

REPORT DOCUMENTATION PAGE				<i>Form Approved OMB No. 0704-0188</i>	
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 the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 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) 03-11-2009		2. REPORT TYPE FINAL TECHNICAL REPORT		3. DATES COVERED (From - To) 15-09-2007 to 14-09-2009	
4. TITLE AND SUBTITLE Development of commercially useable codes to simulate aluminized propellant combustion and related issues				5a. CONTRACT NUMBER FA9550-07-C-0123	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) John Buckmaster				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Buckmaster Research 2014 Boudreau Dr Urbana, IL 61801				8. PERFORMING ORGANIZATION REPORT NUMBER 2009NOV03	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM (Dr. Arje Nachman) 875 N Randolph St Suite 325, Room 3112 Arlington VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2012-0597	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We have modified and developed a large number of codes relevant to the task including: a packing code to generate propellant morphology models; a homogenization code to predict homogenized properties of a heterogeneous propellant; a combustion code that predicts fluctuations in the hot gases coming off the propellant surface; and fluids codes that describe the rocket chamber flow, the rocket nozzle flow, aluminum transport, and nozzle erosion. The packing code is marketable; the homogenization code will be available as an addendum to the packing code in a few months; a fluids code that simulates the chamber flow using approximations valid for slender chambers is marketable, albeit without a user manual at this time; and further development of a nozzle code that accounts for nonuniform inlet conditions (at the nozzle inlet) is under further development under a recently awarded SBIR contract.					
15. SUBJECT TERMS STTR Report; solid propellants, rockets, packing, homogenization, chamber flow, aluminum, nozzle erosion					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			John Buckmaster
U	U	U	UU	27	19b. TELEPHONE NUMBER (Include area code) 217-621-9786

Reset

Development of commercially useable codes to simulate aluminized propellant combustion, and related issues

Final Report

J.Buckmaster dba Buckmaster Research

2014 Boudreau Drive, Urbana, IL 61801

limey@uiuc.edu; tel: 217.621.9786; fax: 217.244.0720

October 23, 2009

Table of Contents

1. Introduction: Solid propellants and their combustion.....	3
1.1 The Solicitation.....	3
1.2 Background	3
2. Phase II Technical Objectives	6
3. Meeting the Technical Objectives	8
4. More Details	4
4.1 Packing.....	4
4.2 Material Properties	5
4.3 Kinetics Fitting	7
4.4 Agglomeration	8
4.5 Morphology effects on the chamber flow.....	10
4.6 SRM nozzle erosion.....	13
4.6.1 Mathematical Formulation.....	15
4.6.2 Obtaining Proper Inlet Conditions	15
4.6.3 Importance of Unsteady Inlet Conditions	16
4.6.4 Importance of Hydrodynamic Instabilities.....	16
4.6.5 Asymmetric Rockets	16
4.6.6 Future Work.....	16
6. Conferences and Publications.....	19
References.....	21
Appendix.....	22

This Page Left Intentionally Blank

Development of commercially useable codes to simulate aluminized propellant combustion, and related issues

Final Report

J.Buckmaster dba Buckmaster Research
2014 Boudreau Drive
Urbana, IL 61801
limey@illinois.edu; tel: 217.621.9786
October 23, 2009

1. Introduction: Solid propellants and their combustion

1.1 The Solicitation

Provide a code to accurately predict the entire flight of aluminum particles from the propellant surface through the nozzle exit plane (for nozzle impact studies and plume signature studies) together with a prediction of the effective properties (thermal conductivity and measures of mechanical strength such as Young's modulus, Poisson's ratio) of the binder, ammonium perchlorate, and aluminum spheroids which together constitute a solid propellant. A code is sought that will accurately model/simulate the effective properties and burning of composite solid rocket propellants. Composite propellants are composed of energetic particulates, typically of ammonium perchlorate (oxidizer) and aluminum (fuel) bound together with a polymeric binder constituting 10-12% of the propellant volume. Particulate packing is critical to both the combustion and mechanical properties, and so the code must contain a robust packing capability. Once the packs have been created numerically, they must be burnt numerically. The code should predict the statistics of the aluminum agglomeration on the surface of an aluminized heterogeneous propellant since the dynamics of these agglomerates (detachment from the surface, transport in the chamber flow, burning, impact of aluminum oxide particulate on the nozzle which abrades the nozzle, and creation of an exhaust signature) is of critical importance to the performance of solid rocket engines.

Phase II: Development of marketable codes.

1.2 Background

Heterogeneous reactive materials are widely used as rocket propellants and in explosives. The solicitation is concerned with propellants, but the tools developed in response to the solicitation can in many cases be of use in the explosive context also. Since the Air Force has interests in explosives as well as in rockets, we have paid some attention to this closely related area with a view to marketing codes to a wider audience than that directly implied by the solicitation. In

addition, there is at least one component here that is likely to be of interest to a wide range of business interests, and that is the packing module. Certain kinds of business need to create heterogeneous materials with prescribed component fractions and that is a non-trivial problem when the added ingredients that are to be placed in the matrix are to occupy a large volume fraction. With these other issues not forgotten, we shall discuss the propellants that are at the heart of the solicitation.

Solid propellant rockets, used as missiles, boosters during the launch phase of the space shuttle, etc., are typically fueled by oxidizer particles (e.g. ammonium perchlorate, AP) embedded in a rubbery fuel-binder (e.g. hydroxyl-terminated polybutadiene, HTPB). Other chemical choices are possible, but it is convenient in our discussion to invoke AP both in its own right, and as a surrogate for the oxidizer, and HTPB both in its own right and as a surrogate for the binder.

Some 80% or so of the volume needs to be AP, i.e. the packing density is high, a challenge to the modeling of the morphology. Often aluminum particles are added, along with the AP, since they burn in the AP-HTPB combustion products to raise the chamber temperature and therefore increase the specific impulse. Note that Tsiolkovsky's famous equation makes it clear that appropriate measures of rocket performance only increase logarithmically with propellant mass so that the smallest increases in performance by other means are highly valued.

Unfortunately, the addition of aluminum exacts a price. The particles are sized in the tens of microns, but sit around on the surface of the propellant (in the HTPB melt layer) where they agglomerate. The agglomerates are sized in the hundreds of microns. After detachment, these agglomerates pass through the chamber and can scour the nozzle walls or accumulate behind the leading lips of re-entrant nozzles. In the latter case, the accumulated mass can be deleterious to rocket performance. The agglomerates also contribute to flow losses within the chamber. The ability to predict the statistics of the agglomerates would be an invaluable tool in propellant development, and is one of the major interests raised in conversations with major players in the rocket industry.

Figure 1 shows a cartoon of a burning aluminized propellant, and gives some clues as to the complexity of the process. The essential physics within the propellant is unsteady three-dimensional heat conduction with different properties (conductivities, densities etc.) for the different components. The gas-phase is governed by the reactive Navier-Stokes equations and the Mach number is small so that a zero Mach number code is required. The number of reactants and reactions is huge, far too many to be precisely modeled within the three-dimensional, unsteady context, and so global kinetics

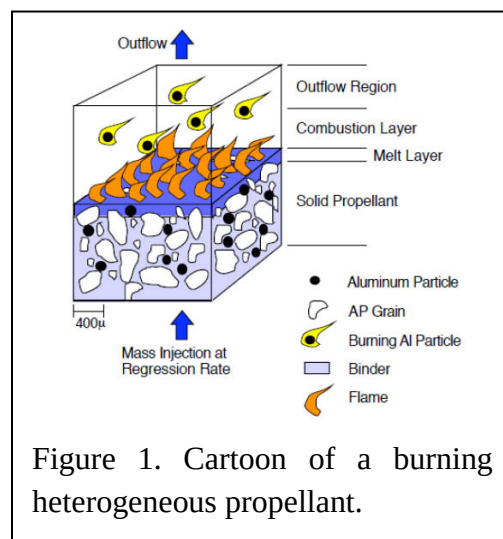


Figure 1. Cartoon of a burning heterogeneous propellant.

schemes must be adopted. The well-known Beckstead-Derr-Price model is one that is suitable, and is characterized by:

- The primary diffusion flame, reaction between AP gases and HTPB gases;
- The AP decomposition flame, exothermic decomposition of primary AP gases to form reactive products;
- The secondary diffusion flame, reaction between HTPB gases and the products of the AP decomposition flame.

A numerical model of Figure 1 can be used for a number of purposes. For example, it can be used to predict: the mean burning rate; gas-phase fluctuations beyond the combustion layer, fluctuations which define boundary conditions for the turbulent chamber flow; variations of the burning rate due to an impinging acoustic wave, a matter of importance in chamber stability studies; and the formation of aluminum agglomerates at the surface, and their subsequent detachment. It hardly needs to be pointed out that these are highly challenging problems, challenging both to the modeler, and to the code constructor.

The modeling of Figure 1 is a crucial component of the solicitation. An equally crucial component is the modeling of the chamber flow, something best carried out in an LES (large eddy simulation) context (DNS is inadequate). This can be used to describe the flight of the aluminum agglomerates, during which they burn to form oxide. The chamber flow itself is not without interest, of course, a notable issue being time fluctuations in the head-end pressure, fluctuations which can be important to rocket-casing fatigue life. Moreover, strong eddies in the chamber flow can impinge on the surface of the burning propellant and might contribute to erosive burning, a phenomenon that is only recently being understood.

The importance of this modeling if it can be used to create a design tool for the rocket industry can hardly be overemphasized. Propellant development and rocket development is a hugely expensive empirical game, presently guided only by crude, back-of-the-envelope modeling, and the experience and genius of the development engineers. The latter component is undoubtedly what makes it work, but real improvement in performance and significant cost-cutting can only come about if more sophisticated design tools are implemented. We live in an age of remarkable sophistication insofar as numerical simulations are concerned, with ready access to large parallel computing platforms, but neither the industry nor the mainstream university propellant community works within this modern framework.

2. Phase II Technical Objectives

The proposed technical objectives were:

1. At the present time, we can pack circles, spheres, ellipses, and ellipsoids; we pack in periodic cuboids. We shall construct marketable codes that can pack in other geometries; and we have good reason to believe that it might be possible to pack arbitrary shapes, something hitherto not accomplished by anyone. The latter would be of great importance for explosives modeling, e.g. HMX. Our goal is to make it possible to execute all packing codes in a laptop environment.
2. The solicitation calls for methods for calculating the effective (large-scale) thermal and mechanical properties of a heterogeneous pack. We have devised a strategy for calculating the thermal properties with accuracy sufficient for our past purposes, but have not examined mechanical properties; AP/HTPB mixtures are essentially visco-elastic. This so-called homogenization problem is not an easy one, and there are no standard strategies in 3-dimensions that can inevitably lead to a general solution (we solved the thermal property problem using *ad hoc* methods which were tested numerically). Moreover, homogenization is not a uniquely defined problem, and the only truly legitimate results are bounds. Strategies for periodic heterogeneities exist, and can provide useful estimates for these bounds. Dr Karel Matous, a noted expert on this subject, will work with us as a consultant, and we expect to generate an addendum to the packing code that can calculate data of value to the engineering community.
3. At the present time we can numerically burn AP/HTPB/Al packs using the 3-step chemistry identified in Section 1. A key component of the model is the specification of the kinetics parameters, parameters for what is, after all, false kinetics, and so cannot be determined from chemistry fundamentals. Instead, “hand-fitting” of certain empirical data is employed, such as one-dimensional burning rates of fine-AP/HTPB blends. Three-dimensional data is never used (except for validation purposes). Now although it is likely that AP propellants have a long future, there is significant interest in other propellant components, such as HMX, and certainly other ingredients are relevant to the explosives problem. And so a key component of a useful marketable code is likely to be a kinetics specification module in which the “fitting” is done in an optimal fashion, by machine, for a prescribed set of target data (one-dimensional burning rates, stability boundaries, and the like).
4. Aluminum plays an important role in the solicitation, and the problem of agglomeration is of enormous importance to the rocket industry. How it comes about is not well understood, and there are different opinions as to the nature of the controlling physical ingredients. In this connection there is a significant lacuna in the DOE funded work, as the DOE goals can be achieved without an accurate determination of the agglomerate statistics. We have worked on agglomeration and we believe that given the mean agglomerate diameter it is possible to accurately predict the standard deviation of the distribution (assuming, as seems true, that the

distribution can be described log-normally) simply from the pack geometry. A key question is whether we can predict the mean diameter.

5. Chamber flow simulations using LES (large-eddy simulations) require boundary conditions at the propellant surface, a specification of the velocity and temperature fluctuation statistics. Calculations carried out so far under DOE auspices use white-noise statistics. By numerically burning a propellant and extending the calculations out to the millimeter scale, beyond the flame zone, it is possible to determine the correct statistics. We believe that this will play an important role in the calculations of the chamber flow.
6. The chamber flow is an important component of the motor description of course, and drives the “flight of the agglomerates” identified in the solicitation. Flow numerical modules have been developed as part of the DOE program, but a number of issues of importance to the present objectives remain unexamined. These include: an examination of the role that different surface statistics plays in the nature of the turbulent flow, and the distribution of the in-flight agglomerates; an examination of whether nozzle impact and scouring by agglomerates can be reasonably quantified; an examination of powerful eddies near the nozzle entrance to see the role they play in defining the near-surface flow, and whether this near-surface flow could affect the burning rate (erosive burning); more accurate treatments of the agglomerate combustion; an accounting of droplet (liquid aluminum oxide) break-up at high Weber numbers.

3. Meeting the Technical Objectives

[The paragraph numbering in this section corresponds to the numbering of section 2.]

The technical objectives identified above are ambitious for a Phase I plus Phase II funding level (2.75 years and \$850,000); consequently, we were not able to meet every objective. However, significant progress and commercialization was done for a number of the items. Below we briefly discuss our accomplishments for each item listed above. More details are provided in Section 4.

1. **Packing:** We aimed to do four things:

- Develop a webpage for commercialization.
- Add various refinements so that exit is always graceful (no Nans).
- Develop sphere packs in volumes other than cuboids, e.g. right circular cylinders. Of particular significance here is that rigid walls define parts of the pack boundary, and very little is known on how such boundaries could affect the pack properties. Conversations with Aerojet engineers in the Spring of 2006 made it clear that the industry is very interested in being able to predict the effect of the chamber boundaries on the burning rate.
- Develop the ability to pack arbitrary shapes, and not just spheres or spheroids, the shapes to be defined using level-sets. This would be an important break-through. It would be invaluable in constructing morphology models of HMX explosives or igniters, for example.

We were able to accomplish all four items. A webpage was built for the general packing code, a number of algorithm refinements were done to improve performance, we are now able to pack spheres in a cylinder and an annulus, and we are now able to pack shapes other than spheres.

The packing code (which we call *Rocpack*) has been commercialized and a web page dedicated for software sales has been created. The link can be found at <http://fermat.cse.uiuc.edu/rocpack/index.html>. The web page allows a potential user to make a variety of small but representative packs before purchasing. A User Manual has been written and is available on the web; a copy is attached in the Appendix. Figure 2 is a screen shot of the home page of *Rocpack*.

More details can be found in Section 4.

Rocpack

[Home](#)
[Demo Program](#)
[Purchase](#)
[Help](#)

About Rocpack

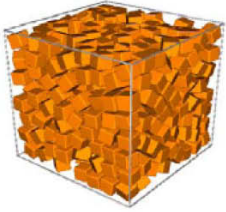
Random packs of spheres are used to model heterogeneous and porous material morphologies such as concrete, sand, coal, porous explosives, and solid rocket propellants, to name a few. The most obvious advantage of using spheres lies in the simplicity of their representation, which facilitates theoretical, experimental, and computational efforts in the physical sciences. Nevertheless, spheres can be a poor approximation to heterogeneous and porous materials when various other properties are of interest. Bulk materials properties such as thermal conductivity and elastic properties strongly depend on the statistical details of the micro-structure, and spheres do not properly replicate these statistics.

In recent years we have constructed closely packed spheres using the Lubachevsky-Stallinger algorithm to generate models of heterogeneous solid propellants. Improvements to the algorithm now allow us to pack non-spherical convex shapes for modeling heterogeneity in complex energetic materials such as plastic-bonded explosives and pressed gun propellants. Rocpack is the name of the resulting packing code. Some examples of packs generated by Rocpack are given in the figure below.

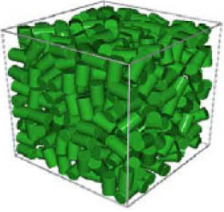
Rocpack is able to pack a variety of convex shapes (spheriods, cylinder, gelcap, icosahedron, cube, crystals) in a container. Currently the container shape can be a cuboid, cylinder, or an annulus. For the annulus a web thickness must be given.

On this site is a GUI that will allow the user to run Rocpack and generate some simple packs of spheres or of a few specific convex shapes. A user manual is also provided that better explains Rocpack, gives useful information with examples of how to run the GUI, and gives system requirements should the reader wish to purchase the program as a single user license.

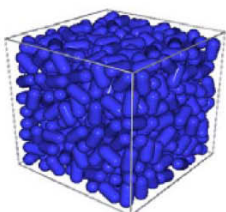
Examples



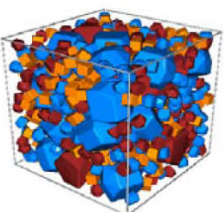
Ending Density : 0.48
Initial Temperature : 1000
Container : Cuboid
Shape 1 :
Shape of Particle : Cube
Number of Particles : 500
Size of Particles : 1
Wall time : 7:58:28



Ending Density : 0.48
Initial Temperature : 1000
Container : Cuboid
Shape 1 :
Shape of Particle : Cylinder
Number of Particles : 500
Size of Particles : 1
Wall time : 0:40:28



Ending Density : 0.48
Initial Temperature : 200
Container : Cuboid
Shape 1 :
Shape of Particle : Gelcap
Number of Particles : 500
Size of Particles : 1
Wall time : 0:40:49



Ending Density : 0.47
Initial Temperature : 1000
Container : Cuboid
Shape 1 :
Shape of Particle : Icosahedron
Number of Particles : 200
Size of Particles : 1
Shape 2 :
Shape of Particle : Icosahedron
Number of Particles : 10
Size of Particles : 3
Shape 3 :
Shape of Particle : Cube
Number of Particles : 200
Size of Particles : 1
Shape 4 :
Shape of Particle : HMX
Number of Particles : 200
Size of Particles : 1
Shape 5 :
Shape of Particle : HMX
Number of Particles : 20
Size of Particles : 3
Wall time : 30:01:24

Figure 2. Home page of *Rocpack* webpage.

2. **Material Properties:** The strategy adopted in the program has three ingredients.

The first is to construct a small pack that is statistically identical (to some appropriate order) to the real (large) pack. The statistics of the large pack can be determined experimentally in the case of a physical pack (using tomomicrography), or numerically from a numerically constructed pack. The latter is easily done; and we have demonstrated that the former is possible. The small pack is used in the subsequent calculations.

The second is a mathematical (analytical) one in which a general framework is constructed which can be used to determine property bounds for any property for which the underlying characteristic operator is elliptical (heat conduction is an obvious example). This is non-trivial, and has been completed.

The third is the actual evaluation of property bounds using the second strategy for a particular property. This is computationally intensive. Results have been obtained for elastic properties, and a paper is presently being written in which these results will be reported in the archival literature. Because this paper is presently only available in draft form, not for public distribution, we merely show the draft abstract here:

“In this report, we use the well-known Hashin-Shtrikman variational principle to obtain the overall mechanical properties of heterogeneous polydisperse particulate composites. The emphasis is placed on the efficient numerical integration of complex three-dimensional integrals and on aspects of the anisotropic material response. In many previous works, only short range elastic interactions have been considered, and most of the work had been done in two-dimensions using artificial and statistically isotropic probability functions. In this work, we consider both short and long range interactions and focus on the efficient numerical integration, which allows us to accurately capture the underlying three-dimensional anisotropic nature of particulate systems. For this purpose, we numerically calculate the complete statistics of real packs, which are numerically or tomographically generated. We use the parallel adaptive sparse Smolyak integration method with hierarchical basis to integrate complex singular integrals containing the product of probability functions and the second derivative of Green’s function. We exploit the geometrical characteristics of our integrand, i.e., we use spherical coordinates to define the basis functions and the integration. Selected examples illustrate both the numerical and physical facets of our work. First, we show the reduction of integral points for spherical systems. Then, we comment on the parallel scalability of our method and on the numerical accuracy associated with the integration of a singular function. To investigate the ability of our scheme to capture the anisotropic nature of packs, we validate the solver using experimental data and solve a lattice type system. Finally, we report on the elastic

constants computed for the modeled particulate system that is tomographically characterized.”

3. **Kinetics Fitting:** We have demonstrated that global kinetics parameters can be determined by using optimization strategies that examine a wide range of data, including surface temperature, burn-rate sensitivity (to temperature changes), burning rates, burn-rate sensitivity for one or more of the propellant components, 1-D linear stability, low pressure deflagration limits, etc. The idea is to obtain this data experimentally for a typical propellant pack; and then, once the kinetics parameters are calculated, using them to explore, numerically, the properties of different packs with the same ingredients but different particle distributions and volume fractions. The principle is that the kinetics parameters do not change just because the morphology changes. The latter has been demonstrated in earlier work that we did in which the parameters were determined using non-optimal ad-hoc strategies. Thus Figure 3 shows burning rate predictions (circles joined by a solid curve) compared to experimental data for four propellants. There are convincing reasons to believe that the experimental data of M21 is not correct.

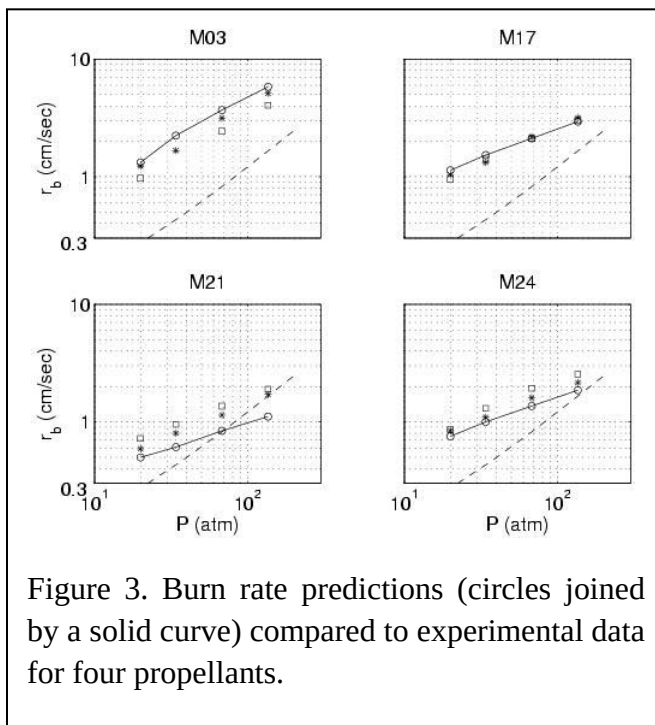


Figure 3. Burn rate predictions (circles joined by a solid curve) compared to experimental data for four propellants.

We are prepared to engage with the rocket industry in this matter, but conversations with various players reveals an unwillingness to believe in the value of global kinetics. This is an odd position to take, as global kinetics is certainly seen as useful in, for example, the automobile industry, but we have no way of overcoming these cultural handicaps. Related to this, one of the major rocket companies has their own packing code, one that is vastly inferior to ours, and cannot be used in the same way (for example, to calculate material properties in the manner we describe in Section 2.2), and yet one they persist in using.

4. **Agglomeration:** We have demonstrated that by using a numerically defined aluminum-loaded pack and using a specified length S_d against which aluminum particle separation distances are measured, it is possible to predict the agglomeration distribution provided the mean is specified. S_d is chosen to match the latter. We had hoped that it might be possible to

identify a pattern from which S_d could be determined from elementary properties of the pack, such as the mean of the distance between large AP particles; we identified this as a high-risk, high-payoff strategy. We have not been able to do this. We are currently of the belief that the problem of agglomeration can only be solved using a detailed simulation of aluminized pack combustion that accounts for the detailed physics acting on each aluminum particle. To carry this out would take a large effort, at the least an exclusively focused STTR.

As evidence of its importance we note that the December 2009 JANNAF meeting (43rd Combustion Subcommittee) has several presentations devoted to aluminum agglomeration. Furthermore, we have heard a manager at one of the leading aerospace rocket companies say that “aluminum agglomeration modeling is one of our highest priorities”.

5. **The effects of propellant morphology on the chamber flow.**

We have shown how the statistics of the fluctuating flow field generated by a burning propellant can be determined. Of particular importance are the joint pdf correlations of temperature and normal velocity which can have a significant effect on head-end pressure oscillations that can lead to motor instability. Because of the enormous amounts of energy that characterize rocket combustion, there is always the concern that motor instabilities can lead to failure.

6. **Chamber flow and nozzle erosion:**

6a. Chamber Flow:

We are able to carry out large scale chamber flow calculations, including the transport of aluminum particles. However, because of the magnitude of the numerical problem, full motor simulations (chamber and nozzle) cannot resolve the nozzle boundary layers. One can question the accuracy of nozzle erosion calculations from such calculations. Moreover, the chamber simulations alone use advanced codes that can only run in a large parallel environment, not available to the rocket industry. Indeed, in the design of the 5 section enlargement of the 4 section solid propellant boosters used on the space shuttle, an enlargement needed for the planned shuttle replacement, ATK contracted with our university partner (CSE at the University of Illinois) to carry out appropriate simulations.

For these reasons we decided to focus on the following issues:

- A simplification of the chamber flow simulations by using an asymptotic strategy valid for high aspect ratio chambers.
- A partial uncoupling of the chamber flow and the nozzle flow in which complete motor simulations sans boundary layer resolution are used to calculate the chamber

flow, and then this solution is used to specify inlet boundary conditions for a fully resolved nozzle flow calculation. The latter focuses on erosion.

- An examination of the effects of non-uniform inlet conditions (arising from propellant morphology and/or chamber flow turbulence) on the nozzle flow.

Finally, because of our long-standing interest in propellant combustion we have long been interested in erosive burning, and the asymptotic strategy that we adopted for the chamber flow turned out to be an ideal tool for an examination of this poorly-understood problem.

More details are provided in Section 4.

6b. Nozzle Erosion:

We made significant progress on the modeling and simulation of erosion of solid rocket nozzles. A more complete description of the work is provided in Section 4. We note here, however, that a key finding of our research is that using turbulent inlet conditions has a significant effect on the time-averaged erosion rate when compared to using uniform inlet conditions. This is important because industry standard is to use uniform inlet conditions when computing the heat flux to the nozzle wall. Furthermore, our work is fully coupled (i.e., we resolve the boundary layer along the nozzle walls and so have complete coupling between the nozzle flow and the wall-normal unsteady heat equation in the solid phase), whereas virtually all industry codes are decoupled. The value of our work has recently been recognized by ATK in that we are writing a follow-up joint STTR proposal with IllinoisRocstar LLC. More details about this collaboration are given in Section 4.6.

4. More Details

4.1 Packing

The packing of arbitrary shape particles was completed. *Rocpack* packs polydisperse particles in cuboids, cylinders, or annuli [1]. The original goal was to pack spherical AP particles to a high packing density. The code can construct packs with prescribed size-distribution statistics, which match those of physically constructed packs. The first version of the code was fully parallel, but recent algorithmic improvements now allow the code to run efficiently on a laptop. Some examples of packs generated by *Rocpack* are shown in Figure 4.

For energetic crystals, we have been working with Dr. Daniel Hooks, Los Alamos National Laboratory (LANL), to generate realistic shapes. Figure 5 shows typical shapes of crystals grown at LANL. Shown below them are corresponding crystal morphologies generated by SHAPE, a software program from Soft Shape (<http://shapesoftware.com/>). The input files were provided by Dr. Hooks. The SHAPE software allows us to generate realistic representations of the energetic crystals. The output from SHAPE is then used as input into our packing code, *Rocpack*. Corresponding packs generated by *Rocpack* are shown in Figure 6. In each pack there are 40 large crystals and 400 smaller crystals, with sizes of 1 and 0.3, respectively. The packs are small and are only meant to be representative of the capabilities of *Rocpack*. The packing fraction is roughly 0.53 for all three packs. The binder regions are shown as voids. Note that our packing code can pack any combination of spheres and crystals. A number of manuscripts detailing the work have been published in archival journals. These are listed in Section 6. A few of the more relevant manuscripts are attached in the Appendix.

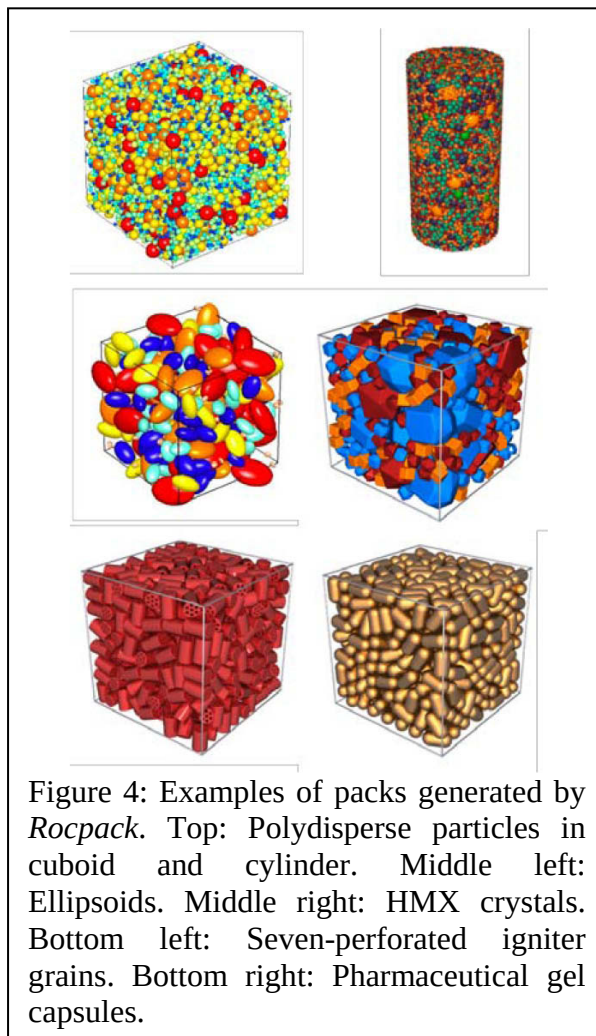


Figure 4: Examples of packs generated by *Rocpack*. Top: Polydisperse particles in cuboid and cylinder. Middle left: Ellipsoids. Middle right: HMX crystals. Bottom left: Seven-perforated igniter grains. Bottom right: Pharmaceutical gel capsules.

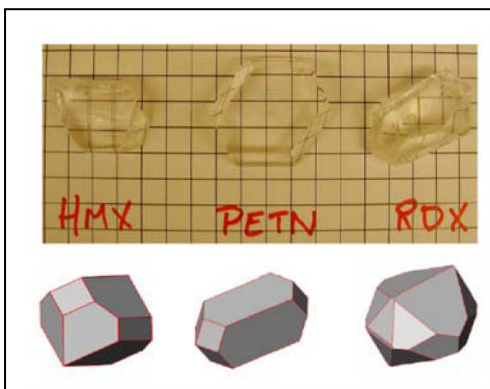


Figure 5. Crystals of HMX, PETN and RDX grown at LANL. Lower images were constructed by *SOFTSHAPE*.

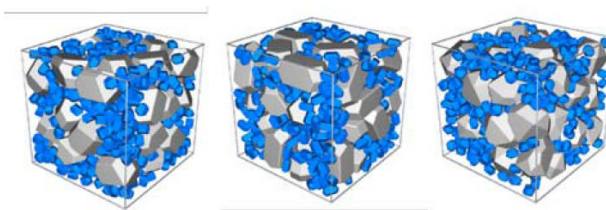


Figure 6. Particle pack morphologies created by *Rocpack*. Packs consist of 40 large and 400 small crystals of HMX (left), PETN (middle), and RDX (right). The packing fraction for all three packs is roughly 0.53.

4.2 Material Properties

Significant progress was made on material modeling and determining the material properties for heterogeneous propellants. To determine the material properties two items are required, namely (i) a representation of the pack, and (ii) a proper mathematical framework that can use the output from the pack (such as 1st, 2nd and 3rd order statistics) as input, and compute the upper and lower bounds of various properties (such as thermal conductivity, Young's modulus, etc.). To address the items two avenues of research were carried out. The first was the development of the computational tools necessary to gather the statistics from either a computational representation of the material, such as using *Rocpack*, or from a micro-CT scan. Figure 7 shows a scanned image of an AP/HTPB propellant. The scan resolution is 2.818 microns per pixel. Since the particle shapes are non-spherical, it was determined that building the proper tools would be easier for spherical glass beads. And so several packs of glass beads were created and then imaged. Voxel data was collected and a histogram constructed. Once the voxel data was collected and analyzed, an equivalent pack of spheres was generated having the same centers as the glass beads. A pack such constructed is hereafter called a "voxel pack", and should be distinguished from a pack generated by *Rocpack*. Since the glass beads are not all exactly spherical, replacing them with spheres creates a pack with some overlap, but this overlap was determined to be less than 0.05%, an acceptable error. Once the pack is constructed the statistics

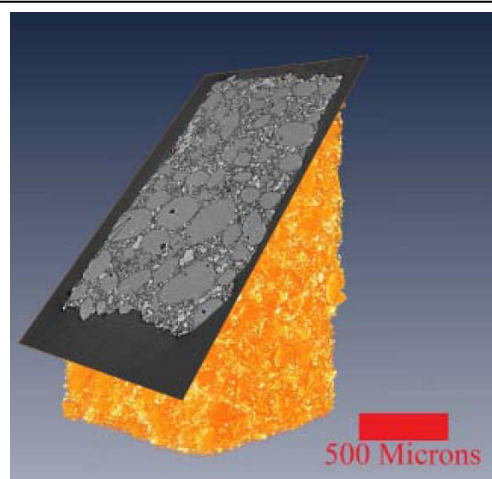


Figure 7. Micro-CT scan of AP-based propellant.

can then be determined. However, the voxel pack is typically large, containing many tens of thousands of particles, and is too large to be used for the material properties code. So a smaller pack is constructed, called a unit cell, which has the same statistics as the voxel pack. Such a pack is constructed using a genetic algorithm with objective functions defined by the statistics. Figure 8 shows the voxel pack constructed from the micro-CT scan of glass beads, the unit cell reconstruction pack, and a histogram comparing the volume fraction of the voxel data, the voxel pack, and the unit cell pack. As expected, the overall agreement is good. Once the unit cell is constructed material properties can be computed. Figure 9 shows Young's modulus for a particular pack. Note that only the lower bound is shown for the numerical results. Computing the upper bound is straightforward and results will be published in the near future.

A manuscript describing the three-dimensional unit cell from micro-CT scans has been written and submitted. The abstract for this paper is given below as Abstract 1, and the submitted manuscript can be found in the Appendix. A manuscript discussing the computation of Young's modulus is currently being written; its abstract appears in Section 3.2.

Abstract 1: In this paper, we present a systematic approach for characterization and reconstruction of statistically optimal representative unit cells of polydisperse particulate composites. Tomomicrography is used to gather rich three-dimensional data of a packed glass beads system. First-, second- and third-order probability functions are used to characterize the morphology of the material, and the parallel augmented simulated annealing

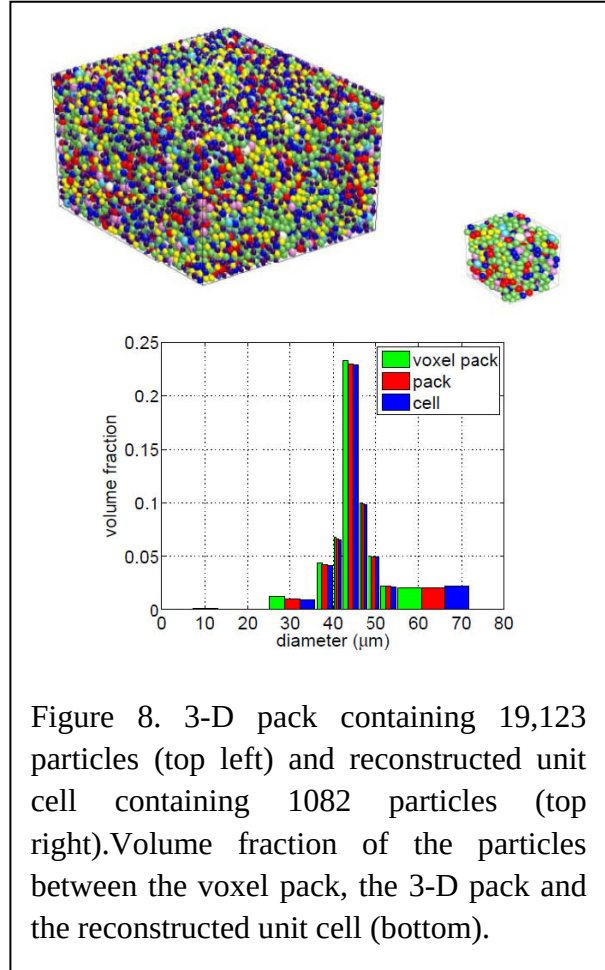


Figure 8. 3-D pack containing 19,123 particles (top left) and reconstructed unit cell containing 1082 particles (top right). Volume fraction of the particles between the voxel pack, the 3-D pack and the reconstructed unit cell (bottom).

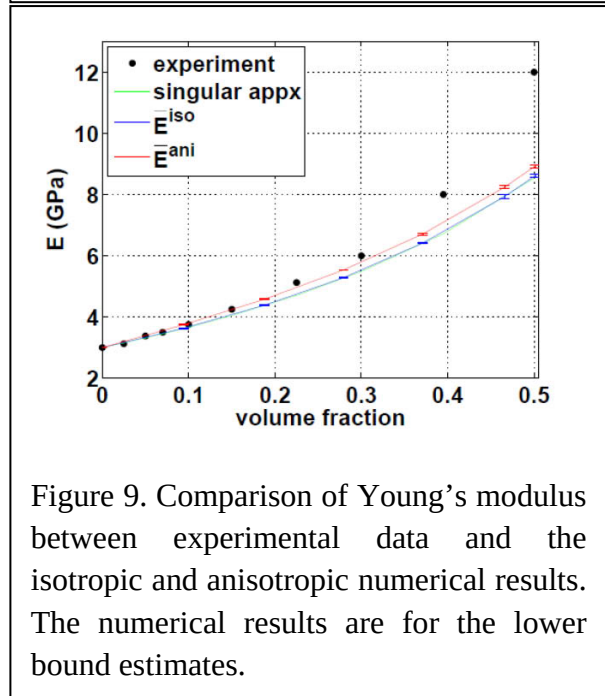


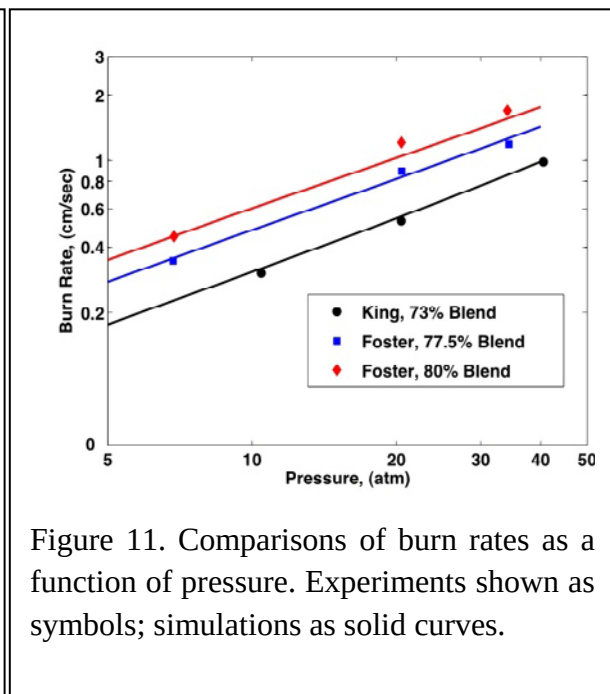
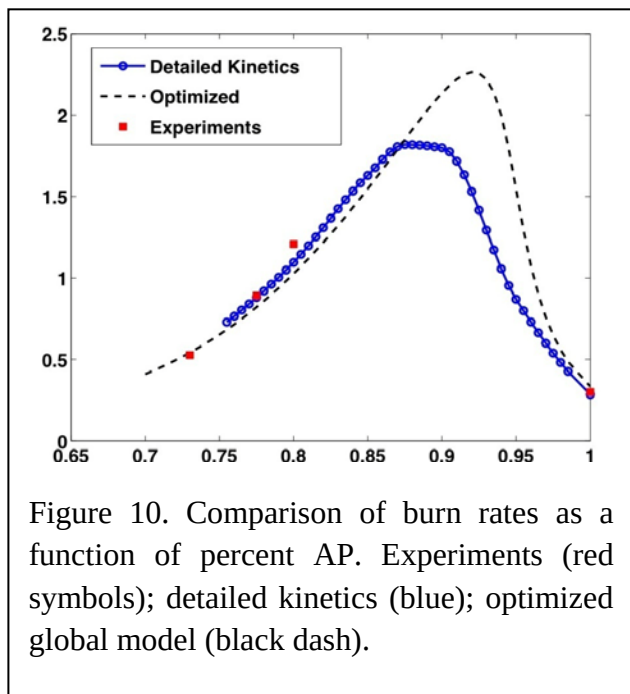
Figure 9. Comparison of Young's modulus between experimental data and the isotropic and anisotropic numerical results. The numerical results are for the lower bound estimates.

algorithm is employed for reconstruction of the statistically equivalent medium. Both the fully resolved probability spectrum and the geometrically exact particle shapes are considered in this study, rendering the optimization problem multidimensional with a highly complex objective function. A ten-phase particulate composite composed of packed glass beads in a cylindrical specimen is investigated, and a unit cell is reconstructed on massively parallel computers. Further, rigorous error analysis of the statistical descriptors (probability-functions) is presented and a detailed comparison between statistics of the voxel-derived pack and the representative cell is made.

4.3 Kinetics Fitting

Most of our work on this problem took place during the Phase I effort, but a brief discussion here is appropriate. The goal is to identify kinetics data (for false kinetics) using a wide variety of available data, and to do this in an optimal fashion, i.e. by minimizing a fitness function. To ensure convergence to the global minimum, rather than a local minimum, a genetics algorithm is used. This is computationally intensive, requiring a parallel environment.

Figure 10 is a comparison of the model burn rates of AP-HTPB blends with detailed kinetics results of Beckstead at a pressure of 20atm. Figure 11 compares burn rates of blends with experimental data.



4.4 Agglomeration

We are able to numerically “burn” three-dimensional propellant packs that contain a modest number of aluminum particles, but with serious limitations on the physics controlling the aluminum. These particles do not melt, they do not reside for any time on the propellant surface (and so do not migrate in the melt layer at the surface) but pass through the surface as the surrounding propellant burns away. But once free of the surface they are subject to fluid mechanical forces, calculated by integrating the surface stress. These forces generate lateral motion and rotation in addition to a mean downstream motion, and videos show that the dynamics is qualitatively similar to experimental observations. A snapshot is shown in Figure 12.

For many propellants the aluminum particles do reside for a time on a surface melt layer, and migrate in that layer. We have a model which examines the surface temperature for a three-dimensional burning pack, and assumes that each aluminum particle is a surface point which migrates because of surface tension variations that arise because of temperature variations. We have shown that, on average (we examine a large number of points) the distance of a point from its starting position is a linear function of time (i.e. a dispersal speed is well defined) for a certain time interval, and then the distance is capped. The concept of a maximum radial migration distance is an important one, for it

means that any one particle, after arriving at the surface at some point, can only come in contact with other particles that come to the surface within a radius from this point that is no more than twice the maximum radial migration distance. Moreover, they must get to the surface in a timely fashion, and the joint migration necessary for contact must occur before particle detachment occurs.

This picture makes it clear that the problem of agglomeration is a complex one, and the numerical challenges are huge. The problem of detachment alone is challenging. Are there short cuts?

Clearly it might be possible to define a small number of fundamental parameters which control agglomeration. These might include: the mean surface residence time of a particle; the mean radial surface migration speed during the time when that speed is well defined; the maximum

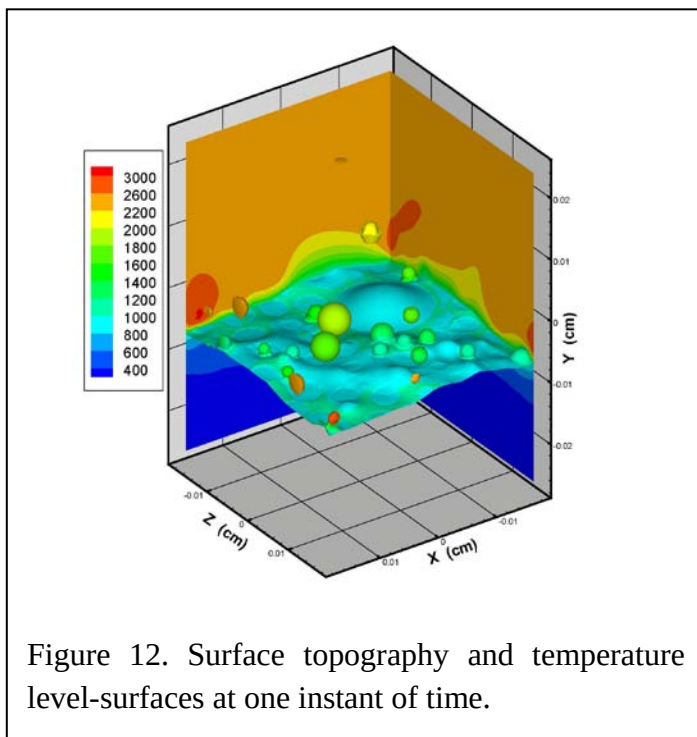


Figure 12. Surface topography and temperature level-surfaces at one instant of time.

radial migration distance; the mean particle diameter; the mean burning rate of the pack. Then, knowing the particle distribution within the pack, determined from *Rocpack*, and the various sizes of these particles, it might be possible to predict the agglomeration statistics, specifically the mean agglomeration size and the distribution about that size. However, it is also quite likely that the distributions about the various means listed here will also play a role. And so the prospects of constructing a predictive tool in this fashion that could be used in the development of a new propellant rapidly fade.

Notwithstanding this, we have examined a one-parameter model within the framework of packs generated using *Rocpack*. This parameter is a distance S_d and whether or not two particles agglomerate depends on their relative locations as measured against S_d , which is chosen to match experimental mean agglomerate sizes. The model predicts the size distribution, essentially log-normal, and, importantly, the standard deviation. Figure 13 shows such a distribution. Clearly the agreement between the measured data and the predictions is good. However, to get such agreement we needed to modify the experimental data by accounting (using a well known model) for aluminum combustion between the surface and the collection point. In general we find that the reported data is often not well defined (pack details important to the theory are not provided), and the role that particle burning might play is not discussed. For these, and other reasons, our hope that we could define a strategy by which S_d could be determined from pack morphology data has not been realized. As we noted in the proposal, this was a high-risk, high-payoff strategy, well worth examining, but one whose failure is not surprising.

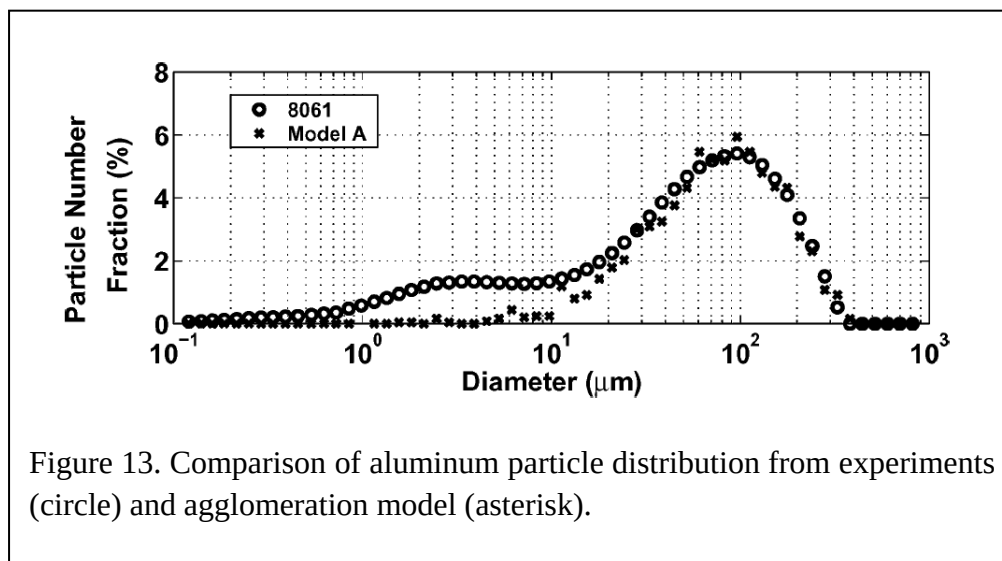


Figure 13. Comparison of aluminum particle distribution from experiments (circle) and agglomeration model (asterisk).

4.5 Morphology effects on the chamber flow

When a heterogeneous propellant burns there are significant flow perturbations (temperature, velocity, etc.) above the surface of the propellant at distances large compared to the extent of combustion, and small compared to the scale of perturbations in the chamber. Figure 14 shows rare images, one obtained at China Lake and one from F. Maggi's laboratory in Italy, of such perturbations, and Figure 15 shows temperature fluctuations that we have obtained in three-dimensional simulations (plots of the normal velocity show the same behavior). That these can affect the chamber flow is apparent from Figure 16 in which time variations in head-end pressure are shown assuming, for the first panel, that the perturbations can be described by white noise (a common assumption in the community), and for the second panel, that the perturbations are generated by propellant morphology. Our primary focus during the STTR has been to examine the effects on the nozzle flow (see section 4.6), and relevant to this is Figure 17 which shows temperature variations at the nozzle inlet of a Bates-15 motor. The latter is a commonly used as a test facility in laboratory work. The perturbations shown here are largely ignored by the community.

