational domain for ICEPIC. . . .	135
95	Staggered Yee grid for FDTD. Placement of the field components in the Yee FDTD method.	136
96	Fields required to push the particle (shown as a star) are shown circled. Additional fields needed to update the push fields one time-step are shown boxed.	136
97	Yee FDTD numerical dispersion relation for propagation along a grid axis, a 2D grid diagonal, and a 3D grid diagonal compared to the physical dispersion relation. The FDTD parameters are $\Delta x = \Delta y = \Delta z = \Delta$ and $\Delta t = \nu \Delta_{t,max} = \nu \Delta / c / \sqrt{3}$	137
98	E_x and E_y values used to compute curl $\mathbf{E}$ to update B_z . The standard Yee FDTD scheme uses the values shown in green. The values shown in red and blue can be used in addition to the values shown in green.	139
99	(a) Extended stencil numerical dispersion relation for $K_1 = -3/8; K_2 = 0; \Delta x = \Delta y = \Delta z = \Delta$; and $\Delta t = \nu \Delta_{t,max} = (2/3)\nu \Delta / c / \sqrt{3}$. (b) Extended stencil numerical dispersion relation for $K_1 = -1/16, K_2 = 2K_1$, $\Delta x = \Delta y = \Delta z = \Delta$; and $\Delta t = \nu \Delta_{t,max} = (6/7)\nu \Delta / c / \sqrt{3}$	139
100	ICEPIC loop which attempts to overlap computation and communication but has multiple synchronization points each time-step.	141
101	Simulated AFRL A6-3 magnetron geometry used in performance testing.	141
102	Redesigned ICEPIC loop with a single communication point each timestep.	142
103	Run-time and efficiency for the AFRL A6-3 magnetron for both the single and multiple synchronization point algorithms. The runs are performed on machine B for a fixed problem size. (a) Run-time. (b) Efficiency.	143
104	Run-time and efficiency for the AFRL A6-3 magnetron for both the single and multiple synchronization point algorithms. The runs are performed on machine with a number of processors maintaining constant $(NX*NY*NZ)/\Delta t \#CPU$. (a) Run-time versus # of processors (b) Efficiency.	144

105	Picture of PML in action. The source of radiation is at the left end, and the PML begins at $z = 0$	146
106	Geometry for tapering of PML parameters.	148
107	Slice of the geometry of a PML in a WR-650 waveguide. The PML occupies the gridded region. The waveguide walls are shown as darkened lines. The dotted line shows the region encompassed by [Bound#].shape. In this case, [Bound#].ndepth is 21, and [Bound#].dir is Z. All of the space inside the waveguide walls is marked as vacuum by the [Shape#] commands, and the PML grows inward (i.e., in the minus Z direction in this example) into the vacuum region.	149
108	(a) Simulated AFRL A6-3 magnetron geometry used in performance testing by ICEPIC [129], simulates the electrodynamics and charged-particle dynamics of microwave RF-systems (Maxwell's equations and the Lorenz force law). (b) Modern continuous wave (CW) industrial heating magnetrons are capable for producing as high as 300 kW of continuous-wave microwave power at frequencies around 900 MHz and are sold commercially [132].	153
109	ICEPIC Simulations of a Pulsed Relativistic Magnetron with Transparent Cathode at UNM. Photograph of transparent cathode with field lines showing cathode priming, electrostatic priming, and magnetic priming [131].	154
110	(a) Simulation model for a 140GHz spatial-harmonic magnetron, [138]. (b) Normalized values of POut /I ² versus $N\theta/\pi$ for Qext varying from 80 to 160.(c) Time-evolved spoke formation plotted in the $r\theta$ plane at (a) 0.4 ns, (b) 1.4 ns, and (c) 5 ns. The results are for $V_D C = 11.3\text{kV}$, $B = 0.79T$, and $Q_{ext} = 160$	155
111	(a) Axial view of the A6 relativistic magnetron with a solid cathode. (b) Grid plot of A6 magnetron in ICEPIC. (c) Computational domain material composition plot, with a slice.	156
112	(a) Results of the ICEPIC simulations for the A6 relativistic magnetron, B-field magnitude. (b) Solution plot on the domain surface.	156
113	Configurations and parameters of bevel- and step-cut Vlasov antennas.	157
114	Vlasov antenna with proposed in [148] cut.	158
115	Bevel-cut Vlasov antenna with reflector.	158
116	Bevel-cut 3D gain patterns: (a) without reflector and (b) with reflector.	159
117	Bevel-cut geometry for simulation of Vlasov antenna with ICEPIC.	160
118	Bevel-cut 3D simulation of Vlasov antenna with ICEPIC. The E-field is shown on the top part of the computational domain.	160
119	Vlasov antenna simulation with ICEPIC. The E- and B-field variations in time at points [0,1,-0.03] (red curve) and [0,1,0.06] (blue curve).	161
120	Vlasov antenna simulation with ICEPIC, PML depth=3 (red curve). The errors in B-field start to grow at time $t=100$, and exceed one order in magnitude at time $t=300$, continuing to grow exponentially then. The lacuna method solution (pink curve) remains stable for any large time.	162
121	Vlasov antenna simulation with ICEPIC, PML depth=3 (red curve). The errors in B-field start to grow at time $t=100$, and exceed one order in magnitude at time $t=300$, continuing to grow exponentially then. The lacuna method solution (pink curve) remains stable for any large time.	162

122	Streamlines of the initial magnetic field showing the vector projections of the magnetic field direction onto the principle planes (red, black, blue). The magnitude decreases from the origin. Also shown are sample 3D streamlines tracing the magnetic field. Blue streamlines are contained within the domain, but green and purple streamlines leave the domain at an oblique angle.	164
123	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$	166
124	WARPX initialization of x -aligned quasi-spherical electromagnetic pulse showing logarithmic contours of the magnetic field magnitude. The magnetic field has B_y and B_z components and $B_x = 0$ as given by Eq. (177). Shown are the two cases representing the (a) standard and (b) compact spatial extents of the pulse. The computational domain sizes are identical.	168
125	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$	169
126	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. The domain size is $3 \times 3 \times 3$. Results are shown at times, $t = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5$	170
127	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. No re-integration is performed for these simulations. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 125(d) and 125(f), which are plotted at the same scale.	171
128	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the standard initial spatial extent and showing logarithmic contours of the magnetic field magnitude. Frequent re-integration is performed for these simulations every 0.05 time. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 125(d) and 125(f), which are plotted at the same scale.	171
129	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. No re-integration is performed for these simulations. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 126(d) and 126(f), which are plotted at the same scale.	172
130	Simulation results from WARPX for the x -aligned quasi-spherical electromagnetic pulse using the compact initial spatial extent and showing logarithmic contours of the magnetic field magnitude. Frequent re-integration is performed for these simulations every 0.05 time. Interior domain size is $1 \times 1 \times 1$. Only one of three auxiliary domains is shown. Results are shown at times, $t = 1.5$ and 2.5 , and can be compared to the upper-right-front portions of Figs. 126(d) and 126(f), which are plotted at the same scale.	172

131	Temporal evolution of the error caused by the boundary condition implementation. Error is computed as the 2-norm in reference to a simulation with an extended domain such that the solution does not reach the boundaries during the simulation time. The errors are shown for the lacuna-based boundary condition implementation (LBC) and a zero gradient boundary condition implementation (ZBC). The lacuna-based boundary condition clearly performs better for both pulse initializations, the standard spatial extent Fig. 131(b) and the compact spatial extent Fig. 131(a). . . .	173
132	(a) The UWB-radar transmits in a wide spectrum with very low power across a much wider frequency than conventional radar systems and are usually very difficult to detect; (b) Minehound radar module system diagram [140].	176
133	(a) Remote-control "Panama" Land Rover with ground-penetrating radar to detect IEDs followed by Mastiff with Choker mine rollers [144]; (b) Although a number of vehicle based GPR systems have been trialled and reported on over the last 5 years the most extensively reported system has been the NIITEK radar shown [140]	177
134	Planewave incident on interface at angle θ at $z = 0$	180
135	Result of numerically solving 1D scalar wave-equation $u_{tt} = c^2 u_{xx} + F(x, t)$ where $F(x, t)$ is chosen to produce one-way chirped planewaves. One point source is located at $x = 3.75$ and a second point source is placed one grid cell in front at $x = 3.75 + \Delta x$.	181
136	Composite $\mathcal{G}^{(4)}$ for the two-dimensional rectangular body embedded in a dielectric. The red and green grids cover the top material ($\epsilon = 1$) while the other grids cover the bottom material $\epsilon = 6$	182
137	Preliminary results for a plane wave (no chirp). E_x and E_y fields at time $t = 10$ (top) and $t = 20$ bottom. Results for $\mathcal{G}^{(8)}$ (periodic BC's in x).	183
138	Figure above shows the planwave initial condition used to compute the scattered field in Figure 139.	184
139	Above is the x -component of the scattered electric field. The boundary conditions for the domain edges are second-order Enquist-Majda absorbing boundary conditions. The body is treated as a perfect electrical conductor.	185
140	An illustration to demonstrate the idea of using two offset sources in order to minimize back propagation of waves that may contaminate the scattered field.	186
141	E_y component of scattered field from PEC cylinder at $t = 7$. Indicator function χ_τ turned off at $t = 6$. Bandwidth = 2	188
142	E_x component of scattered field from embedded body at $t = 16$ with full interface conditions and absorbing boundaries. Bandwidth = 4. Incident angle $\theta = 60$	189
143	Above are zoomed in views of the upper-left corner of the embedded body grid. As the sharp parameter is increased, the corner approaches a right-angle.	190
144	Above are plots of the E_x component of the scattered field for various values of sharp .	191
145	Above shows the E_x field as a line plot across the straight line from $x_a, y_a = (-5, 3)$ to $(x_b, y_b) = (5, 3)$. The field appears to be converging but it remains to define a quantitative metric to measure convergence.	192
146	Figures showing computational setup and results for annular grid. Above left: Overlapping grid with annular boundary. Above right: E_x at $t = 12$. Bottom: Zoomed view of grid interface at right boundary.	194

List of Tables

1	Comparison of the single-thread and parallel execution times.	35
2	Comparison of the one time-step for different boundary conditions in serial implementation.	90
3	Comparison of the one time-step for different boundary conditions in parallel implementation.	91
4	Specification of hardware tested	142
5	Performance improvements due to all changes referenced in this report for the AFRL A6-3 magnetron performed with 48 CPUs.	142
6	Input file labels for PML parameters	150
7	Convergence study for inhomogeneous PEC boundary condition. Columns labeled r contain the ratio of errors. They confirm the expected fourth order convergence. . .	187
8	L1-norm self convergence results. Left column has grid names indicating the refinement factor of the grid, three columns are the error in each component E_x , E_y , and H_z . Next to each component column is the ratio of the errors. Finally, the final row is an estimate of the order of accuracy.	193

References

- [1] S. V. Tsynkov. Numerical solution of problems on unbounded domains. A review. *Appl. Numer. Math.*, 27(4):465–532, 1998.
- [2] Thomas Hagstrom. Radiation boundary conditions for the numerical simulation of waves. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 47–106. Cambridge Univ. Press, Cambridge, 1999.
- [3] Alvin Bayliss and Eli Turkel. Radiation boundary conditions for wave-like equations. *Comm. Pure Appl. Math.*, 33(6):707–725, 1980.
- [4] Alvin Bayliss and Eli Turkel. Outflow boundary conditions for fluid dynamics. *SIAM J. Sci. Statist. Comput.*, 3(2):250–259, 1982.
- [5] Alvin Bayliss and Eli Turkel. Far field boundary conditions for compressible flows. *J. Comput. Phys.*, 48(2):182–199, 1982.
- [6] Bjorn Engquist and Andrew Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31(139):629–651, 1977.
- [7] Björn Engquist and Andrew Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Comm. Pure Appl. Math.*, 32(3):314–358, 1979.
- [8] Björn Engquist and Andrew Majda. Numerical radiation boundary conditions for unsteady transonic flow. *J. Comput. Phys.*, 40(1):91–103, 1981.
- [9] Robert L. Higdon. Numerical absorbing boundary conditions for the wave equation. *Math. Comp.*, 49(179):65–90, 1987.
- [10] O. M. Ramahi. Stability of absorbing boundary conditions. *IEEE Trans. Antennas Propagat.*, 47:593–599, 1999.
- [11] Jean-Pierre Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.
- [12] Jean-Pierre Bérenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 127(2):363–379, 1996.
- [13] S. Abarbanel and D. Gottlieb. A mathematical analysis of the PML method. *J. Comput. Phys.*, 134(2):357–363, 1997.
- [14] R. W. Ziolkowski. Time-derivative Lorenz material model based absorbing boundary conditions. *IEEE Trans. Antennas Propagat.*, 45(10):1530–1535, 1997.
- [15] S. Abarbanel, D. Gottlieb, and J. S. Hesthaven. Long time behavior of the perfectly matched layer equations in computational electromagnetics. *J. Sci. Comput.*, 17(1-4):405–422, 2002.
- [16] Saul Abarbanel, Heydar Qasimov, and Semyon Tsynkov. Long-time performance of unsplit PMLs with explicit second order schemes. *J. of Sci. Comput.*, 41(1):1–12, October 2009.
- [17] Daniel Rabinovich, Dan Givoli, and Eliane Bécache. Comparison of high-order absorbing boundary conditions and perfectly matched layers in the frequency domain. *Int. J. Numer. Methods Biomed. Eng.*, 26(10):1351–1369, 2010.