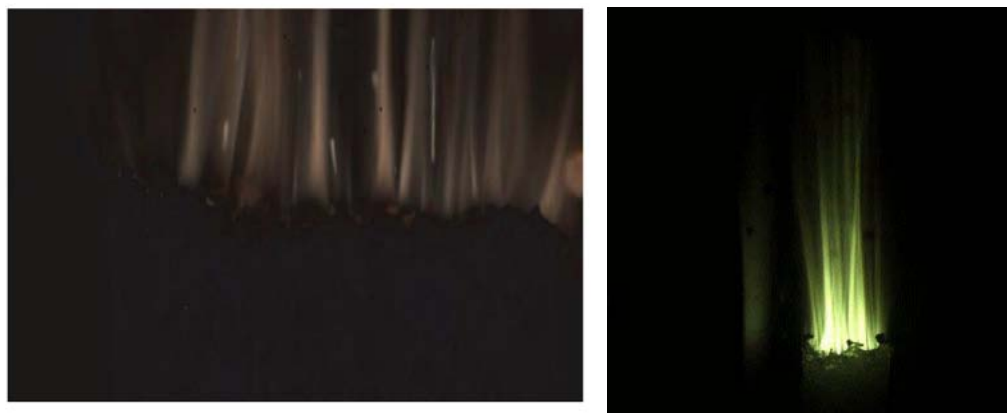
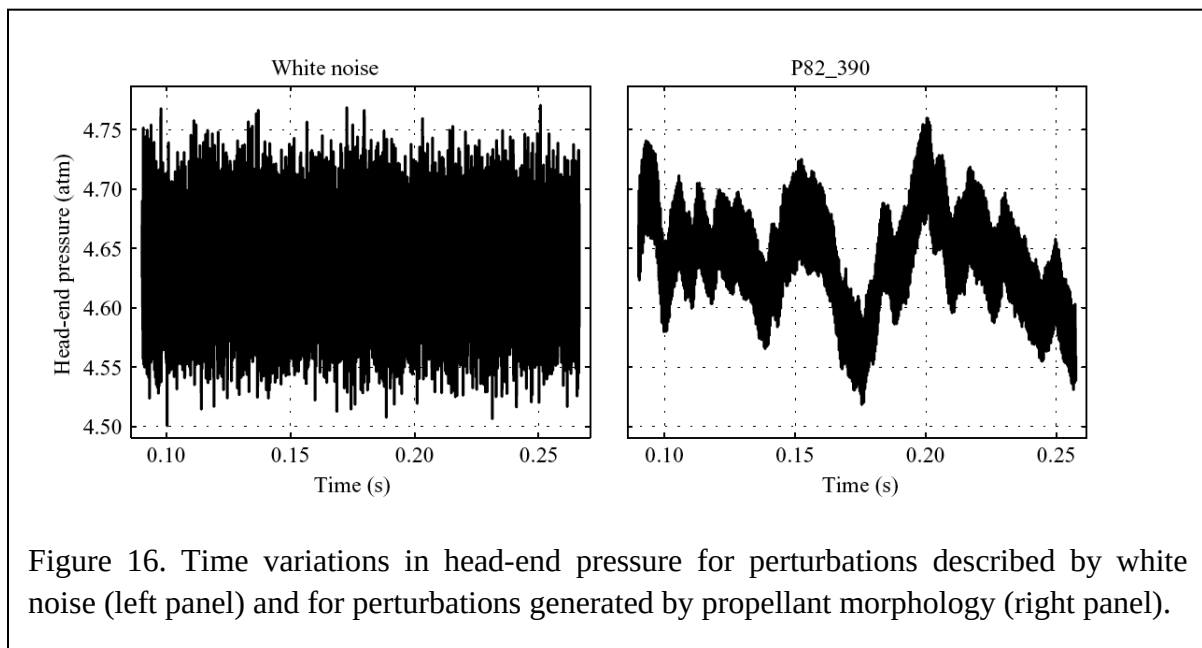
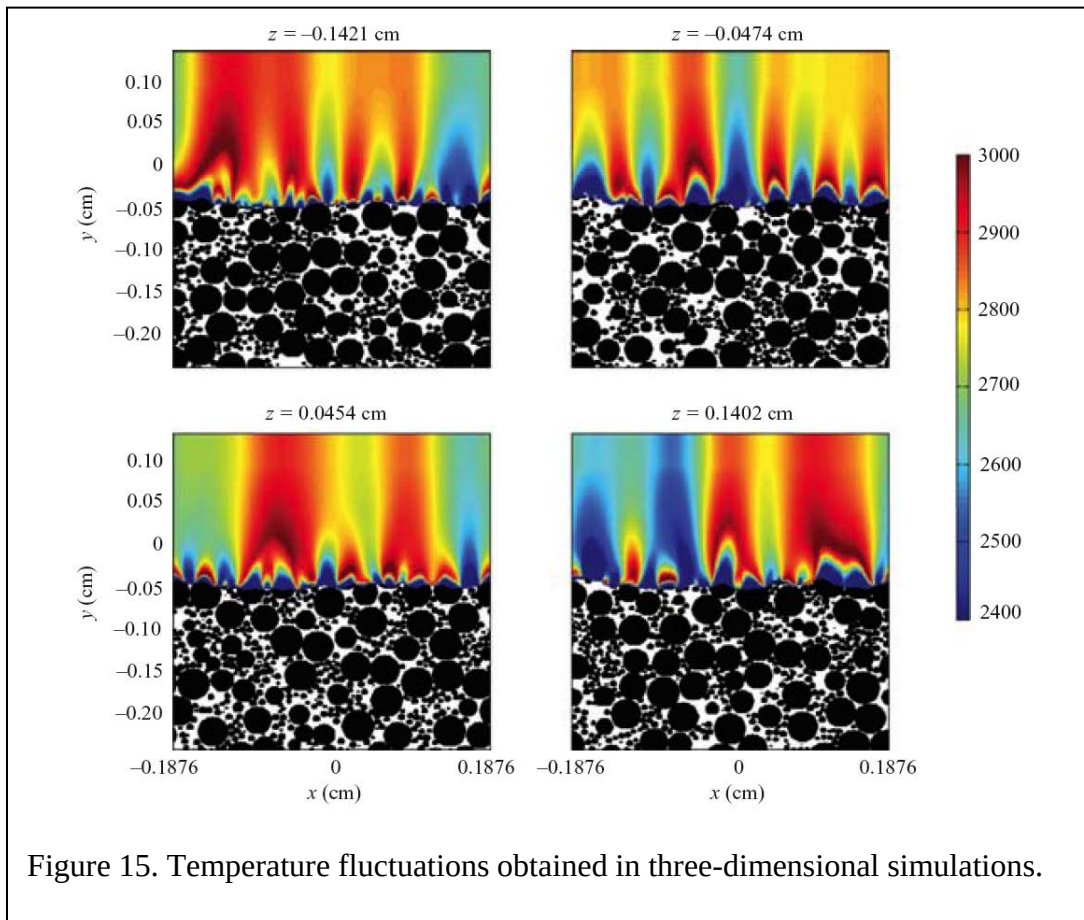
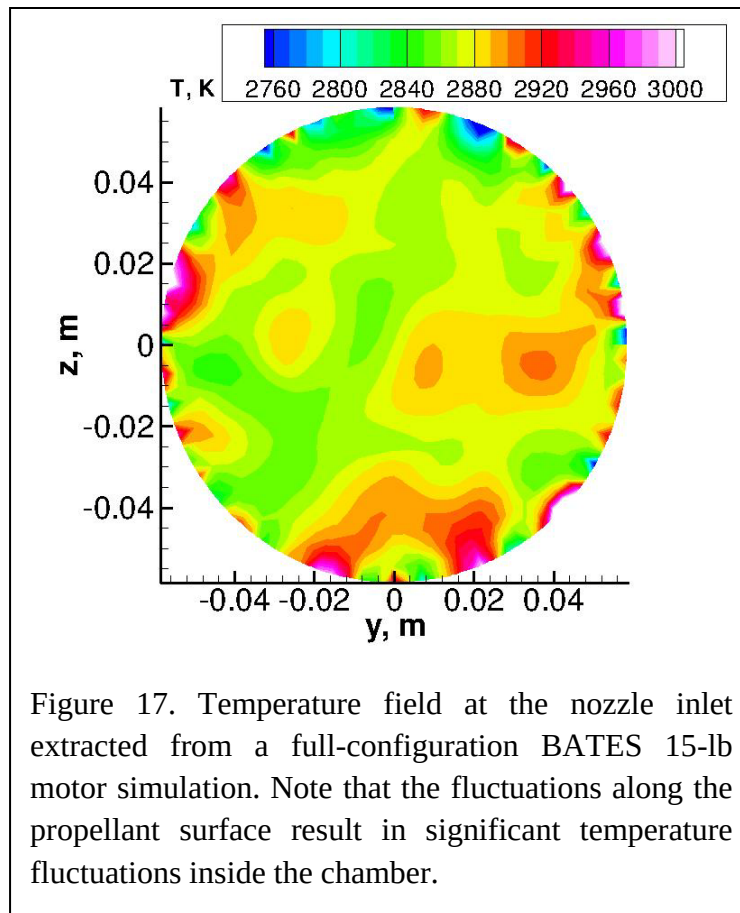


Figure 14. Image from China Lake (left) and image courtesy of F. Maggi (right) showing the near surface fluctuations above a heterogeneous propellant.

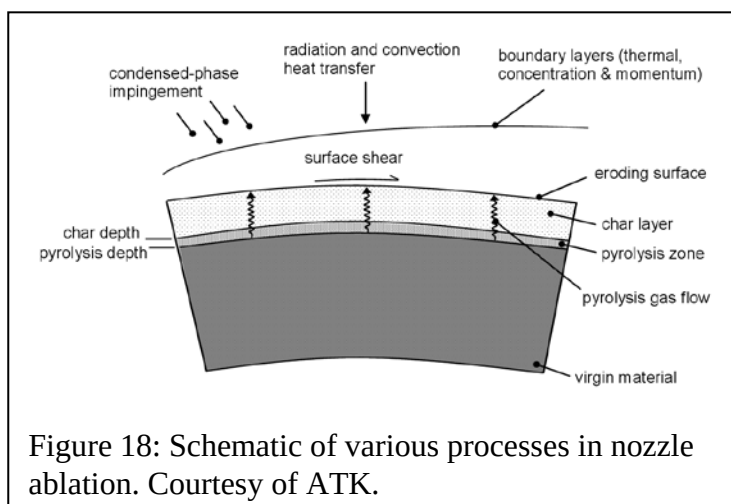




4.6 SRM nozzle erosion

Graphite and carbon/carbon nozzles are often used in solid rocket motors (SRMs) because of their ability to retain structural integrity under extreme thermal environments, and the ease with which they can be machined to achieve a desired geometry [2]. However, when exposed to hot oxidizing chemical species such as H_2O , OH , CO_2 , O , and O_2 , surface erosion can occur. During a long burn this can be sufficient to alter the geometry (including the throat area) to an extent sufficient to alter the rocket performance. For this reason, there is a need for theoretical and numerical studies that can lead to an understanding of the major factors that dictate the erosion rate, and can lead to design improvements. Such studies present serious challenges, for not only must the chamber flow be well described to determine the proper nozzle inlet conditions, itself a challenging problem, but the nozzle flow must be well described, with full resolution of boundary layers.

A simplified schematic is depicted in Figure 18 and summarizes the various physical and chemical processes involved in nozzle erosion/ablation. Chemical, mechanical, and thermal effects coexist within the compressible turbulent boundary layer at the nozzle surface, presenting a challenge to accurately model the momentum and heat transfer to the nozzle walls.



There is a significant literature in the subject, albeit flawed, e.g. [3,4]. Most significantly, in the context of our work, it does not require a description of the chamber flow because the nozzle inlet conditions are assumed to be steady and uniform, only a rough approximation for turbulent flow or for flow generated by the combustion of heterogeneous propellants.

This assumption, part of current industry practice, extends to the inclusion of aluminum oxide particles. Aluminum is commonly used in solid rocket propellants, and oxide particles pass into the nozzle, where their impact on erosion can be significant. The distribution of these particles across the inlet is not uniform, but is assumed to be; and the particles are not all of the same size, but have a size distribution, one that is ignored. We, together with colleagues at the University of Illinois, have developed computational tools that allow for the determination of the detailed chamber flow, along with a detailed description of the oxide particle distribution both in size and location, and so we can use these tools to prescribe nozzle inlet data with a detail heretofore not examined [5,6].

What we cannot do is fully calculate the coupled chamber/nozzle flow whilst accounting for the nozzle boundary layer, simply because of limitations on computational resources. An obvious strategy for dealing with this difficulty is to calculate the chamber flow and the nozzle flow separately: to calculate the chamber flow, including data at the nozzle inlet, using a relatively coarse mesh; and then to use the inlet data, suitably interpolated in space and time, in a fine-mesh calculation of the nozzle flow. The chamber flow can be determined from a full rocket simulation including the nozzle, but with the boundary layers unresolved. See Figure 19 for details.

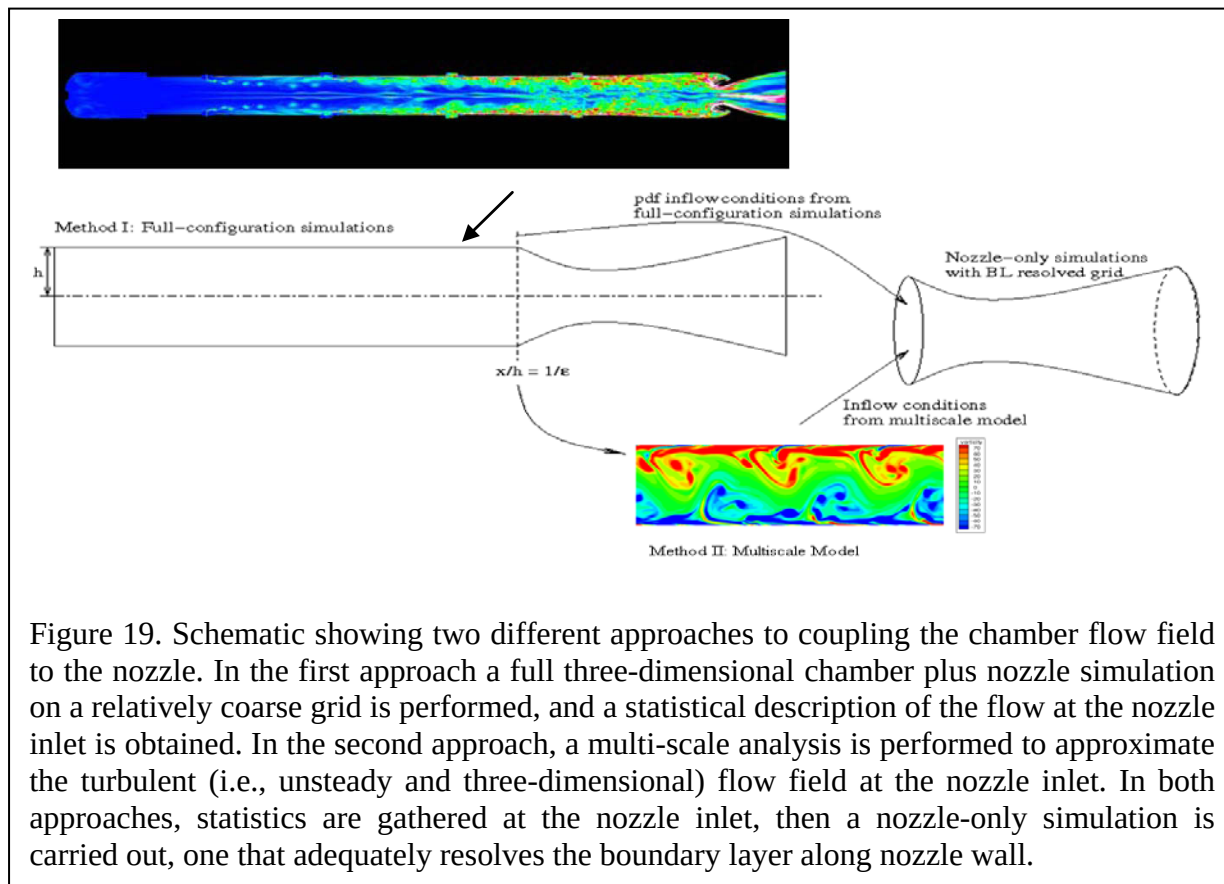


Figure 19. Schematic showing two different approaches to coupling the chamber flow field to the nozzle. In the first approach a full three-dimensional chamber plus nozzle simulation on a relatively coarse grid is performed, and a statistical description of the flow at the nozzle inlet is obtained. In the second approach, a multi-scale analysis is performed to approximate the turbulent (i.e., unsteady and three-dimensional) flow field at the nozzle inlet. In both approaches, statistics are gathered at the nozzle inlet, then a nozzle-only simulation is carried out, one that adequately resolves the boundary layer along nozzle wall.

There are technical difficulties that arise in the pursuit of this strategy, and our efforts presented here are only a partial step towards the ultimate goal. But we have shown that turbulent nozzle inlet data can significantly affect the nozzle flow and the erosion rate; and we have successfully completed a variation of the strategy when the chamber flow is modeled by a three-dimensional cylindrical turbulent flow. In this variation the chamber flow is determined from a chamber-only calculation rooted in the assumption that both the mean flow and the turbulent fluctuations evolve slowly in the axial direction.

The fundamental numerical tool that we have at our disposal, always used when the slowly-varying strategy is not being used (and therefore always used in the nozzle), is *Rocstar*. *Rocstar*

is a tightly coupled fluid-structure-thermal multiphysics solver comprising several physics modules including *Rocburn* and *Rocflo*. *Rocburn* solves the unsteady heat equation in the solid phase in the wall normal direction, allows for gas-solid coupling, and has been modified to accommodate nozzle erosion for our STTR work. Full coupling of the flow field, the unsteady thermal field in the solid, and the retreating solid surface, is not a characteristic of current practice or any published study that we know of.

The code is fully three-dimensional and this enables us to incorporate three-dimensional turbulent flows, and to solve for non-axisymmetric configurations, as, for example, when a thrust vector control nozzle is used. We do make some simplifications. One such is the neglect of gas-phase chemical reactions. Indeed, we account for no gas-phase chemical species, including those oxidizers responsible for the erosion. Instead, we use a familiar pyrolysis law in which erosion is assumed to be a function of the surface temperature and the local pressure. Nor have we yet had time to account for aluminum particles in the gas flow, although we have the tools that would make this possible with modest effort.

4.6.1 Mathematical Formulation

The three-dimensional equations in the gas phase consist of Favre averaged mass, momentum, and energy conservation for a viscous, compressible, ideal gas. A variety of turbulence models, including RANS, LES, and DES are available within the computational framework and provide a closure model for the Reynolds stresses as well as the other turbulent production terms. Although RANS is available, we will not use it since our goal is to investigate unsteady behavior. LES, moreover, is computationally expensive for high Reynolds number flows due to the extremely fine mesh requirement near wall regions, and so we have made limited use of it in our STTR work. Thus our main effort has been the use of a DES (Detached Eddy Simulation) [7].

The nozzle surface is an interface between the condensed phase and the gas phase, and certain connection conditions are imposed there, in addition to the specification of the regression or erosion rate.

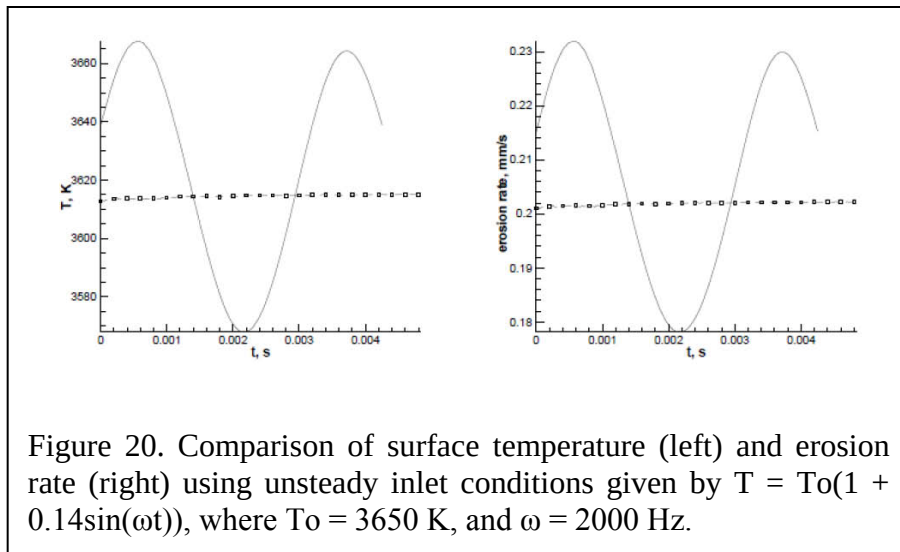
4.6.2 Obtaining Proper Inlet Conditions

As previously stated, the current industry practice is to use steady and uniform inlet conditions. This is a significant shortcoming as the heat transfer at the nozzle wall depends critically on upstream conditions. To remedy this we use one of two approaches. The first approach is based on a full-scale three-dimensional simulation of the rocket chamber plus nozzle; this does not resolve the nozzle boundary layer. The flow field at the nozzle inlet is then captured in space and time as the full-scale simulation proceeds. Afterwards, a statistical description of the flow field at the nozzle inlet, such as joint pdfs in space and time of velocities and temperature, are transferred to the inflow plane of a nozzle-only simulation using a fine enough grid to resolve the boundary layer.

The second approach uses a multi-scale asymptotic model to capture the chamber flow, but otherwise proceeds as in the second approach. Details of the model are given in [8].

4.6.3 Importance of Unsteady Inlet Conditions

To demonstrate the effect of unsteady inlet conditions on the erosion rate on nozzle surface, Figure 20 shows what happens when the injection temperature is perturbed and made periodic in time; i.e., we set $T = T_0(1 + 0.14\sin(\omega t))$, where $T_0 = 3650$ K, and $\omega = 2000$ Hz. The amplitude of 14% of the perturbation in the injection temperature is based on the morphology effect shown in [9]. The temporal evolution of surface temperature and the erosion rate near the throat region are plotted in Figure 20. It is observed that even though the 14% variation in the injection temperature only leads to approximately 1% variations in the nozzle surface temperature, the variation in the erosion rate is as large as 15%.



4.6.4 Importance of Hydrodynamic Instabilities

We have also demonstrated that hydrodynamic instabilities effect nozzle erosion. Figure 21 illustrates unstable chamber flow. Figure 22 shows the consequent fluctuations in temperature at the nozzle inlet. And Figure 23 shows the consequent fluctuations in the erosion rate.

4.6.5 Asymmetric Rockets

We have also examined asymmetric rockets. Thus Figure 23 shows the instantaneous vorticity field inside a gimbaled nozzle with uniform (top left) and turbulent (bottom left) inlet conditions, together with the different erosion results.

4.6.6 Future Work

Although significant progress was made during the Phase II effort, there are still many additional items that are needed before the computational framework can be commercialized and be of use by the rocket industry. Recently a new round of SBIR/STTR request-for-proposals (RFP) were sent have out by DOD (<http://www.dodsbir.net/solicitation/default.htm>) and one of them, MDA A09-T009, specifically requested a commercial tool that can address erosion of nozzles. Several

of the participants of the current Phase II work, namely John Buckmaster and Thomas Jackson, have teamed with IllinoisRocstar LLC (<http://www.illinoisrocstar.com/>) in submitting a new proposal based primarily on the completed work under the current Air Force Phase II work. The proposal details are “High-Fidelity Multiphysics Simulations of Erosion in Solid Rocket Motors”, proposal number MDA-T009-09 #B09B-009-0003. This proposal is accompanied with a letter of support/collaboration with Dr. Mark Ewing of ATK. Had it not been for the current Phase II effort, we would not have been in the position to submit a proposal to MDA.

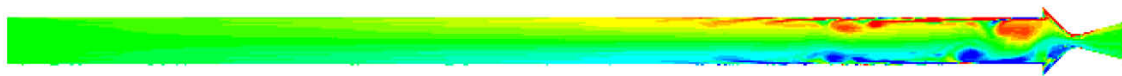


Figure 20. Vorticity contours of planar two-dimensional motor.

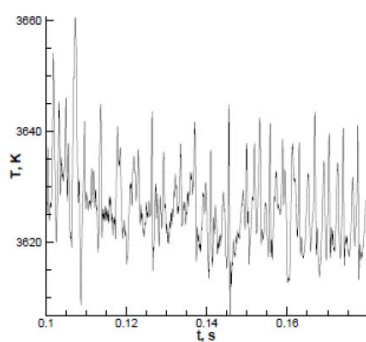


Figure 21. Temperature evolution after the initial transient at nozzle inlet.

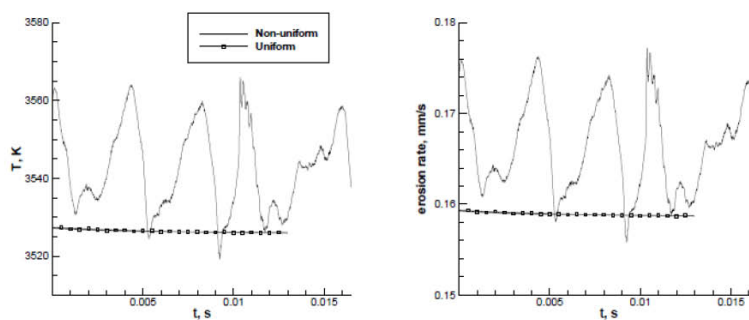


Figure 22. Surface temperature (left) and erosion rate (right) at a point on the nozzle surface near the throat. Also plotted as the dashed line is the solution using uniform inlet conditions.

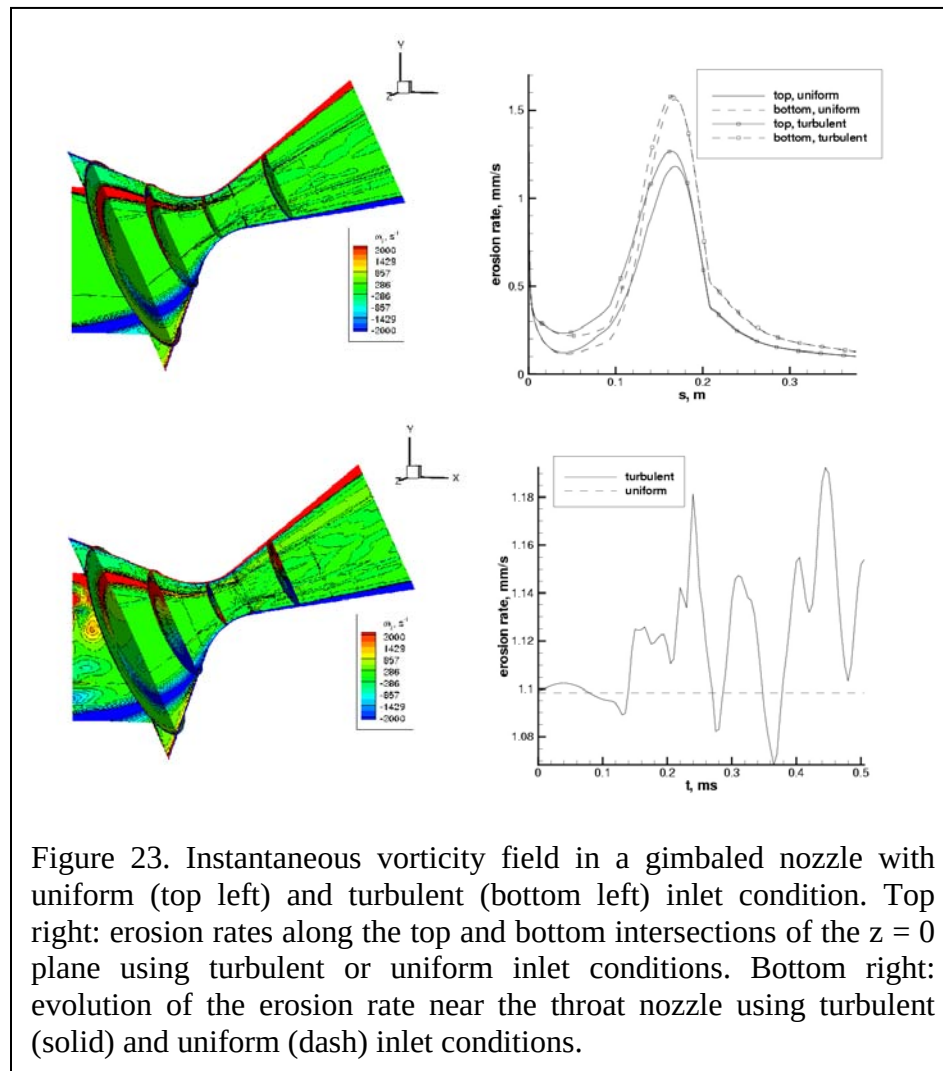


Figure 23. Instantaneous vorticity field in a gimbaled nozzle with uniform (top left) and turbulent (bottom left) inlet condition. Top right: erosion rates along the top and bottom intersections of the $z = 0$ plane using turbulent or uniform inlet conditions. Bottom right: evolution of the erosion rate near the throat nozzle using turbulent (solid) and uniform (dash) inlet conditions.

6. Conferences and Publications

To ensure the work would be as visible as possible to the rocket community at large, a number of conferences were attended and talks given about this work. In addition, several research manuscripts were published in archival journals. The conference publications and archival journal publications are listed separately below.

Conference Publications:

1. K. Matous, F. Maggi, B. Collins, H. Lee and T.L. Jackson, “Experimental and Numerical Modeling of Solid Propellants using Micro-tomography”, 8th World Congress on Computational Mechanics, Venice, Italy, 2008.
2. H. Lee, B. Collins and K. Matous, “Experimental and Numerical Modeling of Heterogeneous Solid Propellants”, ASME International Mechanical Engineering Congress & Exposition, Paper No: IMECE2008-66482, Boston, MA, 2008.
3. B. Collins, F. Maggi, K. Matous, T.L. Jackson and J. Buckmaster, “Using Tomography to Characterize Heterogeneous Propellants”, 46th AIAA Aerospace Sciences Meeting and Exhibit, Paper No: AIAA 2008-941, Reno, NV, 2008.
4. H. Lee and K. Matous, “Computing bounds on thermo-mechanical properties of heterogeneous material using extended Hashin-Shtrikman variational principles with third-order statistics”, 10th US National Congress on Computational Mechanics, Columbus, OH, 2009.
5. M. Brandyberry, A. Tudor, H. Lee and K. Matous, “Three-Dimensional Tomographic Characterization of Heterogeneous Solid Propellants Using n-point Probability Functions”, 10th US National Congress on Computational Mechanics, Columbus, OH, 2009.
6. J. Zhang, T.L. Jackson, F. Najjar and J. Buckmaster (2009) “Multi-physics numerical simulation of erosion in rocket nozzle”, AIAA Paper No. 2009-833, Jan. 2009, in 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL.
7. S. Stafford, F. Maggi, T.L. Jackson and J. Buckmaster (2008) “Propellants Packs”, AIAA Paper No. 2008-0937, Jan. 2008, in 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.
8. F. Maggi, T.L. Jackson and J. Buckmaster (2008) “Aluminum Agglomeration Modeling Using a Packing Code”, AIAA Paper No. 2008-0940, Jan. 2008, in 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.
9. K.N. Hossain, T.L. Jackson and J. Buckmaster (2008) “Three-dimensional Numerical Simulation of Flames Supported by a Spinning Porous Plug Burner”, AIAA Paper No. 2008-1047, Jan. 2008, in 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.
10. T.L. Jackson. (2009) “Issues related to heterogeneous solid-propellant combustion”, Invited paper presented at the 3rd European Conference for Aerospace Sciences (EUCASS), Versailles, France. July 6-9, 2009.
11. J. Zhang, T.L. Jackson, F. Najjar and J. Buckmaster (2009) “Multiphase flow simulations of nozzle erosion in SRM with turbulent inlet conditions. AIAA Paper No. 2009-5499, 45th

AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 2-5 August 2009, Denver, Colorado.

Publications:

1. H. Lee, M. Brandyberry, A. Tudor, and K. Matous, “Three-dimensional reconstruction of statistically optimal unit cells of polydisperse particulate composites from micro-tomography”, Physical Review E, accepted, The American Physical Society (2009).
2. H. Lee and K. Matous, T.L. Jackson and J. Buckmaster, “Computing overall elastic constants of polydisperse particulate composites by Hashin-Shtrikman variational principle from micro-tomographic data”, Composites Science and Technology, in preparation, Elsevier (2009).
3. F. Maggi, S. Stafford, T.L. Jackson and J. Buckmaster (2008) “Nature of packs used in propellant modeling”, Physical Review E., Vol. 77, 046107 (17 pages).
4. J. Zhang and T.L. Jackson (2009) “A high-order incompressible flow solver with WENO”, Journal of Computational Physics, Vol. 228, pp. 2426-2442.
5. Y. Daimon, T.L. Jackson, V. Topalian, J. Freund, J. Buckmaster (2009) “Effect of Propellant Morphology on Acoustics in a Planar Rocket Motor”, Theoretical and Computational Fluid Dynamics, Vol. 23, pp. 63-77.
6. K.N. Hossain, T.L. Jackson, J. Buckmaster (2009) “Numerical Simulations of Flame Patterns Supported by a Spinning Burner”, Proceedings of the Combustion Institute, Vol. 32, pp. 1209-1217.
7. J. Zhang, T.L. Jackson, J. Buckmaster and F. Najjar (2009) “Erosion in Solid-Propellant Rocket Motor Nozzles with Unsteady Non-uniform Inlet Conditions”, submitted to Journal of Propulsion and Power.
8. S. Stafford and T.L. Jackson (2009) “Using level sets for creating virtual random packs of non-spherical convex shapes”, submitted to Journal of Computational Physics.

References

- [1] Maggi, F., Stafford, S., Jackson, T.L. and Buckmaster, J. 2008. "Nature of packs used in propellant modeling". *Physical Review E*, Vol. 77, 046107 (17 pages).
- [2] Acharya, R. and Kuo, K.K., 2007. "Effect of Chamber Pressure and Propellant Composition on Erosion Rate of Graphite Rocket Nozzle," *Journal of Propulsion and Power*, Vol. 23, No. 6, pp. 1242-1254.
- [3] Thakre and Yang, V., 2008. "A Comprehensive Model to Predict and Mitigate the erosion of Carbon-Carbon/Graphite Rocket Nozzles", *Journal of Propulsion and Power*, Vol. 24, No. 4, pp. 822-833.
- [4] Kuo, K. K., S. T. Keswani, 1985. "A Comprehensive Theoretical Model for Carbon-Carbon Composite Nozzle Recession", *Combustion Science and Technology*; 42: 145-164.
- [5] Dick W.A., R.A. Fiedler and M.T. Heath, 2006. "Building *Rocstar*: Simulation science for solid propellant rocket motors". AIAA2006-4590.
- [6] Najjar, F. M., Massa, L., Fiedler, R., Haselbacher, A., Wasistho, B., Balachandar, S., and Moser, R. D., 2005. "Effects of Droplet Loading and Sizes in Aluminized BATES Motors: A Multiphysics Computational Analysis," 41st AIAA JPC, AIAA 2005-3997.
- [7] Wasistho, B., and Moser, R. D., 2005. "Simulation Strategy of Turbulent Internal Flow in Solid Rocket Motor", *Journal of Propulsion and Power*, vol. 21, no. 2, pp. 251-263.
- [8] Zhang, Ju, T.L. Jackson, F. Najjar, J. Buckmaster. 2009. "High-fidelity multiphysics simulations of erosion in SRM nozzles", AIAA 2009-5499, 45th AIAA JPC, Denver CO.
- [9] Massa, L., Jackson, T. L., and Buckmaster, J. 2007. "Fluctuations above a burning heterogeneous propellant," *Journal of Fluid Mechanics*, Vol. 581, pp. 1-32.

Appendix

In this Appendix are attached some relevant publications which gives more details of the work. In order of appearance, these are:

- A1. User Manual for *Rocpack*
- A2. F. Maggi, S. Stafford, T.L. Jackson and J. Buckmaster. Nature of packs used in propellant modeling. *Physical Review E*, 77, 046107 (2008).
- A3. S. Stafford and T.L. Jackson. Using level sets for creating virtual random packs of non-spherical convex shapes. Submitted to *Journal of Computational Science* (2009).
- A4. Ju. Zhang, T.L. Jackson, J. Buckmaster and F. Najjar. Erosion in Solid-propellant Rocket Motor Nozzles with Unsteady Non-uniform Inlet Conditions. Submitted to *Journal of Propulsion and Power* (2009).
- A5. H. Lee, M. Brandyberry, A. Tudor and K. Matous. Three-dimensional reconstruction of statistically optimal unit cells of polydisperse particulate composites from microtomography. Accepted, *Physical Review E* (2009).

User Manual for *Rocpack*

1 Introduction

Random packs of spheres are used to model heterogeneous and porous material morphologies such as concrete, sand, coal, porous explosives, and solid rocket propellants, to name a few. The most obvious advantage of using spheres lies in the simplicity of their representation, which facilitates theoretical, experimental, and computational efforts in the physical sciences. Nevertheless, spheres can be a poor approximation to heterogeneous and porous materials when various other properties are of interest. Bulk materials properties such as thermal conductivity and elastic properties strongly depend on the statistical details of the micro-structure, and spheres do not properly replicate these statistics.

In recent years we have constructed closely packed spheres using the Lubachevsky-Stillinger algorithm to generate models of heterogeneous solid propellants. Improvements to the algorithm now allow us to pack non-spherical convex shapes for modeling heterogeneity in complex energetic materials such as plastic-bonded explosives and pressed gun propellants. ***Rocpack*** is the name of the resulting packing code. Details of the packing algorithm can be found in [1, 2, 3, 4, 5]. An example of a polydisperse pack of spheres is shown in Figure (1).

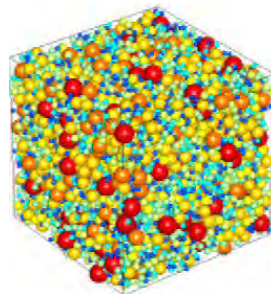


Figure 1: A 10,001 particle pack.

2 Capabilities

Rocpack is able to pack a variety of convex shapes (spheriods, cylinder, gelcap, icosahedron, cube, crystals) in a container. The packing code is nondimensional so that only the relative sizes of the particles are needed as inputs. For example, if one wants to generate a pack of 200 mm, 100 mm,

and 50 mm glass beads (spheres), the relative sizes would be 1, 0.5, and 0.25, respectively. The packing code takes as input either the physical size (then computes relative sizes) or relative sizes.

Currently the container shape can be a cuboid, cylinder, or an annulus. For the annulus a web thickness must be given; the web thickness is the radial thickness of the particles and the hole is therefore 1 - web thickness. The domain for the cuboid is $[-1, +1] \times [-1, +1] \times [-1, +1]$.

3 Inputs for GUI

The parameters and their respective ranges for the GUI are given in Table 1. Some examples of packs generated by **Rocpack** are given in Figure (2). The full code does not have these restrictions.

Initial Temperature	500 – 5000
Number of Particles	1 – 4
Domain Type	Periodic or Fixed
Shape	Cuboid, Cylinder, or Annulus
Random Seed Value	1 – 5000
Aspect ratio (height/width)	1 – 5
Annulus web thickness	0.2 – 0.8
Ending Volume Fraction	0.2 – 0.8

Table 1: Parameter values and ranges for GUI.

Some additional notes are as follows:

1. The Initial Temperature is roughly inversely proportional to the growth rate of a particle, so that the lower the value the faster the growth rate. Having a value set too low (i.e., to 1) will result in a pack that jams quickly. Setting the value too high (i.e., to 100,000) will result in a lattice pack rather than a random pack. The best range is 100 – 1000 for random packs.
2. The web based GUI will only allow the user to input up to 10,000 particles. **Rocpack**, however, can handle an arbitrary number of particles. We have packed as many as 1,000,000 spheres.

3. **Rocpack** can handle any number of particle types and sizes; however, the web based GUI only allows for 4 particle types with corresponding size.
4. To generate different random packs with the same size and size distribution of particles, change the random seeding.
5. The Annulus web thickness flag is only used if the Shape flag has been set to “Annulus”; otherwise it is ignored. Currently only spheres can be packed into an annulus.
6. The web-based GUI has a time limit of five minutes. If the packing fraction is not met within this time limit, the pack will prematurely stop. **Rocpack** in normal mode can run for hours or days.

4 System Requirements

Rocpack is written in c++ and is compiled using g++. There is a Readme file associated with the files that gives specific instructions on how to compile the code. There are actually two versions of **Rocpack**; one packs only spheres and the other packs convex shapes. Having two codes is beneficial because the packing of spheres requires the functional evaluation of an analytic expression (the distance between two spheres) to find the closest neighbor while the code for convex shapes requires a root finding procedure, which is much slower when only spheres are present.

To compile the code for convex shapes, one needs to install a small package in order to compile. “NumPy” is an extension to the Python programming language which is used to calculate various operations in the convex version of **Rocpack**. To install NumPy on Red Hat Linux systems, Fedora, or CentOS, do “yum numpy”. On Debian, its variants and Ubuntu, do “sudo apt-get install numpy”. To manually install numpy, go to <http://numpy.scipy.org/numpydoc/numpy-3.html> and follow the instructions on the website.

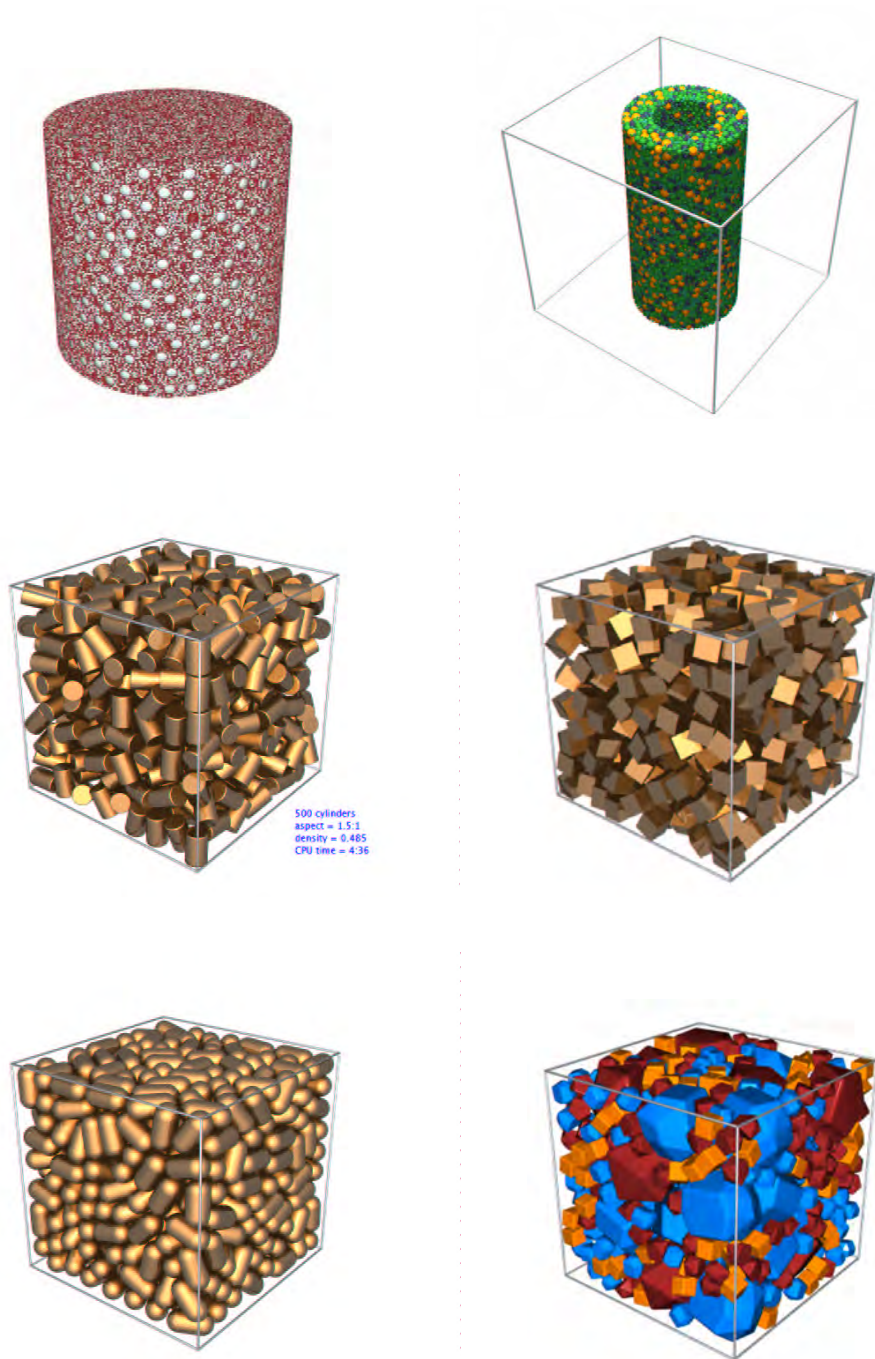


Figure 2: Examples of packs generated by *Rocpack*.

Example 1

The first example has 2 spheres with relative sizes of 1 and 0.6, respectively. The number of particles has been set to 2. There are 100 of the large spheres and 400 of the smaller spheres. The initial temperature is set to 1000, the random seed to 13, and the ending packing fraction to 0.7. The Shape flag has been set to “Cuboid” with aspect ratio of 1. The domain type flag is set to “Fixed Domain”, which means that the walls are treated as solid walls. Figure (3) shows a screen shot of the input. The output is shown in Figure (4). Note that the pack jammed at a packing fraction of 0.591146, which is below the user requested value of 0.7. Increasing the initial temperature to 5,000 results in a packing fraction of 0.595141. Thus we see that increasing the initial temperature, which is roughly equivalent to setting a slower growth rate of the particles, results in a slightly higher packing fraction. If we now set the domain type to “Periodic Domain” and keeping the initial temperature at 5,000, the resulting packing fraction now becomes 0.660592.

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.
 Range of acceptable values in the demo program :
 Initial temperature : [500, 5000]
 Maximum number of particles in each individual fields :
 Sphere : 3000 particles
 Gelcap : 50 particles
 Cylinder : 90 particles
 Icosahedron : 50 particles
 Cube : 50 particles
 Random Seed Value : [1, 5000]
 Aspect Ratio (height/width) : [1, 5]
 Annulus Web Thickness : [0.2, 0.6]
 Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles
 Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	100	Sphere
Particle 2 :	0.6	400	Sphere
Particle 3 :			Sphere
Particle 4 :			Sphere

Domain type : ☐ Periodic Domain ☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria
 Ending Volume Fraction :

Figure 3: Input for Example 1.

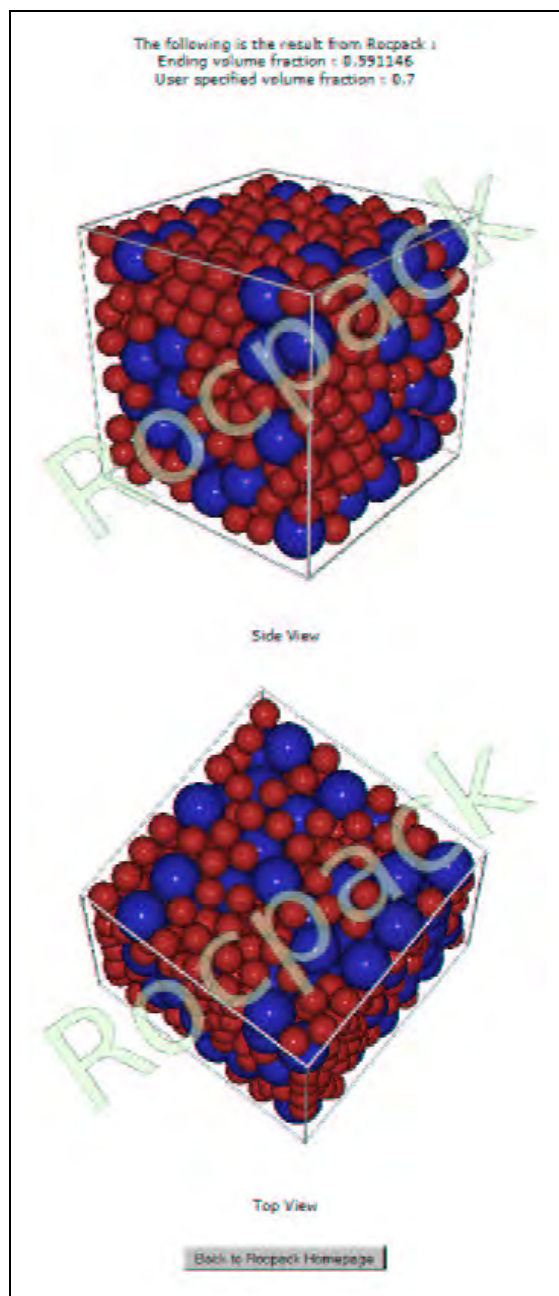


Figure 4: Output for Example 1.

Example 2

The second example has 3 particles, with relative sizes of 1, 0.4, and 0.2 with particle numbers 100, 1000, and 4000, respectively. The number of particles has been set to 3. The initial temperature is set to 1000, the random seed to 13, and the ending packing fraction to 0.7. The Shape flag has been set to “Cuboid” with aspect ratio of 1. The domain type flag is set to “Fixed Domain”, which means that the walls are treated as solid walls. Figure (5) shows a screen shot of the input and the output is shown in Figure (6). Note that the pack jammed at a packing fraction of 0.531005, which is below the user requested value of 0.7. In this case the time limit of five minutes was exceeded and the pack prematurely exited.

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

- Initial temperature : [500, 5000]
- Maximum number of particles in each individual fields :
 - Sphere : 3000 particles
 - Gelcap : 50 particles
 - Cylinder : 50 particles
 - Icosahedron : 50 particles
 - Cube : 50 particles
- Random Seed Value : [1, 5000]
- Aspect Ratio (height/width) : [1, 5]
- Annulus Web Thickness : [0.2, 0.8]
- Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles:

Number of types of Particles

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	100	Sphere
Particle 2 :	0.4	1000	Sphere
Particle 3 :	0.2	4000	Sphere
Particle 4 :			Sphere

Domain type :

☐ Periodic Domain

☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria

Ending Volume Fraction :

Figure 5: Input for Example 2.

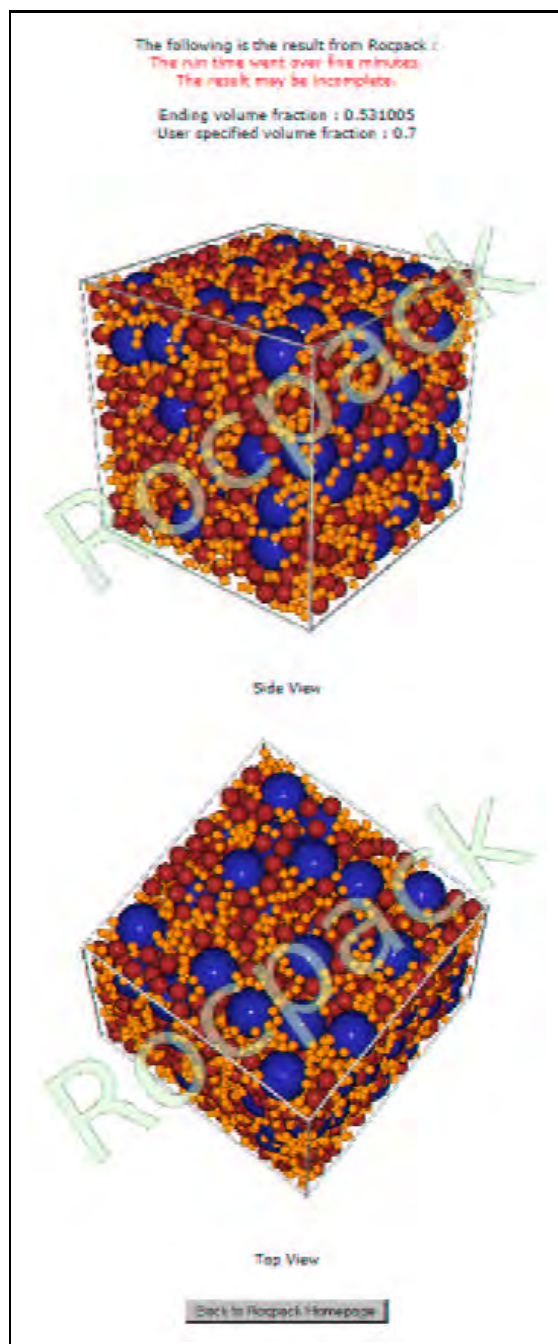


Figure 6: Output for Example 2.

Example 3

The third example has 3 particles, with relative sizes of 1, 0.8, and 0.7 with particle numbers of 50 for each shape. The Shape of Particles flag has been set to “Gelcap”, “Cube”, and “Cylinder”, respectively. The number of particles has been set to 3. The initial temperature is set to 500, the random seed to 13, and the ending packing fraction to 0.7. The Shape flag has been set to “Cuboid” with aspect ratio of 1. The domain type flag is set to “Fixed Domain”, which means that the walls are treated as solid walls. Figure (7) shows a screen shot of the input and the output is shown in Figure (8). Note that the pack jammed at a packing fraction of 0.256684, which is below well the user requested value of 0.7. In this case the time limit of five minutes was exceeded and the pack prematurely exited. It takes significantly more cpu time to generate a pack with convex shapes that to generate a pack of spheres.

Rcpack Demo Program

The demo version of Rcpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

- Initial temperature : [500, 5000]
- Maximum number of particles in each individual fields :
 - Sphere : 3000 particles
 - Gelcap : 50 particles
 - Cylinder : 50 particles
 - Icosahedron : 50 particles
 - Cube : 50 particles
- Random Seed Value : [1, 5000]
- Aspect Ratio (height/width) : [1, 5]
- Annulus Web Thickness : [0.2, 0.6]
- Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles:

Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	50	Gelcap
Particle 2 :	0.8	50	Cube
Particle 3 :	0.7	50	Cylinder
Particle 4 :			Sphere

Domain type : ☐ Periodic Domain ☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria

Ending Volume Fraction :

Figure 7: Input for Example 3.

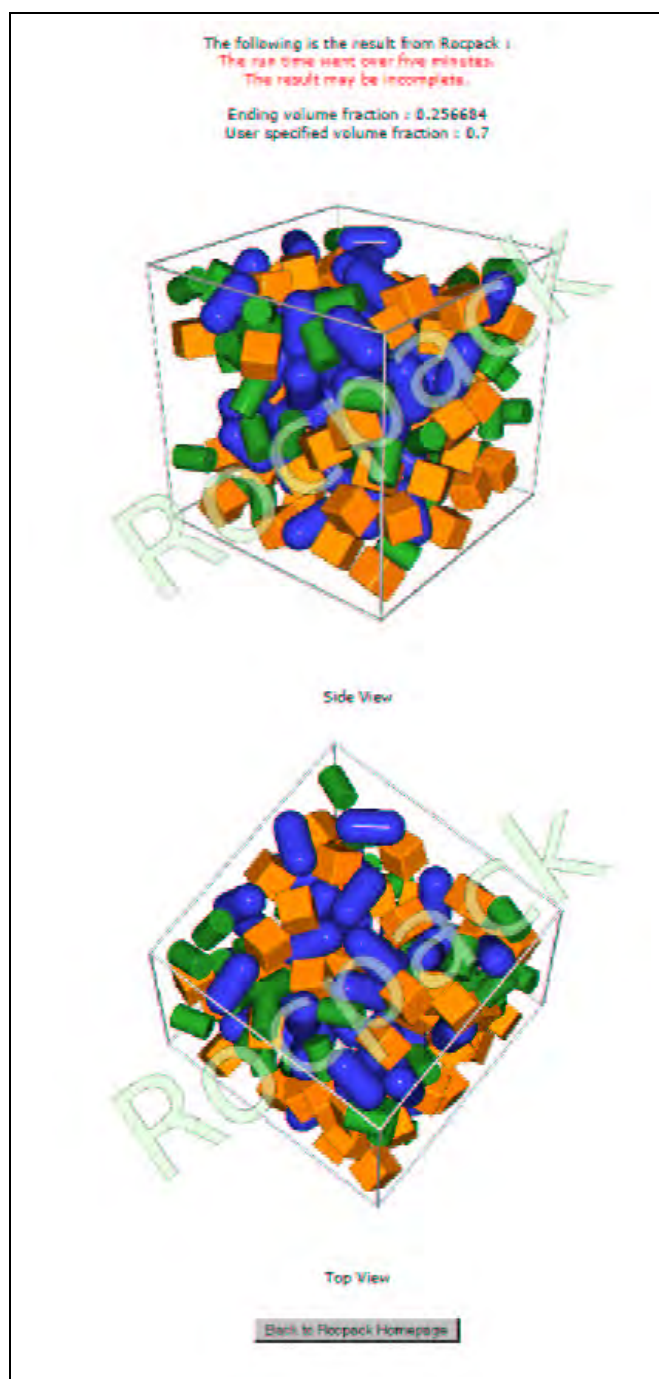


Figure 8: Output for Example 3.

Example 4

The fourth example has 4 particles, with Shape of Particles flag set to “Cube”, “Gelcap”, “Cylinder”, and “Icosahedron”, respectively. Figure (9) shows a screen shot of the input and the output is shown in Figure (10).

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

Initial temperature : [500, 5000]

Maximum number of particles in each individual fields :

Sphere : 2000 particles

Gelcap : 50 particles

Cylinder : 50 particles

Icosahedron : 50 particles

Cube : 50 particles

Random Seed Value : [1, 5000]

Aspect Ratio (height/width) : [1, 5]

Annulus Web Thickness : [0.2, 0.8]

Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles

Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	50	Cube
Particle 2 :	1	50	Gelcap
Particle 3 :	1	50	Cylinder
Particle 4 :	1	50	Icosahedron

Domain type : ☐ Periodic Domain ☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria

Ending Volume Fraction :

Figure 9: Input for Example 4.

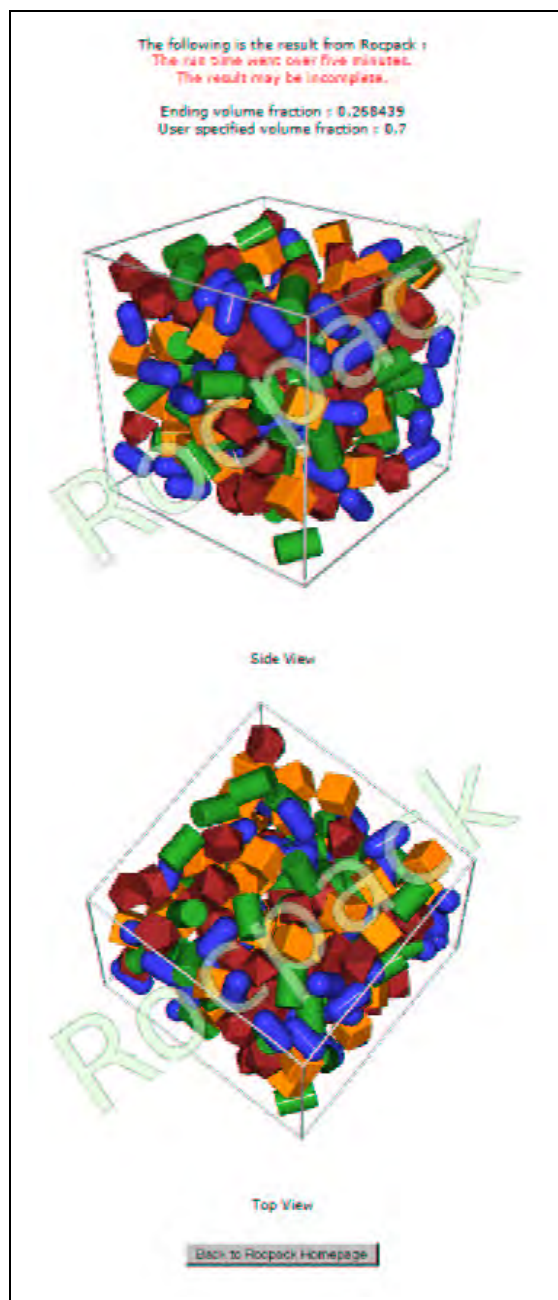


Figure 10: Output for Example 4.

Example 5

The fifth example has 3 particles, with Shape of Particles flag set to “Sphere”. The Shape flag has been set to “Cylinder”. Figure (11) shows a screen shot of the input and the output is shown in Figure (12).

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

Initial temperature : [500, 5000]

Maximum number of particles in each individual fields :

Sphere : 3000 particles

Gelcap : 50 particles

Cylinder : 50 particles

Icosahedron : 50 particles

Cube : 50 particles

Random Seed Value : [1, 5000]

Aspect Ratio (height/width) : [1, 5]

Annulus web Thickness : [0.2, 0.8]

Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles :

Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	100	Sphere
Particle 2 :	6	1000	Sphere
Particle 3 :	5	1000	Sphere
Particle 4 :			Sphere

Domain type : ☐ Periodic Domain ☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria

Ending Volume Fraction :

Figure 11: Input for Example 5.

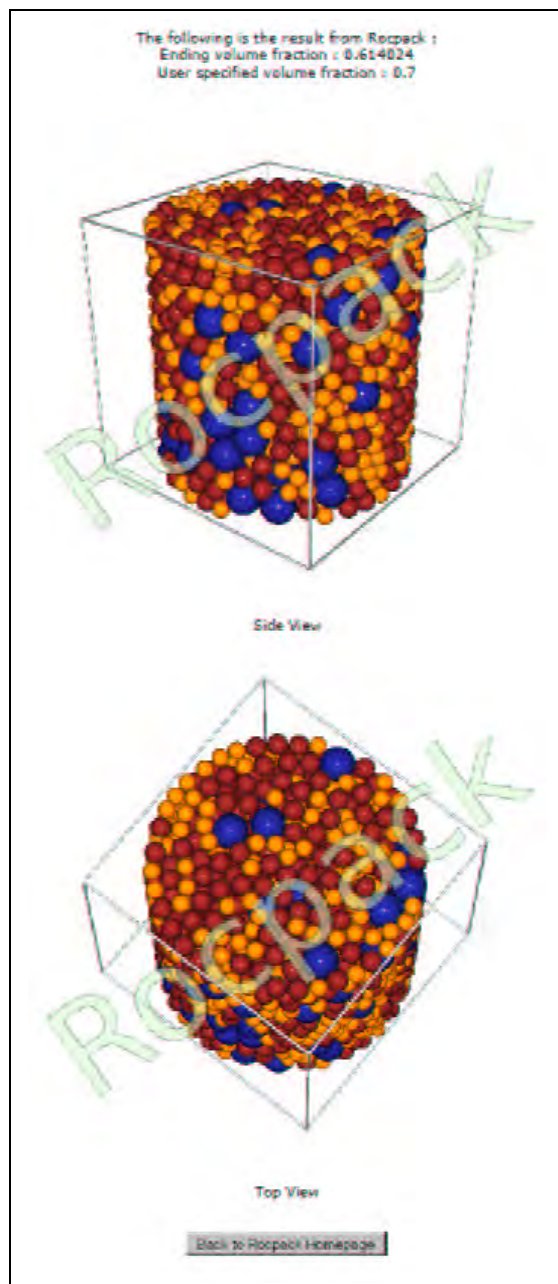


Figure 12: Output for Example 5.

Example 6

The sixth example has 2 spherical particles packed in an annulus. The Shape flag has been set to “Annulus”. The Annulus web thickness flag has been set to “0.4”. Figure (13) shows a screen shot of the input and the output is shown in Figure (14).

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

- Initial temperature : [500, 5000]
- Maximum number of particles in each individual fields :
 - Sphere : 3000 particles
 - Gelcap : 50 particles
 - Cylinder : 50 particles
 - Icosahedron : 50 particles
 - Cube : 50 particles
- Random Seed Value : [1, 5000]
- Aspect Ratio (height/width) : [1, 5]
- Annulus Web Thickness : [0.2, 0.8]
- Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles

Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	1000	Sphere
Particle 2 :	4	3000	Sphere
Particle 3 :			Sphere
Particle 4 :			Sphere

Domain type : ☒ Periodic Domain ☐ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria

Ending Volume Fraction :

Figure 13: Input for Example 6.

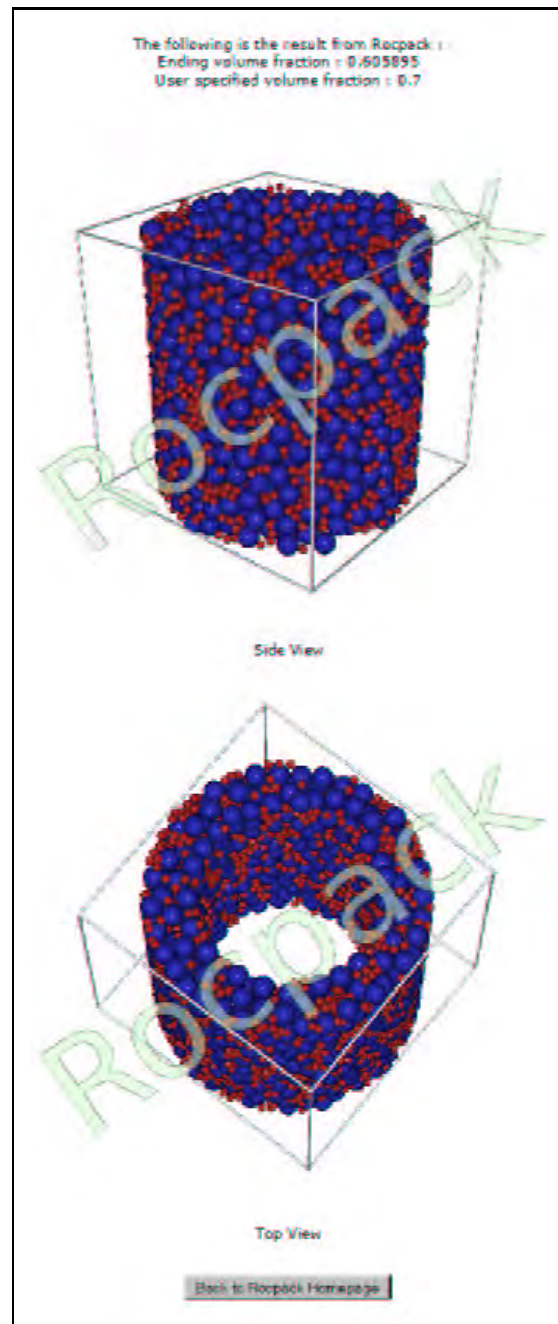


Figure 14: Output for Example 6.

Example 7

The seventh and final example has 2 spherical particles packed in a rectangle with aspect ratio of 2. The Shape flag has been set to “Cuboid”. Figure (15) shows a screen shot of the input and the output is shown in Figure (16).

Rocpack Demo Program

The demo version of Rocpack is limited to five minutes of running time.

Range of acceptable values in the demo program :

- Initial temperature : [500, 5000]
- Maximum number of particles in each individual fields :
 - Sphere : 3000 particles
 - Gelcap : 50 particles
 - Cylinder : 50 particles
 - Icosahedron : 50 particles
 - Cube : 50 particles
- Random Seed Value : [1, 5000]
- Aspect Ratio (height/width) : [1, 5]
- Annulus Web Thickness : [0.2, 0.8]
- Ending Volume Fraction : [0.2, 0.8]

Initial Temperature :

Particles :

Number of types of Particles :

Particle Number	Relative Sizes	Number of Particles	Shape of Particles
Particle 1 :	1	1000	Sphere
Particle 2 :	4	3000	Sphere
Particle 3 :			Sphere
Particle 4 :			Sphere

Domain type :
☐ Periodic Domain
☒ Fixed Domain

Shape :

Random Seed Value :

Aspect ratio (height/width) :

Annulus web thickness :

Stopping criteria :

Ending Volume Fraction :

Figure 15: Input for Example 7.

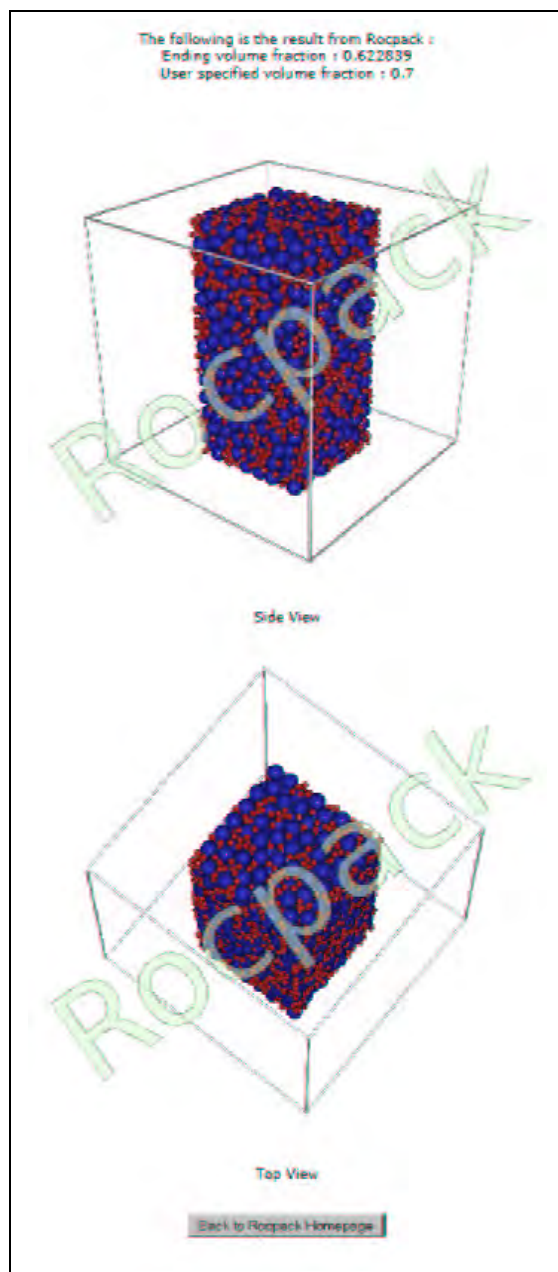


Figure 16: Output for Example 7.

References

- [1] B.D. Lubachevsky and F.H. Stillinger. “Geometric properties of random disk packings”, *J Stat Phys*, **60**(5) 561-583, 1990.
- [2] G.M. Knott, T.L. Jackson, J. Buckmaster. “The random packing of heterogeneous propellants”, *AIAA Journal* **39**(4) 678-686, 2001.
- [3] S. Kochevets, J. Buckmaster, T.L. Jackson, A. Hegab. “Random packs and their use in the modeling of heterogeneous solid propellant combustion”, *Journal of Propulsion & Power* **17**(4) 883-891, 2001.
- [4] F. Maggi, S. Stafford, T.L. Jackson, J. Buckmaster. “Nature of packs used in propellant modeling”, *Physical Review E*, **77**, 046107, 2008.
- [5] S. Stafford and T.L. Jackson. “Using level sets for creating virtual random packs of non-spherical convex shapes”, *Journal of Computational Physics*, accepted.

Nature of packs used in propellant modeling

F. Maggi, S. Stafford, and T. L. Jackson

Center for Simulation of Advanced Rockets, University of Illinois, Urbana, Illinois 61801, USA

J. Buckmaster*

Buckmaster Research, 2014 Boudreau Drive, Urbana, Illinois 61801, USA

(Received 13 August 2007; revised manuscript received 21 January 2008; published 11 April 2008)

In recent years we have constructed closely packed spheres using the Lubachevsky-Stillinger algorithm to generate morphological models of heterogeneous solid propellants. Improvements to the algorithm now allow us to create large polydisperse packs on a laptop computer, and to create monodisperse packs with packing fractions greater than 70% which display significant crystal order. The use of these models in the physical context motivates efforts to examine in some detail the nature of the packs, including certain statistical properties. We compare packing fractions for binary packs with long-known experimental data. Also, we discuss the near-neighbor number and the radial distribution function (RDF) for monodisperse packs and make comparisons with experimental data. We also briefly discuss the RDF for bidisperse packs. We also consider bounded monodisperse packs, and pay particular attention to the near-wall structure where we identify significant order.

DOI: [10.1103/PhysRevE.77.046107](https://doi.org/10.1103/PhysRevE.77.046107)

PACS number(s): 89.90.+n

I. MODELING OF COMPOSITE PROPELLANTS: INTRODUCTION

Heterogeneous solid propellants are widely used in the rocket industry, and are likely to play an important role for as long as rockets are built. Fundamentally, they consist of oxidizer particles of order 1–100 μm diam embedded in a rubbery fuel binder. Various choices are possible, but a common one is ammonium perchlorate (AP) in hydroxy-terminated-polybutadiene (HTPB), and we imply no lack of generality if we place our discussion in this context. These components burn in a thin combustion layer, a few hundred microns thick, in the neighborhood of the propellant surface. It is also common to add metal particles to the binder, 10 μm or so in diameter, aluminum being the most common choice. These burn in the chamber gases at distances well removed from the surface.

Designers of rockets are concerned with a number of propellant-related issues, including the burning rate, the thermal and mechanical properties of the propellant, and, for metallized propellants, the behavior of the metal particles at the surface, including agglomeration. A study of these in a virtual engineering framework starts with a model for the morphology, and the relationship of this model to the physical reality is an important matter. This paper is concerned with some aspects of this problem when the model is generated by the dynamical packing algorithm originating with the work of Lubachevsky and Stillinger [1], as discussed in [2,3].

The aforementioned references discuss packs of spheres, and Ref. [3], in particular, describes strategies for accommodating the wide range of particle sizes that are typical of real propellants in the generation of packs suitable for combustion simulations. Real AP particles are not spherical, of

course, but a study of packs of spheroids [4] suggests that nonsphericity is not an issue insofar as burning rates are concerned, provided the packs are statistically isotropic.

Burning rates for various AP or HTPB packs are calculated in [5], and comparisons made there with experimental data suggest that the effects of morphology on these rates can be satisfactorily predicted. Because the very smallest AP particles cannot be resolved numerically, and so must be homogenized into the binder, the thermal conductivity of the blend must be calculated and this is discussed in [6]. We have not had the need to examine the mechanical properties and behavior of such blends—e.g., effective Young's modulus, stress augmentation, von Mises stress, etc.—but the homogenization literature makes it clear that these kinds of things can depend on the fine statistical details of the packs, e.g., [7].

When aluminum is an ingredient of the propellant, agglomeration can be an important issue. Agglomeration occurs when the particles come to the surface, reside there for a while, and during this residence adhere to other particles. Agglomerates of an order of 100 μm diam are formed in this fashion, and are subsequently carried into the chamber with negative consequences, in addition to the positive energy addition. It would be of great value if the designer could fine-tune the propellant morphology to generate an agglomerate size distribution, which minimizes the negative consequences.¹ No completely predictive strategy has yet been developed, but there is good evidence that knowledge of the pack morphology and of the mean agglomerate size permits the prediction of the size distribution (i.e., the standard deviation, should the distribution be lognormal). This prediction is derived from a proximity model—roughly speaking, particles that are sufficiently close to each other within the solid will agglomerate—discussed in Ref. [8]. Use

*limey@uiuc.edu

¹Frictional energy losses, accumulation in submerged nozzles, nozzle impaction, and exhaust signature.

of this model to explore the effects of morphology requires the construction of large packs, packs with hundreds of thousands of particles, and these need to be generated in, at most, a few hours of CPU time. Also, in this as in all combustion applications, the packing fraction (volume fraction of AP) must be large.

These remarks should make it clear that there is a need for the numerical construction of large polydisperse packs that can be generated quickly and have properties that are close to those of real packs. This paper discusses the issue. We know of only one other packing strategy that is used for propellant modeling, one developed at ATK-Thiokol [9], and at various places we examine the results of this work.

II. PACKING ALGORITHM

The algorithm is described in [1] and the application to propellant packs in [2,3]. For the most part, Stillinger and his co-workers have only been interested in monodisperse packs, but the propellant application is primarily concerned with polydisperse packs, with the largest particles being one or two orders of magnitude larger than the smallest particles.

The algorithm begins with an infinite computational domain defined by the periodic continuation of a cube (or cuboid) in three-dimensional space. Points are randomly assigned to this domain at time $t=0$ with random velocities. For $t>0$ these points grow linearly with time, to generate spheres. The growth rates vary amongst the spheres and their distribution defines the final distribution of diameters (of AP and aluminum particles) in the propellant. The largest particles have a radius $r(t)=at$, where t is time, capped at t_{end} . Typically, t_{end} is not chosen but is defined when the desired packing fraction is achieved, or when the computation terminates because the pack is “jammed.” We shall call a the growth rate; smaller particles grow at a slower rate. Overlapping is prevented by including collisions in the algorithm. Jamming arises when the time between collisions becomes too small, but we note here and discuss later that, strictly speaking, within this dynamic framework jamming never occurs as it does in real packs.

Our first version of the algorithm was in no way optimal, and used a parallel platform message passing interface (MPI) to generate packs within an acceptable time frame. But recently we have had reason to improve our algorithm so that it can generate large packs on a laptop in decent time, and these improvements are described here. We claim no global superiority of our strategy, only that it is vastly superior to our old strategy, and enables us to achieve our application goals.

III. IMPROVED ALGORITHM

The improved algorithm has two major advantages.

(1) It enables us to construct large polydisperse packs on a laptop computer within an acceptable time frame.

(2) It accurately depicts rigid sphere packing, and both packing and statistical results compare favorably with experimental hard sphere packs.

These improvements arise from the use of strategies adopted (in some cases, for the first time) from the molecular dynamics (MD) literature.

Modern implementations of rigid sphere packing take advantage of an event-driven molecular dynamics (EDMD) approach. In EDMD, particles are advanced between “events,” where an event is loosely defined as anything that changes a particle’s state. The event could be a binary collision, a collision between a particle and a domain boundary, or a transfer of a particle across an internal or external boundary. Instead of advancing the particles by a fixed time step as in time-driven MD (TDMD), the particles are always advanced to the next event time. Each EDMD time step therefore requires two tasks.

(1) Find and execute the next predicted event.

(2) Update all event predictions influenced by this event.

These two tasks may be optimized separately. Optimization techniques used for the first task usually rely on an efficient priority queue algorithm. Optimization of the second task is more difficult and can involve neighbor lists or cell methods. A typical EDMD method will use a min heap to find the next event, an upper or lower triangular matrix to store collision time estimates for binary pairs, and a neighbor list or cell for each particle to reduce the number of binary collision prediction calculations. There are many excellent texts that describe these processes in detail [10,11]. For a concise description of a modern EDMD implementation using min heaps and neighbor lists, see Donev *et al.* [12].

Our EDMD-based packing method uses a priority queue for event handling and a hierarchical cell scheme to reduce the number of binary collision prediction calculations for polydisperse packs. We use a fully object-oriented implementation in C++ that provides complete encapsulation of the sphere, boundary, and cell objects. With our method it is not necessary to perform the collision validation checks necessary for our older strategy, nor is it necessary to store a triangular collision time matrix. In practice, our method has demonstrated $O(N)$ run times (for monodisperse packs) and $O(N)$ memory requirements over $10^3 \leq N \leq 10^6$. This makes it possible to generate the very large polydisperse packs we need in less than a day using a serial code on a laptop.

Our priority queue implementation uses a binary heap to process events. Binary heaps are outperformed by Fibonacci heaps for the `find_min` function, which has $O(\ln N)$ complexity for a binary heap, but $O(1)$ for a Fibonacci heap. However, processing each collision requires one `find_min`, one `delete_min`, and one `insert` operation, returning the overall complexity to $O(\ln N)$ for both heap methods. A parallel algorithm might further reduce the run-time cost of the heap operations [13]; unfortunately, the method is based on a triangular storage matrix for the collision times and thus has $O(N^2)$ memory requirements. A more sophisticated parallel algorithm using multiple heaps achieves $O(N)$ memory requirements [14] and the additional complexity could be justified by future requirements for larger packs. However, in our current applications with $N \leq 10^6$ the computation time is dominated by the binary collision prediction calculation. Thus, for the foreseeable future a binary heap will be sufficient for our purposes.

We use a hierarchical cell structure to reduce the number of binary collision prediction calculations that are necessary when updating the collision times. With the cell method, the particles’ centroids are contained within cuboid cells whose

maximum enclosed spheres have larger radii than the contained particles. The cell keeps a list of all contained particles and a list of its boundaries, which may be internal (local cell boundaries) or external (domain boundaries). When predicting collisions, an optimized cell method can reduce the prediction computation time per particle to constant time; this is because only collisions between particles in neighboring cells need be computed. There are additional, inexpensive calculations to determine when the particle passes out of the cell into a neighboring cell or when a particle interacts with a boundary. Both cell transfers and boundary interactions are treated as events, and thus participate in the priority queue.

Although hierarchical cell methods have been implemented in two dimensions [15], we are not aware of any other analysis or implementation of the three-dimensional case. In our implementation, the cells are arranged in a three-dimensional tree structure. When particles outgrow their current cell, they are transferred up the tree to their cell's parent. In the transfer event, the particle is removed from its previous level so that it is always associated with a unique level. Since the binary collision calculation is required for a given particle with all other particles in the neighboring cells, the volume searched is proportional to r^D , where r is the cell radius and D is the dimensionality. Thus, it is advantageous to have each particle in the smallest cell that is practical in order to reduce the volume (and thereby reduce the number of binary pairs). Because of this, our implementation does not require the tree to be an octree. In our implementation, the number of cells at each level and in each dimension need only be integer multiples of the level immediately above it. However, in practice we have found octrees to perform well for very polydisperse packs, so the remainder of the results and discussion in this paper pertain to the octree implementation.

Figure 1 is a one-dimensional cartoon of the cell hierarchy showing search paths for sphere 1 interacting with spheres 2 and 3. Shading denotes cells that must be checked as well as the pathways that are traversed to find these relevant cells: (a) sphere 1 is located in a cell at the bottom of the tree and must search upwards through parent cells recursively; (b) sphere 1 is located at an intermediate level and must search upwards and downwards recursively; and (c) sphere 1 is moving in the direction indicated, and the trajectory calculation is used to recursively rule out subtrees that are not relevant. Were the trajectory method not used, all of the cells in (c) would need to be checked. In three dimensions, the number of ruled-out cells typically exceeds the number of checked cells by a factor of 3.

The computational efficiency of the basic cell hierarchy scheme breaks down when the particles have very different radii. This is due to the r^D search volume needed by the larger particles. For example, the smallest particles need only search their neighboring cells and their parents' neighboring cells, recursively all the way up the tree, with computation time $t_{\text{small}} \propto N_{\text{small}} 3^D h$, where N_{small} is the number of small particles and h is the height of the tree. At worst, t_{small} grows linearly with h , resulting in a very efficient search. By comparison, the largest particles must search their subtree along with all of their neighbors' subtrees, with $t_{\text{large}} \propto N_{\text{large}} 3^D (1 - 2^{-Dh}) / (1 - 2^{-D})$. The search time is proportional to a geomet-

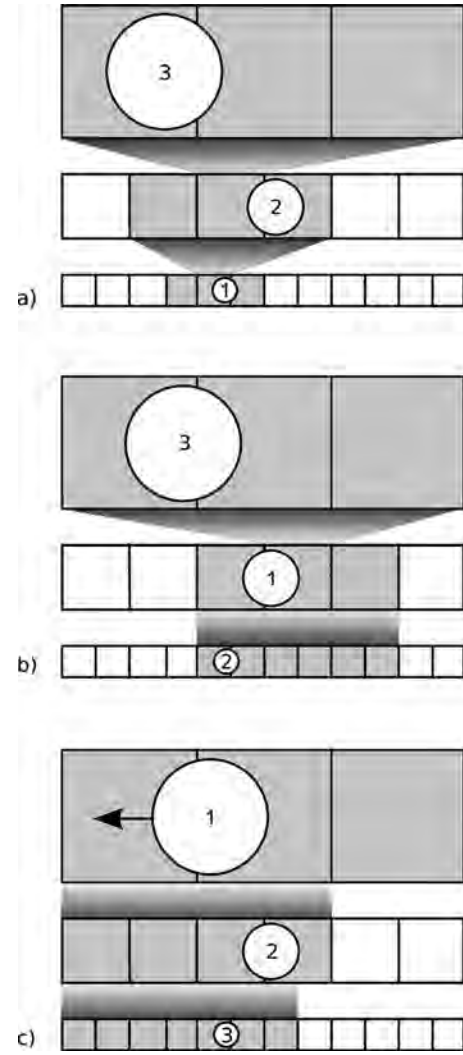


FIG. 1. A one-dimensional cartoon of the cell hierarchy search paths (see the text).

ric series and is very expensive, even for $h=2$. The t_{large} could be partially offset in an optimistic scenario where there is one large particle per cell at the top of the tree. In that case, we might expect $N_{\text{large}}/N_{\text{small}} \propto 2^{-D(h-1)}$, leading to overall computational time

$$t_{\text{overall}} \propto N_{\text{small}} 3^D \left(h + \frac{2^{-D(h-1)} - 1}{1 - 2^{-D}} \right). \quad (1)$$

This situation would arise, for example, if a pack was made by first completely filling the domain (until jammed) with the largest particles and then packing the smallest particles into the voids. The second term in parentheses in Eq. (1) would then be very small, and the run time would be negligibly affected by the polydispersity. In most relevant packings, however, we have fewer large particles than large cells. It is also unlikely that many, if any, of the particles' final radii will exactly match their container cells' radii (perfectly optimized cell sizes) or that any particle size modes will have radii that are a multiple of 2 times any other particle size mode (the best case scenario for an octree). Thus, in more

realistic polydisperse packs the overall run time is dominated by the calculations for large particles with their smaller neighbors. Because of this we construct the octrees by first attempting to fit the final radii of the largest particles as close as possible. After the largest cells are constructed, they are subdivided recursively. Subdivision is ended when the next subdivision would create cells too small to contain the smallest particles.

We use a large particle trajectory cell selection method to avoid the geometric series dependence of the overall computation time. In our method, the subtree cells are treated as (stationary) spheres themselves, and collision times are computed between the cells and the large particle of interest. The cell sphere is constructed as the smallest cell-centered sphere that encloses both the cell and any particle it could possibly contain. If the large particle will not collide with the cell, then any particles in the cell are ruled out. The procedure is recursive through the cell's offspring, so that the complete subtree is instantly also ruled out. To further optimize the calculation, the trajectory of the large particle that is used in the cell collision calculation is bounded by the current best estimate of the next collision time for the large particle. This straightforward trajectory selection process reduced the overall computation time by a factor of 4 for the polydisperse packs in this paper.

The cell scheme has the additional advantage of being well suited for handling various boundary geometries. We use three types of boundary conditions: rigid planar, periodic planar, and cylindrical. Rigid planar boundaries are easily handled by computing the binary collision between a particle and its reflection across the plane. For simulating the very large packs needed for energetic material modeling, we use periodic boundaries. A periodic cuboid has 26 neighboring images which must be included in the calculation, which could increase the computational burden 27-fold. Fortunately, with the cell hierarchy, boundary cells are computed as if they were inner cells, and the computational time is negligibly larger than the equivalent rigid planar calculation.

We also implemented a cylindrical boundary that more accurately reflects the geometry used in many experiments and real-world packs. The cylinder is bounded at each end by either rigid planar or periodic planar boundaries. Collisions with the cylinder are calculated by projecting the sphere onto a plane perpendicular to the cylinder axis and then computing the collision time between the two circles. This boundary is also easy to implement in the cell hierarchy simply by linking the boundary to the cells at the periphery of the cylinder. Figure 2 shows an example of a polydisperse cylindrical pack; it contains 70 008 particles with a ratio of largest diameter to smallest of 29.3 and was constructed on a laptop in 20 h. Note, however, that computational times are strongly dependent on the stopping criterion, the definition of jamming. Thus Kansai *et al.* [16] report the construction of a bidisperse pack of 10 000 particles, size ratio 10, that took 48 h on a 1 GHz Pentium machine, but do not specify the stopping criterion, and so a comparison with our calculation is not possible, although we can note that for their code the scaling with particle number is roughly N^2 . However, our old code, similar in most important respects to that used in [16], is not suitable in a serial version for the generation of the

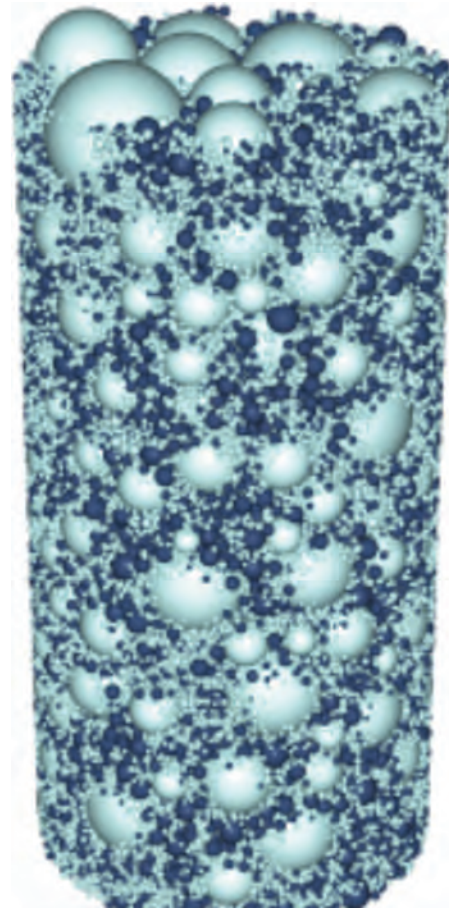


FIG. 2. (Color) A polydisperse cylindrical pack, periodic along the axis, size distributions from Table I.

packs needed for our combustion simulations, and the improvement we have been able to achieve is substantial. Much of this improvement is due to the use of a cell hierarchy, rather than neighbor lists.

Note that although in our application we are interested in large polydisperse packs such as the one shown in Fig. 2, experimental data with which we can compare the properties of our numerically generated packs is only available for relatively simple packs, and these comparisons occupy the significant part of our discussion. All of the packs generated in the paper have periodic planar boundary conditions, unless explicitly specified differently.

IV. COLLISIONS

As we noted earlier, collisions must be accommodated in order to avoid particle overlap. Moreover, these must be pointwise-in-time events to fit within the general framework of the algorithm. Because each particle is growing and there is an outward surface velocity relative to the centroid, a classical elastic collision does not guarantee a pointwise event, and in some cases, depending on the collision speeds and the growth rates, it is necessary to add extra impulses, generating increments to the rebound velocities. The original Lubachevsky-Stillinger algorithm uses the relative velocity

TABLE I. Left columns: coarse and fine AP distributions used in Fig. 2. Right columns: aluminum distribution.

Number	Diameter	Number	Diameter
2	348.320	2	99.685
3	324.110	6	92.135
4	301.580	12	85.155
6	280.615	23	78.705
8	261.110	39	72.745
11	242.960	61	67.235
13	226.070	92	62.140
16	210.355	134	57.340
18	195.730	186	53.080
21	182.125	252	49.060
22	169.465	329	45.345
23	157.685	418	41.910
24	146.725	516	38.735
23	136.525	619	35.800
22	127.035	723	33.090
20	118.205	823	30.585
18	109.990	912	28.265
15	102.345	985	26.125
13	95.230	1037	24.150
10	88.610	1064	22.320
8	82.450	1065	20.630
6	76.720	1039	19.065
4	71.390	989	17.620
3	66.425	918	16.285
2	61.805	831	15.050
1	57.510	733	13.910
1	52.156	632	12.855
2	24.000	530	11.885
44	30.160		
184	26.975		
540	24.125		
1312	21.575		
2792	19.295		
5316	17.260		
9180	15.435		
14464	13.805		
20879	12.350		

of the particle surfaces when computing impulses, and energy is added at each collision; our old version uses the centroid velocities with rebound additions for every collision that depend only on the growth rates, and not on the collision speeds. Of particular relevance is the ratio of the growth rate a_i ($=a$ for the most rapidly growing particles) to the speed $\|\vec{v}_i\|$. After many collisions in which the kinetic energy is enhanced this ratio can become very small so that little growth occurs between collisions, and jamming cannot be achieved within a reasonable time frame. Also, it can become impossible to generate low-density jammed packs. Because of this, it is necessary to manage the kinetic energy of the

pack and, in some sense, minimize its growth. The qualification is necessary because it is only the increase in energy that arises from changes in speed that is of concern, not from the changes in mass.

The literature abounds with methods to counter the increase in kinetic energy. The simplest method is to periodically renormalize the velocities of all particles either by rescaling [11] or by setting the velocities to zero (this method is only relevant when the pack is nearly jammed, see, for example, [1]). In either case, the renormalization requires restarting the priority queue, which is an expensive operation. Rescaling the velocities has also been shown to violate energy equipartition [17], which could be problematic. Another interesting technique borrowed from molecular dynamics is the use of stochastic thermostats that place the pack into contact with an imaginary heat bath. Some of these, such as the Andersen thermostat [10], can be implemented as random collisions with ghost particles, and can thus be integrated into the EDMD scheme as a single-particle event. However, the Andersen thermostat in particular has been shown to pollute the transport coefficients [10] and thus can reasonably be expected to reduce the efficiency of sampling the configuration space during packing. The Lowe-Andersen thermostat removes this shortcoming by considering particle pairs, and it appears to be a promising alternative for efficient packing [18]. Unfortunately, all stochastic thermostats become inefficient as the collision frequency increases because they require additional event predictions at each thermostat collision and also because they require several independent random numbers. When packing spheres, high quality random number generators such as the Mersenne Twister [19] produce a double precision number in the time it takes two binary collisions to be calculated. Thus for high growth rates and the corresponding need for many thermostat collisions, the additional events and random number generation can dominate the computation time.

Our approach is to simply minimize the amount of energy added during each binary collision. When none is needed, none is used; when it is needed an amount is added sufficient merely to cause the surfaces to move away from each other at a minimal speed, say 10^{-30} . The amount of energy added in this way is monitored by defining and calculating a pseudotemperature, a substitute for the set of growth-rate or speed ratios. This pseudotemperature also plays a role in the specification of the initial velocity distribution at the start of the calculation, and links the algorithm more closely to a molecular dynamics framework. In our earlier calculations the velocity components were randomly sampled from the interval $[-1, 1]$; here we sample from a Maxwellian (normal) distribution.

We are only interested in situations when N , the number of particles in the pack, is large, and it makes the discussion of our pseudothermodynamics more agreeable if we suppose that N is an asymptotically large parameter. Then the classical definition of temperature is related to the mean kinetic energy of the particles by

$$\frac{3}{2}kT = \frac{1}{N} \sum_i \frac{1}{2} m_i \|\vec{v}_i\|^2, \quad (2)$$

where k is Boltzmann's constant. Each mass is given by

$$m_i = \frac{4}{3}\rho\pi r_i^3 = \frac{4}{3}\rho\pi(a_i t)^3, \quad (3)$$

and it is this t^3 dependence that is not relevant to us. And so we define a pseudotemperature by

$$T = \frac{1}{3N} \sum_i \frac{a_i^3}{a} \|\tilde{v}_i\|^2, \quad (4)$$

nondimensional when the velocities are appropriately defined. Initial values are defined by

$$\tilde{v}_{i,0} = \sqrt{\frac{T_0}{a_i^3}}(\alpha_1, \alpha_2, \alpha_3), \quad (5)$$

where T_0 is the initial temperature, and the α_i are sampled from independent normal distributions.

Thus the system starts out in thermal equilibrium (small particles travel faster, on average, than large particles) and would remain so were it not for the inelastic collisions. Note that the Maxwellian is consistent with Eq. (2). Note also that this approach could also be applied to systems of particles with rotational degrees of freedom to ensure the packing initially satisfies equipartition of energy.

We use a simple method for determining when a pack has jammed. During the run, the packing fraction is computed after each “pass,” where a pass is defined as a set of N sequential interparticle or solid boundary collisions (particle transfers and other events are not counted). It is of course unlikely that all N particles will participate in collisions in any given pass; nevertheless, the concept of a pass allows us to discuss jamming independently of the number of particles N . We simply stop the pack when the change in the packing fraction over a pass is less than some specified limit value, e.g., the jamming criterion after the i th pass is $(\rho_i - \rho_{i-1})/\rho_i < \varepsilon$, where ε is the limit value.

For monodisperse packs, ε is closely related to the “distance to jamming” $(1 - \rho/\rho_J)$, where ρ_J is the jamming density (see [20] for a detailed discussion of pack properties as this distance approaches zero). To see this, we start with the fact that ρ is proportional to t^3 so that in the last pass,

$$\Delta\rho/\rho = 3\Delta t/t \sim \varepsilon, \quad (6)$$

where Δt is the length of the pass, and we have linearized, since ρ is close to the limit value ρ_J . At the beginning of the pass, when the particle diameter is d , the gap between each particle is $\sim(d_J - d)$, so that since ρ is proportional to d^3 ,

$$(\text{gap}) \sim (d/3)(1 - \rho/\rho_J) = (at/3)(1 - \rho/\rho_J). \quad (7)$$

Thus the time interval before a particle collides with its neighbor is

$$\sim (at/3c)(1 - \rho/\rho_J), \quad (8)$$

where c is the representative surface speed, the sum of the growth rate a , and a positive translational speed. But in this interval $\sim N$ collisions occur, so that it is the length of the pass, and comparing Eqs. (6) and (8) we have

$$\varepsilon \sim (a/c)(1 - \rho/\rho_J). \quad (9)$$

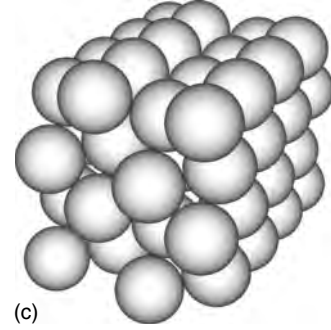
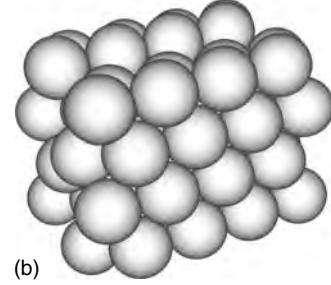
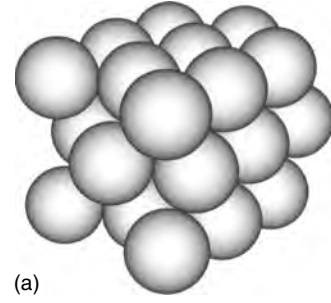


FIG. 3. Lattice packs. (a) Double Nested, (b) Close Pack (Side 1), (c) Close Pack (Side 2)

Throughout the paper we generate most of the packs using the improved algorithm, a serial code with temperature-based initial conditions, what we shall call the T algorithm; growth rates are normalized with a so that the maximum growth rate is 1 and the velocity–growth-rate ratio is controlled by T_0 . But there are a few results obtained using the old parallel code for which velocity components are sampled on the interval $[-1, 1]$ and the ratio is controlled by a ; we shall call this the a algorithm.

V. MONODISPERSE PACKS

In this section we examine monodisperse packs. There is a maximum packing fraction for a monodisperse pack of spheres, well defined for lattice packs, less so for random packs. Lattice packs are characterized by regular repetitive structures. Thus (see [21]) we have cubic lattice—53.36%, orthorhombic—60.46%, double-nested—69.81%, and close-packed—74.05%, the largest attainable for monomodal spheres. The last two arrangements are shown in Fig. 3. As we shall see, the close-packed lattice is relevant to order that we obtain for high-density packs, and near rigid boundaries.

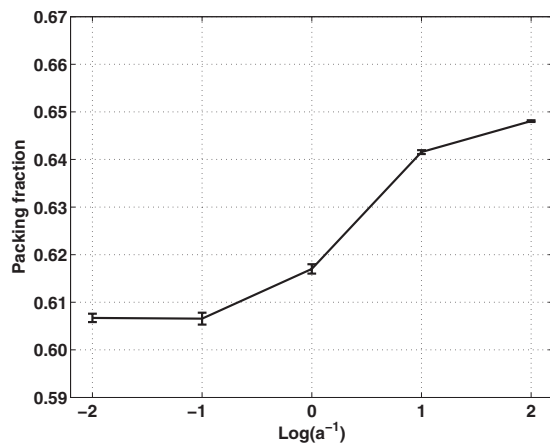


FIG. 4. Final packing fraction as a function of inverse growth rate (a algorithm).

The very concept of random packs is not well defined, because all packs, whether generated experimentally or numerically, have some degree of order. Equally, maximum packing fractions are not well defined. But attempts to create random packs typically lead to packing fractions in the range of 59%–64% where values at the low end are characteristic of what are called loose packs (LRP), and values at the high end are characteristic of what are called dense packs (DRP) [21,22]. (It is a minor syntactical misfortune that dense packs are not called tight packs.)

The a algorithm is capable of modeling both jammed LRP and DRP by varying the growth rate; except for extremely small growth rates, the smaller the growth rate the higher the packing fraction. Figure 4 shows this for three monomodal packs of 3000 spheres, the packs differing because of different initial conditions. The bar for each choice of a shows the maximum, minimum, and average fractions. End values are 60.74% for $a=100$ (an LRP) and 64.78% for $a=0.01$ (a DRP). The ballistic-deposition algorithm used by Webb and Davis [9] yields a value of 60%. Experimental data obtained using mechanical shaking can be found in [21] (62.5%) and [22] (64%).

Figure 5 shows the packing fractions that can be achieved with the T algorithm, to be compared to Fig. 4. We are not

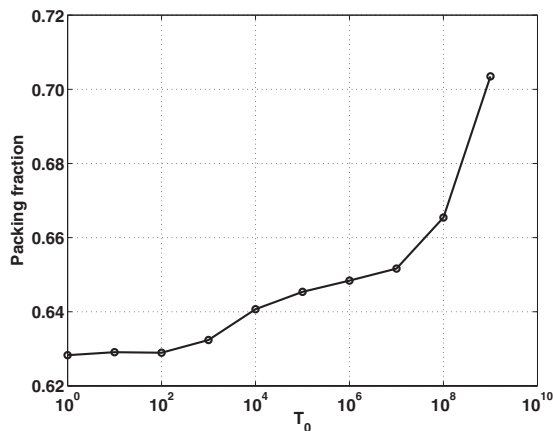


FIG. 5. Packing fraction of a 10 000 particle pack as a function of the initial temperature T_0 (T algorithm).

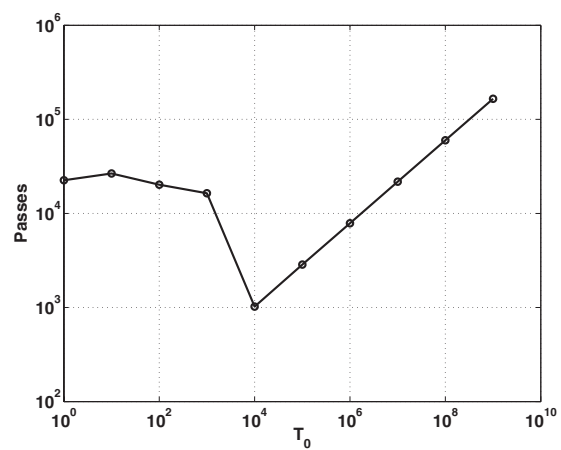


FIG. 6. Number of passes used to generate Fig. 5, $\varepsilon=10^{-7}$.

aware of any other reports of packing fractions greater than 0.7 obtained using the Lubachevsky-Stillinger algorithm. The corresponding number of passes before jamming is achieved are plotted in Fig. 6; ε is equal to 10^{-7} . High temperatures generate packs of higher density than that of a double-nested lattice, implying significant order. Later, we shall see that these ordered packs have many of the statistical characteristics of a close-packed lattice. That packs generated using low temperatures have less order than those generated using high temperatures is clear from Figs. 7 and 8, the former (packing fraction equal to 0.6283) calculated for $T_0=1$, the latter (packing fraction equal to 0.7035) for $T_0=10^9$. Figure 8 looks very much like a lattice pack with dislocations, whereas Fig. 7 is far less regular.

VI. BIDISPERSE PACKS

Typical packing fractions for propellants are in the neighborhood of 78%, and so for this purpose monodisperse packs have little relevance. Here we discuss bidisperse packs, the simplest of the polydisperse variety, and packs for which there is experimental data. The fine component can fill some of the spaces between spheres of the coarse component, and therefore generate higher packing fractions. Not surprisingly, the greater the ratio between the size of the coarse component and size of the fine component, the greater is the maximum packing fraction. The maximum is attained at a

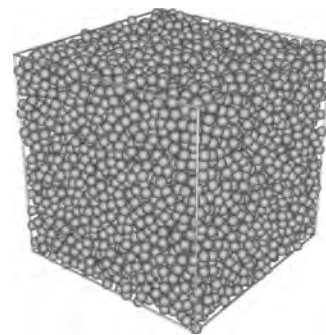


FIG. 7. A 10 000 particle pack for $T_0=1$, packing fraction 0.6283.

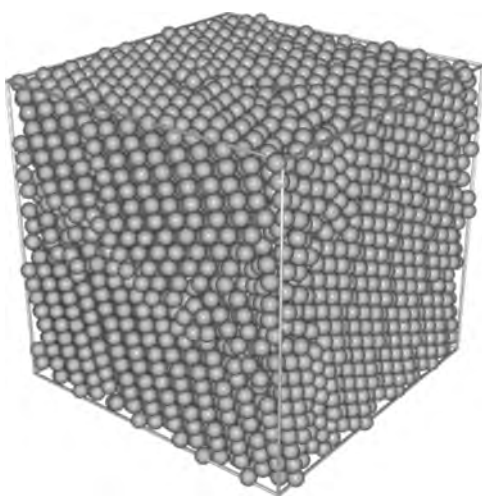


FIG. 8. A 10 000 particle pack for $T_0=10^9$, packing fraction 0.7035.

uniquely defined value of the ratio of the volume fraction of coarse to that of fine.

The earliest reports of polydisperse packs obtained using the Lubachevsky-Stillingier algorithm may be found in [2,3,16], the latter a discussion of bidisperse packs only. Here we compare numerical results for bidisperse packs with experimental data of McGeary [21].

McGeary created a number of bidisperse packs. The size of the particles was defined by using meshes, a standard procedure in the particle sorting business. He used seven-mesh spheres as a coarse component together with several finer components, achieving a maximum packing fraction of approximately 84% for the finest. We compare McGeary's data with numerical results for two growth rates, $a=0.2$, and $a=1$ (a algorithm). We also plot results from [9] generated using a ballistic-deposition strategy.

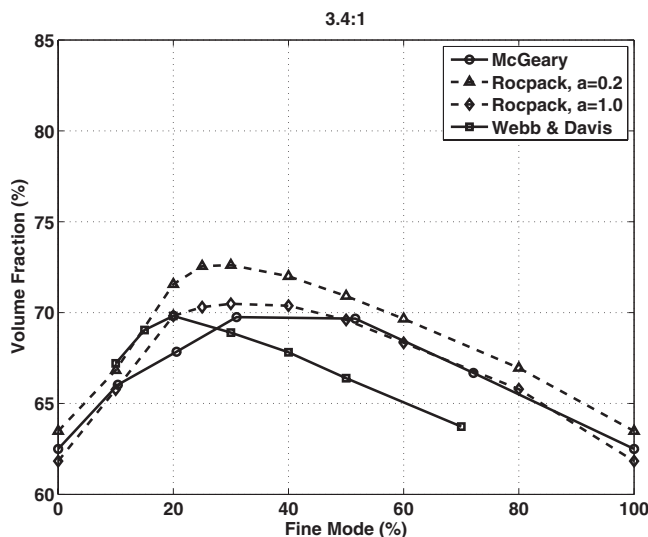


FIG. 9. Packing fraction vs fine-mode percentage for a 3.4:1 coarse-to-fine size ratio.

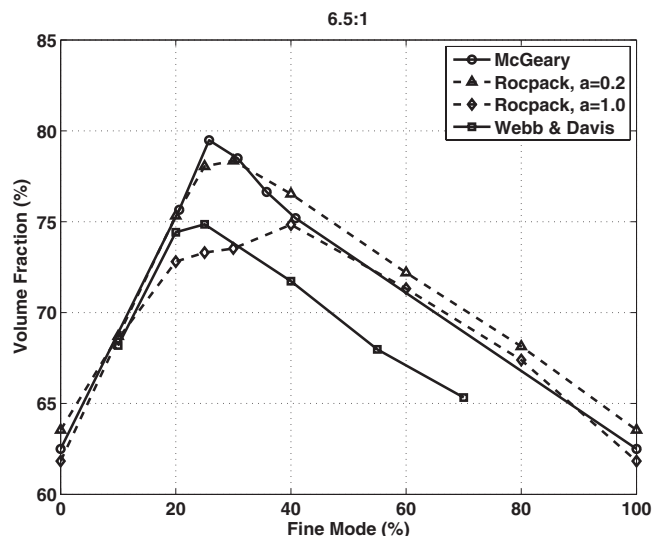


FIG. 10. Packing fraction vs fine-mode percentage for a 6.5:1 coarse-to-fine size ratio.

VII. COARSE-TO-FINE SIZE RATIO 3.4:1

This models the experiments with seven-mesh and 20-mesh, and 20 000 spheres were used in the simulations; comparisons are shown in Fig. 9. Except at a fine-mode fraction of 20, the results for $a=1$ are in excellent agreement with McGeary's data. It is difficult to pin down with precision the location of the maxima, but for $a=1$ the two largest calculated packing fractions correspond to the coordinates (30, 0.7049) and (40, 0.7038); for $a=0.2$ they are (25, 0.7255) and (30, 0.7262). The Webb and Davis results peak early, and then underpredict.

VIII. COARSE-TO-FINE SIZE RATIO 6.5:1

This models the 7/40 mesh study of McGeary, using 50 000 spheres, and the results are shown in Fig. 10. The slower growth rate ($a=0.2$) gives the closest agreement with experiment, with the largest calculated packing fraction of 78.35% at a fine-mode percentage of 30. When $a=1$ the results deviate significantly from the experimental values in the fine-mode percentage interval (20,40)%; the largest calculated value is 74.84% at a fine-mode percentage of 40%.

IX. COARSE-TO-FINE SIZE RATIO 16.5:1

The largest size ratio is achieved with 7/80 meshes and the simulations use 80 000 spheres; only results for $a=0.2$ are shown (Fig. 11). Agreement with experiment is very close, with a largest calculated value of 82.01% at a fine percentage of 25. In all of the cases we discuss here, the ballistic-deposition algorithm typically yields packing fractions significantly lower than the experimental values. In such an algorithm there is, of course, no tuning parameter comparable to a . Whether this matters depends on the application. In the modeling of a propellant with a significant fraction of very fine AP, so fine as to be unresolvable numerically, a relatively low packing fraction for the resolvable

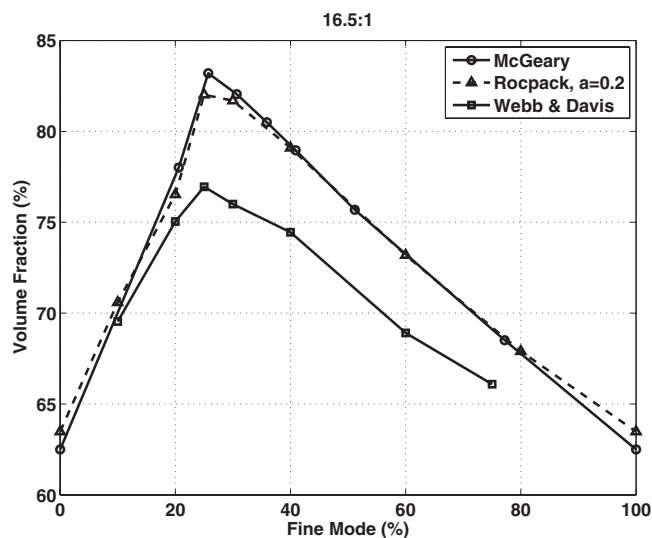


FIG. 11. Packing fraction as a function of fine-mode percentage for a 16.5:1 coarse-to-fine size ratio.

particles might be all that is needed, with the remaining fraction homogenized into the binder. But in other situations, the inability to reach experimentally attainable packing fractions could mean that the strategy is not useful.