- [18] Allen Taflove and Susan C. Hagness. *Computational Electrodynamics: the Finite-Difference Time-Domain Method*. Artech House Inc., Boston, MA, Third edition, 2005.
- [19] R. L. Wagner and W. C. Chew. An analysis of Liao's absorbing boundary conditions. *J. Electromagn. Waves Applicat.*, 9:993–1009, 1995.
- [20] O. M. Ramahi. Stable FDTD solutions with higher-order absorbing boundary conditions. *Microwave Opt. Technol. Lett.*, 15:132–134, 1995.
- [21] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time Dependent Problems and Difference Methods*. Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, 1995. A Wiley-Interscience Publication.
- [22] Victor S. Ryaben'kii and Semyon V. Tsynkov. *A Theoretical Introduction to Numerical Analysis*. Chapman & Hall/CRC, Boca Raton, FL, 2007.
- [23] Lloyd N. Trefethen and Laurence Halpern. Well-posedness of one-way wave equations and absorbing boundary conditions. *Math. Comp.*, 47(176):421–435, 1986.
- [24] Björn Engquist and Laurence Halpern. Far field boundary conditions for computation over long time. *Appl. Numer. Math.*, 4(1):21–45, 1988.
- [25] B. Engquist and L. Halpern. Long-time behaviour of absorbing boundary conditions. *Math. Methods Appl. Sci.*, 13(3):189–203, 1990.
- [26] S. D. Gedney. The perfectly matched layer absorbing medium. In A. Taflove, editor, *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*, pages 263–340. Artech House, Boston, MA, 1998.
- [27] J. A. Roden and S. D. Gedney. Convolution PML(CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339, 2000.
- [28] Eliane Bécache, Peter G. Petropoulos, and Stephen D. Gedney. On the long-time behavior of unsplit perfectly matched layers. *IEEE Trans. Antennas and Propagation*, 52(5):1335–1342, 2004.
- [29] S. Abarbanel, D. Gottlieb, and J. S. Hesthaven. Non-linear PML equations for time dependent electromagnetics in three dimensions. *J. Sci. Comput.*, 28(2-3):125–137, 2006.
- [30] T. Ha-Duong and P. Joly. On the stability analysis of boundary conditions for the wave equation by energy methods. I. The homogeneous case. *Math. Comp.*, 62(206):539–563, 1994.
- [31] Daniel Baffet and Dan Givoli. On the stability of the high-order Higdon Absorbing Boundary Conditions. *Appl. Numer. Math.*, 61:768–784, 2011.
- [32] Max Born and Emil Wolf. *Principles of optics: Electromagnetic theory of propagation, interference and diffraction of light*. With contributions by A. B. Bhatia, P. C. Clemmow, D. Gabor, A. R. Stokes, A. M. Taylor, P. A. Wayman and W. L. Wilcock. Seventh (expanded) edition. Cambridge University Press, Cambridge, 1999.
- [33] M. J. M. Jessel. Sur les absorbeurs actifs. In *Proceedings of 6th ICA*, page 82, Tokio, 1968. Paper F-5-6.

- [34] M. J. M. Jessel and G. A. Mangiante. Active sound absorbers in an air duct. *J. Sound and Vibration*, 23(3):383–390, 1972.
- [35] G. A. Mangiante. Active sound absorption. *J. Acoust. Soc. Amer.*, 61(6):1516–1523, 1977.
- [36] G. Canevet. Active sound absorption in an air conditioning duct. *J. Sound and Vibration*, 58(3):333–345, June 1978.
- [37] M. J. M. Jessel. Some evidences for a general theory of active noise sound absorption. In *Inter-Noise 79: Proceedings*, pages 169–174, Warsaw, Poland, 1979. Institute of Fundamental Technological Research of the Polish Academy of Sciences.
- [38] Gerard Mangiante. The JMC method for 3-D active sound absorption: A numerical simulation. *Noise Control Eng. J.*, 41(2):339–345, 1993.
- [39] S. Uosukainen. Modified JMC method in active control of sound. *Acta Acustica united with Acustica*, 83(1):105–112, January/February 1997.
- [40] S. Uosukainen. Active sound scatterers based on the JMC method. *J. Sound and Vibration*, 267(5):979–1005, November 2003.
- [41] H. Lim, S. V. Utyuzhnikov, Y. W. Lam, A. Turan, M. Avis, V. S. Ryaben’kii, and S. V. Tsynkov. An experimental validation of the noise control methodology based on difference potentials. *AIAA Journal*, 47(4):874–884, April 2009.
- [42] S. Charles, S. Kapotas, and S. Phadke. Absorbing boundaries with Huygens’ secondary sources. In *Proceedings of 64th Annual S.E.G. Meeting*, pages 1363–1366, Los Angeles, USA, 1994.
- [43] Gerard Mangiante and Sylvestre Charles. Absorbing boundary conditions for acoustic waves and Huygens’ principle. *J. Acoust. Soc. Am.*, 103(5):2971–2971, 1998.
- [44] S. A. Schelkunoff. Some equivalence problems of electromagnetics and their application to radiation problems. *Bell System Technical Journal*, 15(1):92–112, January 1936.
- [45] Jean-Pierre Bérenger. A Huygens subgridding for the FDTD method. *IEEE Trans. Ant. Propag.*, 54(12):3797–3804, 2006.
- [46] Jean-Pierre Bérenger. Extension of the FDTD Huygens subgridding algorithm to two dimensions. *IEEE Trans. Ant. Propag.*, 57(12):3860–3867, 2009.
- [47] Fumie Costen and Jean-Pierre Bérenger. Extension of the FDTD Huygens subgridding to frequency dependent media. *Annals of Telecommunications*, 65(3):211–217, 2010.
- [48] Jean-Pierre Bérenger. On the Huygens absorbing boundary conditions for electromagnetics. *J. Comput. Phys.*, 226(1):354–378, 2007.
- [49] Lars Gårding. Sharp fronts and lacunas — some problems and results. In N. H. Ibragimov and L. V. Ovsyannikov, editors, *Proceedings of the Joint IUTAM/IMU Symposium “Group-Theoretical Methods in Mechanics”*, pages 130–144, USSR, Novosibirsk, August 1978. USSR Acad. Sci., Siberian Branch, Institute of Hydrodynamics — Computing Center.
- [50] Henry A. Warchall. Wave propagation at computational domain boundaries. *Comm. Partial Differential Equations*, 16(1):31–41, 1991.

- [51] Henry A. Warchall. Induced lacunas in multiple-time initial value problems and unbounded domain simulation. In Joseph Weiner and Jack K. Hale, editors, *Partial Differential Equations*, volume 273 of *Pitman Research Notes in Mathematics Series*, pages 258–264. Longman Scientific & Technical, Essex, England, 1992.
- [52] V. S. Vladimirov. *Equations of Mathematical Physics*. Dekker, New-York, 1971.
- [53] Thomas Hagstrom and Timothy Warburton. A new auxiliary variable formulation of high-order local radiation boundary conditions: corner compatibility conditions and extensions to first-order systems. *Wave Motion*, 39(4):327–338, 2004. New computational methods for wave propagation.
- [54] Thomas Hagstrom and Timothy Warburton. Complete radiation boundary conditions: minimizing the long time error growth of local methods. *SIAM J. Numer. Anal.*, 47(5):3678–3704, 2009.
- [55] Thomas Hagstrom, Timothy Warburton, and Dan Givoli. Radiation boundary conditions for time-dependent waves based on complete plane wave expansions. *J. Comput. Appl. Math.*, 234(6):1988–1995, 2010.
- [56] Thomas Hagstrom, Dan Givoli, Daniel Rabinovich, and Jacobo Bielak. The Double Absorbing Boundary method. *J. Comput. Phys.*, 259:220–241, 2014.
- [57] Daniel Baffet, Thomas Hagstrom, and Dan Givoli. Double absorbing boundary formulations for acoustics and elastodynamics. *SIAM J. Sci. Comput.*, 36(3):A1277–A1312, 2014.
- [58] John LaGrone and Thomas Hagstrom. Double absorbing boundaries for finite-difference time-domain electromagnetics. *J. Comput. Phys.*, 2016. Submitted for publication.
- [59] M. Medvinsky, S. Tsynkov, and E. Turkel. The method of difference potentials for the Helmholtz equation using compact high order schemes. *J. Sci. Comput.*, 53(1):150–193, 2012.
- [60] M. Medvinsky, S. Tsynkov, and E. Turkel. High order numerical simulation of the transmission and scattering of waves using the method of difference potentials. *J. Comp. Phys.*, 243:305–322, 2013.
- [61] S. Britt, S. Tsynkov, and E. Turkel. A high order numerical method for the Helmholtz equation with non-standard boundary conditions. *SIAM J. Sci. Comput.*, 35(5):A2255–A2292, 2013.
- [62] S. Britt, S. Petropavlovsky, S. Tsynkov, and E. Turkel. Computation of singular solutions to the Helmholtz equation with high order accuracy. *Applied Numer. Math.*, 93:251–241, July 2015.
- [63] S. V. Tsynkov. On the application of lacunae-based methods to Maxwell’s equations. *J. Comput. Phys.*, 199(1):126–149, September 2004.
- [64] S. V. Petropavlovsky and S. V. Tsynkov. Quasi-lacunae of Maxwell’s equations. *SIAM J. Appl. Math.*, 71(4):1109–1122, 2011.
- [65] S. V. Petropavlovsky and S. V. Tsynkov. A non-deteriorating algorithm for computational electromagnetism based on quasi-lacunae of Maxwell’s equations. *J. Comput. Phys.*, 231:558–585, 2012.