It is important to note that the construction of polydisperse packs is much more computationally intensive than the construction of monodisperse packs, and the use of hierarchical cells rather than neighbor lists plays a crucial role in increasing the efficiency of the algorithm so that a serial platform can be used. All of the old work cited in this section [2,3,16], uses neighbor lists, and as a consequence it was necessary for us to partially parallelize the code (not the heaps) in order to generate packs for our combustion studies [5,6].

X. STATISTICAL ANALYSIS OF PARTICLE PACKINGS

The previous section focused on the packing algorithm and the products of this algorithm were characterized merely by the packing fraction and by two related concepts, loose and dense packings (LRP and DRP). The algorithm generates different jammed-state packing fractions according to the choice of the growth parameter a , or the initial temperature T_0 , and variations in the fraction correspond to variations in the inner structure. Thus experimentalists report that a DRP can only be achieved with strong shaking or shocking, whereas this is not necessary for an LRP. The nature of the inner structure is of intrinsic interest, but also, from the propellant modeling perspective, it is relevant to the simulations. Certain thermal and mechanical properties of the pack are likely to depend on it, and bounds and estimates for these quantities in the homogenization literature often depend on its statistics. Thus it is relevant to discuss the statistics and, where possible, compare them with experimental data.

Here and earlier we wrote of “jamming.” From an algorithmic point of view this can refer simply to the state in which the time between collisions is so small that the calculation is effectively stalled, but more precise discussions are

possible. Indeed, Donev *et al.* define the concepts of local jamming, collective jamming, and strict jamming [23]. Local jamming occurs when each particle in a subset of the pack is locally trapped by its neighbors and is unable to translate. Such particles are necessarily touched by at least $d+1$ peers not in the same hemisphere where d is the spatial dimension. In three dimensions this condition is fulfilled for spheres with more than three contacts. Particles with fewer contacts are called rattlers, since they can move.

Donev *et al.* note that computer-generated packs often do not satisfy the local jamming criterion. That is certainly true of packs generated using the Lubachevsky-Stillinger code and its variations, as a small gap exists between all particles when the computation ceases. However, given enough computer time, most gaps can be made arbitrarily small, and the number of contact points between particles can be well defined [20]. For those concerned with the important mathematical questions associated with packing, and for whom packs of modest size are sufficient, this is important. We, however, are concerned with generating large packs in reasonable times, and are not concerned with sophisticated mathematical questions, but with whether the pack statistics match those of experimental packs. In the determination of experimental morphology (by X-ray tomography, for example) there is error, both from measurement uncertainties and the reconstruction algorithm. Then contact is only defined within the constraint of a tolerance. Thus, to make comparisons with the numerical results we also use a tolerance, and particles are said to be in contact if their closest surface separation is smaller than some assigned constant. This is not a new idea; see, for example, Refs. [22,24,25].

Packing structure as well as particle organization can be characterized using spatial statistics, something that has been done for a long time. Most recently, Aste *et al.* generated a number of experimental packs of up to 150 000 monomodal spheres inside a cylindrical container and then analyzed them using XCT (x-ray computed tomography) to identify the placement of centers; statistical analyses were then possible [22]. A number of theoretical studies on ideal or modeled packs are reported in [26,27,24,20,9].

Here we discuss pair correlation and coordination number for monodisperse packs of varying packing fraction; one would expect that the degree of order is related to the packing fraction. We also discuss high-density lattices, experimental results, and correlate certain features of the statistics with certain kinds of order. And we briefly discuss pair correlation for bidisperse packs.

XI. COORDINATION NUMBER

It is known that a single sphere can touch at most 12 equal spheres [28], a condition satisfied by a close-packed lattice. If all spheres in a pack are jammed, each must be in contact with at least four others. The mean number of contacts is called the coordination number (or contact number or kissing number).

Experimental results of Bernal and Mason for monodisperse packs [29] reveal an average number of contacts of 6.4 for a dense packing (packing fraction 62%) and 5.5 for a

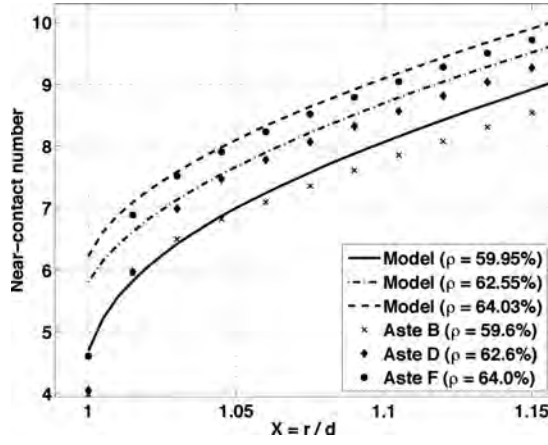


FIG. 12. Average near-contact number as a function of $X \equiv 1 + \epsilon$ (35 000 particles) for three packing fractions, compared with data of Aste *et al.*

loose packing (packing fraction 60%). Also, a near-neighbor (or near-contact) number was determined for these packs, where a near-neighbor to a reference sphere is defined as a sphere whose center is separated by no more than 1.05 diameters from the center of the reference. These numbers are 8.5 and 7.1. In general, near neighbors can be defined using $1 + \epsilon$ instead of 1.05, and it is common, if ϵ (the tolerance) is small (0.01 for example), to identify these near-neighbor numbers with the contact numbers. Clearly the contact number is a function of ϵ , and if ϵ is smaller than ϵ the Lubachevsky-Stillinger algorithm will generate packs with zero contact number.

Figure 12 shows variations in the contact number with $1 + \epsilon \equiv X$ for three different packs of 35 000 particles, the packs differing one from the other because of different choices of a . Similar curves are reported in [30,24], but here we include comparisons with experimental data obtained by Aste and his colleagues. They have studied monodisperse packs using x-ray tomography [31,22] and have provided us with their raw data so that we might make comparisons. Single pack results obtained by us and by Donev *et al.* [20] are shown in Fig. 13.

Table II lists the contact number for some experimental data and for packs that we have generated (the inhouse name

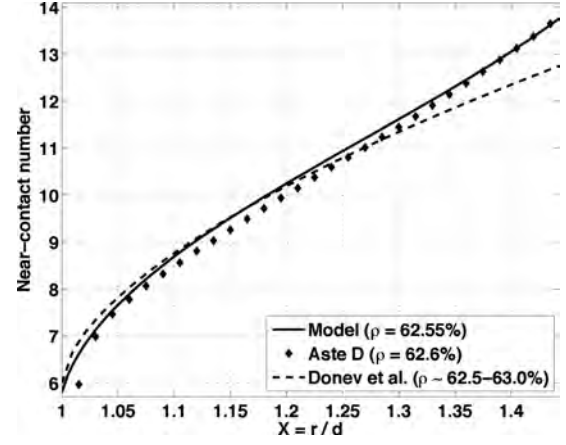


FIG. 13. Average near-contact number as a function of X (35 000 particles) for a single packing fraction, compared with data of Aste *et al.* and results of Donev *et al.* [20].

for our code is Rocpack) for various values of ϵ . Of particular interest is the data of Gotoh as he used a large number of particles, as needed for results to be statistically significant.

XII. RADIAL DISTRIBUTION FUNCTION (RDF)

The concepts of touching number and rattlers (which we do not discuss) are nearest-neighbor concepts, but the statistics of particles at greater distances are of importance, and this brings us to the radial distribution function, also known as the pair distribution or pair correlation. It is defined as the probability of finding a particle center at a distance between r and $r + \Delta r$ from the center of a reference sphere [25]. The discrete definition is

$$g(r, \Delta r) = \frac{V n(r, \Delta r)}{N 4 \pi r^2 \Delta r}, \quad (10)$$

where N is the number of particles in the pack, V is the pack volume, and $n(r, \Delta r)$ is the number of particles in the shell of inner radius r and thickness Δr . This is averaged over all particles; it asymptotes to 1 as $r \rightarrow \infty$.

It is obvious that some separation distances are more likely than others. For example, in a monodisperse pack

TABLE II. Average contacts n_c of Rocpack and experimental packings (data from [29,30,32]). Whether the smallest value of X is an appropriate value for the first column data of Bernal and of Mason is not clear from their reports.

Source	Fraction	N	$n_c(1.005)$	$n_c(1.02)$	$n_c(1.05)$	$n_c(1.1)$
Bernal-LRP	60%	420	5.5	N.A.	7.1	N.A.
Bernal-DRP	62%	476	6.4	N.A.	8.5	N.A.
Mason	N.A.	536	4.7	6.8	8.0	8.9
Gotoh	63.6%	7934	N.A.	7.05	8.0	9.0
Rocpack-LRP	59.95%	35000	5.22	6.04	7.00	8.06
Rocpack-Mid	62.55%	35000	6.15	6.81	7.67	8.69
Rocpack-DRP	64.03%	35000	6.62	7.27	8.10	9.09

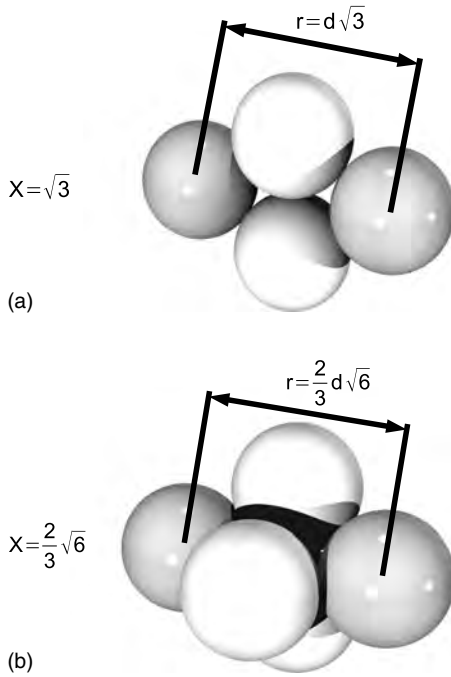


FIG. 14. Possible sphere configurations and the location of corresponding peaks in the RDF. (a) Four close spheres, (b) Tetrahedron configuration.

contact particles will be separated by the diameter d so that we would expect the RDF to have a sharp peak at $X \equiv r/d = 1$. The location of other peaks at $X = \sqrt{3}$ and $\frac{2}{3}\sqrt{6}$ can be understood by referring to the arrangements drawn in Figs. 14(a) and 14(b). The peak at 2 corresponds to three spheres in a row.

XIII. RDF OF MONODISPERSE PACKS

In calculating the RDF for a prescribed pack, it is necessary to make a choice of Δr . If too large, the function is smoothed and peaks are obscured; if too small the noise generated because only a finite number of particles are used can also obscure the peaks. And so it is necessary to make some trial runs to optimize the choice. We do not show the results

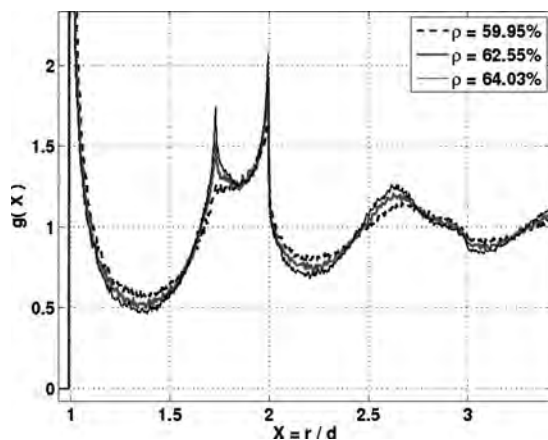


FIG. 15. RDF for different, relatively low, packing fractions.

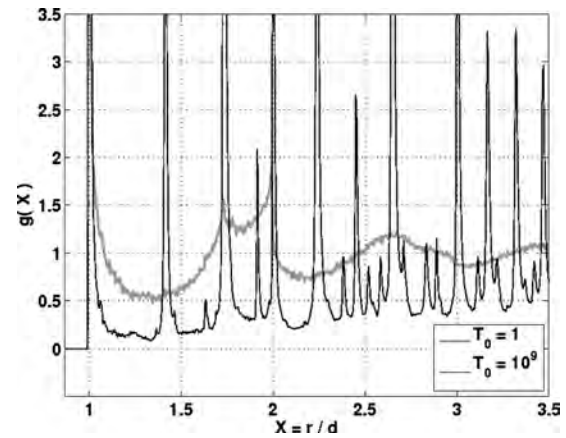


FIG. 16. RDFs for two T -algorithm packs, $T_0=1$ and $T_0=10^9$.

of these trials, but we came to the conclusion that $\Delta r = 0.005d$ lies within an optimal band. We make the same choice for bidisperse packs when d is taken to be the diameter of the smaller spheres.

The growth rate a is an important parameter in the a algorithm; large values prevent the spheres from distributing themselves in a tight fashion; small values permit such a distribution. Similar consequences arise using the T algorithm when T_0 and the tolerance are varied. Figure 15 shows the RDFs for three packs of mass fractions typical of those that can be generated using the a algorithm. It is noteworthy that the peak at $\sqrt{3}$ apparent at the highest packing fraction is completely lost at the lowest. It is also noteworthy that for no pack do we see a peak at $(2/3)\sqrt{6} = 1.633\dots$ [see Fig. 14(b)]. The tetrahedron arrangement is characteristic of the close-packed lattice and does not appear to arise in low-density random packs.

A similar exercise using the full power of the T algorithm yields far more striking results. Figure 16 shows the RDFs for the two packs of Figs. 7 ($T_0=1$) and 8 ($T_0=10^9$). The many peaks for the high-temperature result (including one at $1.633\dots$) reveal the high degree of order in this pack. Later we shall compare it with the RDF of a close-packed lattice.

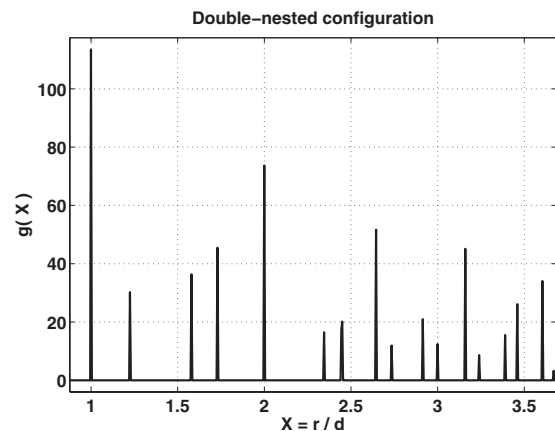


FIG. 17. RDF of double-nested lattice.

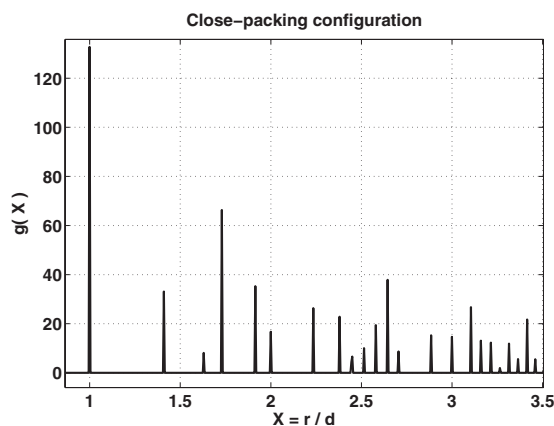


FIG. 18. RDF of close-packing lattice.

XIV. RDFs FOR LATTICES

We have noted that the peaks in figures such as 15 arise because of specific nonrandom structures. Lattices consist entirely of nonrandom structures, and it is of interest to examine the peaks that then arise; in principal, all can be explained by geometry. Since this has been discussed before, we shall just provide a brief description of double-nested and close-packed lattices. The former has a packing fraction typical of packs generated numerically; the latter has a significantly higher packing fraction, but has order that is related to that of high-density numerical packs.

A. Double-nested lattice

The double-nested lattice is shown in Fig. 3(a), and the RDF in Fig. 17. The first six peaks, easily identified from the lattice geometry, are at 1, $\sqrt{\frac{3}{2}}=1.224...$, $\sqrt{\frac{5}{2}}=1.581...$, $\sqrt{3}=1.732...$, 2, and $\sqrt{\frac{15}{2}}=2.738...$.

B. Close-packing lattice

The close-packed lattice is shown in Figs. 3(b) and 3(c), and the following peaks can easily be identified: 1, $\sqrt{2}=1.414...$, $\frac{2}{3}\sqrt{6}=1.632...$, $\sqrt{3}=1.732...$, 2, and $\sqrt{5}=2.236...$.

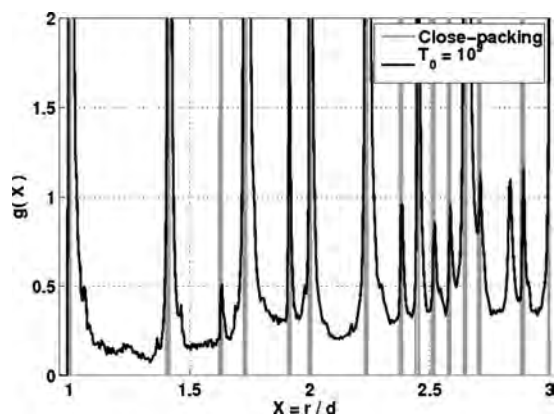


FIG. 19. RDF of the pack of Fig. 8, and close-packing values.

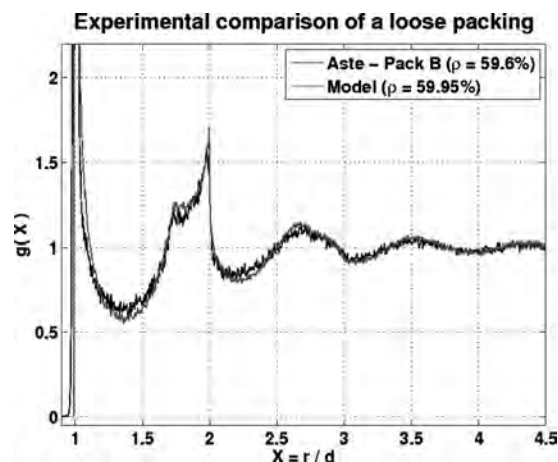


FIG. 20. RDF comparison with experimental data, (a).

Note that the peak at 1.632... arises from the tetrahedron arrangement of Fig. 14(b).

The RDF is shown in Fig. 18. And in Fig. 19 we show the values along with the RDF of the *T*-algorithm pack of Fig. 8. In view of the packing fraction of the pack (70.35%), it might be thought that we should compare its RDF with that of the double-nested lattice (packing fraction 69.81%), but there is little correlation between the two. On the other hand, there is clearly a strong correlation for the close-packing lattice. Thus, although we are still significantly shy of the close-packing density (74.05%), the pack must have a morphology that has many of the characteristics of the lattice.

XV. COMPARISONS WITH THE MONODISPERSE DATA OF ASTE *et al.*

RDFs for packs generated using the Lubachevsky-Stillinger algorithm have been reported before, e.g., [20], but here we compare the calculated results with experimental data. For it is, we believe, important that the packs used in a virtual engineering framework have the same statistics as the real packs. It is possible, of course, that real packs will have different statistics according to the manner in which they are

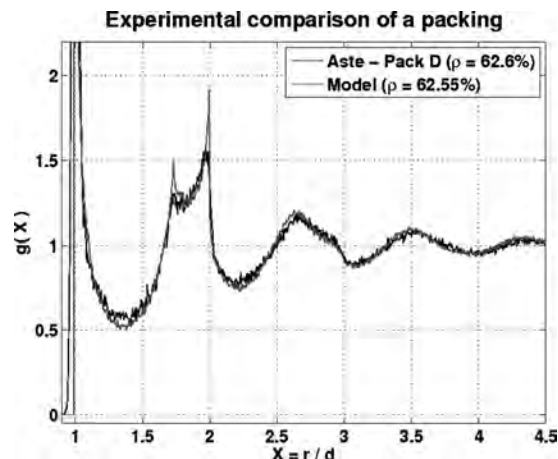


FIG. 21. RDF comparison with experimental data, (b).

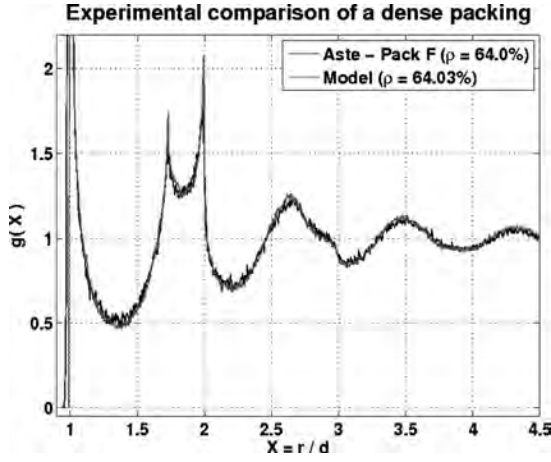
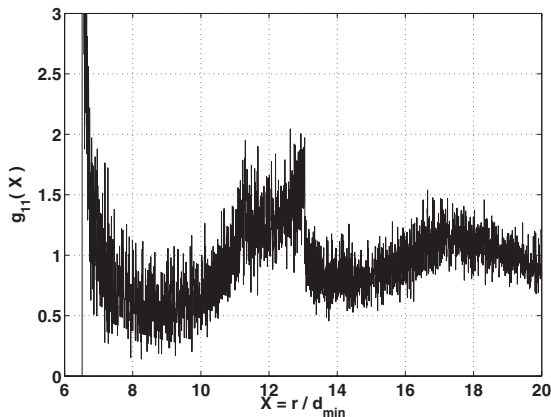
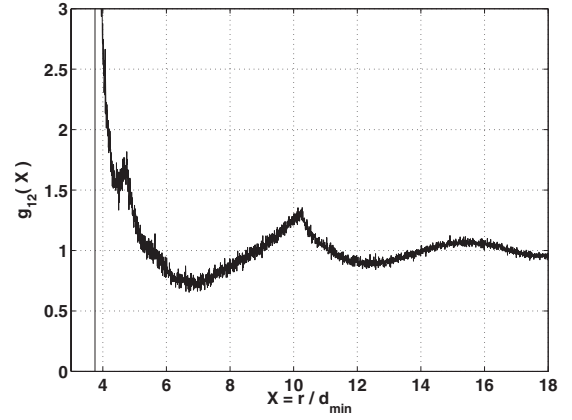


FIG. 22. RDF comparison with experimental data, (c).

generated, a matter that we are not in a position to address at the present time, but Aste and his colleagues at the Australian National University in Canberra have measured the RDFs for a number of monodisperse packs using x-ray tomography [31,22], and have provided us with their raw data so that we might make comparisons, and we do that here. Our packs use 35 000 spheres in a periodic cube and are generated using the *T* algorithm with packing fractions of 59.95%, 62.55%, and 64.03%; the experimental packing fractions are 59.6%, 62.6%, and 64.0%. The comparisons are shown in Figs. 20–22.

We note the coordinates of the first two peaks lying to the right of $X=1.5$, the model coordinates first, the experimental coordinates second: {Pack B: peak 1 (1.735, 1.26), (1.730, 1.22); peak 2 (1.995, 1.71), (1.980, 1.59)}; {Pack D: peak 1 (1.730, 1.50), (1.720, 1.30); peak 2 (1.995, 1.91), (1.985, 1.55)}; {Pack F: peak 1 (1.730, 1.74), (1.730, 1.50); peak 2 (1.995, 2.07), (1.985, 1.77)}. Note that the model peak values are always greater than the experimental ones, significantly so in some cases. But these differences are consistent with errors in the experimental measurements. Aste *et al.* assume a Gaussian uncertainty in the center distance for two touching spheres, and estimate that the average standard deviation is 0.015 diameters. If we randomly adjust the model center coordinates accordingly, and recalculate the RDFs, the

FIG. 23. g_{11} -6.5:1 Bidisperse packing, case A.FIG. 24. g_{12} -6.5:1 Bidisperse packing, case A.

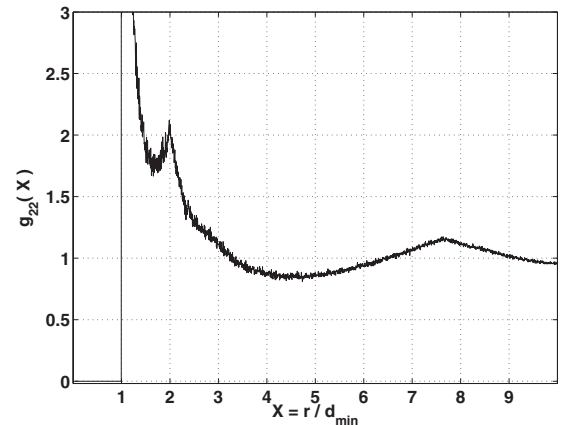
model peaks are decreased and more closely match the experimental values. Similar smoothing would occur if we allowed for a distribution in the sphere diameters.

XVI. RDF OF BIDISPERSE PACKS

For polydisperse structures the counterpart to the RDF function is the partial pair correlation function ($g_{ij}(X)$), the number density of particles of type j that are within shells r to $r+\Delta r$ centered on particles of type i . The function is normalized so that it has the value 1 as $r \rightarrow \infty$, and it defines a symmetric matrix. In its determination there is a problem if the size ratio of the particles is large. For example, for a 16.5:1 ratio in a pack of 50 000 spheres there are only 45 large spheres vs 49 500 small spheres. In this case the statistics involving only the coarse component are nonrepresentative.

A. Pack with size ratio 6.5:1, case A

We consider a pack with 50 000 particles, a 9 to 1 coarse-to-fine mass ratio (i.e., 90% of the mass comprises coarse particles), and a packing fraction of 68.7%. It is close to one of McGeary's data points [21]. Figure 23 is a plot of g_{11} where the index 1 refers to the larger particles. Because, as we have already noted, there are so few of these, the function

FIG. 25. g_{22} -6.5:1 Bidisperse packing, case A.

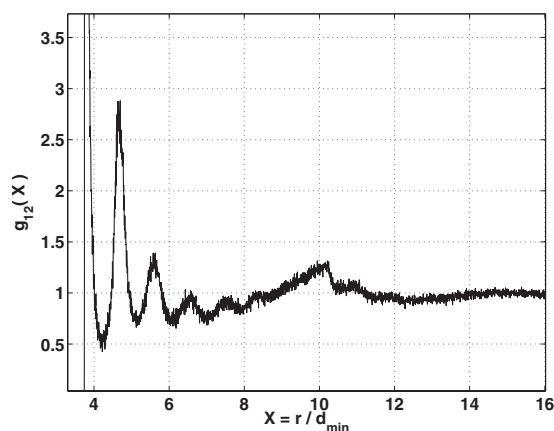


FIG. 26. g_{12} -6.5:1 Bidisperse packing, case B.

is noisy, but trends are apparent. They are similar to those of a monodisperse pack, not a surprising observation. Recall that d is the diameter of the smaller particles. Figure 24 is a plot of g_{12} and reveals structure in the neighborhood of $X = 4.5$ and 10. And Fig. 25, a plot of g_{22} , reveals order at $X = 2$ and in the neighborhood of $X = 7.7$.

B. Pack with size ratio 6.5:1, case B

Here we change the mass ratio to 75% coarse, 25% fine, achieving a 73.1% packing fraction. Now there are so few coarse particles that it is difficult even to observe trends in g_{11} , and so we do not show it. Figure 26 displays five well-defined peaks for X smaller than 8 for g_{12} , and Fig. 27 shows a number of peaks for g_{22} . Some of these can be understood within the monodisperse framework, but others are a consequence of the bidisperse morphology. Thus a layer of small particles in contact with a large one will generate a peak in g_{12} at $X = 3.75$, Fig. 28. The closest small particles in a second layer beyond the first define a peak at 4.61, Fig. 28, the furthest at 4.75, Fig. 29, and so one might expect a single averaged peak between these two limits.

XVII. BOUNDED MONODISPERSE PACKS

In the earlier sections we only considered unbounded periodic packs, but in applications there are always boundaries,

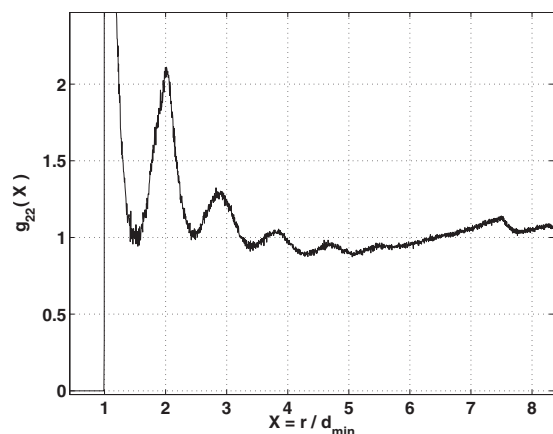


FIG. 27. g_{22} -6.5:1 Bidisperse packing, case B.

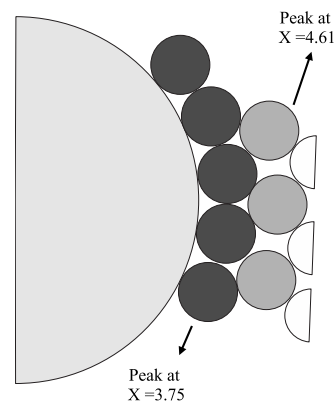


FIG. 28. A possible local configuration for a 6.5:1 bidisperse pack, (a).

and it is of interest to examine the effects of rigid boundaries on the pack morphology. We have examined three different pack geometries: a bounded cube, a bounded right-circular cylinder of infinite length, periodic along the axis, and a finite right-circular cylinder, bounded on all sides, but only present results for the latter configuration.

In the generation of such packs using the Stillinger algorithm, collisions must be accounted for between the spheres and the walls, and the code is easily modified to accommodate these. Our discussion is concerned only with the results. That there will be wall effects (a wall layer) is recognized, for example, in the experimental work of Aste and his colleagues [22]; because they are interested in the properties of unbounded packs, they randomly glue spheres to the walls in an attempt to eliminate the wall layer. Our purpose here is to identify the thickness of the wall layer, and some understanding of its morphology. We only consider monodisperse packs.

There are various ways of characterizing the local pack structure, and we choose to do it in the following fashion. We start by defining a sampling volume or bin. Thus for a cube, for example, a sensible choice would be a sheet of finite thickness parallel to two of the bounding faces; for a cylinder it would be either a cylindrical shell coaxial with the cylinder, or a finite thickness circular disk perpendicular to the axis. In each bin of volume V we count the number of sphere

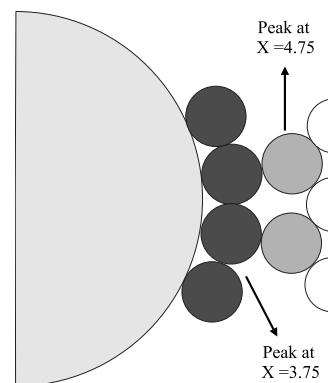


FIG. 29. A possible local configuration for a 6.5:1 bidisperse pack, (b).

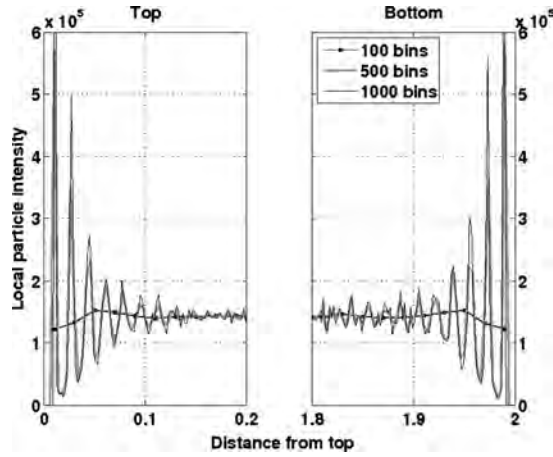


FIG. 30. Center-number density along the axial direction calculated for different bin numbers.

centers located within it, and use this number (N) as a measure of the number of particles within the bin, defining a center-number density by

$$\lambda = \frac{N}{V}. \quad (11)$$

Thus consider a bounded cylinder of height 2, diameter $2/3$, containing 100 000 particles. The packing fraction is 62.57%, the overall number density is 143 239.4, and each particle has a diameter of 0.0202. Figure 30 shows the axial variation of center-number density for different choices of the number of bins. The axial distance is defined by the center of each bin. Note that for 1000 bins each bin has thickness 0.002, for 500 bins the thickness is 0.004, and for 100 bins the thickness is 0.02, essentially the particle diameter.

A couple of observations may be made.

(1) If the particle centers were randomly distributed, the center-number density would be constant. This is approximately true at a distances greater than 0.1 from each end, but not within the wall layer which, accordingly, is approximately 5 particle diameters in thickness.

(2) The large oscillations within the wall layer imply order imposed by the boundary constraint. Indeed, there are 5–6 peaks in the layer, correlating with the particle diameter. Thus, for example, for a cubic lattice (for which the following estimates are easily made) there would be approximately 890 particles in each layer near the end walls, and for a bin thickness exactly equal to the particle diameter the center-number density would be constant with a value of approximately 1.28×10^5 . For a bin of vanishing thickness the center-number density would be zero except at distances $d/2, 3d/2, 5d/2, \dots$ from an end wall ($d=0.0202$ is the particle diameter) where the values would be approximately $2550/\mu$ where μ is the bin thickness, yielding 12.75×10^5 for the 1000 bin case, 6.38×10^5 for the 500 bin case.

Figure 31 shows the radial distribution of the center-number density for 100 bins (shell thickness 0.0033) and 200 bins (shell thickness 0.0017) and, as expected, here also the wall layer is 5–6 particle diameters thick. The expected

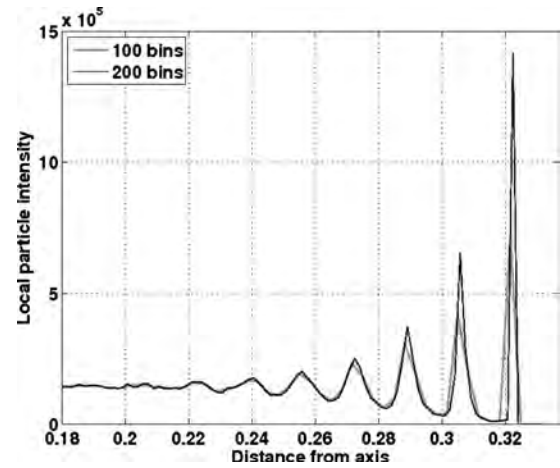


FIG. 31. Center-number density along the radial direction.

peaks for a local cubic lattice structure are approximately 7.73×10^5 for 100 bins, and 15.27×10^5 for 200 bins.

To gain more insight into the structure, we have replotted Fig. 31 in Fig. 32 using 333 bins (shell thickness 0.0010) and measuring the distance from the side wall in units of the particle diameter (scaled shell thickness 0.0496). Also the number of sphere centers is plotted, rather than their density. For a double-nested lattice the peaks would be at $\{0.5, 1.366, 2.232, 3.098, 3.964\}$; for a close-packed lattice they would be at $\{0.5, 1.317, 2.133, 2.950, 3.766\}$. We shall compare our peaks with the latter.

Thus the first peak (the 11th bin) is located in the interval $[0.496, 0.546]$, of which 0.5 is an interior point. The second peak is at the 28th bin, and the one to its immediate left (the 27th) is located in the interval $[1.290, 1.339]$, which includes 1.317. The third peak is at the 45th bin, and the one to its immediate left (the 44th) is located in the interval $[2.133, 2.182]$, which includes 2.133. The fourth peak is at the 61st bin, and the one to its left is located in the interval $[2.926, 2.976]$, which includes 2.950. The fifth peak is at the 78th bin, and the one to its left is located in the interval $[3.769, 3.819]$, which barely excludes 3.766. Thus, as would be expected for an imperfect lattice, there is a small outward displacement in the mean particle locations from what one

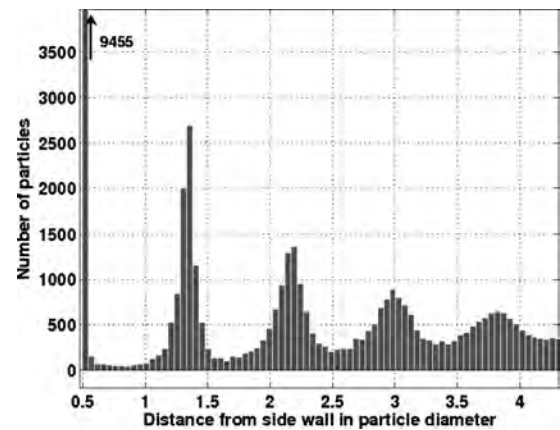


FIG. 32. Particle distribution vs wall distance (in particle diameters).

would get for a perfect lattice, but the correlation with the close-packed lattice is very strong.

XVIII. CONCLUSIONS

In this paper we have used a modified version of the Lubachevsky-Stillinger code to examine sphere packings in a context relevant to the modeling of rocket-propellant morphology. Various algorithmic refinements enable us to pack large numbers of spheres, of different sizes, in an efficient manner. Run times on a laptop computer are sufficiently modest that, with the addition of a graphic user-interface (GUI) (presently under development), the code could be of value to both the university and the industrial community. Packs can be generated in periodic cuboids, bounded cuboids, and in cylinders that are bounded or periodic along the axis.

The initial conditions include the specification of the velocities of randomly distributed points, and we do this in one of two ways: for the T algorithm, for which the maximum growth rate is 1 and the velocity or growth-rate ratio is controlled by a pseudotemperature T_0 , the initial velocity components are proportional to $\sqrt{T_0}$ and are sampled from a normal distribution; for the a algorithm, the ratio is controlled by a and the velocity components are sampled on the interval $[-1, 1]$. The T algorithm is capable of generating monodisperse packs of high density, higher than that of a double-nested lattice, and these high-density packs display significant order.

For bidisperse packs, using the a algorithm, we have compared the variations in packing fraction with the fine-mode percentage with experimental data of McGeary. For the larger coarse-to-fine size ratios (6.5 and 16.5) excellent agreement is achieved with the choice $a=0.2$. For the smaller size ratio (3.5) the choice $a=1$ yields better results, with the smaller a leading to significant overprediction. Apparently,

mechanical shaking is not particularly effective as a packing tool when the size ratio is small.

A study of the radial distribution function for monodisperse packs reveals sharp peaks corresponding to local order. Only a modest number of such peaks are generated using the a algorithm, or for modest values of T_0 using the T algorithm, but when T_0 is large a large number of peaks can be generated. When the latter are compared with the peaks defined by a lattice pack, a strong correlation is achieved with the close-packed lattice. For modest packing fractions (small number of peaks) there exist experimental measurements of the radial distribution function, and we obtain excellent agreement with this data. For bidisperse packs for which the corresponding statistical metric is the partial pair correlation function, certain peaks can be correlated with certain expected arrangements of large and small particles.

For bounded packs (rigid boundaries) it is intuitively clear that the presence of the boundary introduces local order. Indeed, there is a boundary-layer effect, and for monomodal packs the layer thickness is approximately 5 particle diameters. If we examine the pdf of the sphere-center distribution there are 5+ peaks in the neighborhood of a boundary that correlate approximately with integer multiples of the sphere diameter. When examined more closely, the peaks correlate closely with those that would be expected for a close-packed lattice.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy through the University of California under Contract No. B523819, and by the Air Force Research Laboratory under Contract No. FA9550-06-C-0078, program manager Dr. A. Nachman. We are grateful to T. Aste of the Australian National University for providing us with the pack morphology data used in the comparisons of Figs. 11, 12, and 20–22. Ownership of this data resides with ANU.

-
- [1] B. D. Lubachevsky and F. H. Stillinger, *J. Stat. Phys.* **60**, 561 (1990).
 - [2] G. M. Knott, T. L. Jackson, and J. Buckmaster, *AIAA J.* **39**, 678 (2001).
 - [3] S. Kochevets, J. Buckmaster, T. L. Jackson, and A. Hegab, *J. Propul. Power* **17**, 883 (2001).
 - [4] X. Wang, J. Buckmaster, and T. L. Jackson, *J. Propul. Power* **22**, 764 (2006).
 - [5] L. Massa, T. L. Jackson, and J. Buckmaster, *J. Propul. Power* **21**, 914 (2005).
 - [6] L. Massa, T. L. Jackson, J. Buckmaster, and M. Campbell, *Proc. Combust. Inst.* **29**, 2975 (2002).
 - [7] R. Lipton, *J. Mech. Phys. Solids* **47**, 1699 (1999).
 - [8] T. L. Jackson, F. Najjar, and J. Buckmaster, *J. Propul. Power* **21**, 925 (2005).
 - [9] M. D. Webb and I. L. Davis, *Powder Technol.* **167**, 10 (2006).
 - [10] D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic Press, New York, 2002).
 - [11] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, New York, 1987).
 - [12] A. Donev, F. H. Stillinger, and S. Torquato, *J. Comput. Phys.* **202**, 737 (2005).
 - [13] M. Mu, *J. Comput. Phys.* **179**, 539 (2002).
 - [14] S. Miller and S. Luding, *J. Comput. Phys.* **193**, 306 (2004).
 - [15] M. Wackenhut, S. McNamara, and H. Herrmann, *Eur. Phys. J. E* **17**, 237 (2005).
 - [16] A. R. Kansai, S. Torquato, and F. H. Stillinger, *J. Chem. Phys.* **117**, 8212 (2002).
 - [17] S. C. Harvey, R. K. Tan, and T. E. Cheatham III, *J. Comput. Chem.* **19**, 726 (1998).
 - [18] C. P. Lowe, *Europhys. Lett.* **47**, 145 (1999).
 - [19] M. Matsumoto and T. Nishimura, *ACM Trans. Model. Comput. Simul.* **8**, 3 (1998).
 - [20] Aleksandar Donev, Salvatore Torquato, and Frank H. Stillinger, *Phys. Rev. E* **71**, 011105 (2005).
 - [21] R. K. McGeary, *J. Am. Ceram. Soc.* **44**, 513 (1961).

- [22] T. Aste, M. Saadatfar, and T. J. Senden, *Phys. Rev. E* **71**, 061302 (2005).
- [23] Aleksandar Donev, Salvatore Torquato, Frank H. Stillinger, and Robert Connelly, *J. Appl. Phys.* **95**, 989 (2004).
- [24] Alexander Bezrukov, Monika Bargiel, and Dietrich Stoyan, *Part. Part. Syst. Charact.* **19**, 111 (2002).
- [25] K. Lockmann, L. Oger, and D. Stoyan, *Solid State Sci.* **8**, 1397 (2006).
- [26] William M. Visscher and M. Bolsterli, *Nature (London)* **239**, 504 (1972).
- [27] G. Mason and W. Clark, *Nature (London)* **211**, 957 (1966).
- [28] John Leech, *Math. Gaz.* **40**, 22 (1956).
- [29] J. D. Bernal and J. Mason, *Nature (London)* **188**, 910 (1960).
- [30] G. Mason, *Nature (London)* **217**, 733 (1968).
- [31] T. Aste, M. Saadatfar, A. Sakellariou, and T. J. Senden, *Physica A* **339**, 16 (2004).
- [32] Keishi Gotoh and John L. Finney, *Nature (London)* **252**, 202 (1974).

Using level sets for creating virtual random packs of non-spherical convex shapes

D. Shane Stafford*
Thomas L. Jackson†

August 10, 2009

Abstract

Random packs of spheres have been used to model heterogeneous and porous material morphologies during simulations of physical processes such as burning of coal char, convective burning in porous explosives, and regression of solid rocket propellant. Sphere packs have also been used to predict thermo-mechanical properties, permeability, packing density, and dissolution characteristics of various materials. In this work, we have extended the Lubachevsky-Stillinger (LS) sphere packing algorithm to create polydisperse packs of non-spherical shapes for modeling heterogeneity in complex energetic materials such as HMX and pressed gun propellants. In the method, we represent the various particle shapes using level sets. The LS framework requires estimates of inter-particle collision times, and we predict these times by numerically solving a minimization problem. We have obtained results for dense random packs of various convex shapes such as cylinders, spherocylinders, and polyhedra, and we show results with these various particles packed together in a single pack to high packing fraction.

*Theoretical & Applied Mechanics, University of Illinois at Urbana-Champaign

†Center for Simulation of Advanced Rockets at the University of Illinois at Urbana-Champaign

1 Introduction

Heterogeneous solid energetic materials are widely used in the aerospace and defense industries in rockets, explosives, and diverse pyrotechnic devices. The microstructure of these composites are often composed of particles of a crystalline oxidizer, such as ammonium perchlorate (AP) embedded in a polymer such as hydroxyl-terminated polybutadiene (HTPB) or polybutadiene acrylonitrile (PBAN). The polymers serve both as a binding agent and a fuel. Metal flake or powder, usually aluminum, may be added to increase the energetic content of the composite. Other composites may substitute crystals of an energetic material such as HMX for the oxidizer. Particle sizes for the composites range from a few μm for oxidizer particles in the “dirty binder” to 10s of μm for the metal flakes to 100s of μm for the largest oxidizer crystals. In explosives, the solids are loaded to mass fractions beyond 0.90, corresponding to volume fractions of 0.6-0.8. To achieve these high solids loadings, the materials are often compressed before the binder cures during manufacture. This results in particle breakage [1]. The final particle shapes are more polyhedral than spherical, but high aspect ratios seem to be rare [2]. Under conditions supporting deflagration, these composites burn with a reaction zone a few 100 μm thick. In certain scenarios, the deflagration can transition to a detonation [3]. Shown in Fig. 1 is a slice through an HMX-based explosive, illustrating the microstructure of a typical explosive. The HMX crystals range from a few μm to several hundred μm in size and are embedded in a rubbery binder

Porous energetic materials are also common in the aerospace and defense industries, where the porosity is present either by choice, as in the case of rocket igniters, or by nature, as in the case of aging fractured explosives. These materials often also have heterogeneous microstructure, having been manufactured using the composites mentioned above. The grains in igniters and gun propellants are often characterized by a particular shape, such as perforated rods, spherocylinders, or prismatic stars, into which they have been pressed or cast. These grains are poured loosely or packed tightly into rigid containers such as gun shells. The size of propellant grains varies widely, from several 100s of μm to 10s of mm. Solids loading ranges from 50 to 70 % by volume (the remainder is void space). The grains are often coated with deterrents that reduce the initial burn rate and enhance propellant performance [4] [5] [6] [7]. Fractured energetic materials have significantly different properties than gun and igniter propellants, but they share some

common porous material characteristics. The fractures develop over time as the explosive ages, increasing the porosity to the point that the overall device becomes unstable and unsafe. Because fractured explosive began life as solid composites, they typically exhibit much lower porosity than igniters and gun propellants, with volume fractions ranging from 0.7 to 1.0 [8]. In either case, these porous materials have macroscopic burn characteristics quite different than the solid composites [9].

Designers of devices using these materials are concerned with various properties of the materials as they relate to safety and engineering issues. These include thermo-mechanical properties such as Young's modulus, mechanical stability and thermal conductivity, as well as burn rate and metal agglomeration characteristics, among others. These bulk properties are strongly dependent on the morphology of the materials, and it is thus necessary to have a proper model for the morphology. Packs of disks and spheres have been extensively used for this purpose with much success. In particular the dependence of the burn rate on the morphology of composite AP/HTPB has been studied extensively using packs of spheres [10] [11]. At least for burn rate studies, spheres appear to be a reasonable approximation to the shape of the particles, and a study of spheroids showed little dependence of burn rate on sphericity [12]. Likewise, much success has also been reported in aluminum agglomeration model development with packs of polydisperse spheres [13].

Nevertheless, spheres can be poor approximations to heterogeneous and porous materials when various other properties are of interest. Bulk material properties such as thermal conductivity [14] and elastic properties [15] strongly depend on the statistical details of the microstructure, and spheres do not properly replicate these statistics. Efforts to design a statistically optimal periodic unit cell as a model of the microstructure of a composite energetic material also depend on better approximations of the shape of the particulates [16].

Porous flow is another area where spheres may perform poorly as models of morphology. Understanding the fluid flow in porous energetic materials is crucial when designing explosive devices to ensure their proper operation and long term safety [6]. Fluid flow phenomena strongly affect burn rates in porous materials and may play a strong role in the detonation to deflagration transition (DDT) in energetic materials. DDT is an important topic for the safety and reliability of the nuclear stockpile [8]. As with the thermo-mechanical properties, fluid flow through porous energetic materials is also

strongly dependent on the shape of the particle. For example, the well-known Ergun correlation for pressure drop through a packed bed is accurate for monodisperse spheres under many conditions and even works well for monodisperse packs of some non-spherical particles when the particle shape is adequately accounted for [17]. However, when used where the particles have large polydispersity or are moderately non-spherical, the Ergun relation fares poorly [18]. Other empirical correlations for the drag coefficient and permeability of a porous medium also work well for spheres, but fail for more complex structures. In one particularly relevant experiment, empirical relations for predicting the permeabilities of packed beds of spheres were found to over-predict the measured permeability of the non-spherical explosive CP by factors of 5-50 [19].

The latter two classes of problems demand a more accurate representation of the particle structure than can be provided by spheres. Thus, the major focus of this research was to devise a method for generating packs of arbitrary particle shapes with high packing fractions and large particle size ratios.

This article is organized as follows. First, in Section 2, we briefly describe *Rocpack*, our existing code for packing large numbers of polydisperse spheres. We provide further motivation for extending *Rocpack* to handle non-spherical shapes, and survey similar methods that have been presented in the literature. In Section 3, the new method for generating packs of arbitrary shapes is presented. The method builds upon the LS algorithm, introducing a flexible shape representation without changing the original LS framework. A few representative packs of various kinds of shapes are demonstrated, including some packs that resemble some real propellants, in 4. Comparisons are made to the equivalent packs of spheres and some higher-order statistics are shown. Finally, we summarize our conclusions and suggest further research in Section 5.

2 Background for the method

Packings of disks (in 2D) and spheres are the workhorses of the materials science community and have been used for many decades to study physical processes such as liquid flow, crystallizations, hardening, melting, granular flow, gasification, and propellant combustion, among many others [20] [21] [22] [23] [24] [25]. The most obvious advantage of these shapes lies in the simplicity of their representation, which facilitates theoretical, experimental,

and computational efforts in materials science.

As noted above, energetic materials tend to have low porosity, and so a key requirement of any sphere packing algorithm used in this context is the capability of generating packs with high packing fractions. In addition, one more constraint applies for our particular focus: the algorithm must handle polydispersity in an efficient manner. For propellant burn rate studies, the resolution of the numerical simulation of the combustion is a limiting factor, and for these computations packs of only up to a few 100 particles are sufficient. However, the packs will often contain large amounts of polydispersity, and the ratio of the size of the largest particle to the size of the smallest particle can (and often does) exceed 50:1. For concurrent packing methods, this can greatly increase the time required to generate a pack.

In light of these critical constraints, we adapted the Lubachevsky-Stillinger (LS) packing algorithm for spheres and have used it in our code *Rocpack* to successfully model energetic materials for almost a decade [11]. The LS algorithm begins by placing N spheres with zero initial radii (points) at random locations inside the domain of interest. The spheres are given random velocities $\mathbf{v}_i$ sampled from a Maxwell-Boltzmann distribution at temperature Θ_0 and are allowed to grow at a specified growth rate g_i . The particles undergo classical (super-)elastic collision dynamics as they grow to fill the space in the domain. The algorithm stops when either a specified density is reached or when a specified jamming criterion is met.

Modern implementations of rigid sphere packing take advantage of an event-driven molecular dynamics (EDMD) approach. In EDMD, particles are advanced between "events," where an event is loosely defined as anything that changes a particle's state. The event could be a binary collision, a collision between a particle and a domain boundary, or a transfer of a particle across an internal or external boundary. Instead of advancing the particles by a fixed time step as in time-driven MD (TDMD), the particles are always advanced to the next event time.

As an improvement to the LS algorithm, we use a hierarchical cell structure to drastically reduce the computation time when the particle set is very polydisperse. Cell schemes in general are well-known in the MD community, and a dynamic hierarchical cell variant has been implemented in two dimensions by Wackenhut *et al.* [26]. However, we are not aware of any other implementation of a fully three dimensional multi-level cell method. The hierarchical cell scheme allows *Rocpack* to quickly create packs of hundreds of thousands of spheres with size ratios of up to 100:1 on a laptop in a matter

of hours. Details of the algorithm are presented in Refs. [27] and [28] along with validation of the sphere packs and statistical analyses of the results.

The focus of the current article is a recent extension to *Rocpack*: a new packing method for arbitrary convex shapes that can achieve high packing fractions without introducing artificial ordering. The method can, in principle, be applied to pack any set of shapes at any size ratio, provided one can first calculate the signed minimum distance field for each shape. Because the new method is still based on the LS algorithm, it leverages two important properties of the LS method: (1) capability for very high packing fractions, and (2) excellent statistical properties [27].

In the discussion that follows, we will compare various methods for representing the arbitrary shapes in a pack, including other methods for packing arbitrary shapes that are discussed in the literature. Note that *any* of the methods discussed below could be adapted for use within the LS framework. To reiterate, the major desirable properties of the overall method when used to model energetic materials are the following: (1) it should be able to generate packs with high packing fractions $\rho > 0.6$, (2) it should efficiently handle large differences in particle sizes (polydispersity), (3) it should reasonably reproduce the randomness present in real materials, and (4) it should be able to produce packs of $N > 100$ in a reasonable amount of time.

For the representations of the shapes themselves, we also have specific requirements, some which parallel the requirements listed above for the overall method. Our interest is primarily in packs that resemble real rocket propellant and explosive microstructures. These microstructures often consist of many different shapes and many different sizes bound together in a single composite. Some of the shapes may resemble polyhedra; others may be approximated by cylinders, etc.; and all of these must be packed together in the same pack. Thus, we would like to be able to specify the particle size distribution, the particle shape distribution, and the porosity so that we have the opportunity to accurately model these details of real microstructures.

Two other requirements for the shape representation stem from the use of the LS algorithm. First, the shape representation must be consistent or nearly consistent under any arbitrary affine transformation: that is, when we compute the distance to the shape from any point, the computation must yield results that are independent of orientation and translation to within a small tolerance. If there is too much error in the distance estimates, then the LS algorithm may fail. Thus, the shape representation need not be an exact, closed-form analytical relation. Numerical methods are also accept-

able if they can maintain a small enough error in the distance estimate. This tolerance is difficult to determine *a priori* but easy to determine in the implementation of the method.

The second requirement that is due to the use of the LS algorithm is that the distance computation between any two particles must be a fairly efficient process. The LS algorithm is a concurrent algorithm, and as such, is not exceedingly fast. We will show in Section 3 that the LS framework will require numerically finding the root of the distance function between two shapes as they move in time. This root solve is already rather expensive, and adding the additional burden of a numerical distance estimate will increase CPU time. This is, however, the route we choose.

Thus, our requirements for a shape representation can be summarized. (1) The representation must be sufficiently generic to handle various shapes such as spheres, cylinders, and polyhedra. (2) Multiple shapes must be possible within a single pack. (3) We must be able to specify the shapes. (4) The representation must be consistent under affine transformations. (5) The representation must be reasonably efficient. Of course we may choose to trade efficiency to satisfy the other requirements.

There have been very few attempts to create packs of non-spherical shapes reported in the literature. Of the methods that have been published, there are three main classes of methods for representing the 3D shapes: methods for shapes that have convenient analytical representations, voxel based methods, and Voronoi methods. The first class of methods is very useful in cases where the desired particle shapes have clean analytical representations and there exists a tractable method for computing the distance between the shapes (either analytically or numerically). Examples of the use of such methods include packing ellipsoids to high density [29], packing super-ellipses (rounded rectangles) in 2D [30], and packing ellipses and spheroids at various orientations as a model of rocket propellant [12]. These examples are all based on the LS algorithm and thus achieve high packing fraction and can be expected to yield good statistics. A commercial software application, MacroPac [31], uses a “blackberry” model – agglomerations of spheres – to model virtually any shapes in any combination in a single pack. MacroPac is a sequential method and is thus most useful for low packing fractions where the ordering and arrangement of the particles are not crucial in the end result.

Analytical shapes can have major advantages. The first and most obvious advantage is computational efficiency if a closed form solution exists for determining the distance between the two analytical shapes. This is of course

true for sphere-sphere interactions, but is unfortunately not true for even moderately more complex shapes. For example, determining the distance between two cylinders at arbitrary orientation in 3D is extremely difficult because of the need to determine the distance between the 4 circles at the ends of any two cylinders. In such situations, the analytical shape might still be used, along with a numerical minimization or root solver for determining the distance between the shapes. However, as we will discuss further, this can greatly increase computation time.

Another advantage for analytical shapes is that it is possible to represent them exactly. The ability to calculate exact distance to a surface eases the root finding process for determining the distance between two shapes by reducing the overall amount of error in the process. This is especially important for the LS algorithm, which has tolerance issues when the collision prediction is not exact. If the overall error in the process is too large, then the particles might overlap slightly, causing the LS heap to get stuck. Having an exact calculation for the location of a particle surface reduces the likelihood of occurrence of this phenomena.

Even if an analytical representation is not available for a particular shape of interest, an analytical method might still be possible using constructive solid geometry (CSG). With CSG, complex solid shapes are constructed from boolean operations on simple analytical shapes such as spheres, cylinders, and polyhedra. CSG is available in many commercial solid modeling and visualization applications.

If CSG is considered as a viable solution for creating more complex (but still analytical) shapes, then the major drawback of using analytical shapes is an overall lack of flexibility to accommodate new shapes. For each pair of shapes, one must be able to predict collisions, which in turn requires calculating the distance between the two shapes. When the two shapes are spheres or polygons, this is not difficult. However, adding a new shape requires adding new code to compute the distance between the new shape and all existing shapes. Thus, the development cost of adding a new shape becomes prohibitive as the number of shapes increases.

The second drawback of using analytical shapes was mentioned above briefly. There is no guarantee that an analytical solution can be found for the distance between two shapes that have simple, closed form descriptions of their surface locations. For example, computing the distance between two circles in 3D is a notoriously difficult problem [32]. Thus, for some shapes, it may be more simple to compute the distance numerically – even though the

shapes may be very simple. However, if a numerical method is used for the distance calculation, the first disadvantage discussed above still remains, and is in fact made more important because of the limited scope of the numerical method. Also, the computation time can be increased drastically because each collision prediction requires multiple distance computations, which each in turn compute many function evaluations (more on this later).

Still, a CSG method that restricts itself to a particular simple shape or set of shapes, such as MacroPac’s blackberry model, can be very useful for many physical problems. A method similar to MacroPac based on CSG using only voxels (a voxel is the analog of a pixel in 3D) was presented in [33] and commercialized as DigiPac. Although DigiPac is a sequential method, it has evolved to include DEM-like forces and has been able to achieve fairly high packing fractions with small numbers of particles ($N \leq 175$). DigiPac has been successfully validated with experiments (using only the packing fractions ρ) with various particle shapes [34]. In the propellant and energetic materials context, the disadvantages of the method used by DigiPac are that it cannot achieve high enough packing fractions and the shapes are not resolved below the voxel resolution (they are not smooth). The former shortcoming is crucial for our problems; the latter is only important when fine detail is needed on both small and large particles.

Another widely-utilized approach to creating packs of non-spherical shapes is based on the Voronoi tessellation. The Voronoi method begins with a tessellation about random points and then shrinks the resultant polyhedra to achieve a particular packing fraction. This approach has been used in the combustion and energetic materials community [35] [18] for the compelling reason that it can be used to create packs with *any* packing fraction $0 \leq \rho \leq 1$. The method also yields very realistic-looking particle shapes. Among the methods discussed here, the Voronoi-based packing codes are the only ones able to reach the high packing fractions that suit our needs. Unfortunately, the method is not without shortcomings.

First, it is difficult to alter the shape of the resulting polyhedra. In particular, it is impossible to alter the aspect ratios of the particles without affecting the packing fraction. For packs with low packing fractions, this might not be a problem, since there is likely to be ample space to change the aspect ratios while keeping the particles from overlapping. At high packing fractions, however, there is unlikely to be enough space to significantly alter the aspect ratio. A second, but similar problem exists when there is large polydispersity. Unless the particles can be significantly shrunk, all of the

largest particles will be roughly spherical due to the properties of the tessellation itself. Since the shape of the largest particles in a microstructure is of utmost importance, this is a major drawback.

Another problem with Voronoi methods is that although the resulting polyhedra are truly random, their orientations are completely non-random with respect to their neighbors' orientations. In 2D, the faces are seen to fit together like puzzle pieces or the spots of a giraffe, indicating a lack of randomness. Depending on the details of the algorithm, it is also likely that the closest distance between any two neighboring particles is a constant. However, at this point, we cannot explicitly quantify the effect of this lack of randomness would have on computed physical properties.

In light of the fact that none of the available methods meet the requirements detailed above, we propose a new method based entirely on level sets. Level set methods (isosurfaces) have some major advantages. First, it is possible to represent *any* shape for which one can calculate the signed minimum distance field. Also, because the underlying data structure is always a scalar field (an isosurface), it is possible to mix different shapes with no changes to the method. In addition, subgrid resolution enables precise surface mappings with small amounts of discrete data, limited only by the order of the interpolation method. A simplifying property is that the method to calculate the distance between the shapes can be independent of the shapes. Lastly, isosurface scaling (necessary for concurrent packing) is of course trivial if the signed minimum distance is used to represent the shape. Despite these advantages, we are not aware of any other use or proposed use of level sets or other type of isosurface in the context of computer generated random packs.

However, level sets have been used in other discrete element modeling contexts. For example, a few of the properties of level sets were exploited in a stacking algorithm by Guendelman, *et. al*, where a level set was coupled with a triangulated mesh to produce fast inside/outside checks and normal vector computations. This method could possibly be adapted to work within the LS algorithm, but the authors chose not to do so for two reasons. First, the method requires both a triangulation and a level set field for each object. For smooth objects, triangulation reduces the accuracy of the surface representation, and this is unnecessary if the level set can be used alone. Second, the method uses the triangulation vertices as sample points for evaluating the level set field. This is very similar to a direct search method. Although direct search methods may be acceptable for stacking, they are prohibitively expensive within the LS packing framework due to the tolerance require-

ments of the latter method. Thus, we developed an algorithm that relies on level sets as the sole representation of the particle’s surfaces and that also computes inter-particle distances in constant time (independent of the mesh resolution). Of course no method is without its challenges, and these will also be discussed below.

3 Algorithm

Because this algorithm will be used in the context of the LS framework for packing, we can view the shape representation as an unspecified function or a “black box.” We put into the function a pair of particles and the function returns a prediction for the time of the particles’ next collision (or ∞ if they will not collide in their current trajectories). Additionally, if the two particles are actually the next pair to collide, the LS algorithm will also require the location of the contact and a vector normal to the surface at the point of contact so that the proper changes in trajectory can be applied after the collision is processed.

For two particles i and j the black box functions can be approximately described by the pseudocode of Algorithm 7.1. The function `PREDICT-COLLISION` takes as arguments the two particles i and j and returns the predicted time of the next collision between the particles. The member function `UPDATE` simply applies a time step update to a particular particle to calculate its position, velocity, orientation, and angular momentum after the time step Δt . `PREDICT-COLLISION` functions as a root solve, searching for the first time after $t = 0$ that the distance d falls below zero. Note that much of the detail is hidden in the important function call to `COMPUTE-DISTANCE( $a, b$ )`, which returns the instantaneous signed minimum distance between the two particle copies a and b .

The procedure described above is obviously simplified somewhat, but it illustrates the basic computational task: we must integrate the particle positions in time, checking for overlap after each small time step Δt . The rest of the LS framework does not know or care how this is done – a prediction of the next collision time is all that is necessary. This property makes it relatively easy to incorporate any shape representation into the LS framework as long as one can implement a suitable `PREDICT-COLLISION` function for the representation. In the next section, we delve further into the details of the root solver.

One of the most desirable characteristics of the level set representation is that we can easily (in principle) compute a “distance” function that is piecewise smooth and monotonically decreasing at the instant that the shapes begin to overlap. This has important ramifications when attempting to zero in on an impending collision. According to the skeleton definition of $\text{PREDICT-COLLISION}(i, j)$ above, we need only a boolean check on $d_{ij} > 0$. We have presupposed that d_{ij} is available and is also well-behaved as the particles begin to overlap. In fact, we do not need to have a distance function at all; any metric that indicates when particles have overlapped can (in principle) be used to find the exact instant that the particles touch. Other methods for finding the collision point between arbitrary shapes include simply checking for voxel overlap[33] or scaling the particle sizes smoothly until overlap is detected by some other method [29]. Of these methods, the former is simply a boolean check and the latter generates a smoothly varying function that changes sign at the moment of overlap. However, a boolean check is a poor method for finding a root, with a convergence rate of $O(\Delta t)$ (note that the method in Ref. [33] does not use the LS algorithm, and thus never needs to find a root). A smooth function is much more suitable for pinpointing the precise location of an impending collision, but the method used in [29] is not generally available for arbitrary shapes. Fortunately, as we will show below, we can directly compute the signed minimum distance function $d_{ij}(t)$ for two level set shapes i and j , and furthermore, this function is well-behaved at the zero crossing.

Assuming we can compute the instantaneous distance $d_{ij}(t)$ between two shapes represented by level sets, we must then find the next root of $d_{ij}(t)$ numerically. Because we intend for this method to be used with arbitrary shapes, there will be no analytical method for finding the root. We can, however, gain some insight into how the shape of $d_{ij}(t)$ might look by decomposing the shape’s motion into that of its maximum enclosed sphere (with radius r_i) and its non-spherical motion. The distance between the two maximum enclosed spheres $d_{12,\text{sphere}}(t)$ looks something like a parabola

$$d_{12,\text{sphere}} = \sqrt{\|\mathbf{v}_{ij}\|^2 t^2 + 2\mathbf{v}_{ij} \cdot \mathbf{x}_{ij} t + \|\mathbf{x}_{ij}\|^2} - (r_{ij} + g_{ij}t) \quad (1)$$

where $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$, $r_{ij} = r_i + r_j$ and $g_{ij} = g_i + g_j$. This function is nearly parabolic near the point of collision and becomes linear as $t \rightarrow \pm\infty$. Superimposed on this distance will be the periodic oscillations of the rotational degrees of freedom of the two non-spherical shapes. Thus, in general, $d_{ij}(t)$

will be highly nonlinear and oscillating near the next collision $d_{ij}(t) \rightarrow 0^+$.

Shown in Fig. 2 is the computed $d_{ij}(t)$ for two gelcaps (spherocylinders). To make this figure, the two gelcaps were given random velocities, angular velocities, orientations, and positions and were allowed to pass through each other. The overall shape resembles Eq. 1, but there are oscillations due to the changing orientations of the particles. Note that in general, these oscillations do not have a regular period and that they can cause rather sharp peaks to be present in the distance function $d_{ij}(t)$.

To find this root, we resort to an ODE-like curve fit and attempt to find the next t for which $d_{ij}(t) < \epsilon$, where ϵ is some small tolerance below which we are not concerned with overlap. Donev, *et. al* have implemented a method using Hermite polynomials for ellipsoids [29]. For arbitrary shapes, $d'(t)$ is not generally available, and so we choose a piecewise quadratic fit. An error estimate is made at each step and the size of the interval is adapted if the following criterion is not met:

$$\int_t^{t+\Delta t} d_{ij}^*(\theta) d\theta > -\epsilon \quad (2)$$

where d_{ij}^* is the piecewise polynomial fit to d_{ij} . At each interval, the fitted quadratic function is checked for a bracketed positive root. If one is found, the interval is further refined using Brent's method [36]. Thus, although small collisions may be missed due to finite ϵ , any scheduled collision will be predicted to high precision. As previously explained, this precision is critical to the success of the Lubachevsky-Stillinger algorithm.

We now show more detail of the black box in Algorithm 7.2, where we have deferred the shape copying and `UPDATE( $t$ )` member functions into the function `COMPUTE-DISTANCE( $i, j, t$ )` so that the function of the root solve is clear. Next, we explain our algorithm for the `COMPUTE-DISTANCE( $i, j, t$ )` function, which exposes the details of our shape representation.

3.1 Level set shapes

A level set (or isosurface) has some nice properties that make it useful for our purpose. The shape's surface $\Gamma(\mathbf{x}, t)$ (in 3D) is represented implicitly by the zero level set of an auxiliary field $\phi(\mathbf{x}, t)$

$$\Gamma(\mathbf{x}, t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\} \quad (3)$$

where the t dependence is included because our shapes translate, rotate, and grow in time. Our task is simplified if ϕ is the signed minimum distance to Γ

$$\phi(\mathbf{x}, t) = \alpha \min_{\mathbf{y}} \{ \|\mathbf{x} - \mathbf{y}\| \mid \phi(\mathbf{y}, t) = 0 \} \quad (4)$$

where we take the sign $\alpha = 1$ for points $\mathbf{x}$ that lie outside the shape and $\alpha = -1$ for points that lie inside the shape. Then we have $\|\nabla\phi\| \equiv 1$, and $\nabla\phi$ is a normal vector perpendicular to the surface Γ .

The shapes will in general be undergoing affine transformations in three dimensions and will also be growing according to the Lubachevsky-Stillinger algorithm. For a general affine transformation, we have

$$\mathbf{x} = \sigma(t)\mathbf{A}(t)\mathbf{x}_0 + \mathbf{b}(t) \quad (5)$$

where $\mathbf{A}(t)$ is a rotation tensor, $\mathbf{b}(t)$ is a translation vector, $\sigma(t)$ is the scaling factor, and $\mathbf{x}_0$ is the position in the reference frame at $t = 0$. Our shapes will be bounded by their minimum enclosing spheres, so we scale the level set so that $\sigma(t) = R(t)$, where $R(t)$ is the radius of the enclosing sphere. Unfortunately, the signed minimum distance fields are nonlinear and so there is no linear operator that can be applied to map $\phi(\mathbf{x}_0, t = 0) \rightarrow \phi(\mathbf{x}, t)$. Instead, the transformation must be applied pointwise,

$$\phi(\mathbf{x}, t) = \sigma(t) \phi(\sigma(t)\mathbf{A}(t)\mathbf{x}_0 + \mathbf{b}(t), t = 0) \quad (6)$$

which is very expensive.

With the Lubachevsky-Stillinger packing algorithm, we only deal with two shapes at a time, and as discussed in the previous section the kernel of the algorithm requires predicting the time of the next collision between those two shapes via $\text{PREDICT-COLLISION}(i, j, t_{max})$. The level set representation must provide a means to compute the instantaneous distance between two shapes at any time by the function $\text{COMPUTE-DISTANCE}(i, j, t)$. We begin by transforming both shapes back to a reference configuration, computing the inverse of Eq. [5] for each shape. Then, we combine the shapes by taking their intersection. The intersection of the two level sets is just the maximum of the two fields ϕ_1 and ϕ_2

$$\phi_{ij}(\mathbf{x}, t) = \max\{\phi_i(\mathbf{x}, t), \phi_j(\mathbf{x}, t)\} \quad (7)$$

and the resulting field $\phi_{ij}(\mathbf{x}, t)$ is also a signed minimum distance field if the shapes are overlapping. Whether or not the shapes overlap, the intersection

provides useful information because we can use it to calculate the signed minimum distance between the two shapes. To accomplish this, we compute the minimum of their intersection

$$d_{ij}(t) = 2 \min_{\mathbf{x}} \phi_{ij}(\mathbf{x}, t) \quad (8)$$

where a negative d_{ij} indicates that the shapes are overlapping. Thus, as shown in Algorithm 7.3 COMPUTE-DISTANCE(i, j, t) is rather simple. Again, we have deferred most of the work to the function GET-MINIMUM($i, j, \mathbf{x}_0$), which attempts to locate the minimum of the intersected fields. Having found the minimum, we can use Eq. 8 to compute the signed minimum distance.

We now discuss the process of locating the minimum of the intersected level set field $\phi_{ij}(\mathbf{x}, t)$, which is a difficult task. Multivariate, nonlinear minimization is in general a hard problem for smooth functions, but it becomes almost impossible for functions that have strong discontinuities in the gradient [37]. Level sets and their intersections are functions of the latter type, and for intersections in particular, there are guaranteed to be surfaces of gradient discontinuity where the gradient – and thus the direction of steepest descent – changes abruptly. This effect is caused by the `min` functions in Eqs. [4] and [8] and is present whether or not the shapes are smooth. Furthermore, for an intersection, the minimum will always lie on one of these surfaces where the gradient does not exist. An additional complicating factor is that the fields are always locally linear in the direction of $\nabla\phi$. Thus, even if we travel downhill in the direction of steepest descent, we can never gain more information about where the minimum will be located until we pass a location where $\nabla\phi$ actually changes. Locally, the fields have no characteristic length scale. Because of the linearity of the fields and discontinuous gradients, quadratic methods such as Newton-Raphson or BFGS and conjugate gradient methods will fail in this degenerate case.

An example of the two dimensional intersected level set field $\phi_{ij}(\mathbf{x}, t)$ for two different convex shapes is shown in Fig. 3. Outlines of the two shapes are shown in black. The color field and the white contour lines show the values of the intersected field $\phi_{ij}(\mathbf{x}, t)$. The signed minimum distance field has a smoothing effect far from the surfaces of the shapes, but this feature is of limited value in this context. Because of the `max` function in Eq. 7, note that the left side of the figure contains contours (and field values) for the polygon, while the right side of the figure contains the contours for the gelcap on the left side. These contours meet at the ravine passing vertically through the

field. The global minimum lies in this ravine, inside the innermost contour, and the gradient does not exist at the minimum.

In this situation, it would seem that there are two options: 1) try a robust direct search method, or 2) design a new method. We have already achieved partial success with the former option, having applied the Nelder-Mead simplex method [38] as well as the Hooke and Jeeves direct search [39]. Of the two, the simplex method fared better and was able to find the general location of the global minimum quite reliably. The Hooke and Jeeves search tended to require many more function evaluations than Nelder-Mead to achieve the same level of tolerance. However, the Lubachevsky-Stillinger algorithm is not very tolerant of even very small errors in the collision prediction times, and neither method was able to consistently provide good estimates with relative tolerances below 10^{-3} . The poor performance for both methods occurs when the minimizer gets stuck on a surface of gradient discontinuity. It is easier to visualize the problem in 2D, where a height field indicates the value of the objective function. In 2D, the surfaces where the gradient is undefined become ravines where the slope changes abruptly. Both the Nelder-Mead and the Hooke-Jeeves methods move quickly downhill into the nearest ravine. Near the bottom of the ravine, they shrink their respective search templates rapidly to convergence, but are unable to make forward progress in the direction of the ravine. They are thus often unable to find the location of the true minimum.

Therefore, a major focus of this research was concerned with the second option: design of a new minimization method. In spite of the pitfalls discussed above, the signed minimum distance fields do have features amenable to minimization. One of these properties is the aforementioned linearity of the fields. If we perform a line search in the $-\nabla\phi$ direction, then we are guaranteed to travel downhill at a slope of unity until a discontinuity (in the gradient) is reached. In the remainder of this discussion, we will call these discontinuities “ravines” for brevity and because it is easier to visualize the process in 2D. At a ravine, the best search direction is the one that travels *in the direction of the ravine*. The location and direction of the ravine are both easily estimated by linearizing the field on either side of the ravine. We linearize the field separately at two locations $\mathbf{x}_a$ and $\mathbf{x}_b$ that straddle the ravine, dropping the time dependence for clarity

$$\phi_a(\mathbf{x}) = \phi(\mathbf{x}_a) + \nabla\phi(\mathbf{x}_a) \cdot (\mathbf{x} - \mathbf{x}_a) \quad (9)$$

$$\phi_b(\mathbf{x}) = \phi(\mathbf{x}_b) + \nabla\phi(\mathbf{x}_b) \cdot (\mathbf{x} - \mathbf{x}_b) \quad (10)$$

The linearized ravine is a line in 2D (or a plane in 3D). To find an appropriate location to begin a search, we project a line in the downhill direction at $\mathbf{x}_a$,

$$\mathbf{x}(t) = \mathbf{x}_a + t(\mathbf{x}_b - \mathbf{x}_a) \quad (11)$$

and find its intersection with the ravine. With some algebra, we find

$$t = \frac{\phi(\mathbf{x}_b) - \phi(\mathbf{x}_a) + \mathbf{n}_b \cdot (\mathbf{x}_b - \mathbf{x}_a)}{(\mathbf{n}_b - \mathbf{n}_a) \cdot (\mathbf{x}_b - \mathbf{x}_a)} \quad (12)$$

where the $\mathbf{n}_i = -\nabla\phi_i(\mathbf{x}_i)$ are the downhill normals. The estimate of the direction of steepest descent along the ravine is then

$$\mathbf{n}_{ab} = \frac{\mathbf{n}_a + \mathbf{n}_b}{2\|\mathbf{n}_a + \mathbf{n}_b\|} \quad (13)$$

Note that although the intersected field is linear with unity slope along the gradient direction, it is not in general linear in the direction of the ravine $\mathbf{n}_{ab}$. Also, with respect to the magnitude of the gradient, we know only that $\|\nabla\phi_{ij}\| \leq 1$ along the ravine.

We devised a method that uses these properties to find the true global minimum to high precision. Given a starting location $\mathbf{x}_0$, the method first moves downhill in the direction of steepest descent $-\phi_{ij}(\mathbf{x})/\|\phi_{ij}(\mathbf{x})\|$. When the minimizer encounters a ravine, the direction of the ravine is first estimated using Eq. 13, and then a line search is performed along that direction. The process then repeats.

One of the requirements for using this process (and a drawback to the overall level set shape representation) is that the shapes must be convex. The reason for this requirement is illustrated in Fig. 4, where we show the intersected level set field $\phi_{ij}(\mathbf{x})$ for two shapes that have concavities. The outlines of these shapes are shown in black, and the values of $\phi_{ij}(\mathbf{x})$ are again shown in color with white contour lines. The white contour lines show the existence of multiple minima in $\phi_{ij}(\mathbf{x})$ for these shapes. The method we have proposed could easily get stuck in a local minima and miss the global minimum, and so we require all of the shapes to be convex (but not strictly convex).

Simplified pseudocode for the minimization process is shown in Algorithm 7.4. The MOVE-DOWNHILL-INTO-RAVINE($\mathbf{x}, \mathbf{n}$) procedure begins at location $\mathbf{x}$ and moves in the downhill direction $\mathbf{n}$ until the function evaluations $\phi_{ij}(\mathbf{x})$ no longer decrease as expected. This signals the location of a ravine.

In the special case of signed minimum distance fields, the ravine may be found numerically by comparing the expected decrease in the field to the actual decrease in the field. Since signed minimum distance fields have a slope of unity in the search direction, the expected decrease is equal to the step size. The criterion for crossing a ravine is then

$$(\phi_{ij}(\mathbf{x} + \Delta\mathbf{x}, t) - \phi_{ij}(\mathbf{x}, t)) - \|\Delta\mathbf{x}\| > \epsilon \quad (14)$$

where $\Delta\mathbf{x}$ is the step and ϵ is some small tolerance. In practice, we have found $\epsilon = 0.05$ to work well. Larger values are also possible since it is not necessary to identify and search along ravines where the slope has only moderately changed.

Upon finding a ravine, the function will attempt to bracket the location of the ravine. **MOVE-DOWNHILL-INTO-RAVINE** is adaptive. If a particular step is successful, the step size is generally increased; likewise if the step is unsuccessful, the step size is decreased. However, we have found that there is a loss of efficiency if the function tries to find the location of the ravine to high precision. It is better to limit the step size adaptivity to say, 3 decreases, and then to move on to the line search rather than to attempt many function evaluations in this phase.

LINE-SEARCH-FOR-MIN($\mathbf{x}, \mathbf{q}$) begins at location $\mathbf{x}$ and performs a line search in the ravine direction $\mathbf{q}$. Brent's minimization method (a variation of the Golden Search) [36] is used to converge upon the location of the minimum along the line $\mathbf{y} = \mathbf{x} + t\mathbf{q}$. Brent's method requires a bracketed minimum, so before using the method, our algorithm attempts to quickly bracket the minimum by moving along the line with an adaptively increasing step size.

Acceleration is performed in the function **ACCELERATE-IN-DIRECTION**($\mathbf{x} - \mathbf{x}_{accel}$). Acceleration can be very useful in speeding up convergence to the minimum, especially in the (rather common) event that the intersected level set field $\phi_{ij}(\mathbf{x})$ has moderate curvature. If the curvature is significant, the downhill and line searches will tend to zig-zag towards the global minimum when it would of course be more efficient to move in an approximately straight line toward the goal. **ACCELERATE-IN-DIRECTION** thus projects a line search along the direction of a recent rolling average of the overall search progress, for example $\mathbf{x} - \mathbf{x}_0$, again performing an adaptive bracketing and successive Brent search.

As with almost all minimization methods, the method presented here greatly benefits from having a good initial guess for the location of the minimum. Since we already know that the bounding spheres of the pair of

shapes are overlapping, a good initial guess can be quickly found by locating the center of the overlap region of the bounding spheres.

An example run of the minimization routine is shown in Fig. 5, where the search has been restricted to two dimensions for ease of visualization. Also note that for illustrative purposes, the initial guess is not as described above. Outlines of two gelcaps (spherocylinders) are shown in black, while the field values and contours of $\phi_{ij}(\mathbf{x})$ are again shown as color and white lines, respectively. The path of $\text{GET-MINIMUM}(i, j, \mathbf{x})$ is shown in red. The initial guess was below the lower bound for y in the figure, so the path of the minimizer begins at the bottom of the figure. (We would normally start with an educated guess for the initial guess.) The function then moves in the downhill direction until it finds the first ravine. Then, it moves in the direction of the ravine, rapidly approaching the true point of contact. The inset presents a zoomed-in look at the action of the minimizer as the global minimum is approached, where the successive downhill-linesearch process can be seen.

The overall minimization method has been successful when used within the LS framework. The global minimum can typically be found with fewer than 100 function evaluations to a tolerance $\|\mathbf{x} - \mathbf{x}_{old}\| < 10^{-6}$. That is not to say that the minimization routine is optimal; it is not. In particular, although Brent's method can be second order, the linearization of the level set fields by Eqs. 9 and 10 causes the overall method to be at best first order. Nevertheless, as we will now show, the method does indeed work.

4 Results

4.1 Examples of packs

As mentioned previously, the method presented above can in principle be used to pack any convex shape for which one can compute the initial signed minimum distance field $\phi(\mathbf{x})$. Fortunately, these fields are also easy to calculate for many types of shapes including polyhedra, spherocylinders, cylinders, and of course spheres. We have computed the fields for these shapes and have successfully packed variants of each type.

Note that the method presented here is independent of the algorithm used to compute the signed distance fields. These fields may be computed on-the-fly for simple analytical shapes such as spheres and cylinders, or the signed

distances may be pre-computed for more complex shapes and then stored as a mesh (a 3D array) in memory. The mesh method must be coupled with some sort of 3D interpolation technique so that the interpolated values vary smoothly between mesh nodes. We have applied both analytical and mesh representations of the shapes presented in this work. For the meshes, we used trilinear, tri-quadratic, and tri-cubic polynomial interpolation. In all cases, the method was found to work independently of the mesh resolution so long as the interpolated field values were continuous (but the shapes and packing fractions necessarily depend on the ability to resolve the original shape). Higher order interpolations are of course computationally expensive, but this was alleviated somewhat by pre-computing and storing the final polynomial coefficients at each location on the mesh. However, the relatively simple shapes presented in this work all have straightforward analytical solutions for the signed distance field, and these analytical methods were favored for their speed.

Shown in Fig. 6 is a pack of 1000 gelcaps (spherocylinders) with aspect ratio 2:1 (L/D) in a periodic cube configuration. The gelcaps were started with a scaled temperature $\Theta_0 = 10^3$ (see Ref. [27], [40], or [28] for a discussion of packing temperature). Gelcaps are smooth and the intersected level sets are relatively well-behaved, so it is easy for the minimization routine to locate the minimum. Therefore, these represent an almost optimal scenario for the method (spheres would be optimal, but uninteresting), and if only gelcaps are present the code is only about 10 times slower than the LS algorithm for spheres. Interestingly, the gelcaps pack to a higher packing fraction than spheres under the same conditions, $\rho = 0.70$ for gelcaps versus $\rho = 0.63$ for spheres [27]. We can estimate an upper limit for the volume fraction of a pack of spherocylinders based on geometry considerations. If we assume that the gelcaps align themselves as an ordered stack of infinitely long cylinders, then a cross section will reveal a pack of disks that have the maximum theoretical area fraction of 0.906. The gelcaps occupy a volume ratio of 5/6 of their bounding cylinders, so the maximum packing fraction would be $0.906 \times 5/6 = 0.755$, which is higher than the maximum volume fraction of ordered spheres (0.7405). Recently, a similar property was observed with packs of ellipsoids (M&M candies), where MRI experiments revealed that the ellipsoids pack up to $\rho = 0.739$, depending on the aspect ratio [41].

Gelcaps are interesting in their own right, but are not very similar to more the common propellants and energetic materials. Shown in Fig. 7 is a set of 500 right circular cylinders with aspect ratio 2:1, again packed with

$\Theta_0 = 10^3$. This type of particle shape is similar to the 7-perforation grains used in modern cannon propellants [9], and the hope is that such a pack might be one day used to investigate convective burning in a packed gun propellant. Note that the edges of the cylinder are slightly rounded. The rounding is not strictly necessary but was found to increase the efficiency of the minimization drastically. Even a small amount of rounding can decrease the number of function evaluations by ten-fold, so most of the sharp edges of the particle shapes in this work have been slightly rounded.

A pack of 500 metabidiminished icosahedrons in a cubic container is shown in Fig. 8. This shape was chosen because unlike regular polyhedra, it resembles the irregular particle shapes seen in some energetic materials. Shown in Fig. 9 is a pack with icosahedra, gelcaps, and spheres, illustrating the method’s capability of packing any combination of shapes at any size ratio. There were no code changes needed to handle the shapes together in the same pack. We are unaware of any other methods by which polyhedra may be packed alongside non-polyhedral shapes of any kind, including spheres. This is an important capability when modeling energetic materials such as PBX-9501 that consist of large, polyhedral shape particles along with many smaller, spherical or cylindrical particles embedded in a rubberized binder (see Fig. 1 for a cutaway photo of the PBX-9501 microstructure).

Relevant to the same problem are the “HMX crystals” shown in Fig. 11. This model shape is currently being used in packs that resemble crystalline energetic materials and was modeled after the large HMX crystals that are being grown at the Los Alamos National Laboratory, shown in Fig. 10 [42]. The model crystal was constructed by beginning with a hexagonal prism and moving some of the vertices to emulate the structure of the real crystalline shapes. A virtual pack using the crystal model is shown in Fig. 12. The periodic pack consists of small cubes, small and large icosahedrons, and small and large HMX model polyhedra at a packing fraction of $\rho = 0.710$. An identical pack of spheres with the same size ratios, number of particles, and scaled temperature reached a packing fraction $\rho = 0.709$.

4.2 Statistics

For more detailed quantification of the properties of these packs, we turn to multi-point probability distributions. We show here only a few statistics with the simple justification that we cannot discuss the results with respect to experimental results. XCT scans of various particle packs are generally

not available, and although packs of spheres generated by *Rocpack* have been validated against published experimental data [27], the same is not possible here.

If the microstructures of interest are ergodic, isotropic, and homogeneous, we can use simplified one-, two-, and three-point probability functions to describe the microstructure. A complete discussion of these probability functions, their basis, and the reduction to their simple forms when the above requirements are met is beyond the scope of this work (see [43] for a more detailed discussion). We describe the functions only briefly here for completeness.

For the purposes of our brief description, we will assume that the materials of interest have only two phases, are ergodic, are isotropic, and are homogeneous. If we sample the microstructure (our packed bed of spheres or other shapes) at random locations $\mathbf{x}_q$, we can compute the one-, two-, and three-point probability functions S_i , S_{ij} , and S_{ijk} . $S_i(\mathbf{x}_q)$ is the probability that $\mathbf{x}_q$ lies in the i^{th} phase, $S_{ij}(\mathbf{x}_q, \mathbf{x}_r)$ is the probability that $\mathbf{x}_q$ lies in the i^{th} phase while $\mathbf{x}_r$ simultaneously lies in the j^{th} phase, and $S_{ijk}(\mathbf{x}_q, \mathbf{x}_r, \mathbf{x}_s)$ is defined similarly for three points. Fortunately, when the material meets the requirements listed above, these probability functions can be simplified significantly to

$$S_i(\mathbf{x}_q) = \rho_i \quad (15)$$

$$S_{ij}(\mathbf{x}_q, \mathbf{x}_r) = S_{ij}(\|\mathbf{x}_r - \mathbf{x}_q\|) \quad (16)$$

$$S_{ijk}(\mathbf{x}_q, \mathbf{x}_r, \mathbf{x}_s) = S_{ijk}(\|\mathbf{x}_r - \mathbf{x}_q\|, \|\mathbf{x}_r - \mathbf{x}_s\|) \quad (17)$$

Note that we can take the limits of these functions for both small and large $\|\mathbf{x}_r - \mathbf{x}_s\|$ to obtain bulk properties

$$S_{ij}(0) = \delta_{ij}\rho_i \quad (\text{no sum}) \quad (18)$$

$$S_{ij}(\infty) = \rho_i\rho_j \quad (19)$$

with similar consistency checks available for S_{ijk} (see Ref. [44] for general random two-phase media). In the following discussion, we use the subscript 0 to denote sample locations inside the matrix material (or voids) and the subscript 1 to denote sample locations inside the particles. For example, $S_{110}(\mathbf{x}_q, \mathbf{x}_r, \mathbf{x}_s)$ is the probability of simultaneously finding a point $\mathbf{x}_q$ inside a particle, $\mathbf{x}_r$ inside a particle, and $\mathbf{x}_s$ inside the matrix.

We first consider the results of the gelcap pack of Fig. 6. Recall that this pack jammed at a packing fraction of $\rho = 0.70$, which is 10% higher

than the jamming density of spheres under similar conditions. The second order statistics S_{ij} are compared to S_{ij} for an equivalent pack of spheres in Fig. 13. The dashed lines correspond to the gelcap pack while the solid lines correspond to S_{ij} of a pack of spheres that was packed under the same conditions. The characteristic lengths for the two particle shapes are an overall length of 0.162 for the gelcaps and a radius 0.106 for the spheres. The gelcaps have an aspect ratio of 2:1, so their radii are 0.081. The final packing fraction differences can be clearly seen in the figure because they are equal to the intercept of the S_{11} functions. Note also that the first peaks for the gelcaps are moved closer to $\|\mathbf{x}_j - \mathbf{x}_i\| = 0$ than the peaks of the spheres. This is an expected result, indicating the presence of touching neighbors at closer distances than in the sphere pack and is due to the fact that the gelcap has a smaller radius than a sphere of equivalent volume.

The second order statistical metrics S_{ij} for the pack of Fig. 12 are shown in Fig. 14 along with the difference relative to S_{ij} for a very similar pack of spheres. The pack of spheres contains the same number of particles and the same size ratios as the crystal pack. However, although the spheres are jammed at $\rho = 0.71$, the crystal pack is not jammed at that density. Also, because spheres have a larger volume to length ratio, the longest dimension of the largest polyhedra is greater than the diameter of the large spheres: ≈ 0.79 for the polyhedra versus ≈ 0.57 for the spheres. Nevertheless, the S_{ij} functions for the two packs are very similar (within 1 %).

The statistical differences in the two packs are more apparent in the higher order statistical functions. The three-point probability function S_{011} for the same pack is shown in Fig. 15. The distances between the three randomly thrown points x_i , x_j , and x_k are indicated on the axes by $r_1 = \|\mathbf{x}_j - \mathbf{x}_i\|$ and $r_2 = \|\mathbf{x}_k - \mathbf{x}_i\|$. $\theta = 0^\circ$ is the angle between the two vectors $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_k - \mathbf{x}_i$. The large probability along the diagonal (red is high probability) is expected because when $r_1 = r_2$, $S_{011}(r_1, r_2, \theta) = S_{01}(r)$. There is a small, but noticeable difference in the width of the central band; it is wider for spheres. But even with this third-order statistic, the packs seem similar. More differences can be seen in Figs. 16, where $\theta = 72^\circ$ and 17, where $\theta = 180^\circ$, and the contours are moderately different. Despite what seem to be small differences in the packs, there are successful ways to compute, for example, statistically optimal unit cells based on small, but quantifiable differences in these probability functions [43].

5 Conclusions

In this article we have introduced a new method for the creation of virtual packs of arbitrary shapes of particles. The method uses a level set (or isosurface) representation of the shape, which has the advantages of being able to accommodate almost any shape or any combination of shapes in the same framework. We are unaware of any other use of level sets or other type of isosurface in the packing literature. The level set representation also allows us to efficiently predict the next collision between any two particles because we can compute a smooth, and well-behaved distance function $d_{ij}(t)$ for the signed minimum distance between two particles. We also showed why standard minimization algorithms cannot be used with this method and subsequently introduced a new minimization procedure that takes advantage of the specific properties of intersected level sets.

We demonstrated several packs with various shapes that were created using the new method. Some of the shapes of interest to us are gelcaps, cylinders, and irregular polyhedra. We demonstrated the capability of packing these shapes together, in the same container, with no change to the method. In particular, we demonstrated packs that visually resemble HMX crystals and that also have similar packing fractions. We also found that gelcaps, much like ellipsoids, jam at significantly higher packing fractions than spheres.

To reiterate, the primary advantages of this method are that it can pack any combination of convex shapes together and that – thanks to the LS framework – high packing fractions can be achieved. We are unaware of any other available commercial or research tools that have this capability.

Unfortunately, we were unable to validate the results of the packing algorithm by comparing to real materials – except by visual comparisons. The only way to truly quantify how closely our virtual packs resemble the intended microstructures would be to compare them to microstructure data from real materials. For example, second and third order statistics could be computed for tomography experiments from XCT or MRI and quantitatively compared to statistics from our packs. At this time, XCT data is difficult to obtain, but XCT experiments at the Center for Simulation of Advanced Rockets (CSAR) at the University of Illinois are being conducted in conjunction with the Beckman Institute with the goal of quantifying the statistics of both real and simulated energetic materials. Current experiments are focused on gathering information about the packing characteristics of bidisperse glass

spheres.

It is of course most helpful to have information on the real materials of interest, and so later experiments are planned for conducting scans of industrial propellant formulations. These experiments should be carefully designed so that their usefulness is maximized with respect to validation of the packing algorithm. It would be tempting to attempt the largest scan possible so that a fully-representative density plot can be demonstrated. However, to be useful for validation of the packing algorithm, many XCT scans of a single type of material should be obtained so that the variation between samples can be quantified. Also, since the level of polydispersity in real materials is very large, a single scan that attempts to capture many large particles is inappropriate because it will not resolve the details of the small particles. Rather, it would be better to capture many smaller samples that may include only a few of the largest particles (as in Fig. 1) so that smaller particles can be adequately resolved by the XCT equipment. Ergodicity can then be confirmed by comparing the high order statistics of these real samples with the same metrics from virtual packs. This method is also more appropriate for obtaining many small scans that can be directly compared to the statistically optimal unit cell construction algorithms [16].

However, even lacking explicit validation, many computational studies of interest in this field can be carried out without fully understanding how accurately the virtual packs reflect reality. For example, numerical combustion studies have indicated that packs of spheroids produce approximately the same burn rates as packs of spheres in AP/HTPB composite propellants [12]. The same null hypothesis may be true for other phenomena, and such studies could be used to give us further confidence in results obtained with packs of spheres. Furthermore, the development of methods that characterize or estimate the thermal and mechanical properties of real materials can carry on independently of whether or not experimental data sets exist because we can now create virtual packs to test these methods. For example, theories for obtaining bounds on heterogeneous solid material properties such as bulk moduli can be developed by testing done on virtual packs [14], now including packs of non-spherical shapes.

Also, even though bulk burn rate is apparently independent of the particle shape, it is still unclear whether or not this is true for other properties such as the acoustics in the rocket grain. Further studies can answer this question. Furthermore, since the method presented in this work uses level sets, there is a unique ability to smoothly morph any given shape into an equivalent

spherical shape. Using this method, we can not only answer the question of whether the particle shapes influence a given property, but we can also predict at what point the sphericity becomes an important factor. Using this technique, we plan to re-visit the problem of burning spheroidal AP particles in HTPB binder [12], but instead use more realistic shapes such as polyhedra for the burn computations. Since the shapes of some materials are obviously more spherical than other shapes, this type of study is applicable to other areas as well.

One such area is in prediction of pressure rise times in confined packs of loose propellant particles. These predictions are useful when designing solid rocket igniters and gun cartridges [6] [7]. Numerical tools for simulating burn rates in porous materials are not yet widely available (see Ref. [45] for an example where the propellant has simulated cracks), but it is now possible to begin looking at porous burning of packs of realistic shapes of particles like those of Fig. 7.

Yet another direction for future research is the broad area of pack characterization. Packs of spheres have been well-characterized in terms of convergence of the first and second order statistics (volume fraction and $g(r)$, respectively) with respect to packing algorithm variables such as initial scaled temperature Θ_0 , number of particles N , and the distance to jamming δ [27] [28]. These convergence properties are also of interest for packs of arbitrary shapes, but have only been studied for simple shapes such as ellipsoids. Future work with packs created using the algorithms presented here could quantify these properties for packs of monodisperse and polydisperse polyhedra, spherocylinders, etc. present either individually or together in the same pack.

This is by no means an exhaustive list of possible topics for future research. Indeed, our discussion has been almost entirely limited to materials that are used as propellants or energetic materials. Many more opportunities for porous or heterogeneous modeling exist in chemical engineering, materials science, and medicine. It is our hope that the tool presented here will enable researchers to tackle the sorts of computational problems that have been previously impractical because of the difficulty in modeling the complex morphology.

6 Acknowledgements

This work was supported by the US Department of Energy through the University of California under subcontract number B523819. Dr. Jackson was also partially supported by the Air Force Research Laboratory under contract FA9550-06-C-0078, program manager Dr. A. Nachman.

7 Algorithms

Algorithm 7.1 A “black box” that predicts the time for the next collision between particles i and j .

```

PREDICT-COLLISION( $i, j$ )
  ▷ Operate on copies of  $i$  and  $j$ .
   $a \leftarrow i$ 
   $b \leftarrow j$ 
   $t \leftarrow 0$ 
  repeat
     $t \leftarrow t + \Delta t$ 
     $a$ .UPDATE( $t$ )
     $b$ .UPDATE( $t$ )
     $d_{ij} \leftarrow \text{COMPUTE-DISTANCE}(a, b)$ 
  until  $d_{ij} > 0$ 
  return  $t$ 

```

Algorithm 7.2 Details of the collision prediction scheme of our implementation. A piecewise quadratic fit is first formed by advancing copies of the particles forward in time. If a zero-crossing is located, Brent’s method is used to refine the estimate for the root.

```

PREDICT-COLLISION( $i, j, t_{max}$ )
   $t_0 \leftarrow 0$ 
   $d_0 \leftarrow \text{COMPUTE-DISTANCE}(i, j, t_0)$ 
  repeat
     $t_1 \leftarrow t_0 + \Delta t/2$ 
     $t_2 \leftarrow t_0 + \Delta t$ 
     $d_1 \leftarrow \text{COMPUTE-DISTANCE}(i, j, t_1)$ 
     $d_2 \leftarrow \text{COMPUTE-DISTANCE}(i, j, t_2)$ 
    Form piecewise quadratic fit  $d_{ij}^*$  on interval  $[t_0, t_2]$ 
     $error \leftarrow \int_t^{t+\Delta t} d_{ij}^*(\theta) d\theta > -\epsilon$ 
    if  $error > target/2$ 
      then  $\Delta t \leftarrow \Delta t/2$ 
    elseif  $error < 2target$ 
      then  $\Delta t \leftarrow 2\Delta t$ 
       $t_0 \leftarrow t_2$ 
       $d_0 \leftarrow d_2$ 
    ▷ Check for bracketed root.
    if  $d_2 < 0$ 
      then return BRENT-FIND-ROOT( $i, j, t_0, t_2$ )
  until  $t_2 > t_{max}$ 
return  $\infty$ 

```

Algorithm 7.3 Procedure for computing the instantaneous distance between two particles i and j . Copies of the particles are updated to the desired time t by moving them without collisions along their current trajectories. Then, the $\text{GET-MINIMUM}(i, j, \mathbf{x}_0)$ is used to find the distance.

COMPUTE-DISTANCE(i, j, t)

▷ Operate on copies of the shapes
 $a \leftarrow i$
 $b \leftarrow j$
 $a.\text{UPDATE}(t)$
 $b.\text{UPDATE}(t)$
 ▷ Form initial guess for location of minimum.
 $\mathbf{x}_0 \leftarrow (a.\mathbf{x} + b.\mathbf{x})/2$
 $d_{ij} = 2 \cdot \text{GET-MINIMUM}(i, j, \mathbf{x}_0)$
return d_{ij}

Algorithm 7.4 Procedure for locating the minimum of the intersection of two level set fields that represent two convex objects. The procedure first attempts to locate a surface where the gradient is undefined (a ravine) and then performs a line minimization constrained to that surface.

```

GET-MINIMUM( $i, j, \mathbf{x}$ )
   $f \leftarrow \phi_{ij}(\mathbf{x})$ 
   $\mathbf{n} \leftarrow -\phi_{ij}(\mathbf{x}) / \|\phi_{ij}(\mathbf{x})\|$ 
   $count \leftarrow 0$ 
   $\mathbf{x}_{accel} \leftarrow \mathbf{x}$ 
  repeat
     $\mathbf{x}_{old} \leftarrow \mathbf{x}$ 
    MOVE-DOWNHILL-INTO-RAVINE( $\mathbf{x}, \mathbf{n}$ )
     $\mathbf{q} \leftarrow \text{GET-RAVINE-DIRECTION}(\mathbf{x})$ 
     $f \leftarrow \text{LINE-SEARCH-FOR-MIN}(\mathbf{x}, \mathbf{q})$ 
     $count \leftarrow count + 1$ 
    if  $count > accelCount$ 
      then ACCELERATE-IN-DIRECTION( $\mathbf{x} - \mathbf{x}_{accel}$ )
            $count \leftarrow 0$ 
            $\mathbf{x}_{accel} \leftarrow \mathbf{x}$ 
  until  $\|\mathbf{x} - \mathbf{x}_{old}\| < tolerance$ 
  return  $f$ 

```

8 Figures

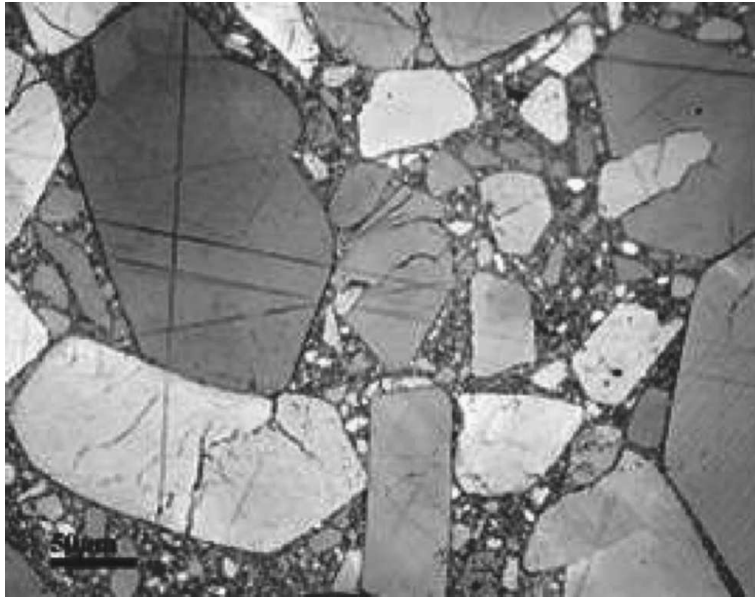


Figure 1: A polarized light micrograph of a plastic bonded explosive (PBX) based on the energetic material HMX. The size scale is 50 μm . (Photo courtesy of C. Skidmore at LANL [42].)

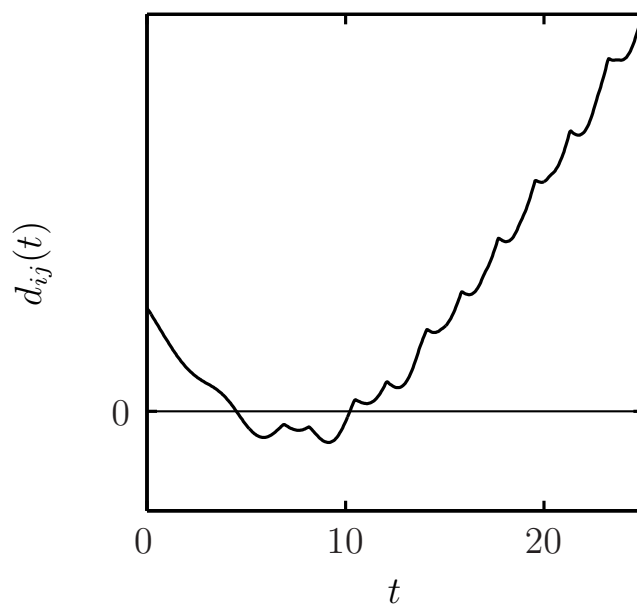


Figure 2: Distance between two gelcaps (spherocylinders) as a function of time. The distance function $d_{ij}(t)$ resembles a quadratic function with superimposed oscillations. The quadratic-like portion (which is actually quartic) is due to the relative movement of the centroids, while the oscillations are due to the rotations as the non-spherical shapes tumble through space.

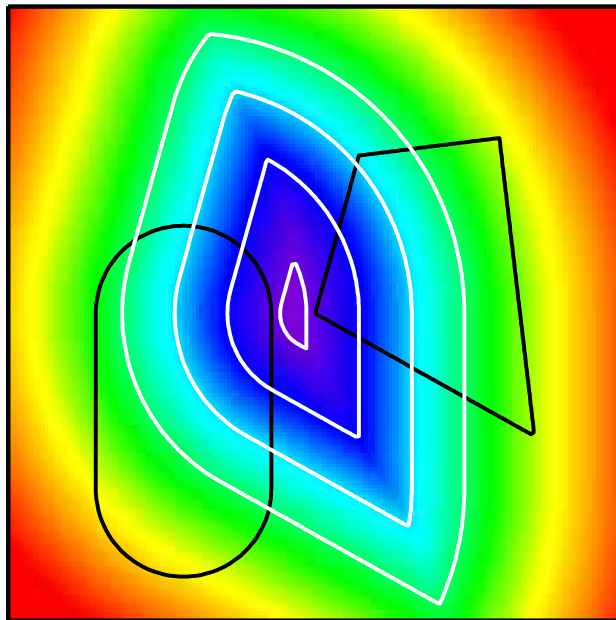


Figure 3: Intersection field $\phi_{ij}(\mathbf{x}, t)$ of the signed minimum distance fields of two convex shapes. The outlines of the 2D shapes (the zero contours of $\phi_i(\mathbf{x}, t)$) are shown in black. The color field and white contours show the intersected field. When the shapes are both convex, a single, global minimum exists. However, note the complexity of the intersected field, particularly the presence of multiple ravines where the gradient jumps. The global minimum can be seen to lie in one of these ravines. In 3D, the ravines correspond to surfaces where the gradient does not exist.