- [66] Robert L. Higdon. Absorbing boundary conditions for difference approximations to the multidimensional wave equation. *Math. Comp.*, 47(176):437–459, 1986.
- [67] Vaughn Betz and R. Mittra. Comparison and evaluation of boundary conditions for the absorption of guided waves in an FDTD simulation. *IEEE Microwave and Guided Wave Letters*, 2(12):499–501, Dec 1992.
- [68] Vaughn Betz and R. Mittra. A boundary condition to absorb both propagating and evanescent waves in a finite-difference time-domain simulation. *IEEE Microwave and Guided Wave Letters*, 3(6):182–184, 1993.
- [69] Stephen D. Gedney. *Introduction to the finite-difference time-domain (FDTD) method for electromagnetics*. Morgan & Claypool Publishers, San Rafael, CA, 2011.
- [70] G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Trans. Electromagnetic Compatibility*, 23:377–382, 1981.
- [71] K. S. Yee. Numerical solution of initial boundary value problem involving Maxwell’s equations in isotropic media. *IEEE Trans. Antennas Propagat.*, 14:302–307, 1966.
- [72] L. Zhao and A. C. Cangellaris. GT-PML: Generalized theory of perfectly matched layers and its application to the reflectionless truncation of finite-difference time-domain grids. *IEEE Trans. Microwave Theory Tech.*, 44:2555–2563, 1996.
- [73] Dan Givoli and Beny Neta. High-order non-reflecting boundary scheme for time-dependent waves. *J. Comput. Phys.*, 186(1):24–46, 2003.
- [74] Thomas Hagstrom, Assaf Mar-Or, and Dan Givoli. High-order local absorbing conditions for the wave equation: Extensions and improvements. *J. Comput. Phys.*, 227(6):3322–3357, 2008.
- [75] Dan Givoli, Thomas Hagstrom, and Igor Patlashenko. Finite element formulation with high-order absorbing boundary conditions for time-dependent waves. *Comput. Methods Appl. Mech. Engrg.*, 195(29-32):3666–3690, 2006.
- [76] S. D. Gedney. An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices. *IEEE Trans. Antennas Propagat.*, 44(12):1630–1639, 1996.
- [77] Allen Taflov and Susan C. Hagness. *Computational Electrodynamics: the Finite-Difference Time-Domain Method*. Artech House, Boston, second edition, 2000.
- [78] J. P. Brooks, K. K. Ghosh, E. Harrigan, D. S. Katz, and A. Taflov. *Progress in Electromagnetics Research*, volume 7, chapter Progress in Cray-Based Algorithms for Computational Electromagnetics”. EMW Publishing, 1993.
- [79] William D. Henshaw. A high-order accurate parallel solver for Maxwell’s equations on overlapping grids. *SIAM J. Sci. Comput.*, 28(5):1730–1765 (electronic), 2006.
- [80] V. S. Ryaben’kii, S. V. Tsynkov, and V. I. Turchaninov. Global discrete artificial boundary conditions for time-dependent wave propagation. *J. Comput. Phys.*, 174(2):712–758, 2001.
- [81] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method*. Prentice Hall, Englewood Cliffs, NJ, 1973.

- [82] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. Reprint of the 1978 original [North-Holland, Amsterdam; MR0520174 (58 #25001)].
- [83] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 2002.
- [84] C. Farhat, I. Harari, and L.P. Franca. The discontinuous enrichment method. *Computer Meth. in Appl. Mech. & Eng.*, 190:6455–6479, 2001.
- [85] C. Farhat, R. Tezaur, and J. Toivanen. A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and Lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 78:1513–1531, 2009.
- [86] Jan S. Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. Algorithms, analysis, and applications.
- [87] Alvin Bayliss, Charles I. Goldstein, and Eli Turkel. An iterative method for the Helmholtz equation. *J. Comput. Phys.*, 49(3):443–457, 1983.
- [88] Ivo M. Babuska and Stefan A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Journal on Numerical Analysis*, 34(6):2392–2423, 1997.
- [89] Frank Ihlenburg. *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences*. Springer, 1998.
- [90] George Biros, Lexing Ying, and Denis Zorin. A fast solver for the Stokes equations with distributed forces in complex geometries. *J. Comput. Phys.*, 193(1):317–348, 2004.
- [91] Shidong Jiang, Shravan Veerapaneni, and Leslie Greengard. Integral equation methods for unsteady Stokes flow in two dimensions. *SIAM J. Sci. Comput.*, 34(4):A2197–A2219, 2012.
- [92] Mark D. Preston, Peter G. Chamberlain, and Simon N. Chandler-Wilde. An integral equation method for a boundary value problem arising in unsteady water wave problems. *J. Integral Equations Appl.*, 20(1):121–152, 2008.
- [93] Tuong Ha-Duong. On retarded potential boundary integral equations and their discretisation. In *Topics in computational wave propagation*, volume 31 of *Lect. Notes Comput. Sci. Eng.*, pages 301–336. Springer, Berlin, 2003.
- [94] Francisco-Javier Sayas. Retarded potentials and time domain boundary integral equations: a road-map. University of Delaware, March 2013.
- [95] Toufic Abboud, Patrick Joly, Jerónimo Rodríguez, and Isabelle Terrasse. Coupling discontinuous Galerkin methods and retarded potentials for transient wave propagation on unbounded domains. *J. Comput. Phys.*, 230(15):5877–5907, 2011.
- [96] Víctor Domínguez and Francisco-Javier Sayas. Some properties of layer potentials and boundary integral operators for the wave equation. *J. Integral Equations Appl.*, 25(2):253–294, 2013.

- [97] Daniel S. Weile, G. Pisharody, Nan-Wei Chen, B. Shanker, and E. Michielssen. A novel scheme for the solution of the time-domain integral equations of electromagnetics. *IEEE Transactions on Antennas and Propagation*, 52(1):283–295, Jan 2004.
- [98] G. Kobidze, Jun Gao, B. Shanker, and E. Michielssen. A fast time domain integral equation based scheme for analyzing scattering from dispersive objects. *IEEE Transactions on Antennas and Propagation*, 53(3):1215–1226, March 2005.
- [99] C. Lubich. Convolution quadrature and discretized operational calculus. I. *Numer. Math.*, 52(2):129–145, 1988.
- [100] C. Lubich. Convolution quadrature and discretized operational calculus. II. *Numer. Math.*, 52(4):413–425, 1988.
- [101] V. S. Ryaben’kii. Boundary equations with projections. *Russian Math. Surveys*, 40(2):147–183, 1985.
- [102] V. S. Ryaben’kii. *Method of Difference Potentials and Its Applications*, volume 30 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2002.
- [103] V. S. Ryaben’kii. Difference potentials method and its applications. *Math. Nachr.*, 177:251–264, 1996.
- [104] V. S. Ryaben’kii. On the method of difference potentials. *J. Sci. Comput.*, 28(2-3):467–478, 2006.
- [105] A. P. Calderon. Boundary-value problems for elliptic equations. In *Proceedings of the Soviet-American Conference on Partial Differential Equations in Novosibirsk*, pages 303–304, Moscow, 1963. Fizmatgiz.
- [106] R. T. Seeley. Singular integrals and boundary value problems. *Amer. J. Math.*, 88:781–809, 1966.
- [107] A. C. Cangellaris and D. B. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in FDTD simulations of electromagnetic phenomena. *IEEE Trans. Antennas and Propagation*, 39(10):1518–1525, 1991.
- [108] R. Holland. Pitfalls of staircase meshing. *IEEE Trans. Electromagnetic Compatibility*, 35:434–439, 1993.
- [109] I. Petrowsky. On the diffusion of waves and the lacunas for hyperbolic equations. *Matematicheskii Sbornik (Recueil Mathématique)*, 17 (59)(3):289–370, 1945.
- [110] V. S. Ryaben’kii, S. V. Tsynkov, and V. I. Turchaninov. Long-time numerical computation of wave-type solutions driven by moving sources. *Appl. Numer. Math.*, 38:187–222, 2001.
- [111] S. V. Tsynkov. Artificial boundary conditions for the numerical simulation of unsteady acoustic waves. *J. Comput. Phys.*, 189(2):626–650, August 2003.
- [112] H. Qasimov and S. Tsynkov. Lacunae based stabilization of PMLs. *J. Comput. Phys.*, 227:7322–7345, 2008.

- [113] R. Courant and D. Hilbert. *Methods of Mathematical Physics. Volume II.* Wiley, New York, 1962.
- [114] A. N. Tikhonov and A. A. Samarskii. *Equations of Mathematical Physics.* Pergamon Press, Oxford, 1963.
- [115] Jean-Pierre Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.
- [116] S. Abarbanel and D. Gottlieb. A mathematical analysis of the PML method. *J. Comput. Phys.*, 134(2):357–363, 1997.
- [117] W. C. Chew and W. H. Weedon. A 3D perfectly matched medium from modified Maxwell’s equations with stretched coordinates. *Microwave and Optical technology Letters*, 7(13):599–604, September 1994.
- [118] Jiayuan Fang and Zhonghua Wu. Generalized perfectly matched layer for the absorption of propagating and evanescent waves in lossless and lossy media. *IEEE Transactions on Microwave Theory and Techniques*, 44(12):2216–2222, Dec 1996.
- [119] Jean-Pierre Bérenger. Evanescent waves in PML’s: origin of the numerical reflection in wave-structure interaction problems. *IEEE Transactions on Antennas and Propagation*, 47(10):1497–1503, Oct 1999.
- [120] Davi Correia and Jian-Ming Jin. Performance of regular PML, CFS-PML, and second-order PML for waveguide problems. *Microwave and Optical Technology Letters*, 48(10):2121–2126, 2006.
- [121] M. Kuzuoglu and R. Mittra. Frequency dependence of the constitutive parameters of causal perfectly matched anisotropic absorbers. *IEEE Microwave and Guided Wave Letters*, 6(12):447–449, 1996.
- [122] J. A. Roden and S. D. Gedney. Convolution PML(CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339, 2000.
- [123] Jean-Pierre Bérenger. Numerical reflection from FDTD-PMLs: a comparison of the split PML with the unsplit and CFS PMLs. *IEEE Transactions on Antennas and Propagation*, 50(3):258–265, Mar 2002.
- [124] Thomas Hagstrom, Manuela L. De Castro, Dan Givoli, and Dina Tzemach. Local high-order absorbing boundary conditions for time-dependent waves in guides. *J. Comput. Acoust.*, 15(1):1–22, 2007.
- [125] E. Bécache and P. Joly. On the analysis of Bérenger’s perfectly matched layers for Maxwell’s equations. *M2AN Math. Model. Numer. Anal.*, 36(1):87–119, 2002.
- [126] Eliane Bécache, Dan Givoli, and Thomas Hagstrom. High-order Absorbing Boundary Conditions for anisotropic and convective wave equations. *J. Comput. Phys.*, 229(4):1099–1129, 2010.
- [127] Andrew D. Greenwood. *Finite-Element Analysis of Complex Axisymmetric Radiating Structures*, IEEE Trans. Antennas Propag., 47(8):1260–1266, 2008.

- [128] Andrew D. Greenwood, Keith L. Cartwright, John W. Luginsland, Ernest A. Baca. *On the elimination of numerical Cerenkov radiation in PIC simulations* J. Comp. Phys. 201:665–684,2004
- [129] Matthew T. Bettencourt and Andrew D. Greenwood. *Performance Improvements for Efficient Electromagnetic Particle-In-Cell Computation on 1000s of CPUs*, IEEE Trans. Antennas Propag., 56(8):2178–2186, 2008.
- [130] M.F. Hadi, M. Picket-May. *A modified (2,4) scheme for modeling electrically large structures with high phase accuracy*, IEEE Trans. Antennas Propagat. 45:254–264,1997.
- [131] C. Mendonca, S. Prasad, E. Schamiloglu, T. Fleming. *3D ICEPIC Simulations of a Pulsed Relativistic Magnetron with Transparent Cathode: A Comparative Study*, Report ADA608936, University of New Mexico, Dept. of Electrical and Computer Engineering Albuquerque, NM, 2012
- [132] Andrey D. Andreev, Kyle J. Hendricks. *Particle-in-Cell (PIC) Simulation of CW Industrial Heating Magnetron*, J. Microwave Power Electrom. Energy, 44(2):114–124, 2010.
- [133] Wynn A.P., Blank D.E., Campbell P.S.,Lentz R.R., Main W.T., Tantawi S.G., et al.(2004). Development of a 300 kW CW L-band industrial heating magnetron, Fifth IEEE International Vacuum Electronics Conference IVEC 2004, April 27-29, 2004, pp.164-165.
- [134] M. J. Al-Hakkak, Y. T. Lo. *Circular waveguides and horns with anisotropic and corrugated boundaries*, Tech. Rep. 73-3, Antenna Lab., Dept. Elect. Eng., Univ. Illinois, Urbana, IL, 1973.
- [135] Boris, J.P. *Relativistic Plasma Simulation-Optimization of a Hybrid Code*, Num. Sim. Plasmas, Naval Research Laboratory, Washington, D.C., 1970, pp. 3–67.
- [136] Okasaki, C. *Purely Functional Data Structures*. Cambridge University Press, 1998.
- [137] Mardahl, Peter; Cartwright, Keith; Greenwood, Andrew; Haworth, Michael; Bowers, Lester; Murphy, Tony; Spencer, Thomas; Gilgenbach, Ronald; Lopez, Mike; Watrous, Jack; Luginsland, John. *Numerical model of the MELBA-C relativistic magnetron using 3D PIC*, American Physical Society, 45th Annual Meeting of the Division of Plasma Physics, October 27-31, 2003, Albuquerque, New Mexico, 2003
- [138] n. nasr esfahani, m. tayarani, and k. schuenemann, Design and 3-D particle-in-cell simulation of a 140 GHz spatial-harmonic magnetron, Progress In Electromagnetics Research, 133:443–458, 2013.
- [139] Hidden agenda: IED detection using ultra-wideband radar. <http://www.army-technology.com/features/featureied-detection-ultra-wideband-microwave-radar-darpa-jieddo>.
- [140] David J. Daniels. UWB radar for the detection of buried ordnance. <http://lib.convdocs.org/docs/index-3591.html>.
- [141] radartutorial.eu: UWB-Radar. <http://www.radartutorial.eu/02.basics/UWB%20Radar.en.html>.
- [142] S. C. Hagness, E. C. Fear, and A. Massa. Guest editorial: Special cluster on microwave medical imaging. *IEEE Antennas and Wireless Propagation Letters*, 11:1592–1597, 2012.

- [143] Anna M. Doro-on. *Risk Assessment and Security for Pipelines, Tunnels, and Underground Rail and Transit Operations*. CRC Press, Boca Raton, FL, 2014.
- [144] Wikipedia: Counter-IED equipment. https://en.wikipedia.org/wiki/Counter-IED_equipment.
- [145] A. S. Lyrantzis. The use of Kirchhoff's method in computational aeroacoustics. *J. Fluids Eng.*, 116(4):665–676, 1994.
- [146] S. N. Vlasov and I. M. Orlova, Quasioptical transformer which transforms the waves in a waveguide having a circular cross section into a highly directional wave beam, *Radiophysics and Quantum Electronics*, vol. 17, no. 1, pp. 115–119, 1974.
- [147] S. N. Vlasov, L. I. Zagryadskaya, and M. I. Petelin, Transformation of a whispering gallery mode, propagating in a circular waveguide, into a beam of waves,” *Radio Engineering and Electronic Physics*, vol. 20, no. 10, pp. 14-17, 1975.
- [148] H. M. El Misilmani, M. Al-Husseini, and K. Y. Kabalan, Improved Vlasov Antenna with Curved Cuts and Optimized Reflector Position and Shape, *International Journal of Antennas and Propagation*, Vol. 2015 (2015), Article ID 193630.
- [149] G. S. Ling and C. W. Yuan, ”Design of a Vlasov antenna with reflector,” *International Journal of Electronics*, vol. 91, no. 4, pp. 253-258, 2004.
- [150] H. Zhou and X. Yang, ”Design of novel VLASOV-type antennas for high power microwaves (HPM),” in *Proceedings of the IEEE 34th International Conference on Plasma Science (ICOPS '07)*, p. 912, Albuquerque, NM, USA, June 2007.
- [151] O. Wada and M. Nakajima, in *Proceedings of the EC6 Joint Workshop on ECE and ECRH*, Oxford, UK, September 1987.
- [152] B. Wang, C.-H. Du, P.-K. Liu, Z.-H. Geng, and S.-X. Xu, ”P3-12: study of a Vlasov mode converter for 94GHz whispering gallery mode gyrotron,” in *Proceedings of the IEEE International Vacuum Electronics Conference (IVEC '10)*, pp. 353-354, IEEE, Monterey, Calif, USA, May 2010.
- [153] Pindera, M.Z., Malosse, J-J, Sun, Y., Vosen, S.R., and Goryca, M.
Multi-Resolution Rapid Prototyping Of Vehicle Cooling Systems: Approach And Test Results, *Proceedings of the 6th Annual Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, Novi, MI, Aug. 12-14, 2014.
- [154] Pindera, M.Z., Sun, Y., Malosse, J-J, and Vosen, S.R., ”Multi-Resolution Rapid Prototyping of Vehicle Cooling Systems” *Proceedings of the 2014 IEEE Aero Conference At the Yellowstone Conference Center in Big Sky, Montana*, Mar 1 - 8, 2014.
- [155] Sengupta, D., Pindera, M.Z., Cole, J.V., and Vaghjiani, G.L. QSPR And Artificial Neural Network Predictions Of Hypergolic Ignition Delays For Energetic Ionic Liquids, *57th JANNAF Joint Subcommittee Meeting*, Springs, CO, 3-7 May 2010.
- [156] Pindera, M.Z. Multi-Resolution Prototyping Of Li-Ion Batteries. *2008 Ground Automotive Power & Energy Workshop*, National Defense Industrial Association of Michigan November 17-19, 2008.

- [157] Pindera, M.Z., Adaptive Flow Control Using Simple Artificial Neural Networks, 40th American Institute of Astronautics and Aeronautics (AIAA) Aerospace Sciences Meeting and Exhibit; 6-9 Jan 2002; Reno, NV. AIAA-Paper2002-0990
- [158] Traian Dogaru, Anders Sullivan, Calvin Le and Chris Kenyon. Radar Signature Prediction for Sensing-Through-the-Wall by Xpatch and AFDTD - Part II, High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC) 2010 DoD, pp. 401-406, 2010.
- [159] Traian Dogaru, Calvin Le. Through-the-Wall Radar Simulations for Complex Room Imaging, ARL-TR-5205, May 2010
- [160] Traian Dogaru, Calvin Le, and Getachew Kirose. Time-Frequency Analysis of a Moving Human Doppler Signature, ARL-TR-4728, February 2009.
- [161] A. Fedoseyev, E. J. Kansa, S. Tsynkov, S. Petropavlovskiy, M. Osintcev, U. Shumlak and W. D. Henshaw, AIP Conf. Proc. 1773, 020001 (2016); [dx.doi.org/10.1063/1.4964955](https://doi.org/10.1063/1.4964955)

APPENDIX A

Overture/cgmx installation and running

A1. Overture/Cgmx-Maxwell solver installation and running

The latest version of the LLNL Overture code has been installed into the ComScis computers, compiled and tested. The compilation and installation of the has been successfully completed. The list and values of the environmental variables to compile, install and run Overture in Ubuntu Linux within VirtualBox are provided in the Appendix B.

A2. Overture and Cgmx Maxwell solver testing

We have generated and ran the test simulation of scattering of a plane wave from a two-dimensional conducting cylinder.

The command file cic.planeWaveBC.cmd was used to compute the scattering of a plane wave from a PEC (perfectly electric conducting) cylinder. The exact solution is known for this case and the errors in the numerical solution are determined while the solution is being computed.

Solution steps:

(1) Generate the grid using the command file cicArg.cmd in Overture/sampleGrids:

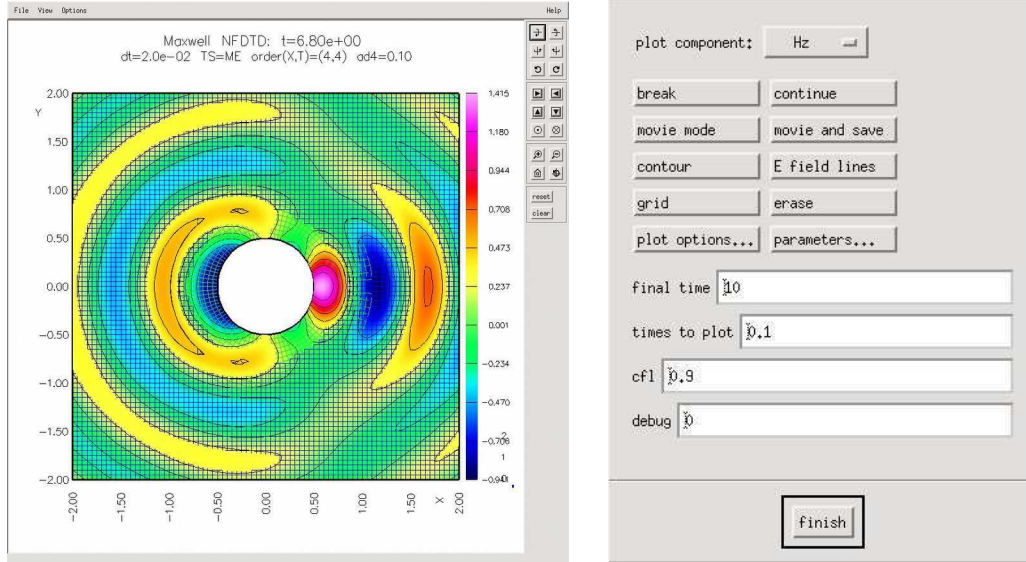
```
$ ogen -noplot cicArg -order=4 -interp=e -factor=2
```

(2) Run cgmx:

```
$ cgmx cic.planeWaveBC -g=cice2.order4.hdf
```

This case was run with fourth-order accuracy.

The figures below show a grid plotted over solution and the simple GUI used in Overture to plot the results during the simulations,



A comparison of our results with the original ones by Dr. Henshaw are shown in the figures below.