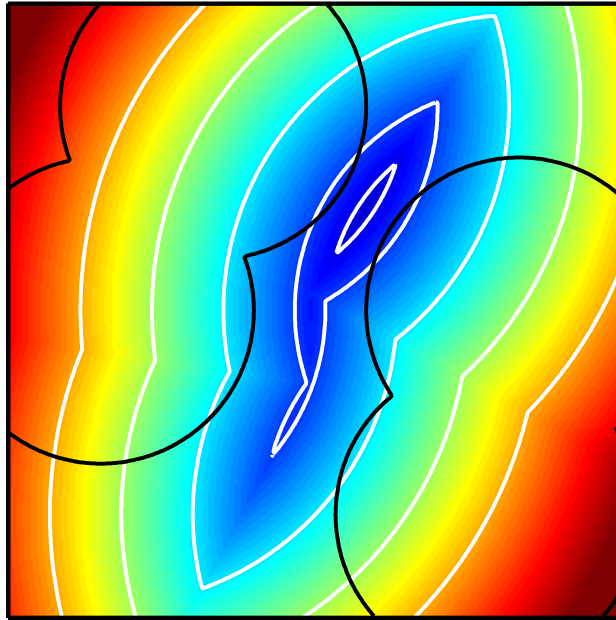


Figure 4: Intersection field $\phi_{ij}(\mathbf{x}, t)$ of the signed minimum distance fields of two non-convex shapes. If the shapes are not convex, the intersected field can contain multiple local minima, which reduces the chances that the minimization routine will converge to the correct global minimum.

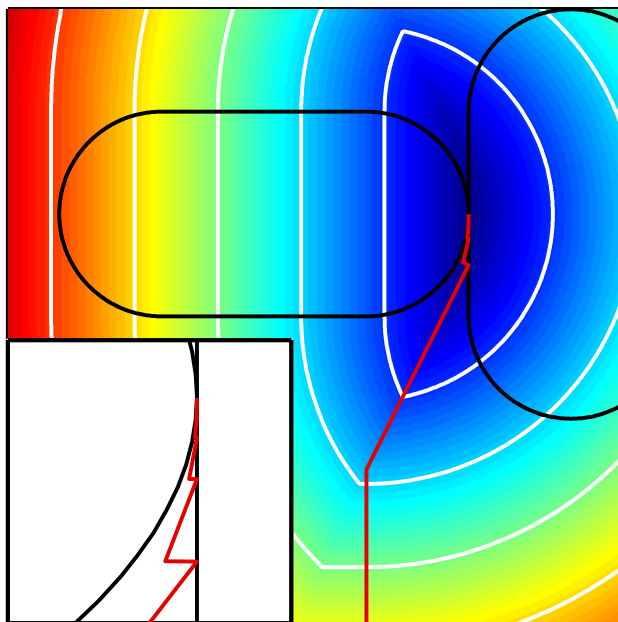


Figure 5: Progress of the minimum search algorithm in 2D. The outlines of the two gelcaps (spherocylinders) are shown in black. The color field and white contour lines show the intersected field $\phi_{ij}(\mathbf{x}, t)$. Shown in red is the path of the minimizer. The inset shows the minimizer's path near the global min, zoomed in and expanded in the x -direction to show more detail.

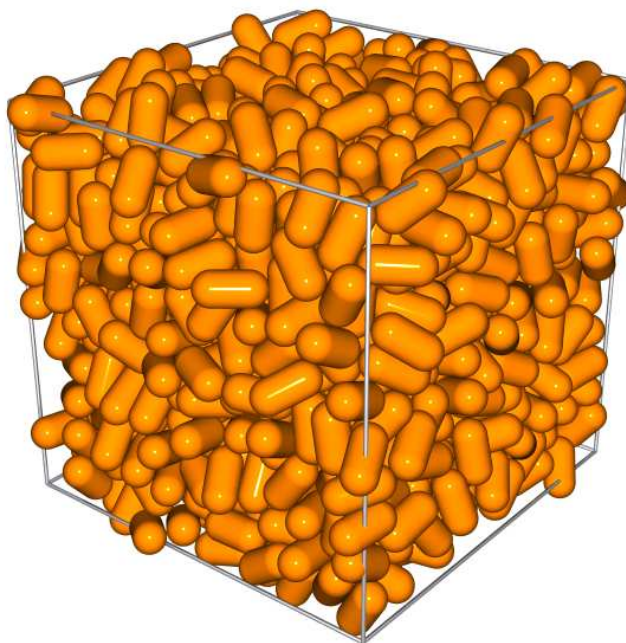


Figure 6: A pack of 1000 gelcaps (spherocylinders) in a periodic cube, packed with $\Theta_0 = 10^3$. Interestingly, the gelcaps pack to a final packing fraction $\rho = 0.70$. A pack of spheres under identical conditions jams at only $\rho = 0.63$

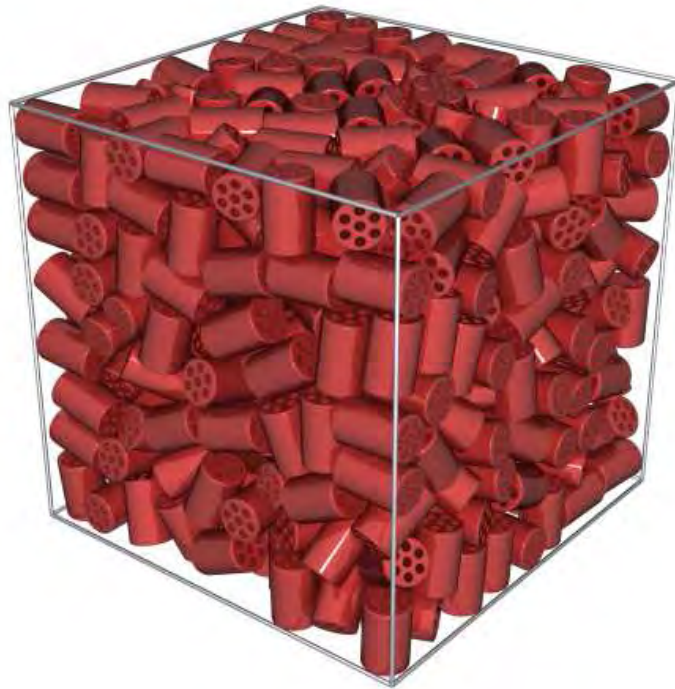


Figure 7: A pack of 500 right circular cylinders rendered as 7-perforation gun propellant grains in a cubic container. The packing fraction of the cylinders is $\rho = 0.485$ (not accounting for the void fraction of the perforations).

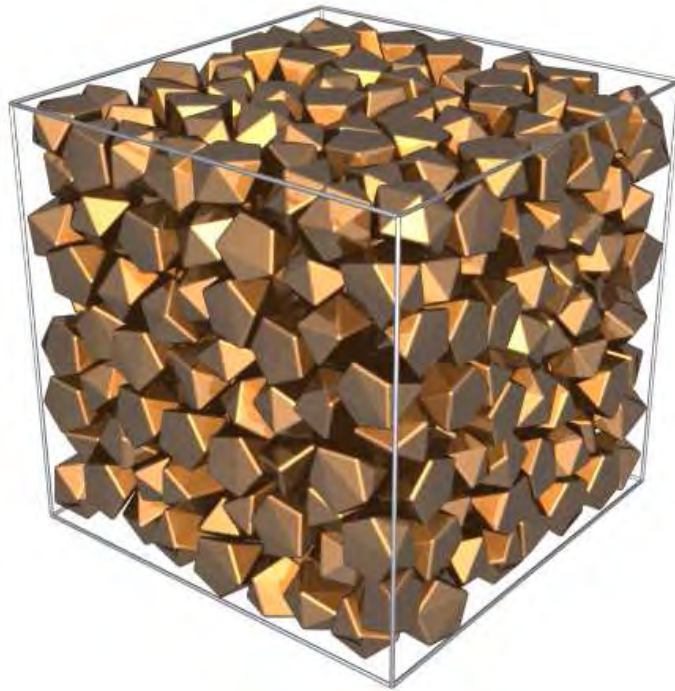


Figure 8: A pack of 500 metabidiminished icosahedrons in a cubic container with packing fraction $\rho = 0.626$. Complex polyhedra such as these are desirable for modeling random packs of crystalline particles.

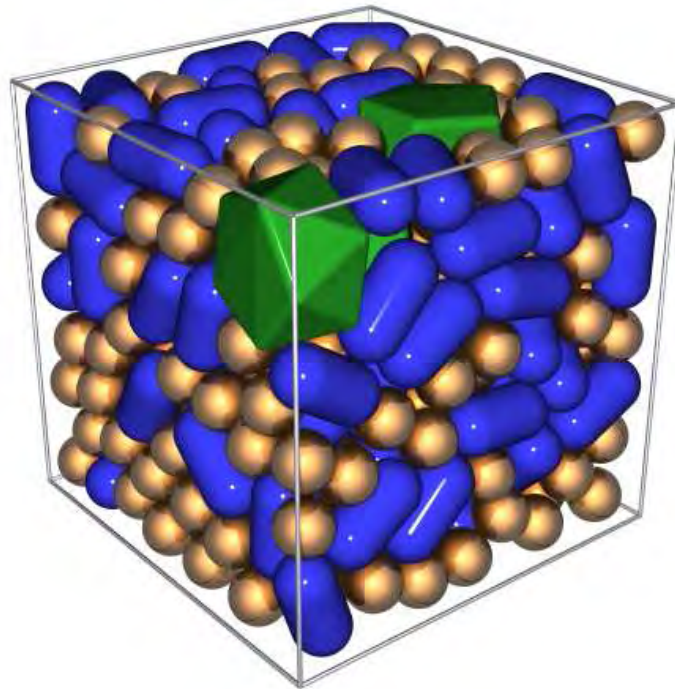


Figure 9: A pack of spheres, gelcaps (spherocylinders), and metabidiminished icosahedrons, illustrating the capability of packing any kinds of convex shapes in any combination. All of these shapes use the same algorithm in the code; their only differences in the code are in their respective level set field values.



Figure 10: Large HMX crystals grown at Los Alamos National Lab. These are single crystals photographed on a 1 cm grid. (Photo courtesy of D. Hooks at LANL [42].)

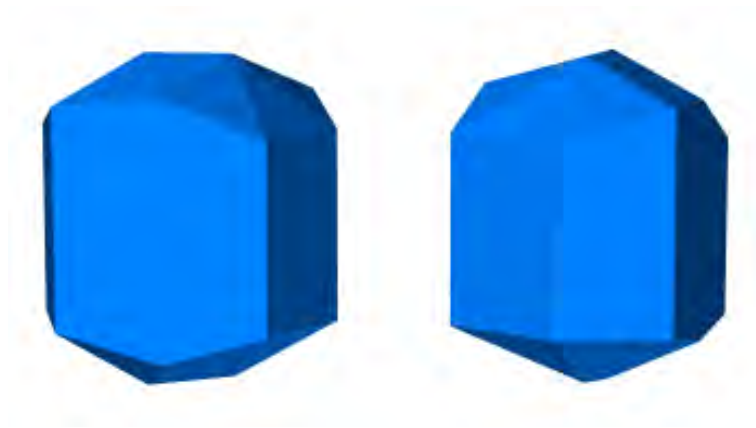


Figure 11: A model of an HMX crystal used in the packing. The shape was modeled from the photos of the large HMX crystals grown at Los Alamos National Lab [42]

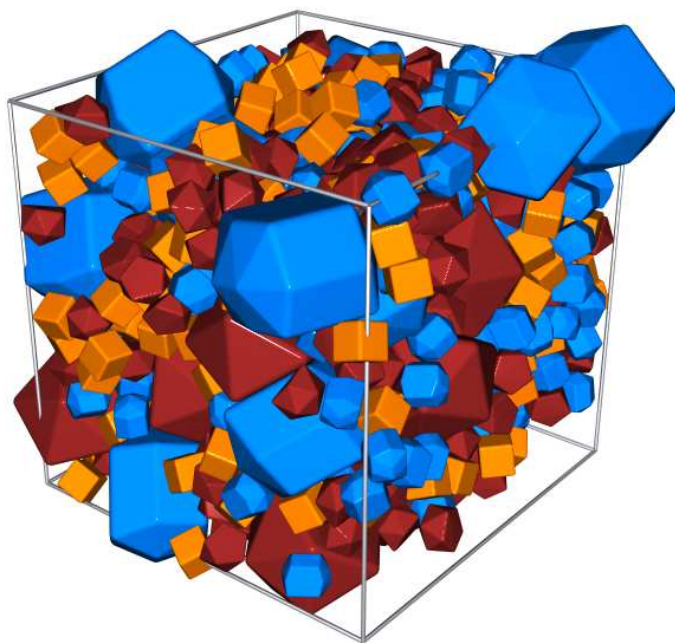


Figure 12: A virtual model of a periodic pack of crystalline HMX. The pack has 200 each of icosahedrons, cubes, and “HMX crystals” of relative size 1 with 10 more icosahedrons and 20 more HMX crystals of relative size 3. The pack reached a packing fraction $\rho = 0.71$ with $\Theta_0 = 10^3$. Only particles whose centroids lie in the periodic domain (shown by the bounding box) are pictured.

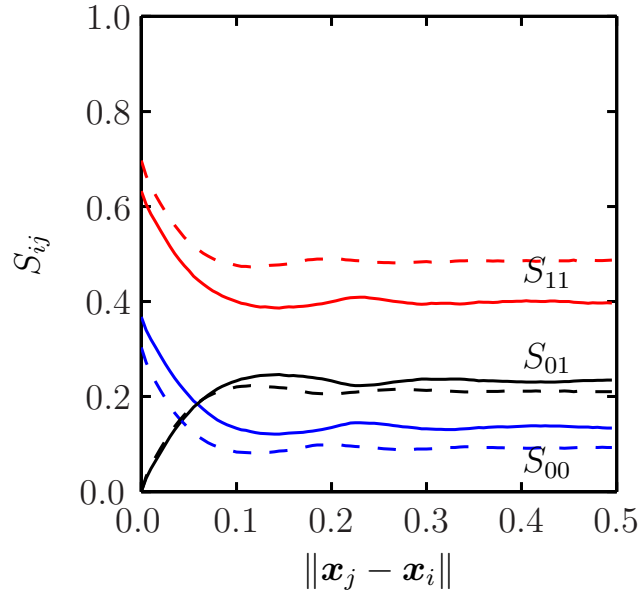


Figure 13: Second order statistics S_{ij} for the gelcap pack of Fig. 6, shown in dashed lines. The solid lines correspond to S_{ij} of a pack of spheres that was jammed under the same conditions. The gelcap length is 0.162 (aspect ratio 2:1) and the sphere radius is 0.106. S_{11} for the gelcaps begins at a higher probability than the spheres because of the higher packing fraction (0.70 for gelcaps versus 0.63 for spheres). Note also that the first peaks for the gelcaps are moved closer to $\|\mathbf{x}_j - \mathbf{x}_i\| = 0$ than the peaks of the spheres, indicating that the gelcap is in contact with closer neighbors. This is the expected result, since the gelcap has a smaller radius than a sphere of equivalent volume.

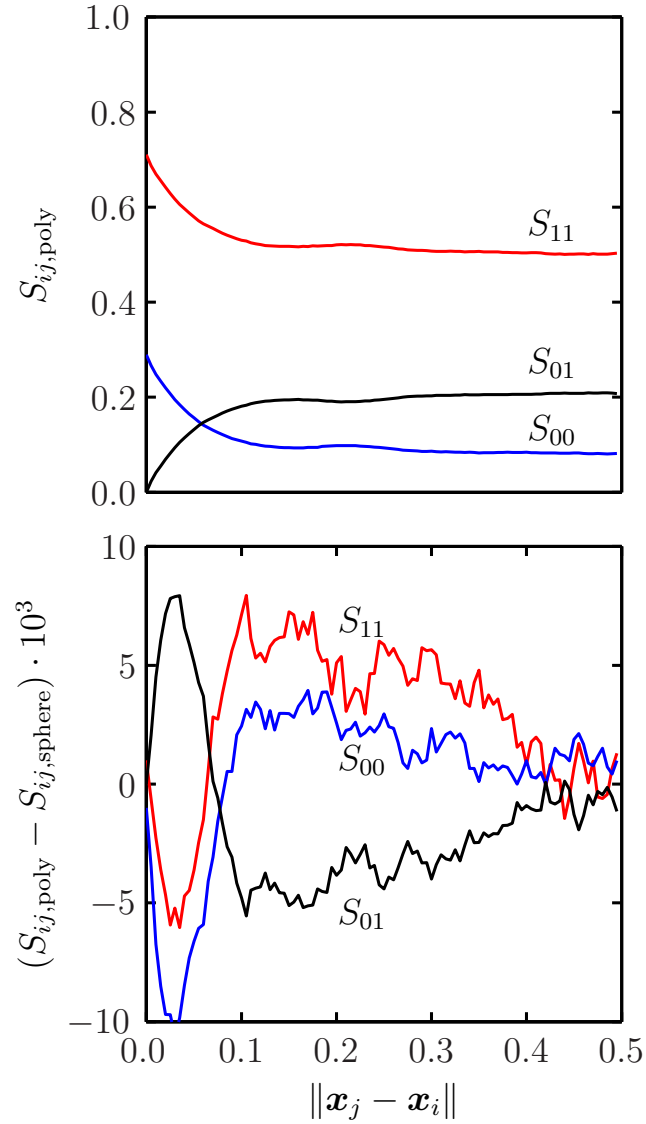


Figure 14: (top) Second order statistics S_{ij} for the HMX model pack of Fig. 12. (bottom) Difference of the statistics relative to S_{ij} of a pack of spheres that was jammed under the same conditions. Note that although the spheres are jammed at $\rho = 0.71$, the crystal pack is not jammed. The longest dimension of the largest polyhedra is ≈ 0.79 ; the diameter of the largest spheres is ≈ 0.57 .

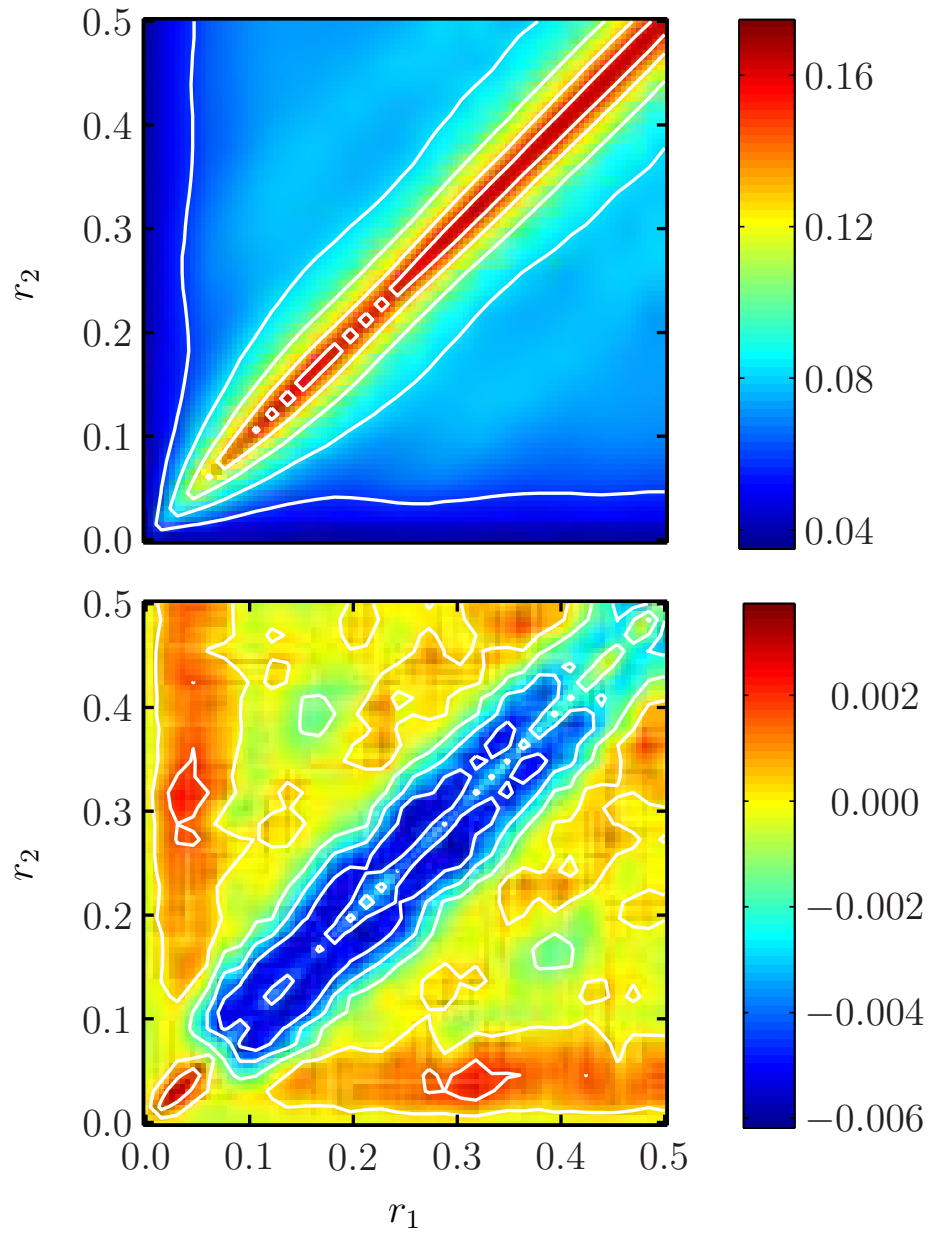


Figure 15: (top) Three-point probability function S_{011} at $\theta = 0^\circ$ for the pack in Fig. 12. (bottom) The difference of S_{011} relative to an equivalent pack of spheres ($S_{011,\text{poly}} - S_{011,\text{sphere}}$). Here, $r_1 = \|\mathbf{x}_j - \mathbf{x}_i\|$, $r_2 = \|\mathbf{x}_k - \mathbf{x}_i\|$, and θ is the angle between the two vectors $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_k - \mathbf{x}_i$. Red corresponds to high probability, while blue corresponds to low probability.

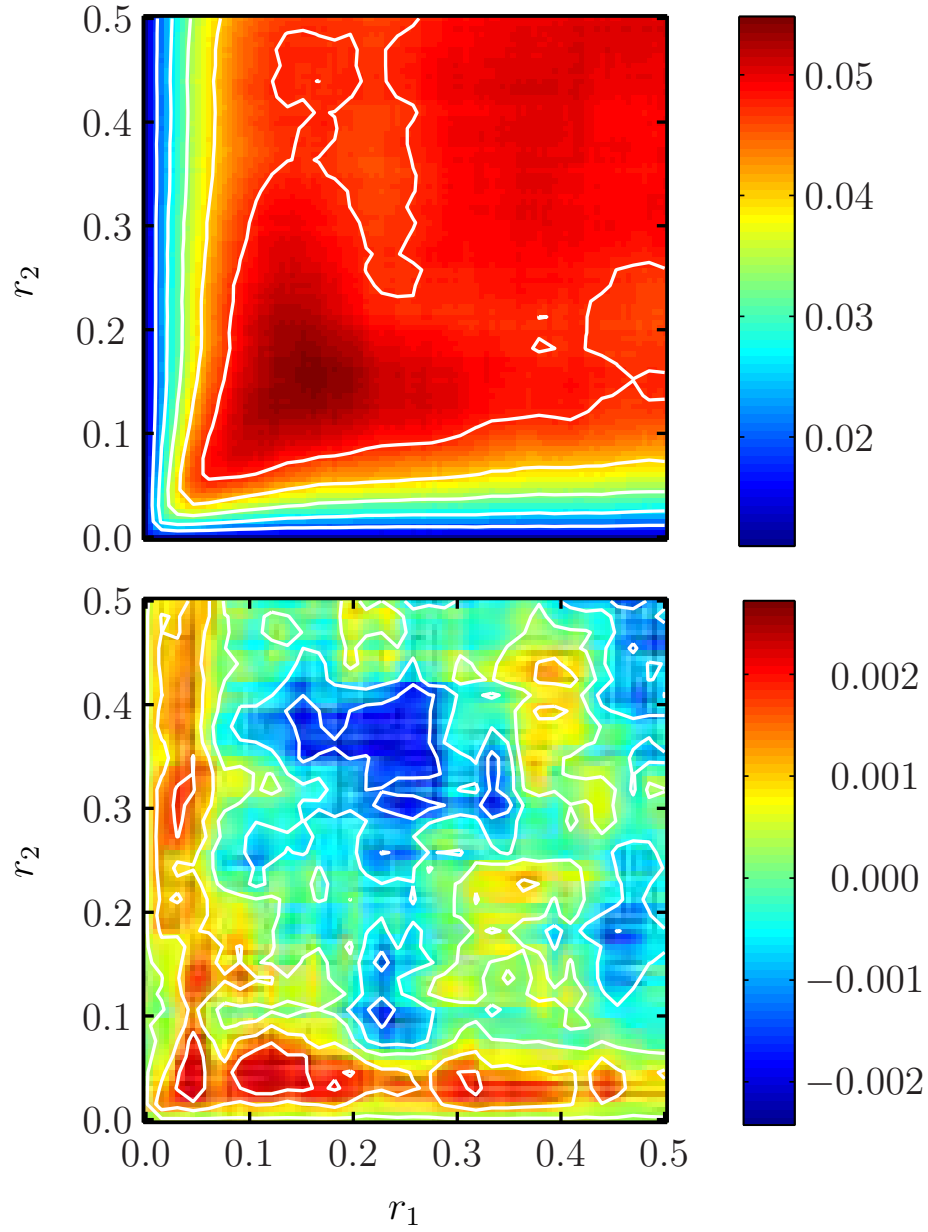


Figure 16: (top) Three-point probability function S_{011} at $\theta = 72^\circ$ for the pack in Fig. 12. (bottom) The difference of S_{011} relative to an equivalent pack of spheres ($S_{011,\text{poly}} - S_{011,\text{sphere}}$).

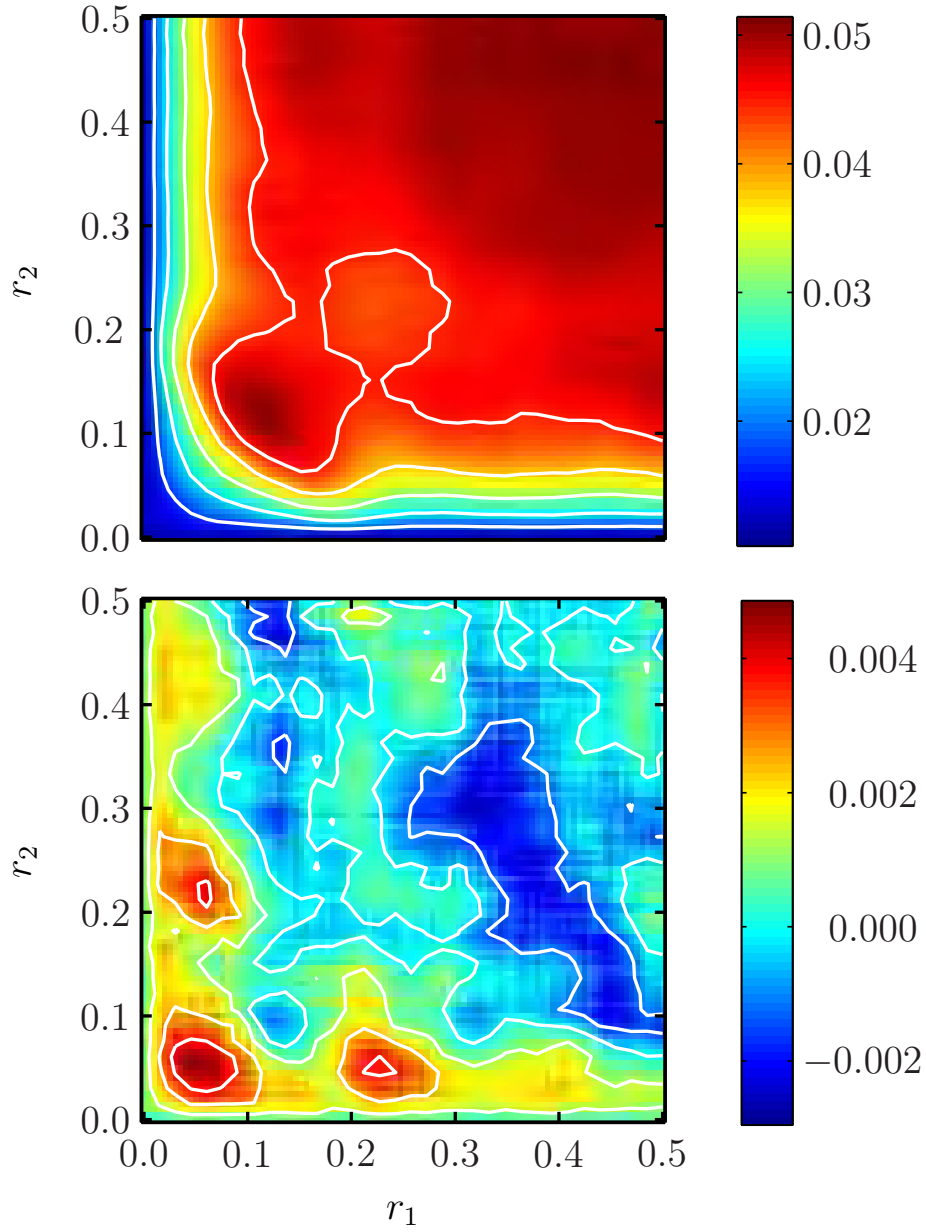


Figure 17: (top) Three-point probability function S_{011} at $\theta = 180^\circ$ for the pack in Fig. 12. (bottom) The difference of S_{011} relative to an equivalent pack of spheres ($S_{011,\text{poly}} - S_{011,\text{sphere}}$).

References

- [1] M Baer, C Hall, R Gustavsen, and D Hooks. Isentropic loading experiments of a plastic bonded explosive and constituents. *Journal of Applied Physics*, Jan 2007.
- [2] R Armstrong and W Elban. Materials science and technology aspects of energetic (explosive) materials. *Materials Science and Technology*, Jan 2006.
- [3] T Davis. *The Chemistry of Powder and Explosives*. Chapman and Hall, 1941.
- [4] D Adams. Igniter performance in solid-propellant rocket motors. *Journal of Spacecraft and Rockets*, Jan 1967.
- [5] T Davis and K Kuo. Experimental study of the combustion processes in granular propellant beds. *Journal of Spacecraft and Rockets*, Jan 1979.
- [6] D Greatrix, J Gottlieb, and T Constantinou. Numerical model for pellet-dispersion igniter systems. *Journal of Propulsion and Power*, Jan 1988.
- [7] K Kuo, R Vichnevetsky, and M Summerfield. Theory of flame front propagation in porous propellant charges under confinement. *stinet.dtic.mil*, Jan 1971.
- [8] B Asay, S Son, and J Bdzil. The role of gas permeation in convective burning. *International Journal of Multiphase Flow*, Jan 1996.
- [9] D Carlucci and S Jacobson. *Ballistics: Theory and Design of Guns and Ammunition*. CRC Press, 2007.
- [10] S Kochevets, J Buckmaster, T Jackson, and A Hegab. Random packs and their use in modeling heterogeneous solid propellant combustion. *Journal of Propulsion and Power*, Jan 2001.
- [11] G Knott, T Jackson, and J Buckmaster. Random packing of heterogeneous propellants. *AIAA Journal*, Jan 2001.
- [12] X Wang, J Buckmaster, and T Jackson. Burning of ammonium-perchlorate ellipses and spheroids in fuel binder. *Journal of Propulsion and Power*, Jan 2006.

- [13] T Jackson, F Najjar, and J Buckmaster. New aluminum agglomeration models and their use in solid-propellant-rocket simulations. *Journal of Propulsion and Power*, Jan 2005.
- [14] R Lipton. Variational methods, bounds and size effects for two-phase composites with coupled heat and mass transport processes at the two-phase interface. *Journal of the Mechanics and Physics of Solids*, Jan 1999.
- [15] R Hill. Elastic properties of reinforced solids: some theoretical principles. *J. Mech. Phys. Solids*, Jan 1963.
- [16] N Kumar, K Matouš, and P Geubelle. Reconstruction of periodic unit cells of multimodal random particulate composites using genetic algorithms. *Computational Materials Science*, Jan 2008.
- [17] I Macdonald, M El-Sayed, K Mow, and F Dullien. Flow through porous media-the Ergun equation revisited. *Industrial & Engineering Chemistry Fundamentals*, Jan 1979.
- [18] M Keyser, M Conradie, M Coertzen, and J Van Dyk. Effect of coal particle size distribution on packed bed pressure drop and gas flow distribution. *Fuel*, Jan 2006.
- [19] J Shepherd and D Begeal. Transient compressible flow in porous materials. *NASA STI/Recon Technical Report N*, 88:22324, January 1988.
- [20] J Bernal and J Mason. Packing of spheres: Co-ordination of randomly packed spheres. *Nature*, Jan 1960.
- [21] T Gruhn and P Monson. Isobaric molecular dynamics simulations of hard sphere systems. *Physical Review E*, Jan 2001.
- [22] I Volkov, M Cieplak, J Koplik, and J Banavar. Molecular dynamics simulations of crystallization of hard spheres. *Phys Rev E*, 66:061401, Jan 2002.
- [23] M Alam and S Luding. Rheology of bidisperse granular mixtures via event-driven simulations. *Journal of Fluid Mechanics*, Jan 2003.

- [24] R Johansson, H Thunman, and B Leckner. Influence of intraparticle gradients in modeling of fixed bed combustion. *Combustion and Flame*, Jan 2007.
- [25] L Massa, T Jackson, and M Short. Numerical solution of three-dimensional heterogeneous solid propellants. *Combustion Theory and Modelling*, Jan 2003.
- [26] M Wackenhut, S McNamara, and H Herrmann. Shearing behavior of polydisperse media. *The European Physical Journal E-Soft Matter*, Jan 2005.
- [27] F Maggi, S Stafford, T Jackson, and J Buckmaster. Nature of packs used in propellant modeling. *Physical Review E*, Jan 2008.
- [28] S Stafford. *Random packs and their use in modeling high speed porous flow*. Ph.D. thesis, University of Illinois at Urbana-Champaign, 2008.
- [29] A Donev, S Torquato, and F Stillinger. Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles. II. Applications to ellipses and ellipsoids. *Journal of Computational Physics*, Jan 2005.
- [30] A Donev, J Burton, F Stillinger, and S Torquato. Tetratic order in the phase behavior of a hard-rectangle system. *Physical Review B*, Jan 2006.
- [31] R Rowe, P York, E Colbourn, and S Roskilly. The influence of pellet shape, size and distribution on capsule filling—a preliminary evaluation of three-dimensional computer simulation using a Monte-Carlo technique. *International Journal of Pharmaceutics*, Jan 2005.
- [32] C Neff. Finding the distance between two circles in three-dimensional space. *IBM journal of research and development*, Jan 1990.
- [33] X Jia and R Williams. A packing algorithm for particles of arbitrary shapes. *Powder Technology*, Jan 2001.
- [34] X Jia, M Gan, R Williams, and D Rhodes. Validation of a digital packing algorithm in predicting powder packing densities. *Powder Technology*, Jan 2007.

- [35] J Reaugh. Checking out the hot spots. *Science and Technology Review*, March:3, Mar 2003.
- [36] R Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, 1973.
- [37] W Press, B Flannery, S Teukolsky, and W Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [38] J Nelder and R Mead. A simplex method for function minimization. *Computer Journal*, Jan 1965.
- [39] R Hooke and T Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8:212–229, 1961.
- [40] A Donev, S Torquato, and F Stillinger. Pair correlation function characteristics of nearly jammed disordered and ordered hard-sphere packings. *Physical Review E*, Jan 2005.
- [41] W Man, A Donev, F Stillinger, and M Sullivan. Experiments on random packings of ellipsoids. *Phys. Rev. Lett.*, Jan 2005.
- [42] D Hooks. Growth and characterization of explosive single crystals at los alamos national laboratory. *CPIAC Bulletin*, 33(2):1, Sep 2007.
- [43] B Collins. *Reconstruction of Statistically Optimal Periodic Unit Cells of Multimodal Particulate Composites*. MS thesis, University of Illinois at Urbana-Champaign, 2008.
- [44] S Torquato and G Stell. Microstructure of two-phase random media. I. The n-point probability functions. *The Journal of Chemical Physics*, Jan 1982.
- [45] X Wang, T Jackson, and L Massa. Numerical simulation of heterogeneous propellant combustion by a level set method. *Combustion Theory and Modelling*, Jan 2004.

Erosion in Solid-Propellant Rocket Motor Nozzles with Unsteady Non-uniform Inlet Conditions

Ju Zhang & Thomas L. Jackson
Computational Science and Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

John Buckmaster
Buckmaster Research
Urbana, IL 61801, USA

Fady Najjar
Lawrence Livermore National Laboratory
Livermore, CA 94550, USA

September 24, 2009

Submission to *J. of Propulsion and Power* as a full length article

Corresponding author contact information:

Ju Zhang
Computational Science and Engineering
University of Illinois at Urbana-Champaign
3243 DCL, MC-278
1304 W. Springfield Ave.
Urbana, IL 61801
Phone: 217-265-7582
Fax: 217-333-1910
Email: juzhang@illinois.edu

Erosion in Solid-Propellant Rocket Motor Nozzles with Unsteady Non-uniform Inlet Conditions

Ju Zhang¹ & Thomas L. Jackson²
 Computational Science and Engineering
 University of Illinois at Urbana-Champaign
 Urbana, IL 61801, USA

John Buckmaster
 Buckmaster Research
 Urbana, IL 61801, USA

Fady Najjar³
 Lawrence Livermore National Laboratory
 Livermore, CA 94550, USA

Abstract

A computational framework is developed to investigate nozzle erosion in solid-propellant rocket motors (SRMs). Highly resolved simulations are performed to understand the effects caused by turbulent inlet conditions, as well as nozzle vectoring, on nozzle erosion. The thermal boundary layer is captured along the nozzle walls to compute the heat flux and erosion rates. To optimize the computational resources, the nozzle configuration is studied by itself and appropriate inflow conditions are imposed. Two complementary formulations to apply these conditions are described: the first is based on extracting the flow field from the full motor configuration; while the second invokes a multiscale asymptotic analysis of turbulent flows inside a rocket motor. Results suggest that specifying turbulent, rather than uniform, inlet conditions can have a significant effect on nozzle erosion.

1 Introduction

Carbon-Carbon/Graphite nozzles are often used in solid rocket motors (SRMs) because of their ability to retain their structural integrity under extreme thermal environments, and the ease with which they can be machined to achieve a desired geometry. However, when exposed to hot oxidizing chemical species such as H_2O , OH , CO_2 , O , and O_2 , surface erosion can occur. During a long burn

¹Corresponding author. Visiting Scholar; juzhang@illinois.edu. Member AIAA

²Senior Research Scientist; tlj@illinois.edu. Associate Fellow, AIAA.

³Code Physicist; najjar2@llnl.gov. Senior Member AIAA

this can be sufficient to alter the geometry (including the throat area) to an extent sufficient to alter the rocket performance. For this reason, there is a need for theoretical and numerical studies that can lead to an understanding of the major factors that dictate the erosion rate, and can lead to design improvements. Such studies present serious challenges, for not only must the chamber flow be well described to determine the nozzle inlet conditions, itself a challenging problem, but the nozzle flow must be well described, with full resolution of the boundary layers.

Early work goes back to the 1960s [1, 2], and accounts for the diffusive fluxes of oxidizing species to the nozzle surface. Later more comprehensive studies include work by Kuo and Keswani [3], further developed some 20 years later by Acharya and Kuo [4]; and recent work by Thakre and Yang [5]. Not surprisingly, the last two studies are rich, and justify meaningful validation tests. But also, not surprisingly, they can be improved upon. Most significantly in the context of our work, they do not require a description of the chamber flow because the nozzle inlet conditions are assumed to be steady and uniform, only a rough approximation for turbulent flow or for flow generated by the combustion of heterogeneous propellants.

This assumption is part of current industry practice, and extends to the inclusion of aluminum oxide particles. Aluminum is commonly used in solid rocket propellants, and oxide particles pass into the nozzle, where their impact on erosion can be significant. The distribution of these particles across the inlet is not uniform, but is assumed to be so; and the particles are not all of the same size, but have a size distribution, one that is ignored. We, together with colleagues at the University of Illinois, have developed computational tools that allow for the determination of the detailed chamber flow, along with a detailed description of the oxide particle distribution both in size and location, and so we can use these tools to prescribe nozzle inlet data with a detail heretofore not examined [6, 7].

What we can not do is fully calculate the coupled chamber/nozzle flow whilst accounting for the nozzle boundary layer, simply because of limitations on computational resources. (Amongst other limitations we do not use an adaptive timestep strategy although the problem is one for which the temporal resolution required is high). An obvious strategy for dealing with this difficulty

is to calculate the chamber flow and the nozzle flow separately: to calculate the chamber flow, including data at the nozzle inlet, using a relatively coarse mesh; and then to use the inlet data, suitably interpolated in space and time, in a fine-mesh calculation of the nozzle flow. The chamber flow would actually be determined from a full rocket simulation including the nozzle, but with the boundary layers unresolved.

There are technical difficulties that arise in the pursuit of this strategy, and the work presented here is only a partial step towards the ultimate goal. But we shall present evidence that non-uniform and unsteady nozzle inlet data can significantly affect the nozzle flow and the erosion rate; and we shall successfully complete a variation of the strategy when the chamber flow is modeled by a turbulent flow. In this variation the chamber flow is determined from a chamber-only calculation rooted in the assumption that both the mean flow and the turbulent fluctuations evolve slowly in the axial direction. In later work we shall carry out the complete strategy for a three-dimensional turbulent flow, with inlet boundary conditions for the nozzle flow generated from pdf descriptions of the coarse-mesh results. (Examples of turbulent inlet conditions for a jet flow can be found in [8]). In doing this we shall be able to accommodate complex chamber features including grain shapes such as stars and finoculs, inhibitors, slots, and submerged regions. We anticipate that this will generate results with better fidelity than those generated using current industry practice, and will provide important insights into thermal management questions of importance to the Air Force and to rocket builders.

A simplified schematic, depicted in Figure 1, summarizes the various physical and chemical processes involved in nozzle erosion/ablation. Chemical, mechanical, and thermal effects coexist within the compressible turbulent boundary layer at the nozzle surface, presenting a challenge to accurately model the momentum and heat transfer to the nozzle walls.

The fundamental numerical tool that we have at our disposal, always used when the slowly-varying strategy is not being used (and therefore always used in the nozzle), is *Rocstar*, a code developed at the Center for Simulation of Advanced Rockets (CSAR) as part of the DOE funded ASC centers [6, 9]. *Rocstar* is a tightly coupled fluid-structure-thermal multiphysics solver. It

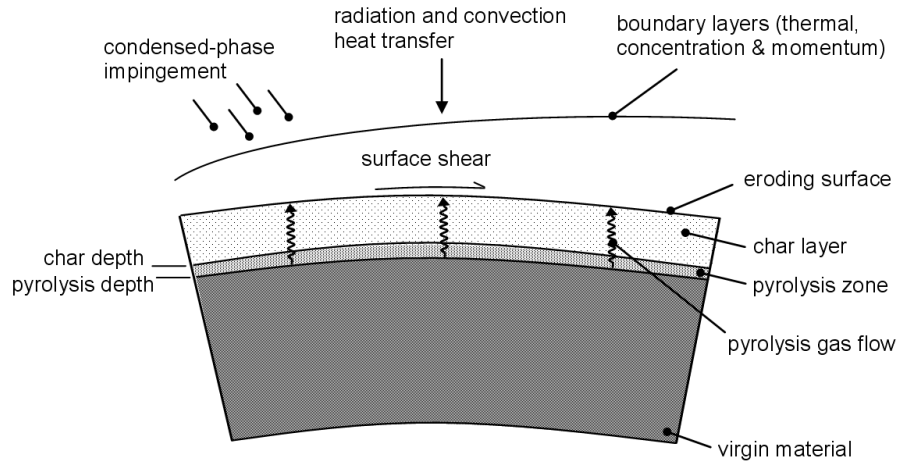


Figure 1: Schematic of various processes in nozzle ablation. *Courtesy of ATK.*

consists of several physics modules including Rocburn [10] and Rocflo [11]. Rocburn solves the unsteady heat equation in the solid phase in the wall normal direction, allows for gas-solid coupling, and has been modified to accommodate nozzle erosion. Full coupling of the flow field, the unsteady thermal field in the solid, and the retreating solid surface is not a characteristic of current practice or any published study that we know of.

The code is fully three-dimensional and this enables us to incorporate three-dimensional turbulent flows, and to solve for non-axisymmetric configurations, as, for example, when a thrust vector control nozzle is used.

We do make some simplifications. One such is the neglect of gas-phase chemical reactions, and we note that only insignificant effects are reported in [5]. Indeed, we account for no gas-phase chemical species, including those oxidizers responsible for the erosion. Instead, we use a familiar pyrolysis law in which erosion is assumed to be a function of the surface temperature and the local pressure. Nor do we account, in this preliminary study, for aluminum particles in the gas flow, an important matter to be investigated in the future. Coarse-mesh full rocket simulations with aluminum particles are reported in [7, 12].

The structure of the paper is as follows. Section 2 gives the mathematical formulation. In

particular, the governing equations for the gas flow in the nozzle and for heat conduction in the nozzle walls are given, along with a methodology for gas-solid phase coupling. Section 3 discusses the numerical strategies for computing the erosion of the nozzle. Included is a discussion about how to prescribe the nozzle inlet conditions, one of the important aspects of this work. One method is based on a multiscale model, and this is presented here. Section 4 presents several verification tests. Section 5 presents results for the BATES motor, including a case with a vectorized nozzle, and Section 6 gives concluding remarks.

2 Mathematical Formulation

2.1 Gas-Phase

The three-dimensional equations in the gas-phase consist of Favre averaged mass, momentum and energy conservation for a viscous, compressible, ideal gas and are written as

$$\partial_t \bar{\rho} + \partial_j (\bar{\rho} \tilde{u}_j) = 0, \quad (1)$$

$$\partial_t (\bar{\rho} \tilde{u}_i) + \partial_j (\bar{\rho} \tilde{u}_i \tilde{u}_j) + \partial_i \bar{p} - \partial_j \tilde{\sigma}_{ij} = -\partial_j (\bar{\rho} \tau_{ij}), \quad (2)$$

$$\partial_t (\bar{\rho} \tilde{E}) + \partial_j ((\bar{\rho} \tilde{E} + \bar{p}) \tilde{u}_j) - \partial_j (\tilde{\sigma}_{ij} \tilde{u}_i - \tilde{q}_j) = -\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4, \quad (3)$$

where Reynolds averaging is denoted by an overbar and Favre averaging is denoted by a tilde. $\tilde{\sigma}_{ij}$ and $\tilde{q}_j$ are respectively the viscous stress terms and heat flux terms, expressed in terms of averaged variables. $\tilde{E}$ is the specific total energy,

$$\tilde{E} = \tilde{h} - \frac{\bar{p}}{\bar{\rho}} + \frac{\tilde{u}_j \tilde{u}_j}{2}. \quad (4)$$

And $\bar{\rho} \tau_{ij}$ is the Reynolds stress,

$$\bar{\rho} \tau_{ij} = \overline{\rho u_i'' u_j''}. \quad (5)$$

where $(\cdot)''$ terms are perturbations from the Favre mean. The turbulence terms in (3) are given by

$$\alpha_1 = \tilde{u}_i \partial_j (\bar{\rho} \tau_{ij}), \quad (6)$$

$$\alpha_2 = \partial_j (\overline{p' u_j''}) / (\gamma - 1), \quad (7)$$

$$\alpha_3 = \overline{p' \partial_j u_j''}, \quad (8)$$

$$\alpha_4 = \overline{\sigma_{ij} \partial_j u_i''}. \quad (9)$$

where $(\cdot)'$ terms are perturbations from the Reynolds mean.

A variety of turbulence models, including RANS, LES and DES, are available in the computational framework and provide a closure model for the Reynolds stresses as well as the $\{\alpha_i\}$ [13, 14]. LES, however, is computationally expensive for high Reynolds number flows due to the extremely fine mesh requirement near wall regions. Thus, in the results shown in this paper, the DES (Detached Eddy Simulation) model is used [13]. DES is a simple modification of Spalart-Allmaras (SA) turbulent model and switches from RANS(SA) to LES-like away from solid surfaces. This saves computational cost by increasing the resolution in the wall-normal direction whilst keeping the resolution in other directions relatively coarse. It is especially suitable for resolving the nozzle boundary layer.

2.2 Solid-Phase

In the solid-phase the nozzle material (both graphite and carbon-carbon) is usually modeled as a homogeneous material. Then a one-dimensional heat conduction equation is solved in the direction normal to the nozzle surface, a strategy justified when the thickness of the thermal layer is much smaller than the nozzle radius. For example, for a nominal erosion rate of 0.02 cm/s and a thermal diffusivity of 0.02 cm²/s, the thermal layer has a thickness of ~ 1 cm whereas a typical nozzle radius is ~ 1 m. Note, however, that for carbon-carbon nozzles the material is inhomogeneous as woven strands of carbon are embedded in resin so that the material is locally anisotropic [15]; a more realistic model would take that into account. Such complexities, along with ingredients such

as thermal expansion, are beyond the scope of this preliminary study.

In a frame attached to the nozzle surface, heat conduction is governed by the one-dimensional equation

$$c_c \left(\rho_c \frac{\partial T_c}{\partial t} + M \frac{\partial T_c}{\partial n} \right) = \frac{\partial}{\partial n} \left(\lambda_c \frac{\partial T_c}{\partial n} \right), \quad (10)$$

subject to the following boundary conditions:

$$\left. \frac{\partial T_c}{\partial n} \right|_{surface} = g, \quad (11)$$

$$\left. \frac{\partial T_c}{\partial n} \right|_{(n \rightarrow -\infty)} = 0. \quad (12)$$

Here, T_c is the solid-phase temperature, λ_c the thermal conductivity, ρ_c the density, c_c is the specific heat, $M = \rho_c \dot{r}_c$ the mass flux with $\dot{r}_c$ the erosion rate, and n is the distance along the wall normal. The term g is related to conditions in the gas phase, discussed in the next subsection. A variation on this formulation, one used by Acharya & Kuo [4] and Thakre & Yang [5], assumes that the normal distance can be well approximated by the radial distance. This is clearly reasonable if the slope of the nozzle wall is small, but that is often not the case. The question is examined in the Appendix in the context of the BATES motor, a commonly used test bed.

2.3 Gas-Solid Phase Coupling

The nozzle surface is an interface between the condensed phase and the gas phase, and certain connection conditions are imposed there, in addition to the specification of the regression or erosion rate. Thus the tangential velocities are equal, as are the normal mass fluxes in a frame tied to the surface:

$$\tilde{u}_t = 0, \quad (13)$$

$$\bar{\rho}_g(\tilde{u}_n - \dot{r}_c) = \rho_c \dot{r}_c. \quad (14)$$

Energy conservation has the form

$$[c] M \tilde{T}_s = \lambda_g \vec{n} \cdot \vec{\nabla} \tilde{T}_g - \lambda_c \vec{n} \cdot \vec{\nabla} T_c + M Q_s, \quad (15)$$

where $[\cdot] \equiv (\cdot)_g - (\cdot)_c$, and Q_s is the surface heat release. The temperature is continuous, $[T] = 0$. In all of these formulae the subscript g refers to a gas-phase quantity whilst the subscript c refers to a condensed-phase quantity. Also, $\vec{n}$ is the surface normal pointing into the gas. The value of Q_s is chosen to be 100 kcal/kg in the current work.

The connection conditions are essentially identical to that of [5]. Even so, we knowingly neglect terms related to turbulent fluctuations. For example, equation (14) would assume the following form,

$$\bar{\rho}_g(\tilde{u}_n - \dot{r}_c) + \overline{\rho'_g u''_n} = \rho_c \dot{r}_c, \quad (16)$$

if terms related to turbulent fluctuations are accounted for. The same applies to equation (15), and terms like $[c] \overline{M' T''_s}$ and $\overline{\lambda''_g \vec{n} \cdot \vec{\nabla} T''_g}$ (if temperature dependent transport properties are assumed like in [5]) are neglected. To keep these terms would require closure modeling, and at the moment we do not know how to model these terms nor do we have the tools to perform a direct numerical simulation to develop appropriate subgrid models for the terms.

The erosion rate is determined from a pyrolysis law,

$$\dot{r}_c = A_c \bar{p}^\beta \exp\left(\frac{-E_c}{R_u \tilde{T}_s}\right), \quad (17)$$

where A_c is an empirical pre-exponential constant to be calibrated, E_c is the activation energy in the condensed-phase, R_u is the universal gas constant, and $\tilde{T}_s$ is the surface temperature. Representative values are $\beta = 0.5$ and $E_c/R_u = 34,625$ K [16]. The empirical parameter A_c is chosen such that a desired erosion rate is reached under certain operating conditions, and can be determined from experiments or more detailed numerical simulations. For example, for a measured erosion rate of 0.05 cm/s at a surface temperature of 2750 K and a chamber pressure of 10 atm, A_c is 4646 cm/s-atm^{0.5}. We comment here that the particular form of the pyrolysis law, equation (17),

neglects temperature fluctuations in the exponential term, and assumes them to be small compared to the mean.

At steady state, equation (10) can be integrated, and the solution substituted into equation (11) to give

$$\left. \frac{\partial T_c}{\partial n} \right|_{surface} = g_o = (\dot{r}_c / \alpha_c)(\tilde{T}_s - T_{-\infty}), \quad (18)$$

where g_o represents the steady-state value, $\dot{r}_c$ is given by equation (17), and α_c is the thermal diffusivity in the solid-phase. Substituting this equation into the energy balance equation (15), we obtain

$$\lambda_g \left. \frac{\partial \tilde{T}}{\partial n} \right|_g = \lambda_c g_o - M Q_s + [c] M \tilde{T}_s \approx \lambda_c g_o - M Q_s = \lambda_c (\dot{r}_c / \alpha_c)(\tilde{T}_s - T_{-\infty}) - M Q_s. \quad (19)$$

Here, we neglect the difference in the specific heats for convenience. This equation relates the surface temperature to the gas-phase surface heat flux at steady state. The equations in the gas-phase are now closed with the boundary conditions defined by equations (13), (14), and (19).

We use the steady state description in our calculations, and also neglect geometry changes due to surface regression. Rocflo can routinely accommodate both effects, if necessary.

3 Numerical Strategies

3.1 Rocflo

The gas-phase equations (1-3), subject to the boundary conditions (13), (14), and (19), are solved using Rocflo [11] with Detached Eddy Simulation (DES) for the turbulence model [13]. The equations are discretized in time using an explicit four-step Runge-Kutta method. A second-order finite volume central scheme is used for spatial discretization. Specifically, the scheme is based on cell-centered control volumes for the convective terms. The code provides provision for added second- and fourth-order artificial dissipation for enhanced stability, but artificial dissipation is not used in the present simulations when using DES. Convective fluxes through the control volume faces are

approximated using the average of variables.

The viscous terms are discretized in a similar fashion, making use of an auxiliary control volume shifted by a half grid cell in each computational direction. The auxiliary control volumes are used for the first derivatives and the primary ones (grid cells) for the second derivatives.

We use Rocflo for all of the nozzle calculations, and for some full-rocket simulations. It will be used in the future for coarse-mesh full-rocket turbulent simulations as part of the proposed two-step strategy.

3.2 Multiscale Formulation of Rocket Chamber Flows

The use of Rocflo in the two-step strategy is computationally intensive, and so we sought an easier strategy first, in order to verify our belief that turbulent fluctuations at the nozzle inlet plane need to be accounted for in the study of nozzle erosion. The strategy we have chosen for this purpose uses a multiscale asymptotic analysis to construct a description of the turbulence in a high aspect ratio chamber, and the outflow from this calculation can be used as nozzle inlet plane data. It has been formulated and used by Spalart for turbulent boundary layers [17] and by Venugopal *et al* [18] (and references therein) for planar SRM computations. Within the SRM context, it determines the flow dynamics at any plane with streamwise location $x/h = 1/\epsilon$ where ϵ is a small parameter, x is the distance from the head-end, and h is the half-width (radius) of the chamber for a planar (cylindrical) geometry. Asymptotic analysis valid in the limit of vanishing ϵ (which, as in all asymptotic analyses, we hope is accurate for small but finite values of physical interest), removes the overall axial gradients and generates a problem with periodic boundary conditions at the upstream and downstream ends in which the gradient effects appear as source terms in the evolution equations [18]. ϵ is a geometric parameter, but it is also the ratio of mass injection rate to the mean flow rate [18] at the location x/h . As noted above, the strategy was used by Venugopal *et al.* [18] for plane rocket chambers. It was recently applied to combustion simulations in two dimensions [19]. Here we use a formulation for incompressible (low Mach number) flow, restrictions of limited physical value but ones that enable us to easily examine whether or not

chamber fluctuations need to be accounted for in the evaluation of nozzle erosion.

The nondimensional conservation equations to be solved in vector, non-conservative form in the low Mach number limit are

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{V} = S_\rho, \quad (20)$$

$$\rho \frac{D\mathbf{V}}{Dt} + \nabla p - \frac{1}{Re} \nabla \cdot \tau_{ij} = \mathbf{S}_v, \quad (21)$$

and

$$\rho \frac{DT}{Dt} - \frac{\gamma}{Pr Re} \nabla^2 T + (\gamma - 1) \nabla \cdot \mathbf{V} = S_e, \quad (22)$$

with the equation of state

$$1 = \rho T. \quad (23)$$

Here, $Re = V_{inj}h/\nu$ is the Reynolds number based on the injection velocity, V_{inj} , and the chamber half-width (radius), h , for a planar (cylindrical) geometry. The source terms on the RHS arise as part of the multiscale analysis [18], and all spatial operators take either their planar or cylindrical (axisymmetric or three-dimensional) form.

These equations are solved using the high-order incompressible scheme described in [20]. Thus, the convective terms are discretized using the 5th-order weighted essentially non-oscillatory (WENO) scheme of [21], making it possible to resolve difficult applications with strong shear or turbulence. And the fractional-step scheme of [22] together with the optimized two-step alternating 4-6 low-dissipation and low-dispersion Runge-Kutta (LDDRK) scheme of [23] is used to improve temporal accuracy. The solver accurately captures the Proudman–Culick analytical solution for the steady flow in a slender chamber [24, 25].

The numerical behavior at $r = 0$ in both axisymmetric and three-dimensional cylindrical geometries need to be regularized so as to avoid the singularity. For velocity components at $r = 0$, an averaging over the first grid points away from $r = 0$ over 2π in the $r\theta$ -plane is carried out. To this end, u_x is simply averaged from u_x on those grid points. The other velocity components, u_r

and u_θ , are first transformed into Cartesian geometry as u_y and u_z , i.e., for the j th grid point in θ ,

$$u_{y,j} = u_{r,j} \sin(\theta) + u_{\theta,j} \cos(\theta), \quad (24)$$

$$u_{z,j} = u_{r,j} \cos(\theta) - u_{\theta,j} \sin(\theta), \quad (25)$$

and averaged in Cartesian geometry,

$$u_y|_{r=0} = \frac{1}{n_\theta} \sum_{j=1}^{n_\theta} (u_{r,j} \sin(\theta) + u_{\theta,j} \cos(\theta)), \quad (26)$$

$$u_z|_{r=0} = \frac{1}{n_\theta} \sum_{j=1}^{n_\theta} (u_{r,j} \cos(\theta) - u_{\theta,j} \sin(\theta)). \quad (27)$$

This transform is necessary because these grid points can be seen as close neighbors in both y - and z -directions but not in the θ -direction. The unique velocity vector obtained in this way can be transformed back into the cylindrical geometry as necessary.

The numerical scheme of [20] requires that the pressure update be solved for using a Poisson equation. To avoid the singularity at $r = 0$, instead of solving

$$\nabla^2 p = \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial x^2} \right\} p = RHS, \quad (28)$$

we solve

$$r^2 \nabla^2 p = \left\{ r \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{\partial^2}{\partial \theta^2} + r^2 \frac{\partial^2}{\partial x^2} \right\} p = r^2 RHS. \quad (29)$$

von Neumann conditions are imposed at the boundaries.

We show some results for the three-dimensional cylindrical geometry; a more detailed presentation will be presented elsewhere [26]. The normal injected velocity is set to one; i.e., $V_{inj} = 1$. Figure 2a shows a grid resolution study for the u_x rms velocity component for grids of 32^3 , 64^3 , and 128^3 . Note the relatively small differences between the two finest grids. Similar convergence is observed for the other rms velocity components and are not shown here. Profiles of all the rms velocities are shown in Figure 2b for the 128^3 grid. It is observed that the rms of the axial velocity

has the largest magnitude, while the remaining two have similar magnitudes with the radial rms velocity being the smallest. It is also interesting to note that near the axis, the rms of the radial and the circumferential components are almost identical asymptotically, showing some “degeneracy” as may be expected due to the singularity there. The circumferential vorticity ω_θ field is shown in Figure 3 where it is seen that there are small scale structures indicating a turbulent flow.

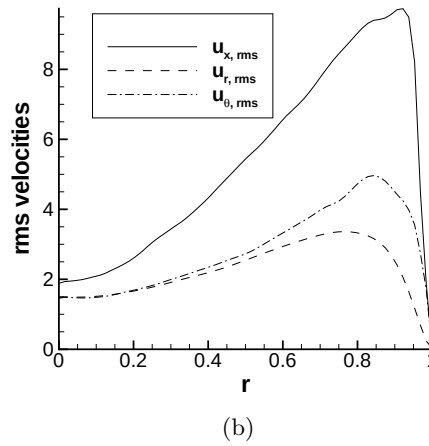
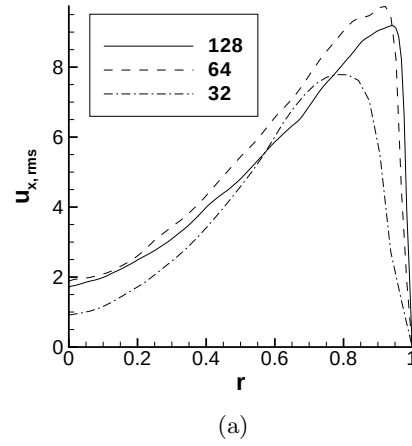


Figure 2: (a) Convergence of u_x velocity component and (b) profiles of rms velocities in a three-dimensional cylindrical multiscale simulation with Re of 1000 and ϵ of 0.04.

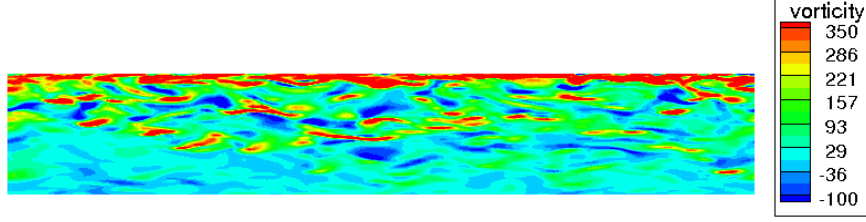


Figure 3: Circumferential vorticity ω_θ field in a three-dimensional cylindrical multiscale simulation with Re of 1000 and ϵ of 0.04. The mesh in the plane is 128×128 .

3.3 Patching Solutions in the Mean Flow Direction

Our two-step strategy requires the extraction of data from a coarse-mesh full length simulation at some midplane, which is then used as upstream data for a fine-mesh simulation downstream of the midplane. In particular, the full-configuration simulation is first integrated in time until initial transients disappear. The simulation is then further integrated in time while the flow variables such as density, velocity, etc. for the inlet conditions in the subsequent fine-mesh simulation downstream of the midplane are extracted during the course of the simulation at the midplane. The same applies to the extraction from the multiscale simulation, except that the extraction is only done for the velocity components since the multiscale formulation is currently restricted to the incompressible equations. The extraction is done uniformly in both space and time for convenience in the interpolation during the subsequent fine-mesh simulation. During the subsequent fine-mesh simulation downstream of the midplane, the data previously collected can be easily read in and linearly interpolated in both space and time. For example, the interpolation in time uses the linear relation

$$u(t) = \frac{(u_{t1} - u_{t0})}{(t_1 - t_0)}(t - t_0) + u_{t0} \quad (30)$$

to update the velocity $u(t)$ at one particular location on the inlet plane, where t_0 and t_1 are two adjacent extraction times. These values are then used to derive the stagnation quantities in the characteristic inflow boundary conditions implemented in Rocflo [11] that imposes stagnation pressure, stagnation temperature, and the direction of the velocity vector at the midplane.

4 Verification

In this section we describe calculations that provide some verification of the numerical schemes. We have already noted that the multiscale solver captures the Proudman–Culick solution, and will here describe some tests of `Rocflo` together with an evaluation of appropriate inlet data to be used when, as in our two-step strategy, flow calculations are divided into an upstream calculation and a downstream calculation.

4.1 The Compressible Laminar Boundary Layer

In examining nozzle erosion, the wall boundary layer must be properly captured. One key ingredient in the nozzle erosion simulation, namely a boundary condition that imposes normal heat flux along the nozzle boundaries, was missing in `Rocflo` and is implemented and verified here. As a sample problem we consider flow over a flat plate with a Mach number of 0.5, compute the temperature profile at a Reynolds number of 5000, and compare this profile with existing theoretical results through the Illingworth transformation [27]. The length of the computational domain is 2.5 m and the height is 2.0 m. A flat plate is placed at the bottom boundary from 0 to 2.0 m and a “buffer” region upstream of the plate. The number of grid points is 120 in x and 64 in y , and the grid is stretched in both x - and y -directions. The smallest cell size in y inside the boundary layer is 0.5 mm at $y = 0$ and stretched to ~ 75 mm at the top boundary while the smallest cell size in x is also 0.5 mm at the leading edge of the plate $x = 0$ and stretched to ~ 75 mm at both the upstream and downstream boundaries. The flat plate is treated as a no-slip wall with heat transfer, specified by $\frac{\partial T}{\partial y} = T_e \sqrt{\frac{U_e}{2\nu x}} g'$, where $T_e = 288$ K, $U_e = 170$ m/s, $\nu = 0.04$ m/s², and g' is a constant to be specified. A similarity solution using the Illingworth transformation exists for these boundary conditions for any constant value of g' . Here, we choose g' to be 0.01. A far-field boundary condition is imposed at the upstream and top boundaries of the domain, and an outflow boundary condition is imposed at the downstream boundary. The bottom boundary upstream of the plate is treated as a symmetry plane. The comparison is shown in Figure 4.

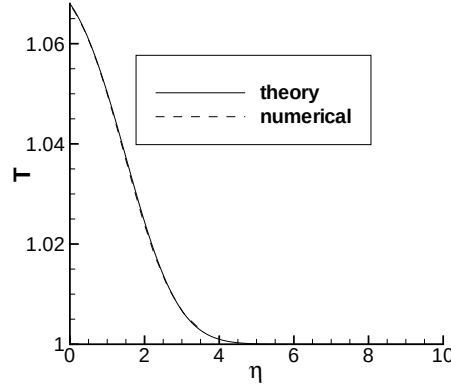


Figure 4: Profiles of temperature (normalized by the freestream temperature) at $x = 0.5$ m for a laminar boundary layer over a flat plate with heat transfer at Mach number of 0.5.

4.2 Flow in the Thakre and Yang nozzle

Thakre and Yang designed a nozzle for use in both experiment and numerical simulations [5]. They showed that the center-line flow is accurately predicted by an isentropic flow analysis. In particular, the analytical solution is given by

$$\frac{A}{A^*} = \frac{1}{M} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma + 1}{2(\gamma - 1)}}, \quad (31)$$

$$\frac{T}{T^*} = \frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right), \quad (32)$$

and the isentropic relations are used to find the relations between flow variables and the cross-section area, A , along the nozzle. Here we consider the case with a stagnation pressure of 6.9 MPa and a stagnation temperature of 3000 K at the inlet shown in Figures 3-5 of [5]. The specific heat ratio is 1.2, and the specific heat at constant pressure is 1719 J/kg-K. The latter is a species-averaged value.

Comparisons between the isentropic relations and the simulations are shown in Figure 5. The nozzle has an extended throat region, and this is responsible for the small step in each of the analytical solutions. It should be noted that the flow is not expected to be fully isentropic especially

near the wall due to friction. For example, the temperature along the surface, shown in Figure 6, is different from the isentropic solution.

4.3 Solution Patching

The patching procedure described in Section 3.3 applies to all of the verification tests in this subsection, where we will address the verification of this procedure for both steady and unsteady flows.

4.3.1 Laminar Steady Flows

Consider first a plane chamber, nozzle free, of length 0.581 m and height 0.02 m. There is uniform steady injection at the top surface, imposed symmetry at the centerline, and slip (no boundary layer) at the head end. The Proudman-Culick asymptotic solution is relevant. A 100×50 mesh is used, and from a full-length simulation flow variables such as the pressure and velocity are extracted at the half-way plane, $x = 0.29$ m. This data is then used as inflow data for the second simulation, from the half-way plane to the exit plane. To verify the interpolation scheme from a coarse mesh to a fine mesh as described above, a mesh of 100×100 is used in the simulation from the half-way plane to the exit plane. Note that, in both cases, the meshes are stretched in the wall-normal direction to exercise a non-trivial spatial interpolation. Also, since the flow is steady, no interpolation in time is necessary. Agreement between the first and second simulations in the latter domain is good as seen in the comparison illustrated by Figure 7.

4.3.2 Laminar Unsteady Flows

A second test uses the AFRPL (Air Force Rocket Propulsion Laboratory) BATES (Ballistic Test and Evaluation Systems) motor, specifically the 70-lb BATES (hereafter referred to as BATES70), described by Geisler and coworkers, [30, 28, 29]. A schematic is shown in Figure 8. We shall consider both the planar and axisymmetric geometries. Geometry details and other relevant parameters can be found in [7]. In the absence of imposed perturbations of significant amplitude, the flow in both

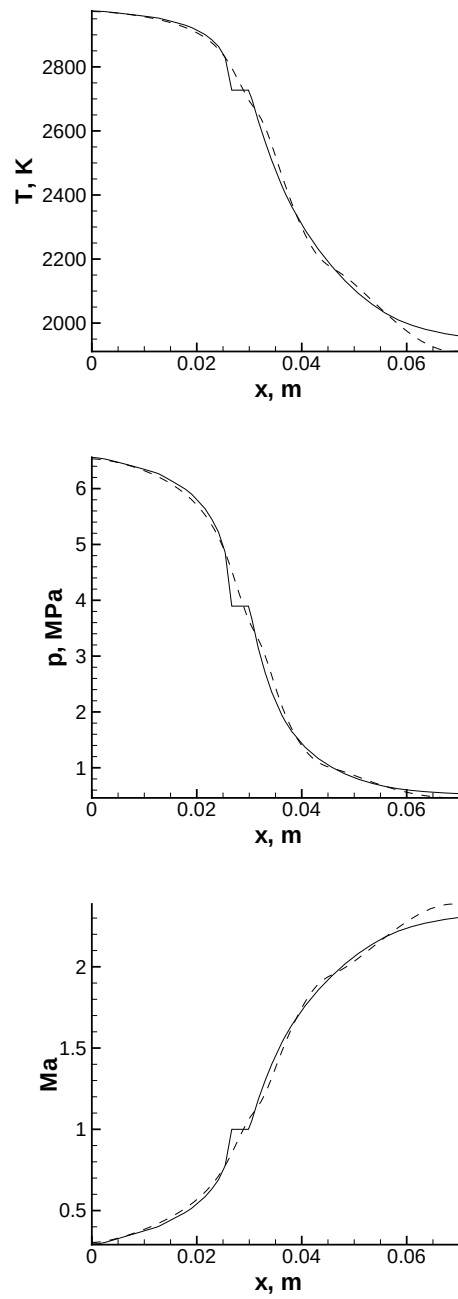


Figure 5: From top to bottom: profiles of static temperature, pressure, and Mach number (all shown as dash) and 1-D isentropic (solid) solutions along the nozzle center line. Note that the “step” in the analytical profile is due to the extended throat.

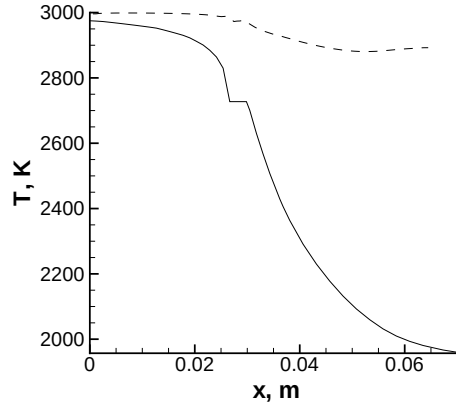


Figure 6: Profiles of static temperature (dash) and 1-D isentropic (solid) along the nozzle surface.

geometries is laminar.

Since our focus is the effect of turbulent inlet condition on nozzle erosion, which is obviously unsteady, the next verification test for the solution patching procedure is, therefore, for an unsteady flow. In this test a two-dimensional plane version of the BATES70 motor is used with an injection mass flux, $\dot{m}$, that is periodic in time,

$$\dot{m} = \dot{m}_o(1 + 0.1 \sin(\omega t)), \quad (33)$$

where $\dot{m}_o = 25.7 \text{ kg/m}^2\text{-s}$ and $\omega = 2000 \text{ Hz}$. The full-configuration simulation is first integrated in time until the initial transients disappear. The simulation is then further integrated in time while the flow variables such as density, velocity, etc., at the nozzle inlet plane are extracted for the subsequent nozzle-only simulation. The extraction is done for 40 uniform sampling points in the y -direction at every $10 \mu\text{s}$ and lasts for 20 ms. The extracted data is then used as the inlet conditions for the subsequent nozzle-only simulation. To this end, the data is linearly interpolated in both space and time, as neither the sampling points in space nor that in time coincides with the grid points at the inlet or time step in the nozzle-only simulation. Figure 9 shows the comparison of the density of both the full-configuration (solid) and the nozzle-only (dash with symbols) simulations

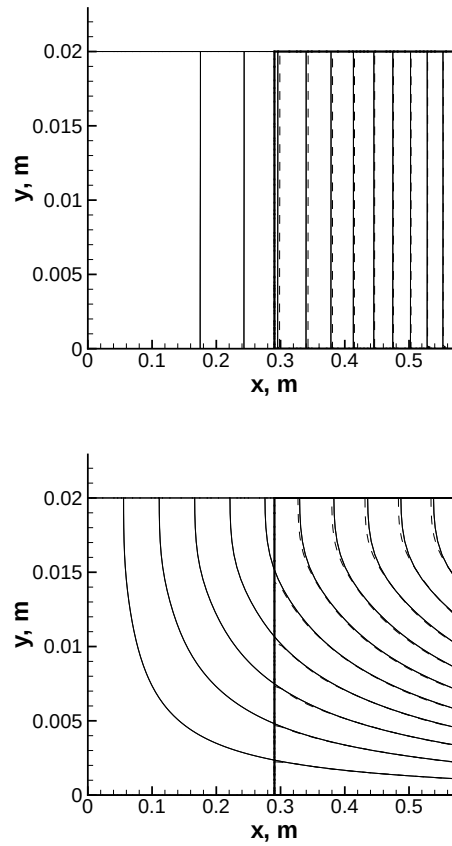


Figure 7: Pressure (top) and u -velocity (bottom) field in a full- and half-configuration (enclosed in bold edges, from $x = 0.29$ to 0.581 m). The solution in the full-configuration is plotted by solid lines while that of the half-configuration by dashed lines. The inflow boundary condition at $x = 0.29$ m in the half-configuration run is provided by the steady state solution obtained from the full-configuration run.

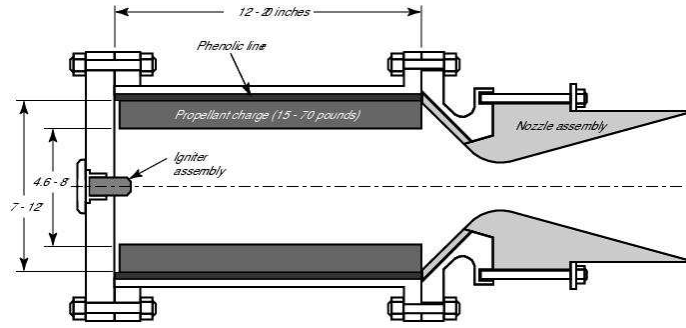


Figure 8: Schematic of BATES Configuration.

at locations indicated in Figure 10. Only two representative probes are shown in Figure 9 but good agreements are obtained at all the locations, especially downstream in the diverging section. The small differences between the two solutions observed for locations P2 is attributed to the effect of inflow boundary condition.

The above test problem is for an inviscid flow and the boundary layer along the nozzle surface is unresolved, as it was targeted for short turnaround time to verify the patching procedure within the nozzle region but outside the boundary layers. Here, we further verify the patching approach by a problem that involves viscous flow and resolves the boundary layer. The geometry setup is the same as before. A resolution of ~ 0.02 mm in the wall-normal direction is achieved by putting 36 uniform grid points within boundary layer of thickness 0.85 mm. Our focus in this test is the solution within the boundary layer. Figure 11 shows a comparison of the density evolution of the full-configuration (solid) and the nozzle-only (dash) simulations. Note that the two solutions agree reasonably well. Only two representative probes at $x = 0.59$ and 0.75 m are shown but we have also examined probes at a few other streamwise locations and similar agreements are obtained. It is also worthwhile to note that the inviscid solution described above with the boundary layer unresolved seems to capture the resolved solution here reasonably well away from the surface. This justifies our use of the coarse-resolution simulation to extract the inlet data for fine-mesh nozzle-only simulation.

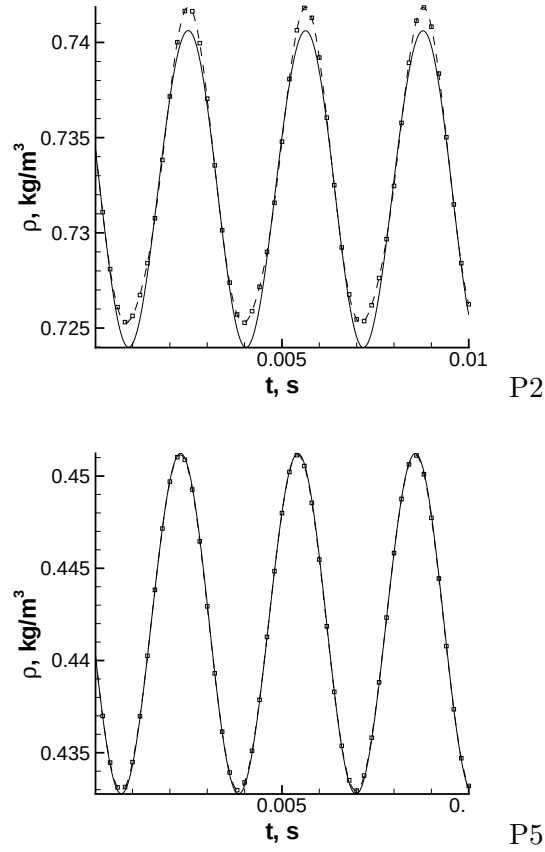


Figure 9: Comparison of density evolution of both full-configuration (solid) and nozzle-only (symbols) BATES70 simulations at quasi-steady state for two representative locations. These probe locations are shown in Figure 10. Other probes show similar agreements. Maximum error among the six probes is at probe 2 and is 0.17%.

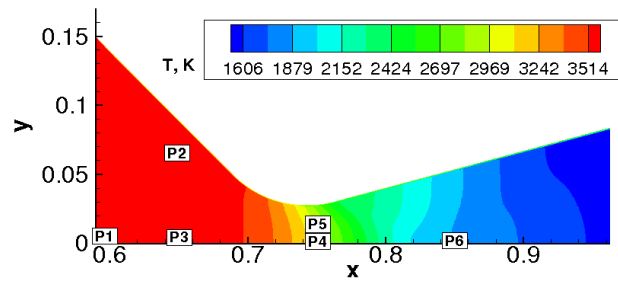


Figure 10: Positions of the stations for the histories shown in Figure 9. P1–P6: (0.59,0), (0.65,0.068), (0.65,0), (0.75,0), (0.75,0.012) and (0.85,0) m.

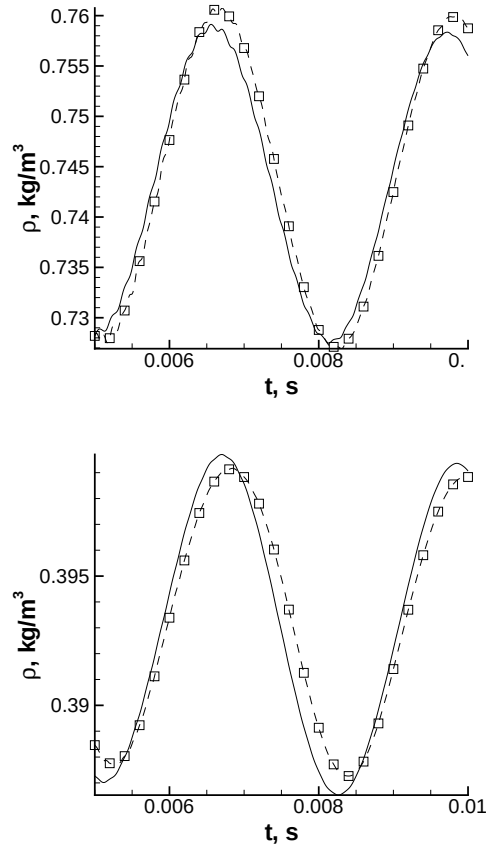


Figure 11: Comparison of density evolution of the full-configuration (solid) and nozzle-only (symbols) BATES70 simulations at quasi-steady state for two representative locations. The boundary layer along the nozzle surface is resolved and both stations are 0.1 mm from the surface and within the boundary layer. The probes in the top and bottom panels are located at $x = 0.59$ and 0.75 m, respectively.

For the axisymmetric motor configuration we compare the pressure fields between the full-configuration and the nozzle-only configuration; see Figure 12. No interpolation is used in this particular case since the flow is steady. Excellent agreement is obtained, as expected.

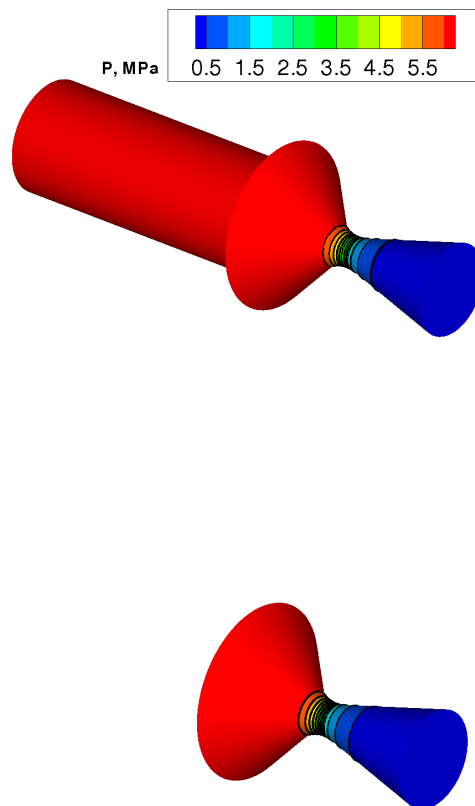


Figure 12: Pressure field in the nozzle of an axisymmetric BATES70 for the full-configuration (top) and the nozzle-only (bottom) simulations. The inlet conditions for the nozzle-only calculation is based on the steady state flow field extracted from the full-configuration calculation.

5 BATES Nozzle Simulations

In this section we describe a number of flow simulations using the BATES70 nozzle, for different inlet conditions. In one simulation we generate strong three-dimensional effects by tilting the nozzle.

5.1 Grid Issues

We start by examining possible numerical effects in a planar version of BATES, using uniform inlet conditions. We are particularly concerned with the sensitivity to parameters such as grid size and cell aspect ratio. Four sets of grids with 12, 24, 36 and 48 points in the wall-normal direction inside a wall layer 0.85 mm thick are considered. An example of the flow field generated for the 36 point grid, together with a close-up of the mesh near the surface, are shown in Figure 13. In Figure 14 the temperature and heat flux profiles are plotted at an axial location of $x = 0.692$ m for the four different grids. And in Figure 15, erosion rates for the different resolutions are shown. The results converge at a first order rate near the surface, and the rate resumes the expected second order of the central scheme in the interior. As a result, the erosion rate at the surface also converges at first-order.

Since the mesh near the nozzle surface is highly stretched as seen in Figure 13, an additional numerical effect to examine is that of the aspect ratio of the computational cells. This is done for the 36 point case by refining the grid resolution in the x -direction in the neighborhood of the throat ($0.68 \leq x \leq 0.8$ m). Three sets of grids with aspect ratios of 30, 15 and 7.5 are studied, with the 7.5 aspect ratio being the finest. The temperature and heat flux in the gas phase along the wall normal direction at $x = 0.692$ m are extracted and shown in Figure 16. The results are seen to be essentially insensitive to the aspect ratio.

We also examine the various grids for the axially symmetric BATES70 nozzle. Results are shown in Figures 17 and 18. In these calculations the empirical and physical parameters are calibrated to a throat erosion rate of ~ 0.3 mm/s for a metalized propellant with 15% aluminum in a BATES motor [31].

5.2 Unsteady Effects

We next examine the effect of imposing unsteady inlet conditions on the erosion rate along the nozzle surface. The test here is again a planar two-dimensional version of the BATES70 motor. The boundary layer along the nozzle surface is resolved with the same grid as in the two-dimensional

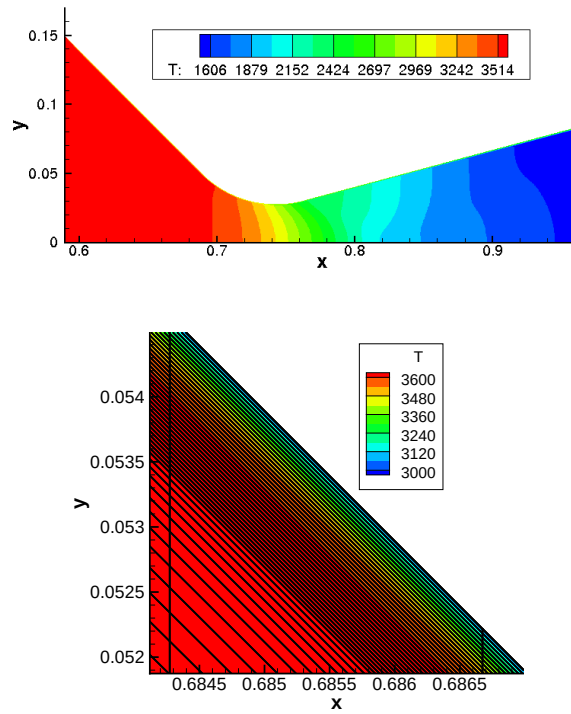


Figure 13: Temperature field of a planar BATES70 nozzle with uniform inlet condition ($P_o = 20$ MPa, $T_o = 3650$ K) and 36 grid points in the wall layer of 0.85 mm thick along the surface. The bottom panel shows the close-up view in the boundary layer at the wall. The mesh used is also plotted.

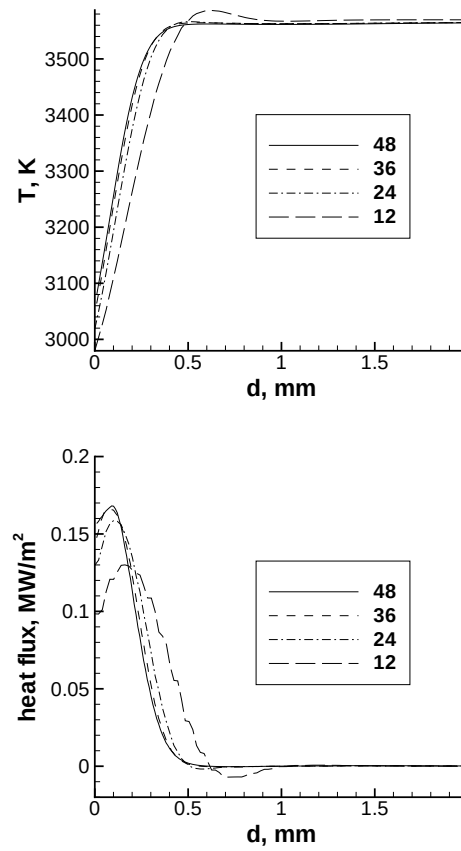


Figure 14: Profiles of temperature (top) and heat flux (bottom) at $x = 0.692$ m along the wall-normal direction for the case shown in Figure 13.

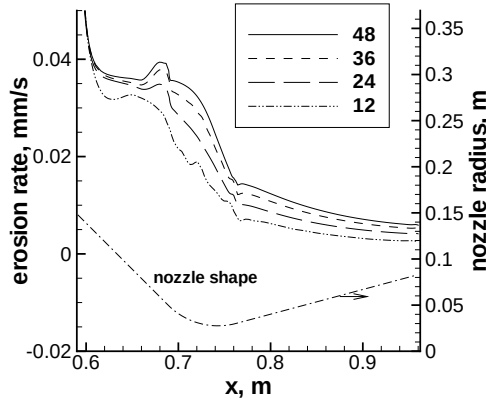


Figure 15: Grid convergence for the nozzle shown in Figure 13. The number refers to the number of grid points in the wall-normal direction inside a 0.85 mm layer along the nozzle surface.

36 grid case shown in Figure 13. The injection temperature is perturbed and periodic in time,

$$T = T_o(1 + 0.14 \sin(\omega t)), \quad (34)$$

where $T_o = 3650$ K and $\omega = 2000$ Hz. The amplitude of 14% of the perturbation in the injection temperature is based on morphology effects as discussed in [32]. A full motor is used for simplicity and is adequate to demonstrate the effect of unsteadiness in the inlet condition on nozzle erosion. The temporal evolution of the surface temperature and the erosion rate at $x = 0.65$ m are plotted in Figure 19. It is observed that even though the 14% variation in the injection temperature only leads to $\sim 1\%$ variations in the nozzle surface temperature, the variation in the erosion rate is as large as $\sim 15\%$. Note that this amplification effect is ultimately due to the large activation temperature of the surface reaction (equation 17). One obtains

$$\frac{d\dot{r}_c}{\dot{r}_c} = \frac{-E_c}{R_u \tilde{T}_s} \frac{d\tilde{T}_s}{\tilde{T}_s}, \quad (35)$$

by differentiating equation (17). We comment that the time-averaged erosion rate does not seem to differ much between the perturbed and the unperturbed cases. However, this does not hold

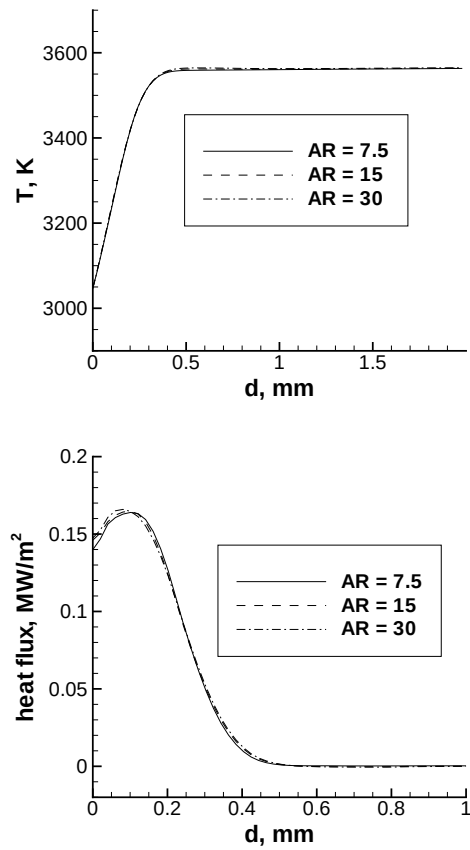


Figure 16: Profiles of temperature (top) and heat flux (bottom) at $x = 0.692$ m along the wall-normal direction for three different aspect ratios of computational cells in the boundary layer. Note that the grid shown in Figure 13 corresponds to aspect ratio of 30.

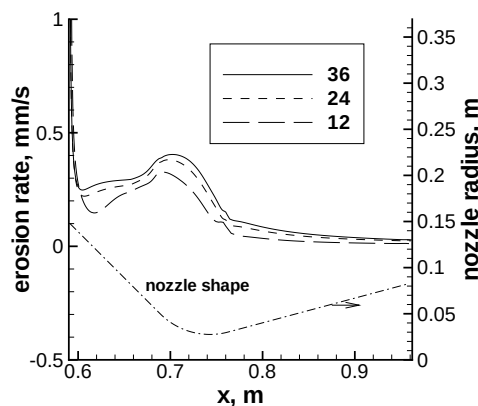


Figure 17: Erosion rate along the BATES70 nozzle surface at different resolutions. The number of grid points refers to that inside a 0.5 mm nominally boundary layer along the nozzle surface. Note that the inlet condition is uniform. Here, $P_o = 6.5$ MPa, $T_o = 3650$ K.

when hydrodynamic instabilities or when turbulence effects are present, as we shall see in the next subsection.

5.3 Hydrodynamic Instability Effects

Encouraged by the unsteadiness effect shown in the previous subsection, we investigate the effect of hydrodynamic instabilities on the erosion of a nozzle surface. Again, a planar two-dimensional version of the BATES70 motor is used. However, the chamber of the motor is extended to have a length of 4.5 m in order to excite self-sustained hydrodynamic instabilities. A 10% white noise is added to the injection rate for this purpose. Figure 20 illustrates the excited hydrodynamic instability via instantaneous vorticity contours during the statistically steady state. The instability manifests itself as parietal vortex shedding [33, 34, 35, 36]. The data at the nozzle inlet is collected during the statistically steady state for the solution patching for the subsequent nozzle-only simulation. An example of the data collected is shown in Figure 21 for the temporal evolution of temperature. Non-trivial temporal behavior is indeed seen as a result of the hydrodynamic instability. The nozzle-only configuration is next simulated with data previously collected from $t = 0.16 - 0.18$ s. The boundary layer along the nozzle surface is again resolved with the same grid

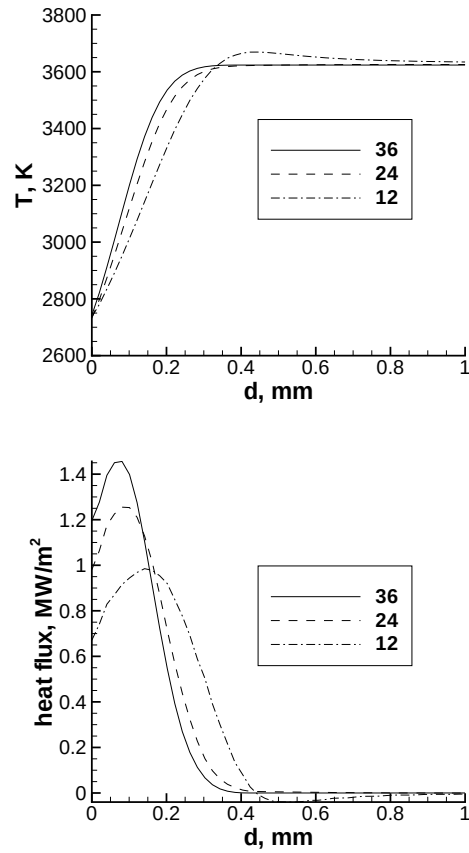


Figure 18: Profiles of temperature (top) and heat flux (bottom) at $x = 0.692$ m along the wall-normal direction for the three resolutions for the BATES70 nozzle.

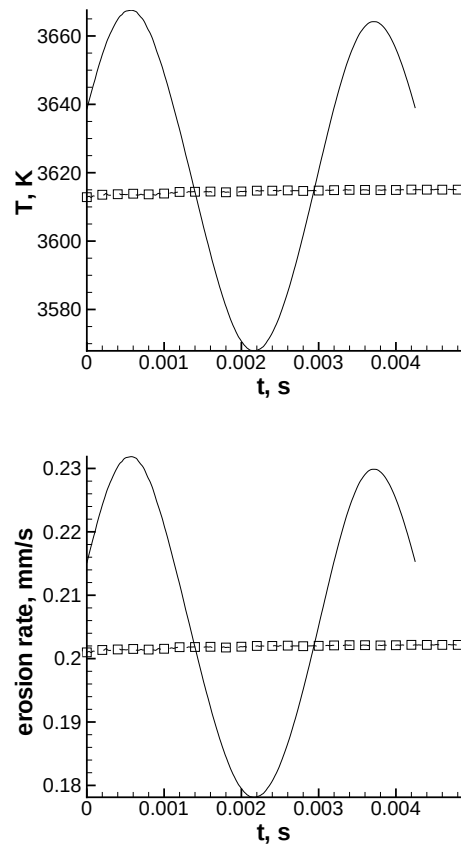


Figure 19: Comparison of surface temperature (top) and erosion rate (bottom) evolution at $x = 0.65$ m of BATES70 nozzle using perturbed (solid) and unperturbed (dashed) temperature injection conditions along the propellant surface.

as in the two-dimensional 36-grid case shown in Figure 13. The solution is compared with that by applying uniform inlet condition in Figure 22. The temporal variation in the erosion rate is $\sim 10\%$. Also, the erosion rate value using the uniform inlet solution is different from the time-averaged value using non-uniform inlet conditions by $\sim 5\%$. This simulation shows the importance of accounting for the unsteady behavior of the flow upstream of the nozzle for calculating the nozzle erosion rate when using nozzle-only simulations.

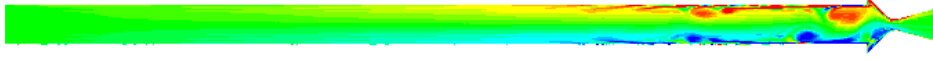


Figure 20: Vorticity contours of *modified* planar two-dimensional BATES. The length of the chamber is extended to 4.5 m instead of the original 0.5 m. In addition, 10% random white noise is added to the injection rate. The boundary layer along the nozzle is unresolved.

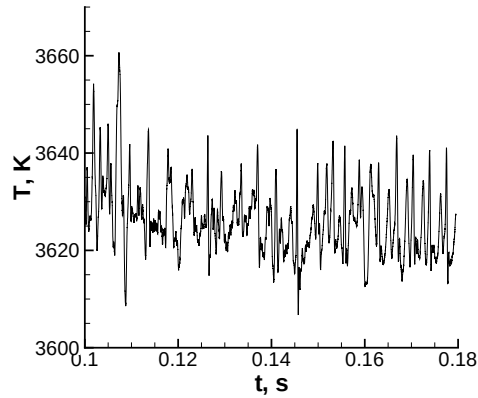


Figure 21: Temperature evolution after initial transient at the nozzle inlet at (0.589, 0.05) m of the modified BATES shown in Figure 20.

5.4 Three-dimensional (non-axisymmetric) effects

Three-dimensional effects arise in the nozzle region as the result of either nonuniform inlet conditions or a non-trivial geometry that leads to three-dimensional flows even when the nozzle geometry itself is axisymmetric. Thus for the Onera-86 rocket [13] the flow at the nozzle inlet is turbulent. It is

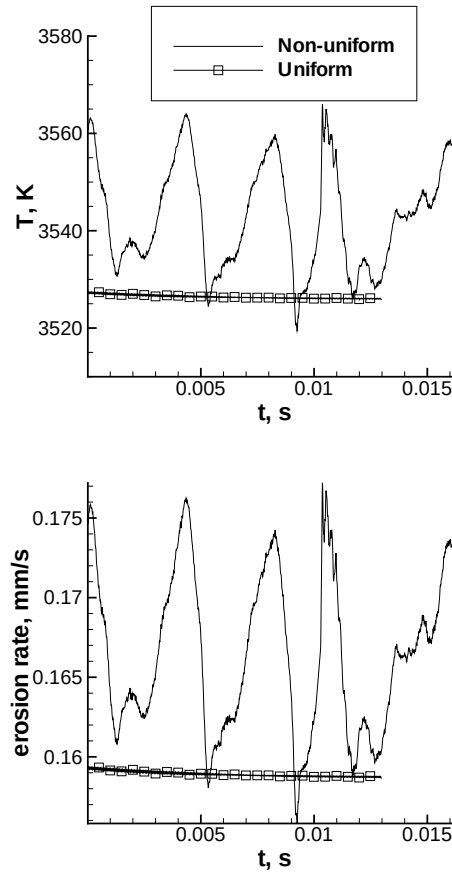


Figure 22: Surface temperature (top) and erosion rate (bottom) evolutions along the nozzle surface of the modified BATES motor near the throat at $x = 0.7$ m from a nozzle-only simulation with resolved boundary layers. Also plotted by dash line is the solution using uniform inlet conditions, where the stagnation temperature is set to the injection temperature of 3650 K and the stagnation pressure is based on the time averaged (isobaric) value at the inlet from the turbulent simulation.

worth noting that significant temperature fluctuations can also arise merely because a heterogeneous propellant is used [32]. Thus Figure 23 shows the temperature field at the nozzle inlet generated by a full-configuration (chamber and nozzle) BATES15 simulation; the flow is assumed to be laminar and future work will examine the effect of turbulence. And the multiscale rocket simulation results shown in Figure 3 reveal turbulence sufficiently far downstream.

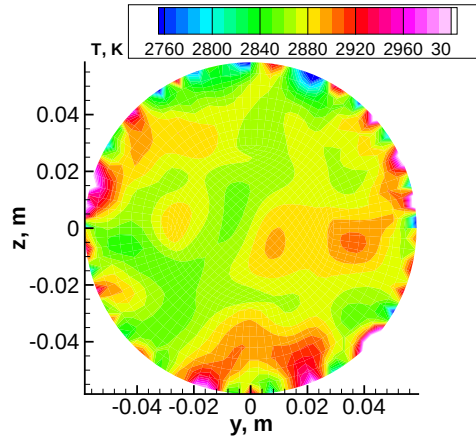


Figure 23: Temperature field at the nozzle inlet extracted from a full-configuration BATES15 simulation [37].

Our eventual goal is to demonstrate that three-dimensional behavior in the chamber region can have an effect on nozzle erosion, and must be taken into account for meaningful erosion predictions. One way to do this is to extract the flow field at the inlet plane in a full-configuration three-dimensional turbulent simulation and then use that data in a fine-resolution nozzle-only simulation, as discussed in Section 5.3 for the two-dimensional planar geometry. An alternative approach, as we have already noted, is to calculate the inlet conditions using the multiscale strategy. For example, the data obtained in the multiscale rocket simulation discussed in Section 3.2 can be used as the turbulent inlet condition for a three-dimensional BATES70 nozzle simulation. The three-dimensional cylindrical multiscale simulation discussed in Section 3.2 is used here, and proper reference scales based on the physical parameters of the BATES motor are used to obtain dimensional values. In particular, the time histories of the velocity components are collected at the

middle plane with a uniform time interval between each frame or sample. The patching method discussed in Section 3.3 and verified in Section 4 is then used in the subsequent Rocflo nozzle-only simulation. In this test, the velocity and temperature are directly imposed at the inflow boundary. The velocity is from the three-dimensional cylindrical multiscale simulation as described above. The evolution of the extracted u -velocity at one particular location on the nozzle inlet plane is shown in Figure 24 to demonstrate the nontrivial temporal behavior. The temperature is simply perturbed and periodic in time,

$$T = T_o(1 + 0.1 \sin(2\pi ft)), \quad (36)$$

where $T_o = 3650$ K and $f = 2000$ Hz. We specify the temperature independent of the velocity because, at the moment, we are solving the incompressible multiscale equations. In the future we plan on developing a high-order numerical code to solve the compressible equations. In this case the velocities and temperature fields will be dependent, and will be described as joint pdfs along the nozzle inlet plane. In addition to imposing unsteady inlet conditions, we also examine three-dimensional effects induced by a non-trivial geometry by considering a “gimbaled” or thrust vector control nozzle. The configuration is generated by rotating the axisymmetric BATES nozzle 25° about the z -axis whilst retaining a small portion of the unrotated chamber. This generates a three-dimensional flow-field even when the inlet data is uniform. For the turbulent simulations, DES discussed earlier is used.

The z -vorticity field is plotted in Figure 25 to demonstrate the combined effect of a three-dimensional geometry with turbulent inlet conditions on nozzle erosion. The erosion rates along the top and bottom intersections of the $z = 0$ plane and the nozzle surface are shown in Figure 26. Significant differences between the top and bottom intersections and between the turbulent and uniform inlet conditions are seen. For instance, the erosion rate near the throat for the turbulent inlet condition is $\sim 10\%$ higher than that of the uniform inlet condition. Also, the erosion rate is nearly two times higher near the inlet region due to the enhanced heat transfer caused by the turbulence injected along the inlet plane. Finally, the unsteadiness of the erosion rate due to the

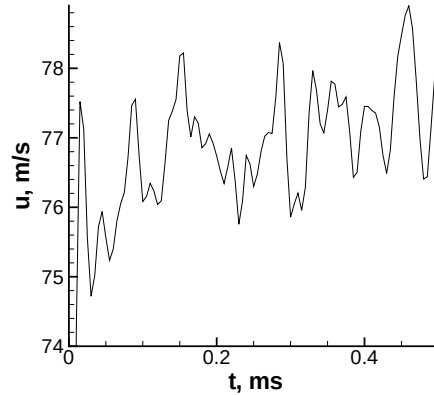


Figure 24: Evolution of the u -velocity at one location in the nozzle inlet plane, extracted from the three-dimensional cylindrical multiscale simulation.

turbulent inlet conditions is demonstrated in Figure 27.

6 Concluding Remarks

A computational framework is presented to investigate nozzle erosion in solid-propellant rocket motors. Currently, a numerical simulation of the full motor geometry, while resolving the boundary layer in the nozzle region, is computationally prohibitive. To optimize the computational resources, the nozzle configuration is studied by itself and appropriate inlet conditions must be imposed. Current industry practice is to apply uniform and steady inlet conditions. Here, we propose two complementary formulations to apply proper inlet conditions: the first is based on extracting the flow field from a coarse-resolution (i.e., fine enough to model the turbulence in the bulk geometry, but not resolve the nozzle boundary layer) from the full motor configuration, and the second invokes a multiscale asymptotic analysis of turbulent flows inside a rocket. A patching procedure is described which takes data from either a full-configuration simulation or from a simulation using the multiscale equations and imposes that data at the inlet plane for a resolved nozzle-only simulation. The patching procedure, which uses interpolation in space and time, is then verified by a number of test problems.

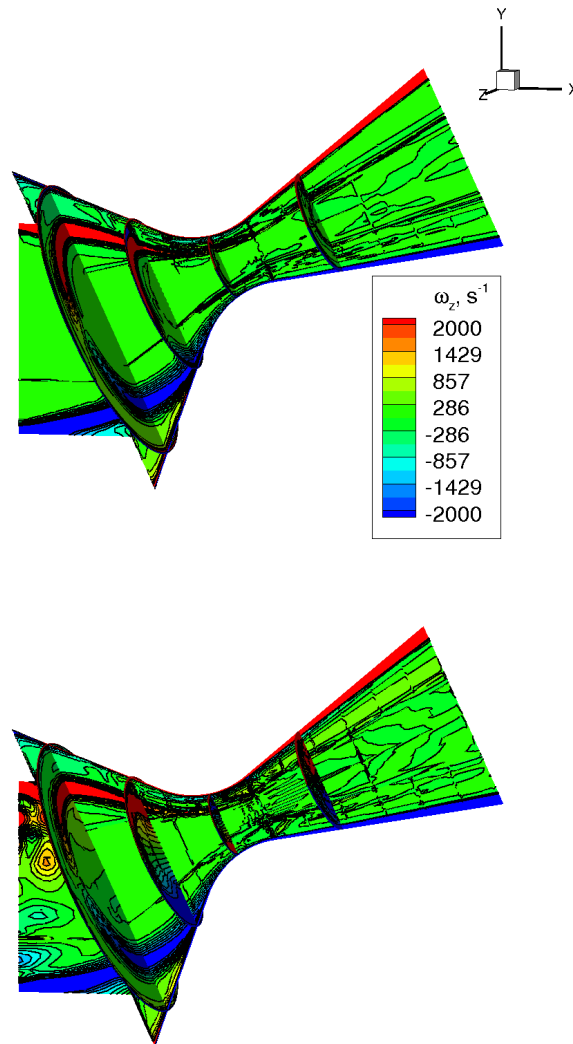


Figure 25: Instantaneous z -vorticity fields in the “gimbaled” BATES nozzle with uniform (top) and turbulent (bottom) inlet conditions. For the uniform (top) inlet conditions, $P_o = 6.5$ MPa, $T_o = 3650$ K. For the turbulent (bottom) inlet conditions, the inlet temperature is given by equation (36), the velocities from the multiscale simulation, and pressure is determined from outgoing characteristics and oscillates around 6.5 MPa.