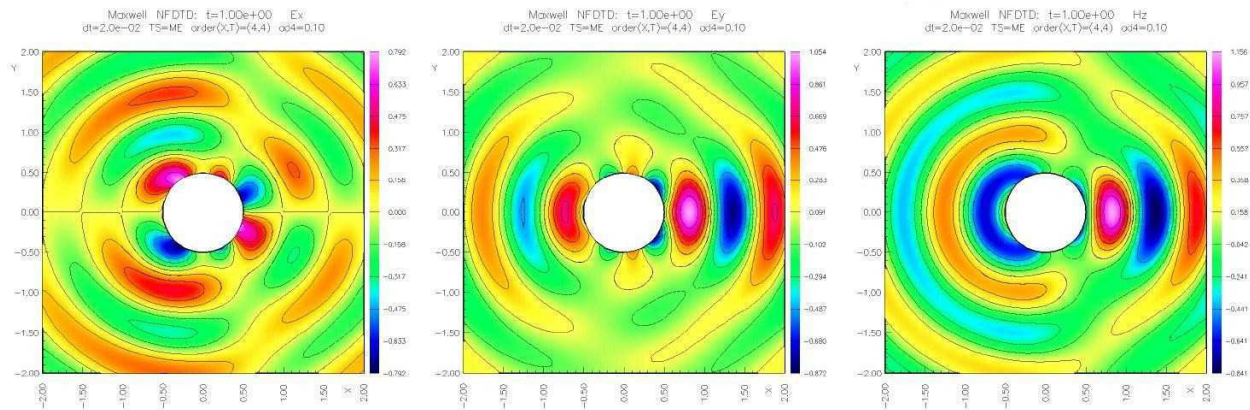
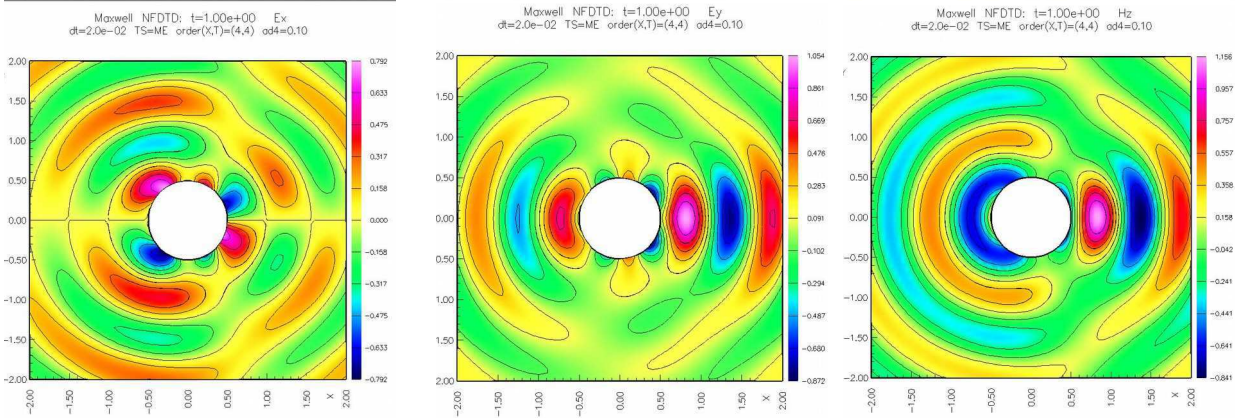
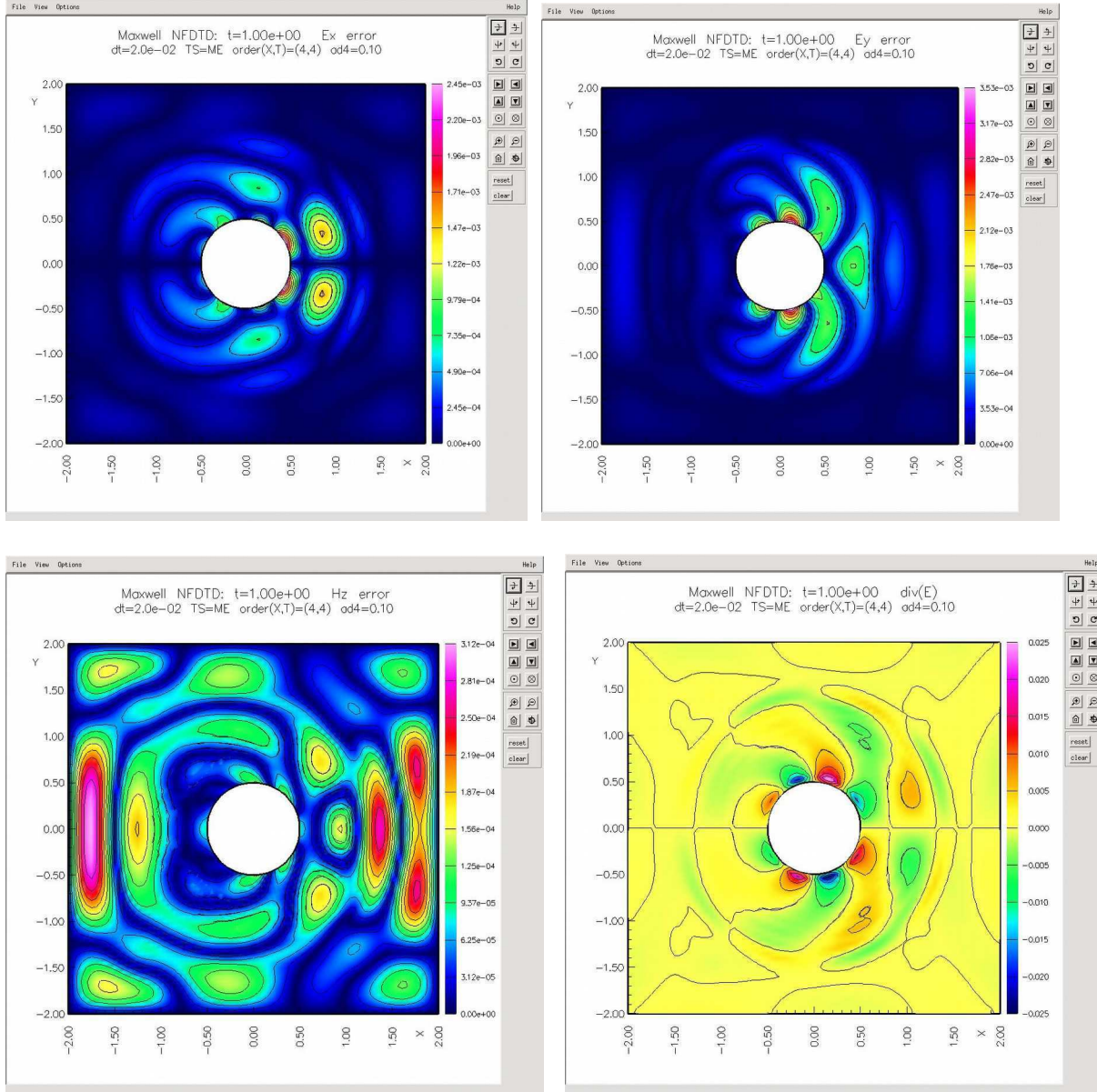


Figure 1: Scattering of a plane wave by a PEC cylinder. Computed solution at $t = 1.0$, E_x , E_y and H_z .



One can observe an excellent agreement in comparison of solutions at time $T=1.00$ for E_x , E_y and H_z for a PEC test case. Our results are at the bottom figure row, and the original ones by Dr. Henshaw are at top figure row.

The plot of solution errors for time $T=1.00$ for E_x , E_y and H_z for a PEC test case is presented below. The numerical solution error corresponds to the 4-th order scheme for this problem.



The plots of solution errors for time $T=1.00$ for E_x , E_y and H_z , and $div(E)$ for a PEC test case is presented below.

REFERENCES

1. W. D. Henshaw, SIAM J. Sci. Comput. 28, 1730 (2006).

Appendix B

B1. Environmental variables to compile, install and run Overture in the Ubuntu 12.04 LTS Linux within VirtualBox

```
export XLIBS=/usr
export OpenGL=/home/cs/over2/Mesa-7.6.1/
export MOTIF=/usr/
export HDF=/home/cs/over2/hdf5-1.8.8
export APlusPlus=/home/cs/over2/A++P++-0.8.1/A++/install
export Overture=/home/cs/over2/Overture.v25d
export CG=/home/cs/over2/cg.v25d
export CGBUILDPREFIX=/home/cs/over2/cg.v25d/bin
export LAPACK=/home/cs/over2/lapack-3.3.1

export LD_LIBRARY_PATH=/home/cs/over2/hdf5-1.8.8/lib64/:${MOTIF}:${
{XLIBS}}:${HDF}/lib:${OpenGL}:${Overture}/lib:${
{APlusPlus}}/lib:/home/cs/over2/A++P++-0.8.1/A+
+/install/lib64:/usr/lib/x86_64-linux-gnu:/home/cs/over2/lapack-
3.3.1/

$ echo $PATH
//home/cs/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/games:/opt/vagrant_ruby/bin:/home/cs/over2/Overture.v
25d/bin:/home/cs/over2/cg.v25d/bin/mx/bin
```

B2. Overture

An Object-Oriented Toolkit for Solving Partial Differential Equations in Complex Geometry

HOME
DOCUMENTATION

OVERVIEW
DOCUMENTATION

RESEARCH GROUP
PRESENTATIONS

OGEN
PUBLICATIONS

CGINS
DOWNLOAD

CGCNS
DOWNLOAD

CGMX
CONTACT

CGSM

CGMP

Download Overture Software

Disclaimer: this software comes with absolutely no warranty.

Latest releases

1. [Overture.v25.tar.gz](#) Latest version of Overture.
2. [cg.v25.tar.gz](#) The CG PDE solvers. Requires the same version of Overture to be installed.
3. [AP-0.8.1.tar.gz](#). The A++/P++ array class library for serial and parallel arrays. This version should compile with newer versions of the C++ compilers.

Alpha releases - use at your own risk

1. [Overture.v25k.tar.gz](#) Latest alpha release of Overture. Use this version for Mac's running Mavericks.
2. [cg.v25k.tar.gz](#) The CG PDE solvers. Requires the same version of Overture to be installed.
3. [Overture.v25h.tar.gz](#) Alpha release of Overture.
4. [cg.v25h.tar.gz](#) The CG PDE solvers. Requires the same version of Overture to be installed.
5. [AP-0.8.2.tar.gz](#). The A++/P++ array class library for serial and parallel arrays. This version should compile with newer versions of the C++ compilers. Use this version for Mac's running Mavericks.

Software dependencies

Here are versions of the software that you may need to install if you don't already have them on your system.

1. [hdf5-1.8.8.tar.gz](#). A version of HDF that works with the latest version of Overture. NOTE: newer versions may be incompatible.
2. [MesaDemos-7.2.tar.gz](#), [MesaLib-7.2.tar.gz](#) Version of Mesa (OpenGL) that works with the latest version of Overture. NOTE: newer versions may be incompatible.
3. [petsc-lite-2.3.2-p6.tar.gz](#). A version of PETSc that works with the latest version of Overture. PETSc is recommended if you need to solve linear equations (e.g. for Cgins). NOTE: newer versions may be incompatible.
4. [lapack.tgz](#) the LAPACK library of linear algebra routines (needed for the CG PDE solvers).

Past releases

1. [Overture.v24.tar.gz](#) Latest version of Overture.
2. [cg.v24.tar.gz](#) The CG PDE solvers. Requires the same version of Overture to be installed.
3. [Overture.v23.tar.gz](#). Overture version 23.
4. [cg.v23.tar.gz](#). CG version 23.
5. [AP-0.8.0.tar.gz](#). The A++/P++ array class library for serial and parallel arrays.
6. [AP-0.7.9d.wdh090307.tar.gz](#). The A++/P++ array class library.

Installation Instructions

Here are installation instructions for Overture and CG [install.pdf](#).

Here are instructions for installing Overture on Ubuntu and also for installing on a Windows machine using VirtualBox [Installing Overture on Ubuntu 10 04 LTS.pdf](#) (Thanks to Chris Jarvis and Mohammad Ahoiali).

[Missing Overture on Search Engine Expert](#) (Thanks to Chris Davis and Mohammad Hossain).

Overture mailing list.

You can join the **Overture mailing list** by going to <http://www.freelists.org/list/overture>. Then post mail to overture@freelists.org. Use this forum to ask questions about Overture.

The searchable list archive of previous posts to the overture mailing list is found at

<http://www.freelists.org/archives/overture> (you don't need to join the mailing list to view the archive).

The old Overture mailing list (until September 2007) was at <http://www.topica.com/lists/overture>. You can go here if you want to see old posts and old questions/answers.

```
Counter datafile "/var/cpanel/Counters/overture2.dat" display=Counter%7Cft=6%7Cid=10%7Cfrgb=100;139;216%7Cdd=E" must be created in cPanel first!
```

[Home](#) [Overview](#) [Research group](#) [Ogen](#) [Cgins](#) [Cgcns](#) [CgmX](#) [Cgsm](#) [Cgmp](#) [Documentation](#) [Presentations](#)
[Publications](#) [Download](#) [Contact](#) [Valid CSS](#) [Valid HTML](#)

B3.Overture Installation Instructions

W.D. Henshaw,
Department of Mathematical Sciences,
Rensselaer Polytechnic Institute, Troy NY, 12180, USA.

Overture

Abstract

This document contains step-by-step instructions on how to install Overture and the supporting packages.

Contents

1	Machines and Compilers where Overture has been built	2
2	Installing Open Motif	2
3	Installing Mesa OpenGL	3
4	Installing HDF	4
5	Installing A++	5
6	Installing P++	6
7	Installing PETSc (optional)	8
8	Installing Overture	9
8.1	More comments on installing Overture	10
9	General comments	11
10	Installing Overture on Mac OS/X	12
10.1	OS X 10.9 - Mavericks	12
10.2	Installing Overture on Mac OS/X – *OLDER VERSIONS*	13
11	Installing Overture on the Blue Gene	14
11.1	Installing parallel HDF	14
11.2	Installing P++	14
11.3	Installing Overture	14
12	Installing LAPACK	15
13	Installing CG	16
14	Notes for running Overture in parallel	17

1 Machines and Compilers where Overture has been built

Platform (OS)	C++/C	Fortran
Intel/AMD (Redhat) (32/64 bit)	gcc 4.3.2	gfortran/pgf77/ifort
Mac OS X	gcc 4.3.2	gfortran/xlf

Table 1: Overture has been built on these machines with these compilers. Notes: pgf77 is the fortran compiler from PGI, ifort is the intel fortran compiler.

Our current development platform is Linux-Intel 64-bit with gcc and pgf77.

2 Installing Open Motif

Note: Mac OS/X users should skip ahead to Section [10](#).

You may need to install Motif if you do not already have a version. Most Linux systems now come with Motif in `/usr/lib/libXm.so` (or `/usr/lib64/libXm.so` for 64 bit machines) in which case you can skip this step.

The Motif library is called `libXm.a` or `libXm.so`. In older versions of Linux the default version of Motif is probably LessTif. I have had trouble with LessTif so you should install open Motif which you can get for free from [open Motif](#). I have tried version 2.2.1. There is an rpm available with precompiled binaries. Here are the steps I took to install open Motif from the source files:

1. `gzip -d openMotif-2.2.1.tar.gz`
2. `tar -xf openMotif-2.2.1.tar`
3. `cd openMotif-2.2.1`
4. `setenv CC gcc` : set CC to your favourite C compiler, usually gcc on linux
5. `./configure --prefix="this directory"` : where "this directory" is the current directory (or the place you want to install Motif).
6. `make`
7. `make install`

3 Installing Mesa OpenGL

You may need to install **Mesa** (OpenGL). Most Linux systems should already have OpenGL. However, if you would like off-screen rendering support (for making high-resolution hard copies) then you should install Mesa. I normally run with Mesa rather than the native OpenGL.

I currently use version Mesa-7.2.

Assuming you have downloaded `MesaDemos-XXX.tar.gz` and `MesaLib-XXX.tar.gz` in directory MESA (e.g. `/home/henshaw/Mesa`)

1. `tar xzf MesaLib-XXX.tar`
2. `tar xzf MesaDemos-XXX.tar.gz` (optional)
3. `tar xzf MesaGLUT-XXX.tar.gz` (optional)
4. `cd Mesa-XXX`
5. `make` (to see a list of available machines)
6. `make <machine-name>` where `<machine-name>` is the appropriate choice, such as `linux-x86` (`linux-x86-64`). I usually use dynamic libraries.
7. Goto to one of the example directories (such as `demos`) and run a test. If you have built dynamic libraries then you will first need to set your `LD_LIBRARY_PATH` as indicated by the message from Mesa when it finished building.
8. In previous versions of Overture one needed to build the `libGLw` library in the `cd widgets-sgi` directory. This is no longer necessary since these widgets will be built with Overture.
9. Mesa has off-screen rendering support (for creating hard-copy's) found in `libOSMesa.a` which should automatically be built with Mesa. Overture also needs access to some of the source files in Mesa (such as `Mesa-XXX/src/mesa/main/context.h`) so your OpenGL environmental variable should point to the Mesa source (as opposed to `/usr/lib`) if you want this off screen rendering. Overture can also do off-screen rendering using X pixmaps, but this may not work if you have a graphics card that does direct rendering.
10. You may increase the maximum possible resolution that an image can be saved as. Mesa is compiled with certain default maximum values and the highest resolution depends on the parameters that were used to compile Mesa. To increase the Mesa resolution you should change the definitions for `MAX_WIDTH` and `MAX_HEIGHT` in `Mesa/src/config.h`. For example you may set these to be 2048 or more. Mesa must be re-compiled after these changes are made.

4 Installing HDF

HDF (<http://www.hdfgroup.org/HDF5/>) is a data base package that you may need to install. It may already be installed on your system.

Here are the installation steps for HDF5. I currently use hdf5-1.8.8 (1.6.5 should also work). You will need to build a serial version and/or parallel version depending on which version of Overture you are building.

1. `tar xzf hdf5-1.8.8.tar.gz`
2. optionally rename the directory `hdf5xxx`
3. `cd hdf5xxx`
4. `unsetenv CC`
5. `unsetenv cc`
6. For the serial version : `setenv CC gcc` (if you want to use gcc).
7. For the parallel version : `setenv CC $MPI_ROOT/bin/mpicc`.
8. `./configure --prefix='pwd'`
9. Sometimes in parallel I instead need to use `setenv CC gcc` and then `./configure --prefix='pwd' --enable-parallel`.
10. `make`
11. `make install`

5 Installing A++

A++/P++ is the serial/parallel array class. To get A++/P++ you will need to go to the [Overture home page](#) and register under software.

The A++ installation process requires the gnu autoconf package and gmake, available from the gnu site <http://www.gnu.org/order/ftp.html>.

If you have downloaded AP-nnn.tar.gz into directory XXX then the basic steps are as follows (if something goes wrong, consult the documentation that comes with A++).

1. `cd XXX`
2. `gzip -d AP-nnn.tar.gz`
3. `tar -xf AP-nnn.tar`
4. `cd A++P++nnn`
5. `configure --enable-SHARED_LIBS --prefix='pwd'`
6. `make`
7. `make install`
8. Optionally type `make check` to run the A++ test codes.

The libraries will be installed in `XXX/A++P++nnn/A++/install/lib`. You should use `setenv APlusPlus XXX/A++P++nnn/A++/install` when building Overture.

6 Installing P++

P++ is the parallel version of the array class. NOTE: many but not all features of Overture run in parallel. **Note** that we build the version of P++ without the newer PADRE implementation.

To build the P++ library (only required if you are going to build the parallel version of Overture)

1. `cd A++P++nnn` (goto the A++P++ directory)
2. Set the environment variable `MPI_ROOT` to point to the mpi directory such as `setenv MPI_ROOT /usr/local/mpi`
3. Here is the configure statement I used to compile with gcc version 4.3.2 on my dual processor workstation

```
configure --enable-PXX --prefix='pwd' --enable-SHARED_LIBS \
--with-mpi-include=-I${MPI_ROOT}/include \
--with-mpi-lib-dirs="-Wl,-rpath,/usr/apps/gcc/4.3.2/lib -L${MPI_ROOT}/lib" \
--with-mpi-libs="-lmpich -lmpich -lc -lm" \
--with-mpirun=${MPI_ROOT}/bin/mpirun --without-PADRE
```

4. Here is a configure statement I used on a Linux cluster that did not use mpirun to start parallel jobs. P++ expects to find mpirun by default but you can override this and provide another script. In the example below I use `mpirun-wdh`, a script that I wrote which calls the appropriate routine to launch parallel jobs.

```
configure --enable-PXX --prefix='pwd' --enable-SHARED_LIBS \
--with-mpi-include=-I${MPI_ROOT}/include \
--with-mpi-lib-dirs="-Wl,-rpath,${MPI_ROOT}/lib -L${MPI_ROOT}/lib" \
--with-mpi-libs="-lmpi++ -lmpi" \
--with-mpirun=${HOME}/bin/mpirun-wdh --without-PADRE
```

5. Here is the configure statement I used when I couldn't run mpirun on a front end machine so I disabled the checks that parallel jobs can be executed:

```
configure --enable-PXX --prefix='pwd' --enable-SHARED_LIBS \
--with-mpi-include=-I${MPI_ROOT}/include \
--with-mpi-lib-dirs="-L${MPI_ROOT}/lib" \
--with-mpi-libs="-lmpi" \
--disable-mpirun-check \
--without-PADRE
```

Here is the configure statement Robert Read used to install P++ on his laptop running fedora and Open MPI

```
configure --enable-PXX --prefix='pwd' --enable-SHARED_LIBS \
--with-mpi-include=-I/usr/include/openmpi-x86_64 \
--with-mpi-lib-dirs="-Wl,-rpath,${MPI_ROOT}/lib -L${MPI_ROOT}/lib" \
--with-mpi-libs="-lmpi -lmpi_cxx -lc -lm" \
--with-mpirun=${MPI_ROOT}/bin/mpirun --without-PADRE
```

6. `make`
7. `make install`
8. Optionally type `make check` to run the P++ test codes.

The libraries will be installed in `XXX/A++P++nnn/P++/install/lib`. You should use `setenv PPlusPlus XXX/A++P++nnn/P++/install` and `setenv APlusPlus XXX/A++P++nnn/P++/install` when building the parallel version of Overture.

Note: To change the maximum number of parallel processes that can be used by P++, change the file `A++P++nnn/P++/include/maxProcessors.h` before building the P++ library.

Here are more example builds of P++:

- To build on a Linux cluster at LLNL I first ran `mpic++` on a C++ file with the “-v” option to see all the libraries that were being linked to:

```
mpic++ -v -o conftest -I/usr/local/tools/mvapich2-gnu/include conftest.C
```

I then configured P++ using these libraries:

```
configure --enable-PXX --prefix='pwd' --enable-SHARED_LIBS \
--with-mpi-include=-I${MPI_ROOT}/include \
--with-mpi-lib-dirs="-Wl,-rpath,${MPI_ROOT}/lib -L${MPI_ROOT}/lib" \
--with-mpi-libs="-lmpichcxx -lmpich -lpmi -lopa -lmpl -lpthread -lrdmacm \
                -libverbs -libumad -ldl -lrt -lnuma -lstdc++ -lm" \
--with-mpirun=${HOME}/bin/mpirun-wdh --without-PADRE
```

7 Installing PETSc (optional)

Overture can optionally use PETSc to solve sparse matrix problems. You will probably want to get PETSc if you want to solve the 3D incompressible Navier-Stokes equations with cgins.

The version of PETSc that works with Overture.v25 or v22 is PETSc 2.3.2[petsc-lite-2.3.2-p6.tar.gz](#). Here are the steps I took to install the (uni-processor version) on my Linux machine where I use the gcc and PGI fortran compilers:

1. `tar xzf petsc-lite-2.3.2-p6.tar.gz`
2. `mv petsc-2.3.2-p6 petsc-2.3.2`
3. `cd petsc-2.3.2`
4. `setenv PETSC_DIR `pwd``
5. `setenv PETSC_ARCH linux-gnu-opt`
6. `setenv PETSC_LIB $PETSC_DIR/lib/$PETSC_ARCH`
7. `setenv CC gcc`
8. (for help type: `./config/configure.py --help`)
9. `./config/configure.py --PETSC_ARCH=linux-gnu-opt --with-debugging=0 --with-fortran=0 --with-matlab=0 --with-mpi=0 --with-shared=1 --with-dynamic=1`
10. `make`

Notes for Mac: Follow the above instructions but use:

1. `setenv PETSC_ARCH macx`
2. `./config/configure.py --PETSC_ARCH=linux-gnu-opt --with-debugging=0 --with-fortran=0 --with-matlab=0 --with-mpi=0 --with-shared=0 --with-dynamic=0`

8 Installing Overture

Here are instructions to install the Overture library. See also the README file that comes with Overture. To get Overture you will need to go to the [Overture home page](#) and register under software.

Overture requires perl for configuration and running tests. If you don't have perl you can get it for free from <http://www.perl.com>

Overture needs a collection of environment variables to be set. These variables tell Overture where to look for other libraries. You can edit the file **Overture/defenv** to define these variables and then type “source defenv” to actually set the variables. You might want to add them to your .cshrc so they get assigned when you open a window (shell).

- **setenv XLIBS xxxx** : Directory holding the **include** and **lib** directories for the X window system. For example you might type “**setenv XLIBS /usr**”. If you then type “**ls \$XLIBS/lib**” you should see the X libraries such as **libX11.so** or **libX11.a**
- **setenv OpenGL xxxx** : If you type “**ls \$OpenGL/lib**” you should see the OpenGL libraries such as **libGL.so** (**libGL.a**), or **libMesaGL.so** (**libMesaGL.a**) if you use Mesa.
- **setenv MOTIF xxxx** : If you type “**ls \$MOTIF/lib**” you should see the Motif library **libXm.so** or **libXm.a**
- **setenv HDF xxx** : If you type “**ls \$HDF**” you should see the HDF directories including **include** and **lib**. If the HDF variable contains the sub-string “hdf5” in it, then the Overture configure script will know to use HDF5 (otherwise you will need to use the configure option useHDF5).
- **setenv APlusPlus xxxx** : if you type “**ls \$APlusPlus**” you should see the A++ directories **include** and **lib**; if you type “**ls \$APlusPlus/lib**” you should see the A++ libraries and if you type “**ls \$APlusPlus/include**” you should see the A++ include files. Here is an example: **setenv APlusPlus /home/dilbert/A++P++/A++/install**.
- **setenv Overture xxxx** : location of the Overture directory that we are just going to install. Here is an example : **Overture /home/henshaw/Overture.v23.d**.
- **setenv CG xxxx** : If you install the CG solvers this is the location of the CG source directory. Here is an example : **setenv CG /home/henshaw/cg.v23**.
- **setenv CGBUILDPREFIX xxxx** : This is the location of the CG executable and object files. You can make this the same as **CG** or choose a different directory location. This option is used to compile multiple versions of CG (e.g. serial and parallel) from the same source tree. Here is an example : **setenv CGBUILDPREFIX /home/henshaw/cg.v23.d**
- **setenv LAPACK xxx** : Overture and CG need the LAPACK and BLAS libraries. This env variable is the directory that holds the **liblapack.a** and **libblas.a** libraries. Example: **setenv LAPACK /home/henshaw/lapack**
- Set the **LD_LIBRARY_PATH** environment variable which is used by the loader when you run an Overture program in order to find dynamic libraries (such as **libX11.so** or **libOverture.so**):

```
setenv LD_LIBRARY_PATH ${MOTIF}/lib:${XLIBS}/lib:${HDF}/lib:${OpenGL}/lib:$Overture/lib:${APlusPlus}/lib
```

For machines with the “-rpath” like load flag, it may not be necessary to set the **LD_LIBRARY_PATH** since this information is stored with the executable.

Here now are the steps for installation:

1. **tar xzf Overture.xxx.tar**
2. **cd Overture.xxx**

3. **configure** : (or **configure [option1] [option2] ...**) (or type **configure --help**). If perl is in a funny place then type **perl configure**. The default is to build a debug version with double precision (recommended).
4. **make** : this will build Overture, takes about 1/2 hour for the default options, a couple of hours for an optimized version. You can also type “**make -j 2**” on machines with multiple processors in order to perform a parallel make.

If you would like to **test the distribution** type

check.p

from the main Overture directory to build a collection of grids and run a set of regression tests. You can also run these test separately (as well as other tests) by:

1. **cd sampleGrids** : goto the sample grids directory
2. **make** : will build a set of overlapping grids and check to see if they are correctly built. This is a good test that Overture is installed and working.
3. **cd ../primer**
4. **make** : to make the primer examples. Now run some of the primer examples, such as **mappedGridExample1** or **mappedGridExample2** etc.
5. **cd ../tests** : go to the tests directory which holds other tests
6. **make** : make the test routines. Type **checkop.p** to run a perl script to test the operators.
7. There are other demos and tests in the **sampleGrids**, **sampleMappings** and **examples** directories.

8.1 More comments on installing Overture

Overture uses the perl script **configure** to build configuration files. Type

configure --help

to see the configure options. If perl is not installed where I think it should be then you may have to type

perl configure --help

The default options are “precision=double” for double-precision and “debug” for compiling with debug information so that you can diagnose problems with a debugger. In most cases performance is not seriously degraded by the “debug” option although some programs such as the grid generator “ogen” will run quite a bit faster if you configure with the “opt” option for optimization:

configure opt

Generally the configure script will automatically determine the machine type. If not you may have to type

configure <machine-name> [options]

The configure script will read the file **config/Makedefs.<machine-name>** and build Makefile’s in each directory. If you need to change the Makefile’s for your setup then you should usually change the **config/Makedefs.xxx** file and re-configure, rather than changing all the Makefile’s.

9 General comments

Here are some steps that I would take to track down bugs.

1. Do the A++ test codes run?
2. If you are having trouble with the graphics try building the **paperplane** example in the **Overture/tests** directory. This example tests whether OpenGL and Motif are working (and does not even link to any Overture libraries).
3. The perl script **Overture/check.p** will run a series of regression tests.
4. Some Overture test codes are in the directory **Overture/tests**. The perl script **Overture/tests/checkop.p** will run regression tests on the operators (these tests are also run by **Overture/check.p**).
5. Use gdb or your favourite debugger to see what the problem is. If you don't know how to use a debugger you've come to the wrong party.

10 Installing Overture on Mac OS/X

10.1 OS X 10.9 - Mavericks

Stage I: The default compiler for Mavericks is *clang* which replaces gcc and g++. DO NOT install gcc.

1. Install Xcode from the apple web site.
2. Install the command line tools in Xcode:

```
xcode-select --install
```

This command will ask you whether you want to install the command lines tools. You will know if you have the command lines tools if you can issue the command `gcc --version`.

3. Install gfortran from <http://gcc.gnu.org/wiki/GFortranBinaries#MacOS>
4. Install XQuartz (if you do not already have it). This should install X11 libraries in `/opt/X11/lib`. This is needed so you can pop up X windows but since there is no Motif here we will need to install another version using Macports.
5. Install Macports
6. Use Macports to install Open Motif and OpenGL:

```
sudo port install openmotif
sudo port install mesa
```

This will install another version of X11 in `/opt/local/lib` in addition to the Open Motif and Mesa libraries.

Stage II:

1. Install HDF following the normal instructions.
2. Install A++ following the normal instructions but, you may need to make a symbolic link for gmake (somewhere in your path)

```
ln -s /usr/bin/make /home/dilbert/bin/gmake
```

and configure with

```
./configure --disable-SHARED_LIBS --prefix='pwd'
```

3. Install Overture and cg following the regular instructions and using

```
setenv XLIBS /opt/local
setenv MOTIF /opt/local
setenv OpenGL /opt/local
```

as the locations for X11, Motif and OpenGL (do NOT accidently link to `/opt/X11`).

10.2 Installing Overture on Mac OS/X – *OLDER VERSIONS*

Overture and cg have been build on Mac OS/X. Thanks to Kyle for working this out. The installation process is very similar to the process for Linux after some initial steps to install the appropriate compilers etc..

Stage I:

1. Install and update the Apple developer tools (an extra DVD that came with the mac).
2. Install macports, see, <http://www.macports.org/>
3. Using macports, install hdf, mesa, glw, petsc, etc using:

(a) `port install gcc4.3 hdf5 petsc mesa glw motif`

Here we have installed version 4.3 of gcc.

4. It is convenient to make soft links for the 4.3 compilers that macports installs. Go to the MacPorts bin directory and make links like

(a) `ln -s gcc-mp-4.3 gcc`

Do this for gcc, g++ and gfortran.

Stage II: Install A++P++:

1. Download A++P++, Overture and cg from the Overture web site.
 - (a) Unpack A++: `tar xzf AP-nnn.tar.gz`
 - (b) Rename A++ directory if desired: `mv A++P++-nnn A++P++-nnn-gcc4.3`
 - (c) `setenv CC gcc; setenv CXX g++; setenv FC gfortran`
 - (d) `./configure --with-CC="$CC -I/usr/include/malloc -mlongcall" --with-CXX="$CXX -I/usr/include/malloc -mlongcall" --disable-SHARED_LIBS --prefix='pwd'`
 - (e) `make MAKE=make install`
 - (f) `setenv APlusPlus 'pwd'/A++/install; setenv PPlusPlus $APlusPlus`

Stage III: Install Overture and CG: follow the instructions in Sections 8 and 13 for installing Overture and CG.

Notes:

1. You should add the Overture/lib directory to your dynamic library path: `setenv DYLD_LIBRARY_PATH $DYLD_LIBRARY_PATH:$Overture/lib`

11 Installing Overture on the Blue Gene

Here are notes on installing Overture on a Blue Gene/Q at RPI.

I used the gnu compilers:

```
module load gnu
```

11.1 Installing parallel HDF

Set RUNSERIAL to be the command to run a program on the Blue Gene.

```
tar -xzf hdf5-1.8.8.tar.gz
mv hdf5-1.8.8 hdf5-1.8.8-parallel
cd hdf5-1.8.8-parallel
```

```
unsetenv CC
unsetenv cc
```

```
./bin/yodconfigure configure
```

```
setenv RUNSERIAL "srun -N1 -n1 --time 1 --partition debug"
setenv LDFLAGS "-dynamic"
```

```
configure --prefix='pwd' --enable-parallel
```

```
make
make install
```

11.2 Installing P++

```
tar xzf AP-0.8.2.tar.gz
mv A++P++-0.8.2 A++P++-0.8.2-gcc.4.4.6-parallel
```

```
./configure --enable-PXX --enable-SHARED_LIBS --prefix='pwd' \
            --with-CC=mpicc --with-CXX=mpicxx --disable-mpirun-check --without-PADRE
```

11.3 Installing Overture

Overture was built without graphics and without support for Perl.

```
configure bg parallel CC=mpicxx bCC=g++ cc=mpicc bcc=gcc FC=mpif90 bFC=gfortran\
            --disable-X11 --disable-perl --disable-gl
```

12 Installing LAPACK

CG needs the LAPACK and BLAS libraries. If you need to build these libraries you should find a version `lapack.tgz` from the web.

Here are the steps for installation:

1. `tar xzf lapack.tgz` : Untar the lapack library.
2. `cd LAPACK`
3. Copy the appropriate `make.inc.XXX` file from the `INSTALL` directory into `make.inc` in the main directory. Example: `cp INSTALL/make.inc.LINUX make.inc`
4. Edit the `make.inc` file and define the compiler and compiler options.
5. Type `make blaslib` to make BLAS.
6. Type `make lapacklib` to make LAPACK.
7. CG expects the libraries to be named `liblapack.a` and `libblas.a` so you may need to rename the libraries that are built in the main directory.

13 Installing CG

Here are instructions to install the CG solvers. CG uses the same environmental variables as Overture. There is no need to configure CG since it finds all the information it needs from Overture. See the cg **Readme** file for more information.

Here are the steps for installation:

1. `tar xzf cg.xxx.tar` : untar the cg files.
2. Define the CG and CGBUILDPREFIX environmental variables as described in Section 8.
3. `cd cg.xxx`
4. `make` : this will build the default CG solvers. You can also type “`make -j 2`” on machines with multiple processors in order to perform a parallel make. Sometimes the parallel make gets confused so you may have to type `make` again after the first one finishes.
5. Instead of making all the default CG solvers as above, you can also just build individual ones. If you only want to build cgins, for example, go to the `$CG/ins` directory and type `make`.
6. By default the Maxwell solver is not built. Go to the `$CG/mx` directory and type `make` if you want to use it.

Here are some tests you can run

1. From the main cg directory type `make check` to run regression tests for each solver.
2. To test just one solver, say cgins, go to the `$CG/ins/check` directory and type `make`.

If you want to change some CG files and rebuild a solver you can use the `$CG/user` directory to do this. See the `$CG/user/Readme` file for more information.

14 Notes for running Overture in parallel

The parallel version of Overture is in beta release to friendly users. Not everything works in parallel but many things do.

To configure and build the parallel version (you need to install P++ first):

1. `setenv PPlusPlus xxxx` where xxxx is the location of the P++ installation.
2. `setenv APlusPlus xxxx` where xxxx is ALSO the location of the P++ installation.
3. `configure parallel`
4. `make`

General notes:

- You will need to have both a serial and parallel version of Overture built.
- It is safer to use the serial version of `plotStuff` to plot grids and plot results in show files, although some of the graphics, such as plotting grids and 2d/3d contours, should work in parallel.
- Build grids with the serial version of `ogen` unless you have to build a very large grid. (The parallel version of `ogen` works in many cases but is still under development).
- Parallel moving grids work (but not yet parallel deforming grids).
- Parallel AMR works (but not yet parallel AMR and parallel moving grids).
- Some Mapping's don't work in parallel since I haven't gotten around to writing the new `mapS` and `inverseMapS` functions that take serial arrays as arguments – these are usually easy to write.
- Generally it is best to convert some Mapping's (e.g. `DataPointMapping`, `HyperbolicMapping`) into `NurbsMappings` for use with parallel (see, for example, `Overture/sampleGrids/loftedBox.cmd`).

PDE solvers that should run in parallel:

- The primer example `pwave` is a good example of a small parallel code. Many of the other primer examples may run in parallel but are generally not good examples of how to write efficient parallel code.
- The cg solvers `cgens`, `cgins`, `cgad`, `cgm`, `cgs`, and `cgmp` run in parallel but not all options will work. `cgasf` does not run in parallel yet.

Developer notes: If you are writing Overture code to run in parallel here are some notes:

- Avoid scalar indexing of a parallel array, `u(i1,i2,i3)` since this will use communication by default. Instead, access the local serial array (see examples in `pwave.C` for example).
- Avoid P++ array operations except for simple copies. It is more efficient and robust to grab the serial arrays, operate on the serial arrays and then call `updateGhostBoundaries`.
- There are a variety of parallel utility routines for dealing with parallel array operations, grid functions and parallel copies etc. See the files `App.{h,C}` and `ParallelUtility.{h,C}`.

APPENDIX C. WARPX Build Instructions

C1.Build Instructions

Before you begin you must have an account on the WarpX Git repository. You can get access to this by filling out this [form](https://sites.google.com/site/warpxwiki/download-request-page) : <https://sites.google.com/site/warpxwiki/download-request-page>. A caveat: on some systems it is best to consult your system-administrator for build instructions or have him/her build the key software, specially the MPI libraries. You are **strongly** encouraged to build all needed software (except MPI) yourself.

These instructions have been tested on Mac OS X and a Fedora Linux box. Significant work may be needed to get the dependencies working on other platforms. Note that WarpX itself is very easy to build once you have the dependencies figured out. We have built it from desktops to latest supercomputer clusters.

You can cut and paste all instructions to build on your machine. Ideally no other instructions should be needed. If your system does not have the "curl" program to download the software needed by WarpX you can use "wget" instead. If you do not have either of these, then cut-and-paste the URLs to a web browser and copy the archives to the appropriate directory.

It is recommended that you use the system MPI on a cluster. In that case you **do not need to reinstall MPI yourself**. If you do use the system MPI then in the instructions given below make sure that you replace the MPI directory locations to your system MPI.

These installation instruction process does not need administrator permissions. First, make two directories in your \$HOME directory

```
cd
mkdir software
mkdir buildDir
```

The buildDir directory will have the sources and the software directory will be the directory in which all software is installed.

Getting and installing MPI

In general MPI should be built by the system administrator. If this already exists just add the location of the directory in which mpicc lives to you [config.py](#) file. You **must** build parallel HDF5 and parallel WarpX with the same MPI.

Get it first. We will use OpenMPI.

```
cd $HOME/buildDir
curl http://www.open-mpi.org/software/ompi/v1.2/downloads/openmpi-1.2.5.tar.gz
-O
```

Now unzip and untar

```
gunzip openmpi-1.2.5.tar.gz
tar -xvf openmpi-1.2.5.tar
```

Now build and install it

```
cd openmpi-1.2.5
./configure --prefix=$HOME/software/openmpi-1.2.5
make
make install
```

Now go back to the software directory and softlink for ease of use.

```
cd $HOME/software
ln -s openmpi-1.2.5 mpi
```

Put your MPI location in your path

```
export PATH=$HOME/software/mpi/bin:$PATH
```

Getting and installing HDF5

Get it first.

```
cd $HOME/buildDir
curl http://www.hdfgroup.org/ftp/HDF5/prev-releases/hdf5-1.8.2/src/hdf5-1.8.2.tar -O
```

Now unzip and untar::

```
gunzip hdf5-1.8.2.tar.gz
tar -xvf hdf5-1.8.2.tar
```

We will need both parallel and serial builds.

Serial HDF5 build

We will build serial HDF5 out-of-place.

```
cd hdf5-1.8.2
mkdir serial
cd serial
../configure --prefix=$HOME/software/hdf5-1.8.2
make
make install
```

Now go back to the software directory and softlink for ease of use.

```
cd $HOME/software
ln -s hdf5-1.8.2 hdf5
```

Parallel HDF5 build

We will build parallel HDF5 out-of-place.

```
cd $HOME/buildDir/hdf5-1.8.2
mkdir parallel
cd parallel
export CC=$HOME/software/mpi/bin/mpicc
export CXX=$HOME/software/mpi/bin/mpicxx
export LD_LIBRARY_PATH=$HOME/software/mpi/lib/
../configure --prefix=$HOME/software/hdf5-1.8.2-openmpi-1.2.5
make
make install
```

Now go back to the software directory and softlink for ease of use

```
cd $HOME/software
ln -s hdf5-1.8.2-openmpi-1.2.5 hdf5mpi
```

Optional HDF5 build with SZIP library

You can build extra compression capability into HDF5 using the SZIP library. First, download and install SZIP. See the project website at

When you run the HDF5 configure command, add the flag `--with-szlib=$RESEARCH_SOFTWARE_BASE/szip` where the path points to the library directory.

Optional: Getting and Installing PETSc

The PETSc libraries are required for some of the subsolvers in WarpX, specially ones that deal with solving a large sparse system of linear/non-linear equations.

For version 3.0.0-p3 (which was the latest by the time of this writing), you can use:

```
cd $HOME/buildDir
curl ftp://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.0.0-p7.tar.gz -O
```

Extract the tar file and set the environmental variable PETSC_DIR to the PETSc installation folder

```
tar -xzf petsc-3.0.0-p7.tar.gz
cd petsc-3.0.0-p7
export PETSC_DIR=$PWD
cd $PETSC_DIR
```

For parallel build, make sure that MPI has already been installed.

Serial Petsc build

For ease of use, have PETSc download, configure, compile and install using

```
./config/configure.py --with-shared=0 --with-mpi=0 \
--prefix=$HOME/software/petsc-3.0.0-p7 \
--with-debugging=0 --COPTFLAGS='-O2 -g' --with-x=0 \
```

```
--download-superlu --download-f-blas-lapack=yes
```

Once the configuration step is complete (it may take a while) set the environment variables as instructed by PETSC. On my Mac I have to do

```
PETSC_ARCH=darwin9.6.0-c-opt; export PETSC_ARCH
```

Then build the code using

```
make all
make install
```

Now go back to the software directory and softlink for ease of use

```
cd $HOME/software
ln -s petsc-3.0.0-p7 petsc
```

Parallel Petsc build

First go back to your petsc directory

```
cd $PETSC_DIR
```

To build in parallel you must have MPI installed. Configure PETSC using the following command

```
./config/configure.py --with-shared=0 --with-mpi-dir=$HOME/software/mpi \
--prefix=$HOME/software/petsc-3.0.0-p7-openmpi-1.2.5 \
--with-debugging=0 --COPTFLAGS='O2 -g' --with-x=0 \
--download-hypre --download-parmetis --download-blacs \
--download-superlu_dist \
--download-scalapack --download-mumps --download-sundials
```

MPI may be installed on a different folders for different machines. Adjust the --with-mpi-dir flag accordingly. Make sure that you compile PETSc with the same MPI installation that you use for WarpX.

Once petsc has been configured it can be built using the instructions which the build script prints on successful configuration. Unfortunately, due some strange problem in PETSC the instructions are not so clear. Read the output carefully to figure this out. On my Mac I need to do:

```
export PETSC_ARCH=darwin9.6.0-c-opt
export PETSC_DIR=/Users/ammam/buildDir/petsc-3.0.0-p7
```

Now build the code using

```
make all
make install
```

Now go back to the software directory and softlink for ease of use

```
cd $HOME/software
ln -s petsc-3.0.0-p7-openmpi-1.2.5 petscmpi
```

Getting Python

In general you will not need to install Python as it comes standard with almost all Mac and Linux systems. On the Mac you may want to upgrade to Python 2.5 by downloading MacPython 2.5 from the Python website.

Getting and installing Scons

Scons is the WarpX build system. First get it from

<http://prdownloads.sourceforge.net/scons/scons-1.0.1.tar.gz>

Save it in your \$HOME/buildDir directory. Then

```
cd $HOME/buildDir
gunzip scons-1.0.1.tar.gz
tar -xvf scons-1.0.1.tar
```

Now build it

```
cd scons-1.0.1
python setup.py install --prefix=$HOME/software/scons-1.0.1
```

Now go back to the software directory and softlink for ease of use.

```
cd $HOME/software
ln -s scons-1.0.1 scons
```

Add the path to scons to your PATH and PYTHONPATH variables. You can do this by adding the following lines to you .bashrc or .profile files

```
export PATH=$HOME/software/scons/bin:$PATH
export PYTHONPATH=$HOME/software/scons/bin:$PYTHONPATH
```

Getting and installing WarpX

To obtain WarpX you need need an Git account on the WARPX repository described elsewhere. Obtain a copy of repository by doing

```
git clone git@warpx.sourcerepo.com:warpx/WARPX.git warpx_git
```

The final argument warpx_git is arbitrary, but will be the name of the top-level directory created. If your RSA key was setup correctly and you have internet connectivity, the repository should now copy or clone itself to your machine. Once complete, your working copy is the most recent version in the repository.

As a final basic configuration for Git, tell it who you are and your email address, so that later commits can be handled appropriately. Use the two commands below with your name and email address.

```
git config --global user.name "Your Name"
```

```
git config --global user.email you@yourserver.com
```

More instructions for working with the Git repository are outlined on the [Working Collaboratively](#) page.

Change to the WARPX src directory

```
cd warpx_git/src
```

Create a file called config.py

```
touch config.py
```

Now open this in Emacs or some other text editor and add the following lines to it

```
import os

mpi = os.environ.get("HOME") + "/software/mpi"
hdf5s = os.environ.get("HOME") + "/software/hdf5"
hdf5p = os.environ.get("HOME") + "/software/hdf5mpi"
```

Your MPI path may be different if you are using a MPI already installed on the system. Now you are ready to build WarpX.

Serial WarpX build

To build WarpX in serial, from the warpx/src directory do

```
scons parallel=no debug=no
```

This will build WarpX in serial without any debugging symbols. The compiled code is put into a directory (in warpx/src) called build. The executable is in the build/warpx directory and is called xwarpx.

You can also build with debugging enabled

```
scons parallel=no debug=yes
```

The compiled code is put into a directory (in warpx/src) called build-deb. The executable is in the build-deb/warpx directory and is called xwarpx.

Parallel WarpX build

To build WarpX in parallel, from the warpx/src directory do

```
scons parallel=yes debug=no
```

This will build WarpX in parallel without any debugging symbols. The compiled code is put into a directory (in warpx/src) called build-par. The executable is in the build-par/warpx directory and is called xwarpx.

You can also build with debugging enabled

```
scons parallel=yes debug=yes
```

The compiled code is put into a directory (in warpx/src) called build-par-deb. The executable is in the build-par-deb/warpx directory and is called xwarpx.

Building WarpX with PETSc

Use

```
scons usepetsc=yes
```

to build WarpX with PETSc. Other options can also be used as usually. WarpX will then check for PETSc, BLAS and LAPACK libraries and headers in the specified folders. If any of these libraries is not found, scons will stop the building process and output an error message.

Updates to the instructions can be available at

<https://sites.google.com/site/warpxwiki/warpx/warpx-documentation/build-instructions>

C2. Configuring Build Parameters

WarpX needs to be configured before it can be built. For this, you need to prepare a file called config.py in the warpx/src directory. The first time after you download the code do

```
cd warpx/src  
touch config.py
```

This will create an empty file. In this file you can add locations for various libraries and tools. Most commonly you will need to add the location of your MPI library, your serial HDF5 build and your parallel HDF5 build. If you intend on using PETSc, you will also need to specify your PETSc installation folder, and also folders for SUPERLU, SUPERLU_DIST, or HYPRE if requested. PETSc is dependent on LAPACK and BLAS, and these two packages are also needed if you want to use PETSc.

An example config.py looks like the following

```
##  
# Configure options for WarpX  
##  
mpi = "/Users/ammam/software/mpi"  
hdf5s = "/Users/ammam/software/hdf5"  
hdf5p = "/Users/ammam/software/hdf5mpi"
```

If you want to use PETSc, you will have to add folders linking to PETSc installation folder by adding

```
petscser=$HOME/software/petsc-3.0.0-p3  
petscpar=$HOME/software/petsc-3.0.0-p3
```

And depending on where the BLAS and LAPACK libraries are, you might also need to specify their locations. If you install PETSc to then \$HOME/software/petsc-3.0.0-p3, then most likely you will need to add the following to config.py

```
lapack_libdir = "$HOME/software/petsc-3.0.0-p3/lib"  
blas_libdir = "$HOME/software/petsc-3.0.0-p3/lib"
```

If you install BLAS and LAPACK through PETSc (by including them in the configure line), you need to manually specify the names of their libraries by adding the following to config.py

```
lapacklib = "flapack"  
blaslib = "fblas"
```

This step is needed because the LAPACK and BLAS libraries as installed by PETSc have non-conventional names, and for WarpX to be able to recognize the libraries, this modification is needed.

Obviously, the paths in the config.py file will be different for your system.

Other variables accepted by the system are CC, which specifies the C compiler to use and CXX which specifies the C++ compiler to use. Sometimes it is necessary to specify these explicitly if you are using non-standard compilers or compilers located in non-standard places.