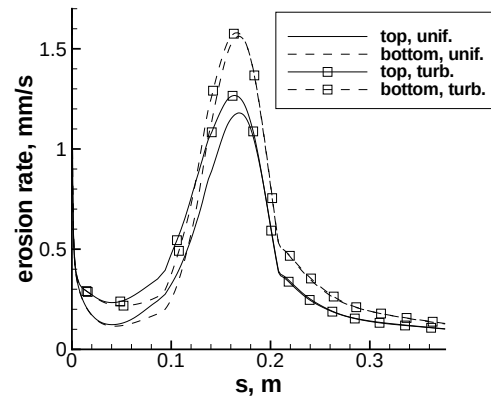


Figure 26: Erosion rates along the top and bottom intersections of the $z = 0$ plane and the nozzle surface for simulations using either turbulent or uniform inlet conditions. The s -axis is the distance from the respective leading edge of the intersections along the nozzle.

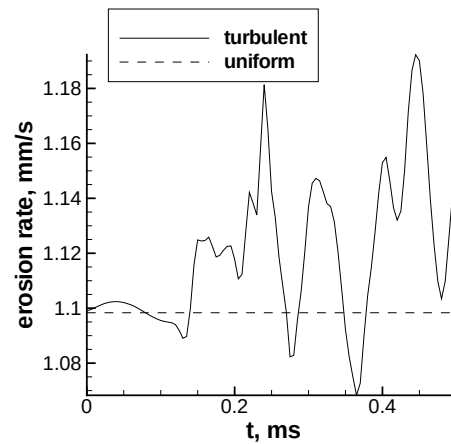


Figure 27: Evolution of the erosion rate near the throat of the BATES nozzle for simulations with turbulent (solid) and uniform (dash) inlet conditions.

Highly resolved simulations are performed in a BATES 70lb motor to understand the effects caused by non-uniform and unsteady inlet conditions. In planar two-dimensional geometry, the temporal variation in the erosion rate is shown to be as large as $\sim 15\%$ with a 14% variation in the injection temperature. When hydrodynamic instabilities are present in a slender rocket (large L/D), the value of the erosion rate using uniform inlet conditions are different from the time-averaged value using non-uniform and unsteady inlet conditions by $\sim 5\%$. Finally, three-dimensional effects is demonstrated in the context of non-axisymmetric geometry for a “gimbaled” or thrust vector control nozzle. All of the results presented here suggest that specifying turbulent, rather than uniform, inlet conditions can have a significant effect on nozzle erosion.

Future work will include such effects as non-equilibrium chemistry, particles, and more advanced modeling of the nozzle material, to investigate nozzle erosion.

Appendix: Effect of Radial direction being Different from the Surface Normal Direction

Note that equation (10) is different from the heat transfer governing equation in both Acharya and Kuo [4] and Thakre and Yang [5]. In their case, the heat transfer is assumed to be dominant in the radial direction (which is not necessarily the surface normal direction) and the equation that is solved in the r -direction is given by

$$\rho_c \frac{\partial h_c}{\partial t} + \frac{\rho_c}{r} \frac{\partial(r h_c \dot{r}_c)}{\partial r} = \frac{1}{r} \frac{\partial}{\partial r} \left(\lambda_c r \frac{\partial T_c}{\partial r} \right). \quad (37)$$

This potentially could have an impact on the boundary conditions imposed. In the planar version of BATES, for example, the heat flux at the boundary should have components in both x - and y -directions that are decomposed from the heat flux in the predominant wall-normal direction as can be seen to be the case in Figure 13. The difference between the heat flux in the wall normal and y -direction along the nozzle surface is plotted in Figure 28.

Similarly, the gas velocity at the gas-solid interface should have velocity components in both x -

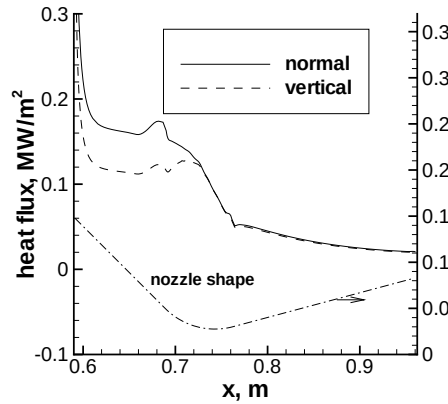


Figure 28: Heat flux in the wall-normal (solid) and y -direction (dash) along the BATES nozzle surface for the case shown in Figure 13.

and r -directions that are decomposed from the normal velocity. To illustrate the effect of neglecting the difference between the radial and surface normal direction, let us consider a two-dimensional planar geometry and a straight nozzle surface curve that has an inward surface normal vector of $(\cos(\theta), \sin(\theta))$ pointing downward. Instead of being $-u_n$, u_y should actually be $u_n \sin(\theta)$ and instead of being zero, u_x should actually be $u_n \cos(\theta)$. In case of the converging section of BATES70, θ is $\sim 3/4\pi$ and the magnitude of u_y could be overestimated by about 40%, and u_x is equal to u_y instead of zero.

Acknowledgments

This work was supported through Buckmaster Research under grant FA9550-07-C-0123 under a Phase II program with the Air Force, program manager Dr. A Nachman. This work was partially supported by the US Department of Energy through the University of California under subcontract B523819, and by NASA Constellation University Institutes Project under grant NCC3-989, and through the University of Maryland under subcontract Z634015, with Claudia Meyer as the project manager. We would like to thank Dr. Piyush Thakre and Professor Vigor Yang for providing the nozzle shape data in their study.

References

- [1] L. J. Delaney, L. C. Eagleton, W. H. Jones. “A Semiquantitative Prediction of the Erosion of Graphite Nozzle Inserts”, *AIAA Journal* 1964; 2 8: 1428, 33.
- [2] A. J. McDonald, P. O. Hedman. “Erosion of Graphite in Solid-Propellant Combustion Gases and Effects on Heat Transfer”, *AIAA Journal* 1965; 3 7: 1250, 57.
- [3] K. K. Kuo, S. T. Keswani. “A Comprehensive Theoretical Model for Carbon-Carbon Composite Nozzle Recession”, *Combustion Science and Technology* 1985; 42: 145, 64
- [4] Acharya, R. and Kuo, K.K., “Effect of Chamber Pressure and Propellant Composition on Erosion Rate of Graphite Rocket Nozzle,” *Journal of Propulsion and Power*, Vol. 23, No. 6, 2007, pp. 1242-1254.
- [5] Thakre and Yang, “Chemical Erosion of Carbon-Carbon/Graphite Nozzles in Solid-Propellant Rocket Motors”, *Journal of Propulsion and Power*, Vol. 24, No. 4, 2008, pp. 822-833.
- [6] Dick, W., Heath, T., Fiedler, R., and Brandyberry, M., “Advanced Simulation for Solid Propellant Rockets”, 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2005-3990, July 10-13, 2005.
- [7] Najjar, F. M., Massa, L., Fiedler, R., Haselbacher, A., Wasistho, B., Balachandar, S., and Moser, R. D., “Effects of Droplet Loading and Sizes in Aluminized BATES Motors: A Multiphysics Computational Analysis,” 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2005-3997, July 10-13, 2005.
- [8] Glaze, David J. and Frankel, Steven H., “Stochastic Inlet Conditions for Large-Eddy Simulation of a Fully Turbulent Jet,” *AIAA Journal*, 41(6):1064-1073, 2003.
- [9] Jiao, X., Zheng, G., Lawlor, O. S., Alexander, P. J., Campbell, M. T., Heath, M. T., and Fiedler, R. A., “An Integration Framework for Simulations of Solid Rocket Motors”, 41st

- AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2005-3991, July 10-13, 2005.
- [10] Massa, L., Jackson, T. L., and Buckmaster, J., “Using Heterogeneous Propellant Burning Simulations as Subgrid Components of Rocket Simulations”, AIAA Journal, vol. 42, no. 9, 2004, pp. 1889-1900.
- [11] Blazek, J., “Flow Simulation in Solid Rocket Motors Using AdvancedCFD”, AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit ,AIAA-2003-5111, 2003.
- [12] Najjar, F. M., Ferry, J., Haselbacher, A. and Balachandar, S., “Simulations of Solid-Propellant Rockets: Effects of Aluminum Droplet Size Distribution,” *Journal of Spacecraft and Rockets*, Vol. 43, 6, pp. 1258-1270, 2006.
- [13] Wasistho, B., and Moser, R. D., “Simulation Strategy of Turbulent Internal Flow in Solid Rocket Motor”, *Journal of Propulsion and Power*, vol. 21, no. 2, 2005, pp. 251-263.
- [14] Wasistho, B., Balachandar, S., and Moser, R. D., “Compressible Wall-Injection Flows in Laminar, Transitional, and Turbulent Regimes: Numerical Prediction,” *Journal of Spacecraft and Rockets*, Vol. 41, No. 6, 2004, pp. 915-924.
- [15] Sutton, G.P. and Biblarz, O. “Rocket Propulsion Elements.” 7th Edition, John Wiley and Sons (2001).
- [16] Chelliah, et al. *Combustion and Flame* 1996, 104.
- [17] Spalart, P. “Direct Simulation of a Turbulent Boundary Layer up to $Re_\theta=1410$,” *J. Fluid Mech.*, Vol. 187, 1988, pp. 61-98.
- [18] Venugopal, P., Moser, R.D. and Najjar, F.M. “Direct Numerical Simulation of Turbulence in Injection Driven Plane Channel Flows.” *Phys. of Fluid*, **20**, 105103, 2008.

- [19] A. H. G. Isfahani, Zhang, J. and Jackson, T.L. “The Effects of Turbulence-induced Time-Periodic Shear on a Flame Anchored to a Propellant” *accepted by Comb. & Flame*, 2008.
- [20] Zhang, J. and Jackson, T.L. “A High-Order Incompressible Flow Solver with WENO.” *J. of Comp. Phys.*, 228, (2009), pp. 2426.
- [21] G.S. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202.
- [22] Kim, J. and Moin, P., Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (1985), 308-323.
- [23] F.Q. Hu, M.Y. Hussaini, J.L. Manthey, Low-dissipation and -dispersion Runge-Kutta schemes for computational acoustics, *Journal of Computational Physics* 124 (1996) 177-191.
- [24] Proudman, I., “An Example of Steady Laminar Flow at Large Reynolds Number,” *Journal of Fluid Mechanics*, Vol. 9, 1960, pp. 593-602.
- [25] Culick, F. E. C. 1966 “Rotational axisymmetric mean flow and damping of acoustic waves in solid propellant rocket motors”, *AIAA J.* 4, 1462-1464.
- [26] Zhang, J. and Jackson, T.L. “Direct Numerical Simulation of Turbulence in Injection Driven Three-Dimensional Cylindrical Flows”, in preparation.
- [27] Illingworth, C. R., “Some Solutions of the Equations of Flow of a Viscous Compressible Fluid,” *Proc. Cambridge Phil. Soc.*, vol. 46, pp. 469-478.
- [28] Geisler, R.L., Beckman, C.W., and Kinkead, S.A., “The Relationship between Solid Propellant Formulation Variables and Motor Performance”, AIAA-75-1199, 11th AIAA/SAE Propulsion Conference, Anaheim, CA, 1975.
- [29] Geisler, R.L., and Beckman, C.W., “The History of the BATES Motors at the Air Force Rocket Propulsion Laboratory”, AIAA-98-3981, 11th AIAA/SAE Propulsion Conference, Anaheim, CA, 1998.

- [30] Geisler, R.L., "A Global View of the Use of Aluminum Fuel in Solid Rocket Motors", AIAA-2002-3748, 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Indianapolis, IN, 2002.
- [31] Geisler, R.L., "The Relationship Between Solid-Propellant Formulation Variables and Nozzle Recession Rates," JANNAF Rocket Nozzle Technology Subcommittee Meeting, Lancaster, CA, 1213 July 1978.
- [32] Massa, L., Jackson, T.L., Buckmaster, J. and Najjar, F. "Fluctuations above a burning heterogeneous propellant," *Journal of Fluid Mechanics* Vol. 581, 2007, pp. 1-32.
- [33] Lupoglazoff, N. and Vuillot, F. "Parietal vortex shedding as a cause of instability for long solid propellant motors - Numerical simulations and comparisons with firing tests." AIAA Paper 96-0761 (1996).
- [34] Lupoglazoff, N. and Vuillot, F. "Numerical simulations of parietal vortex-shedding phenomenon in a cold flow set-up." AIAA Paper 98-3220 (1998).
- [35] Avalon, G., Ugurtas, B., Grisch, F. and Bresson, A. "Numerical computations and visualization tests of the flow inside a cold gas simulation with characterization of a parietal vortex shedding." AIAA Paper 2000-3387 (2000).
- [36] Apte, S.V. and Yang, V. "A large-eddy simulation study of transition and flow instability in a porous-walled chamber with mass injection." *Journal of Fluid Mechanics*, 477:215-225 (2003).
- [37] A. Nevill, Private Communication.

Three-dimensional reconstruction of statistically optimal unit cells of polydisperse particulate composites from microtomography*

H. Lee², M. Brandyberry², A. Tudor² and K. Matouš¹ [†]

October 26, 2009

¹ Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA.

² Computational Science and Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA.

Abstract

In this paper, we present a systematic approach for characterization and reconstruction of statistically optimal representative unit cells of polydisperse particulate composites. Microtomography is used to gather rich three-dimensional data of a packed glass beads system. First-, second- and third-order probability functions are used to characterize the morphology of the material, and the parallel augmented simulated annealing algorithm is employed for reconstruction of the statistically equivalent medium. Both the fully resolved probability spectrum and the geometrically exact particle shapes are considered in this study, rendering the optimization problem multidimensional with a highly complex objective function. A ten-phase particulate composite composed of packed glass beads in a cylindrical specimen is investigated, and a unit cell is reconstructed on massively parallel computers. Further, rigorous error analysis of the statistical descriptors (probability functions) is presented and a detailed comparison between

*Manuscript accepted for publication in *Physical Review E*

[†]Corresponding author, kmatus@nd.edu.

statistics of the voxel-derived pack and the representative cell is made.

Key words: random heterogeneous particulate composite, representative unit cell, microtomography, probability functions, parallel augmented simulated annealing algorithm.

1 Introduction

Computational methods based on particulate packs are commonly used in variety of scientific disciplines. For example, particulate packs have been used in modeling of heterogeneous materials, such as solid propellants [23, 31], granular media [1], protein folding [25] and low-temperature phases of matter such as liquids, crystals and glasses [9]. Moreover, packing problems are common in information theory [39] and many different branches of pure mathematics [13]. A study of these systems in a computational framework usually starts with a model of the morphology, such as a packing algorithm. Therefore, a packing algorithm to guide these models has been in the forefront of mathematical and scientific investigations for many decades, and this fascination led to development of several packing codes that are capable of producing high-quality polydisperse heterogeneous packs [22, 28, 19].

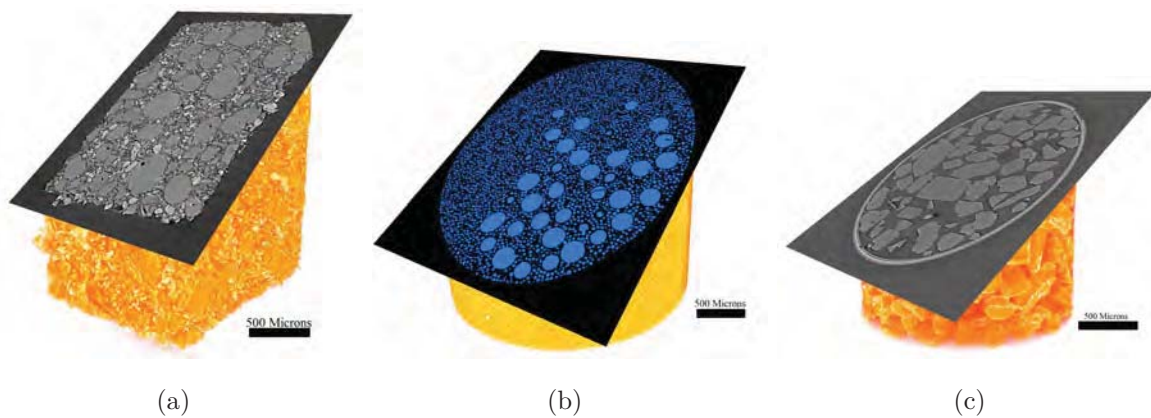


Figure 1: (Color online) Typical members of the micro-CT based ensemble. a) Heterogeneous propellant. b) Glass beads system. c) Table sugar.

Recently, new developments in three-dimensional imaging using microtomography (micro-CT) have also ushered in the rapid expansion of statistical modeling techniques that investigate the morphology and the microstructure characterization of widely used material systems, such as propellants [17, 11], glass beads [4], paper [15], and engineered cementitious composites [8], just to name a few. An example of the complicated microstructures obtained from the micro-CT can be seen in Figure 1.

The subsequent statistical characterization is usually performed, for both computationally

and/or tomographically obtained packs, in order to understand the internal structure of these systems. The need for such understanding and importance of the higher-order statistics start with early work of Bernal [6] who investigated the geometrical structure of liquids using the radial distribution function. Significance of statistical description galvanized several research communities in condensed matter physics and far beyond, with applications in non-Gaussian noise as a tool to study disordered materials [46], and application of Minkowski functionals in analysis of background cosmic radiation [37], just to name a few. Moreover, the analysis of higher-order statistics, in the guise of the analysis of x-ray speckle, is increasingly gaining attention among condensed matter physicists [47].

Unfortunately, both computationally and/or tomographically derived packs are often too large to be uniformly resolved in practical numerical simulations of combustion phenomena [23], nonlinear viscoelastic response of a binder [2] or damage evolution along the particle-matrix interface [30]. Therefore, many researchers have devoted their attention to finding a statistically optimal unit cell. Povirk [35] proposed a method for determining periodic microstructures in two-dimensions that are statistically similar to more complex, random, two-phase microstructures by using a certain statistical descriptor function. Yeong and Torquato [48, 49] proposed a method for the reconstruction of random media based on two-point probability functions using Simulated Annealing (SA) and two-dimensional material slices. Bochenek and Pyrz [7] also used the Simulated Annealing procedure in conjunction with a pair correlation function and a stress interaction parameter to reconstruct a unit cell in three-dimensions. However, the simple pair correlation function used in their work did not represent the disparate particle modes, and thus, the probability spectrum optimized in [7] was restricted. Zeman and Šejnoha [50] have examined three considerably different material systems: a fiber composite, a woven composite and a masonry, to demonstrate the reconstruction using two-point probability functions and the lineal path function, yet again, only two-dimensional images of microstructures have been employed.

Recently, Jiao *et al.* [20, 21] have reconstructed a three-dimensional realization of Fontainebleau sandstone and a boron-carbide/aluminum composite from two-dimensional tomographic images using second-order statistics and a “lattice-point” algorithm. In their work, a lattice or a voxel based reconstruction is used for a two-phase medium in conjunction with a basis function approximation of the correlation functions for a faster probability function evaluation. Fullwood *et al.* [16] have also used a voxel-based phase-recovery algorithm for reconstruction of three-dimensional polycrystals. Their method is based on a fast discrete Fourier transform, and thus, its parallelization is cumbersome and the domain analyzed in the paper is relatively small, as pointed out later. Although relatively simple, the pixel/voxel based representation of microstructure simplifies the geometrical considerations, since any geometrical object, such as a sphere, a rhombus, etc., is approximated by a cluster of voxels, and thus, the resolution is highly

pixel/voxel dependent. Moreover, for many physical phenomena, such as a particle decohesion, the voxel based representation is inappropriate due to the step-like interface representation.

Similar to our approach described hereafter is the work of Seidler *et al.* [38]. In their paper, granules are first mapped to spheres, as in our work, and several statistical measures, such as cylindrical density, radial distribution function and bond orientational correlation function are computed. However, only limited micro-CT data of an almost ideally monodisperse system (glass spheres with $d = 63\ \mu\text{m}$ and $>95\%$ having polydispersivity of $\pm 4\ \mu\text{m}$) was used in their work with only 2,000 granules in the system, and only 359 granules used for statistics evaluation. In our work, we scan 19,892 polydisperse particles and statistically characterize 19,123 of them. We also show that a statistically optimal, and thus, the smallest possible unit cell consists of 1,082 spheres. Therefore, it is unlikely that 359 inclusions can be used for detailed morphology characterization, even though the monodisperse system likely requires fewer particles in the cell compared to our polydisperse one. Moreover, the selected statistical descriptors in [38] are limited to spherical geometries, whereas our concept based on n -order probability functions is general and easily applicable to general non-spherical shapes, such as crystals.

Different from the sampling based methods described above is the analysis based on Minkowski functionals [3, 32]. In three dimensions the functionals are related to the familiar measures of volume fraction, surface area, integral mean curvature and Euler characteristic, such as connectivity of pores. The heterogeneous material is represented by a Boolean model, where *overlapping grains* of various shape and sizes are used to reconstruct a material morphology. Although mathematically elegant, Minkowski functionals are better suited for two-phase porous media, such as soils or sedimentary rocks, due to the overlapping nature of the algorithm, and this elegance would be lost if a constraint on the particle inter-penetration were to be introduced.

Another approach introduced by Sundararaghavan and Zabaras [41] employs support vector machines for three-dimensional reconstruction of microstructures using limited statistical information available from planar images. Quintanilla and Max Jones used convex quadratic programming to model random media with Gaussian random fields [36].

In this work, we reconstruct a fully three-dimensional polydisperse medium, such as a heterogeneous solid propellant (Figure 1(a)), from three-dimensional tomographic data. We make no assumptions about the probability functions and discretize them numerically with high detail. The full probability spectrum, not just a single function, and the geometrically exact shape of particles, not just a voxel simplification, are used in our approach, yielding a multidimensional optimization problem with a highly complex objective function. Therefore, we perform the reconstruction on massively parallel computers. The present paper is an extension on our earlier work [24, 12], where two-dimensional periodic cells of trimodal fibrous composites and three-dimensional trimodal particulate composites were reconstructed, emphasizing the discrete nature of the entire probability spectrum and the proper geometrical particle shape represen-

tation. Here, we advance this methodology to the reconstruction of polydisperse particulate systems from rich three-dimensional tomographic data. Moreover, we compute the third-order probability functions, comment on their basic features and compare their characteristics for the tomographic data and our statistically equivalent cells. (Analysis of higher-order statistics was often omitted in the papers mentioned above.) Our codes, *Stat3D* and *Recon3D*, compute the statistics and reconstruct a representative unit cell (RUC) in parallel using Message Passing Interface (MPI), allowing for large domains to be efficiently evaluated (~ 66 million voxels), higher statistical moments to be computed accurately and the optimal computational domains to be generated quickly. Their parallel linear scalability has been shown in [12].

The remainder of the paper is organized as follows. In section 2, we discuss the sample preparation, the tomographic imaging and the image processing. For clarity of presentation, Section 3 describes the basic concepts of the probability functions, and details the morphology of the glass beads system to mimic a complex heterogeneous solid propellant (Figure 1(a)). In Section 4, we comment on the reconstruction procedure and compare the statistically representative unit cell and its statistics to the original tomographically observed medium with the statistical functions described in Section 3. Finally, some conclusions are drawn in Section 5.

2 Sample preparation, micro-CT scanning and image processing

We prepare a sample pack of spherical glass beads with average diameter of $44\text{ }\mu\text{m}$ in a 2.052 mm diameter scanning tube with one end blocked with epoxy (Figure 2). The glass beads were manufactured by *Crystal Mark Inc.* and 98% of the beads by weight are of diameters between $35\text{ }\mu\text{m}$ and $58\text{ }\mu\text{m}$. In order to prepare as uniform sample as possible, the beads were filtered in two steps. First, to remove hollow particles, the beads were submerged in a water bath and stirred. The hollow beads floating on the water surface were removed and the remaining ones were completely dried. Second, the non-spherical particles were removed by rolling them down on an inclined plane and removing beads that would not roll. After this preparation, we poured the filtered beads slowly into the tilted tube where a piece of thread was placed in the middle of the container before pouring. The thread was pulled out slowly. A similar method to randomize a pack by pulling a stick out was previously used by Aste *et al.* [4]. The tube was tapped on the side, the bottom and the top. The beads were slightly compressed from above during the tapping procedure.

The final sample is scanned by an *Xradia* micro-CT machine. The resulting resolution for this scan is $2.818\text{ }\mu\text{m}$ per voxel. The three-dimensional image reconstruction is shown in Figure 2(b). The 64-bit Quantification Pack for *Amira 5.2* is used to gather the information specific to

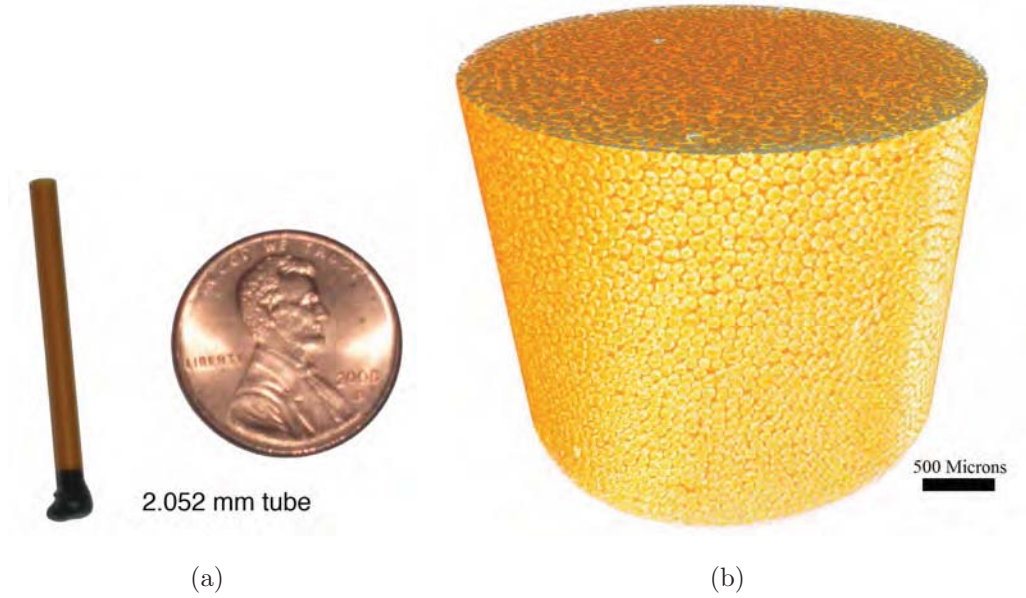


Figure 2: (Color online) a) Plastic tube used in scanning experiments. b) Polydisperse glass beads system.

each particle, i.e., the position of the particle center and the individual particle volume. To characterize this random medium, we first acquire a block ($1445.372 \times 1287.892 \times 789.106 \mu\text{m}^3$) from the core of the 3D scanned image in Figure 2. This represents $513 \times 457 \times 280 = 65,643,680$ voxels, which is 29.89 times higher resolution to that used in Fullwood *et al.* [16], where only $130 \times 130 \times 130 = 2,197,000$ voxels were analyzed.

The particles in contact are identified and separated through the separation procedure in *Amira*. This leads to a small volume removal. In the present case, the packing fraction from the original voxel pack is 0.574 before the separation and 0.553 after the separation, and thus we have 2.1% loss of the packing volume due to the separation. It is known that the packing fraction for a stable conglomeration of polydisperse spheres ranges from 0.55 to 0.64, and even up to 0.74 in the crystalline state [34]. Considering the role of the inter-particle forces like friction [14, 18], the packing fraction of our pack is well ranged. In Figure 3(b), most of the particles are now clearly separated from each other. At the final stage of the process 19,892 particles are identified in the region of observation.

Despite the physical filtering to remove extra small or hollow particles, we observe debris of very small sizes forming a peak around $4 \mu\text{m}$ diameter in Figure 3(a). However, the volume fraction of these particles is very small and can be neglected. Here we remove the beads with diameters less than $7 \mu\text{m}$. The remaining particles are separated into nine discrete bins as depicted in Figure 3(a). Next, a pack based on the sizes and the locations of the individual beads is generated and used to obtain the statistical properties of the actual material. In generating

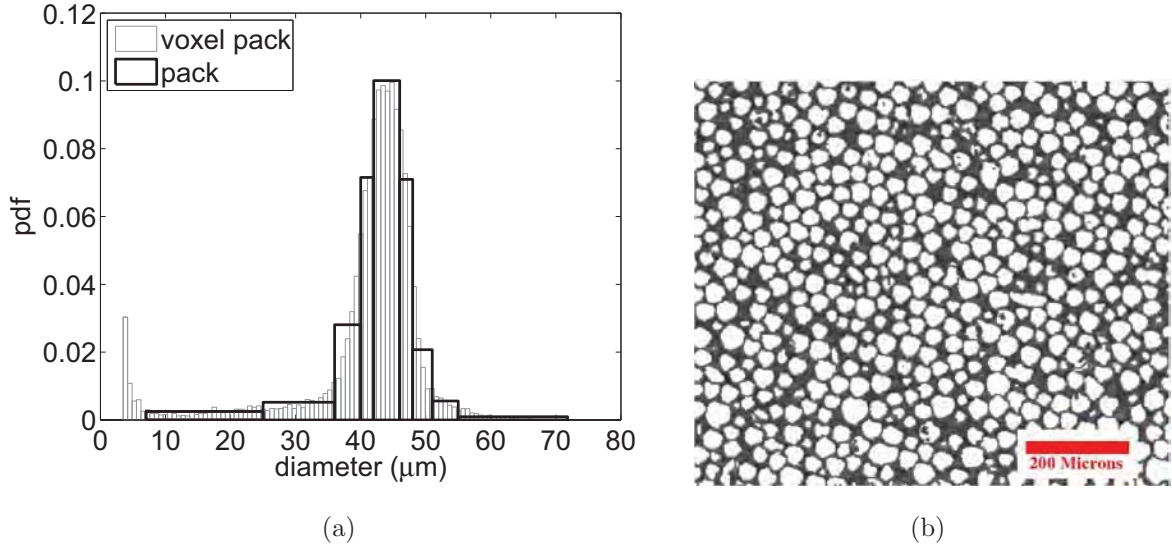


Figure 3: a) Probability density function and a new binning of the particle sizes. Nine discrete bins are used to capture the original distribution. Note the debris for $d < 7 \mu\text{m}$. b) Two-dimensional slice through the voxel pack. Particle separation is done in *Amira*.

the pack, we replace all the beads with spheres, which causes 0.05% overall particle overlap. The overlap is calculated as $V_o/V_p \times 100$, where V_o is the volume of overlap between particles and V_p is the total volume of the particles. The final pack has 19,123 spherical particles. We note that the tomographic pack after the image processing is called the ‘voxel pack’ and the pack generated based on the voxel data by replacing the inclusions with spheres is called the ‘pack’ hereafter.

3 Statistical morphology of composites

It is generally accepted that the effective properties of a heterogeneous material do not depend only on the properties of each phase, but also on the interactions between the material phases. In a statistical sense, both the volume fraction of each phase and various correlation functions between the phases are fundamental in determining the material properties [44, 45]. In this section, we discuss one-, two- and three-point probability functions to characterize the morphology of a heterogeneous medium [5, 45].

3.1 Probability functions

To describe the probability functions, we consider a phase indicator function at a position $\mathbf{x}$ in a sample α of an ensemble space $\mathcal{E}$:

$$\chi_r(\mathbf{x}; \alpha) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ in phase } r, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The ensemble average is given by

$$\overline{\chi_r(\mathbf{x})} = \int_{\mathcal{E}} \chi_r(\mathbf{x}; \alpha) p(\alpha) d\alpha, \quad (2)$$

where $p(\alpha)$ is a probability density function. The n -point probability function, $S_{r_1 r_2 \dots r_n}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, is defined as

$$S_{rs\dots q}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) = \overline{\chi_r(\mathbf{x}_1) \chi_s(\mathbf{x}_2) \dots \chi_q(\mathbf{x}_n)}, \quad (3)$$

and it represents the probability of finding phases $r, s, \dots, q$ at points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, simultaneously.

The probability functions for a heterogeneous material are spatially complex in general. If the material satisfies ergodicity, statistical homogeneity and isotropy, the ensemble average can be replaced by the volume average and the probability functions are translation and direction invariant [5]. With these three simplifications, the one-point probability function is reduced to the volume fraction, and the higher-order probability functions depend on the distance only:

$$\begin{aligned} S_r(\mathbf{x}) &= c_r, \\ S_{rs}(\mathbf{x}, \mathbf{x}') &= S_{rs}(|\mathbf{x} - \mathbf{x}'|), \\ S_{rsq}(\mathbf{x}, \mathbf{x}', \mathbf{x}'') &= S_{rsq}(|\mathbf{x} - \mathbf{x}'|, |\mathbf{x} - \mathbf{x}''|), \end{aligned} \quad (4)$$

where c_r is the volume fraction of the phase r . We also observe the limit cases of the two- and three-point probability functions which can be expressed as

$$S_{rs}(\mathbf{x}, \mathbf{x}') \rightarrow \begin{cases} c_r \delta_{rs} & \text{if } |\mathbf{x} - \mathbf{x}'| \rightarrow 0, \\ c_r c_s & \text{if } |\mathbf{x} - \mathbf{x}'| \rightarrow \infty, \end{cases} \quad (5)$$

$$S_{rsq}(\mathbf{x}, \mathbf{x}', \mathbf{x}'') \rightarrow \begin{cases} c_r \delta_{rs} \delta_{rq} & \text{if } |\mathbf{x} - \mathbf{x}'|, |\mathbf{x} - \mathbf{x}''| \rightarrow 0, \\ S_{rq}(\mathbf{x}, \mathbf{x}'') & \text{if } |\mathbf{x} - \mathbf{x}'| \rightarrow 0, \\ c_r c_s c_q & \text{if } |\mathbf{x} - \mathbf{x}'|, |\mathbf{x} - \mathbf{x}''| \rightarrow \infty, \end{cases} \quad (6)$$

where δ_{rs} is the Kronecker delta.

3.2 Morphology of a pack

To obtain the statistical properties of the pack that is based on the voxel pack as discussed in Section 2, we discretize the probability space. The probability functions are numerically sampled through a Monte Carlo-like method. A spherical sampling template depicted in Figure 4 is used. This template is especially advantageous for isotropic cases as pointed out in [40].

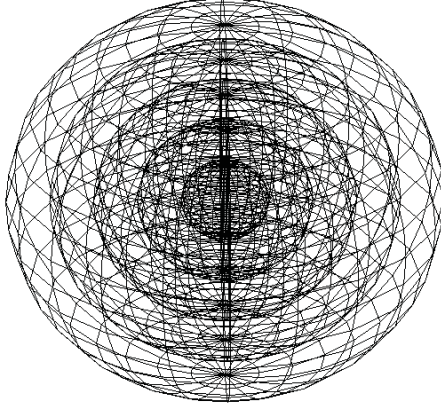


Figure 4: Three-dimensional sampling template with 5 radial points and 20 circumferential points, for illustration.

The probability functions described in Section 3.1 are computed by our parallel code, *Stat3D*, allowing for a large domain to be efficiently evaluated [12]. For the present pack, we sample the one- and two-point probability functions by throwing a sampling template with 1,000 radial points and 20 circumferential points 500,000 times. These sampling parameters have been selected after a convergence study to assure the spatial convergence.

The one-point probabilities (volume fractions) of the pack are shown in Figure 5(b). The nine bins are based on those of the probability density function plotted in Figure 3(a) and each bin is assigned a different mode number (1 – 9). For the total volume to be conserved, one has $\sum_1^N c_i = 1$, where N represents the total number of phases. In our case, $N = 10$ including the matrix (m) (voids in the prepared pack). We evaluate $c_m = 0.46$ numerically for our particulate medium, which is different from $1 - \sum_1^{N_p} c_i$ by 0.2% where N_p is the number of the particle modes ($N_p = N - 1$). However, considering the image resolution ($2.818 \mu\text{m}$ voxel) and the error due to the separation procedure (2.1% in the total packing fraction), the volume conservation error is well controlled.

Figure 6(a) shows the two-point probability function, S_{mm} , zoomed at the core in the four-dimensional space. The colored sphere shows the probability of finding the phase m (matrix), when its center lies in the matrix as well. We find that the two-point probability functions of the pack satisfy isotropy with the standard deviation, $\sigma_{S_{rs}}$, remaining less than 1% of the

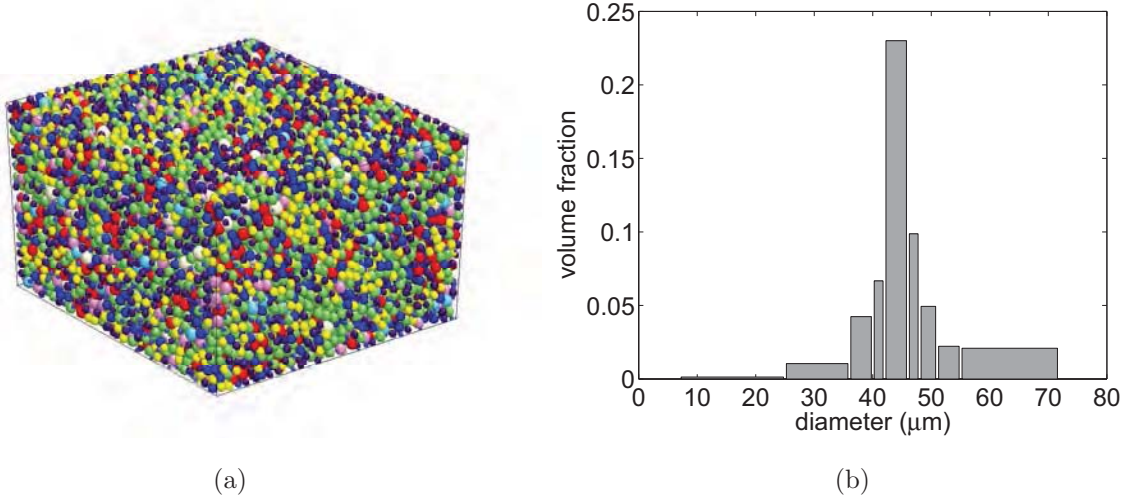


Figure 5: (Color online) a) Polydisperse pack obtained from the voxel pack by replacing voxel-based inclusions with spheres. Each color of particle in the pack belongs to a different mode according to pdf shown in Figure 3(a). b) The corresponding one-point probabilities (volume fractions).

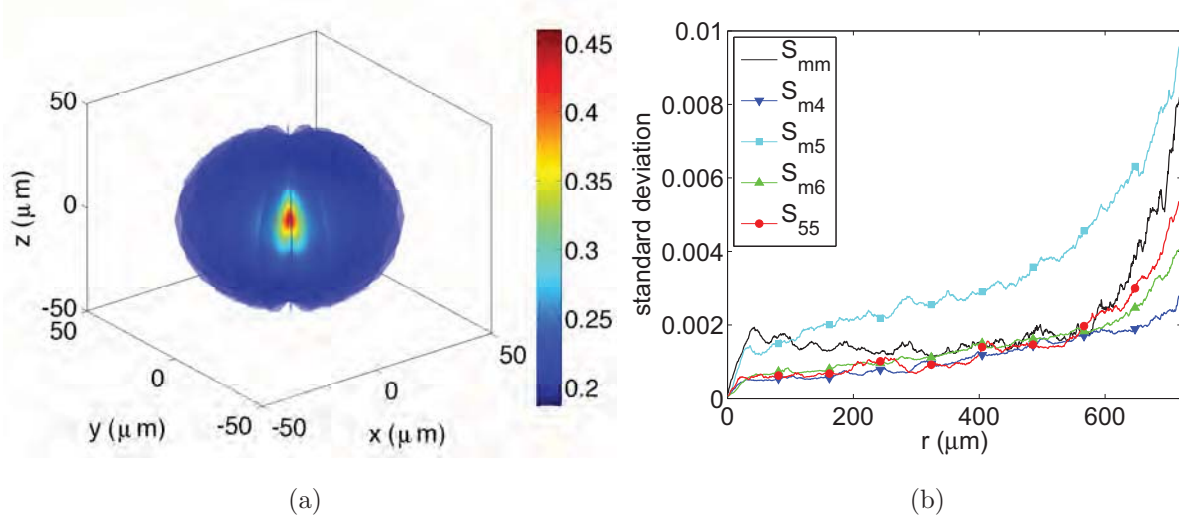


Figure 6: (Color online) a) Full two-point probability function S_{mm} . Observe the sphericity of S_{mm} with the value at the origin converging to the volume fraction and value at infinity converging to c_m^2 . b) The corresponding standard deviation of isotropy, $\sigma_{S_{rs}}$, for selected second-order probability functions. Although the standard deviation is small over the whole x -range, observe the increase of $\sigma_{S_{rs}}$ due to the wall effect.

magnitude of the actual probability function, shown in Figure 6(b). The exponential behavior of the deviation at the large radii can be attributed to the wall effect, which produces a significantly different packing structure near the boundary of the container as pointed out in [43, 28]. Based

on the observation of isotropy, we can average the two-point probability functions over all the possible orientations since it is direction-invariant. Statistical homogeneity and ergodicity of the particulate composites under consideration have been verified in [10]. In Figure 7, the isotropic

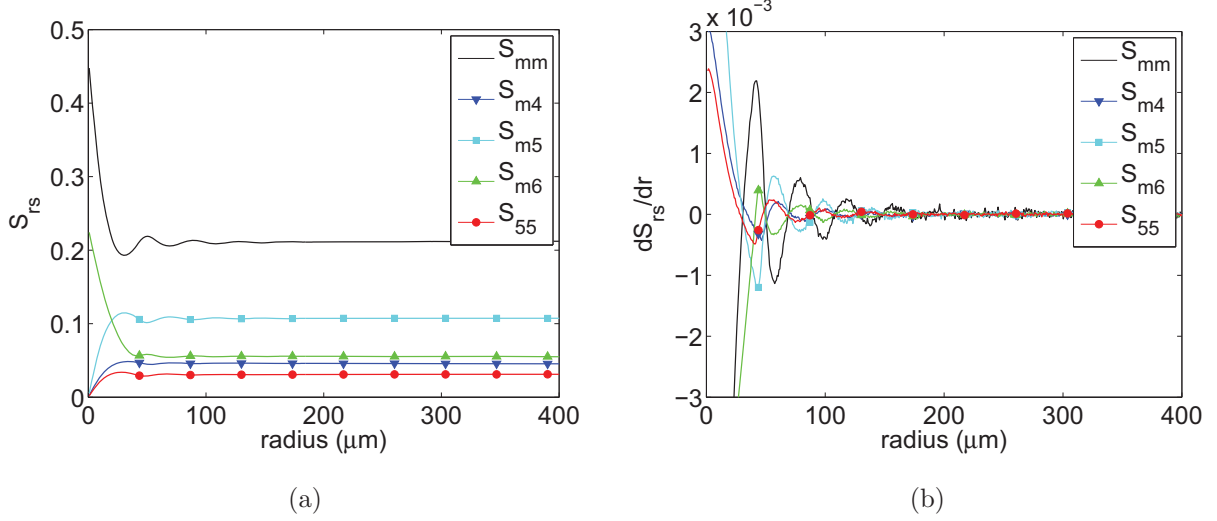


Figure 7: (Color online) a) Isotropic two-point probability functions S_{rs} computed by radial averaging of complete probability functions in Figure 6(a). Only selected S_{rs} functions are shown for clarity. Note that probability functions are plotted for radius $\leq 400 \mu\text{m}$ to make their behavior close to the origin more visible. However, full probability functions are computed for radius $723 \mu\text{m}$ (one half of the longest edge of the pack in Figure 5(a)). Detailed structure of S_{rs} governed by spatial interactions of particles and their corresponding diameters can be seen. b) Gradients of S_{rs} . The gradients of the two-point probability functions show where the functions reach the asymptotic values. This distance provides a characteristic material length scale of the second-order, and we use it to construct the size of the RUC.

two-point probability functions and their gradients are shown. We note that the isotropic two-point probability functions saturate when the radius is greater than $200 \mu\text{m}$, which provides a characteristic material length scale, i.e., the optimal size of an unit cell: twice the radius, or $400 \mu\text{m}$. Note that there are one-hundred two-point probability functions in our computations, since we have ten different phases (the nine particle modes and the matrix). Figure 7 shows only five selected second-order probability functions. However, the trends are the same for the other second-order probability functions.

To validate the probability functions evaluated on the pack, we use the limit cases of the second-order probability functions in Equation (5), i.e., the two-point probability functions are examined for convergence to one-point probability functions in the limit cases, and for numerical error due to sampling. In order to quantify the physical errors for the pack, we define two error

functions for two-point probabilities:

$$\varepsilon_{S_{rs}(0,0)} = |S_{rs}(0,0) - c_r \delta_{rs}|, \quad (7)$$

$$\varepsilon_{S_{rs}(0,\infty)} = |S_{rs}(0,\infty) - c_r c_s|. \quad (8)$$

In our analysis, we have maximum errors 0.001 and 0.004 over the one-hundred two-point probability functions, measured by expressions (7) and (8), respectively. Given that $\varepsilon_{S_{rs}(0,0)}$ and $\varepsilon_{S_{rs}(0,\infty)}$ range in $[0, 1]$, these errors are negligible. The numerical error is calculated by

$$\varepsilon_{S_{sr}|S_{rs}} = \frac{\|S_{rs} - S_{sr}\|_{L_2^D}}{\sum_{r=1}^N \sum_{s=1}^N \|S_{rs}\|_{L_2^D} / N^2}, \quad (9)$$

where $\|\cdot\|_{L_2^D}$ is the discrete L_2 norm and N represents the number of phases. Since S_{rs} and S_{sr} are symmetric, $\varepsilon_{S_{sr}|S_{rs}}$ represents a relative numerical error. For the current pack, the maximum value of the error based on Equation (9) is 0.035 and the mean is 0.006. This error criterion provides confidence in the spatial convergence of our analysis.

For many physical processes, the higher-order statistics play an important role in capturing the interactions between particles, such as a case where a small particle lies between two big particles. For example, such a scenario helps to promote interfacial debonding. Therefore, we also investigate three-point probability functions. As expected, the three-point probability functions tend to be more sensitive to the geometrical properties of a pack than the two-point probability functions. We recall that the three-point probability functions depend on two vectors in general. Thus, the function $S_{rsq}(\mathbf{x} - \mathbf{x}', \mathbf{x} - \mathbf{x}'')$ is defined in a seven-dimensional space, considering two vectors in three-dimension and the function value itself. Since the numerical evaluation of the full three-point probability function is computationally expensive, we use a semi-isotropic assumption in terms of the orientation of a plane determined by the two randomly generated vectors. This assumption reduces the space dimensionality from seven to four, since $S_{rsq}(\mathbf{x} - \mathbf{x}', \mathbf{x} - \mathbf{x}'')$ becomes $S_{rsq}(|\mathbf{x} - \mathbf{x}'|, |\mathbf{x} - \mathbf{x}''|, \theta)$, where θ is the angle between the two vectors $\mathbf{x} - \mathbf{x}'$ and $\mathbf{x} - \mathbf{x}''$. We again use the sampling template, shown in Figure 4, and average on planes generated by discrete rays (semi-isotropy). Note that for the perfectly statistically isotropic material, the three-point probability functions are angle independent, except when they degenerate to the lower-order probability functions:

$$S_{rsq} \rightarrow \begin{cases} S_{rs} \delta_{rq} & \text{on } r_1 - \text{axis} \\ S_{rq} \delta_{rs} & \text{on } r_2 - \text{axis} \\ S_{rs} \delta_{sq} & \text{on } r_1 = r_2, \end{cases} \quad (10)$$

where $r_1 = |\mathbf{x} - \mathbf{x}'|$ and $r_2 = |\mathbf{x} - \mathbf{x}''|$.

There are one thousand different three-point probability functions for a ten-phase medium. However, due to the symmetries, $S_{rs} = S_{sr}$ and $S_{rsq} = S_{rqs}$, etc., several functions can be

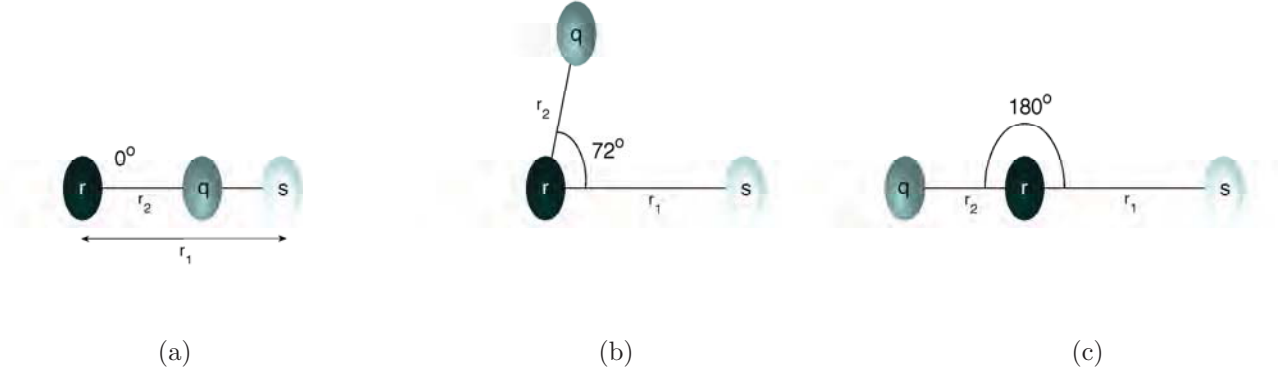


Figure 8: Alignments of three points in phases r , s and q . a) $\theta = 0^\circ$. b) $\theta = 72^\circ$. c) $\theta = 180^\circ$.

omitted. In this work, we do not investigate all third-order probabilities, and rather focus on the generic findings that are typical for the third-order statistics. Thus, we present symmetric and unsymmetric types of the three-point probability functions. To compute the third-order statistics of the pack, we use 50,000 throws of a template with 1,000 radial and 20 circumferential points. Sampling accuracy is again determined by the limit cases in Equation (10). Figure 8 shows the different arrangements of the three randomly generated points in phases r , s and q for S_{rsq} at the three different angles, 0° , 72° and 180° , for instance. For a symmetric case, S_{355} is selected to represent the S_{rss} -type of three-point probability functions. Figure 9 shows S_{355} at the three angles mentioned above. Since s and q are the same in this case, S_{355} comes to zero on both r_1 - and r_2 -axes, while it degenerates to S_{35} for $r_1 \equiv r_2$ (Equation (10)). Figure 10 shows an unsymmetric three-point probability function S_{775} which goes to zero on r_1 -axis and $r_1 \equiv r_2$ while it converges to S_{75} on r_2 -axis. In Figure 9 and Figure 10, the widths of the three characteristic bands at $\theta = 0^\circ$ depend on the sizes of the particles in the bins r , s and q of S_{rsq} . Even though the present pack is almost isotropic based on the two-point probability functions, the three-point probability still captures some anisotropic aspects of the pack, showing some dependency on angles. We will return to elaborate on the anisotropy and on the degeneracy of the third-order statistics in section 4.2.

4 Reconstruction of a representative unit cell

In this section, we proceed to construct the representative unit cell (RUC) that retains the statistical characteristics of the pack shown in Figure 5(a). The two main steps in the reconstruction are to find the length scale of the RUC and to optimally locate the particles with a minimum (or no) particle overlap. Finally, the statistical results of the cell are compared with

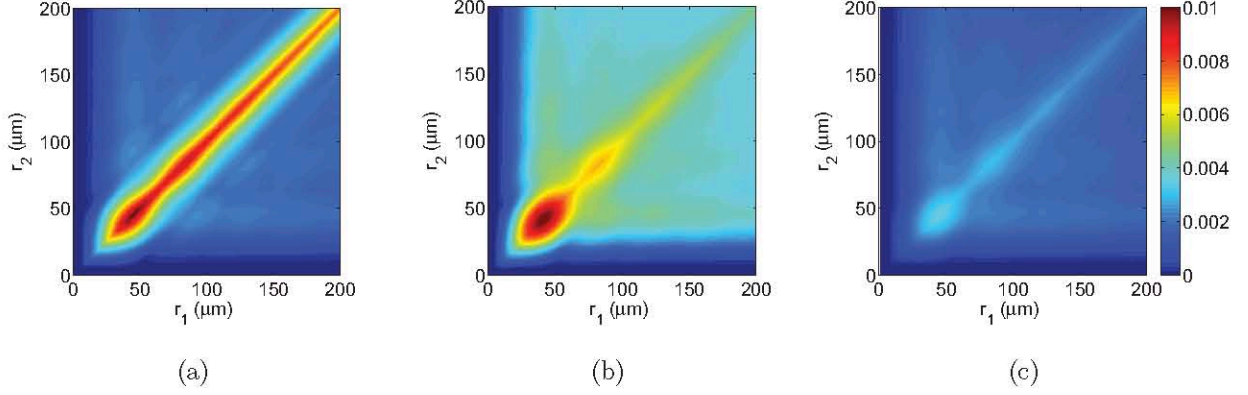


Figure 9: (Color online) Three-point probability function S_{355} of the pack depicted in Figure 5(a). A diagram of S_{rsq} computation is shown in Figure 8. We use a semi-isotropic assumption in evaluation of S_{rsq} . a) S_{355} for $\theta = 0^\circ$. Note the symmetric degeneracy of $S_{355} \rightarrow S_{35}$ on $r_1 = r_2$ ray for $\theta = 0^\circ$. b) S_{355} for $\theta = 72^\circ$. c) S_{355} for $\theta = 180^\circ$. A certain degree of anisotropy in third-order statistics can be observed for b) and c), since for a perfectly isotropic medium $S_{rsq}(|\mathbf{x} - \mathbf{x}'|, |\mathbf{x} - \mathbf{x}''|)$ is an angle θ independent except for degenerate cases (10) as shown in a).

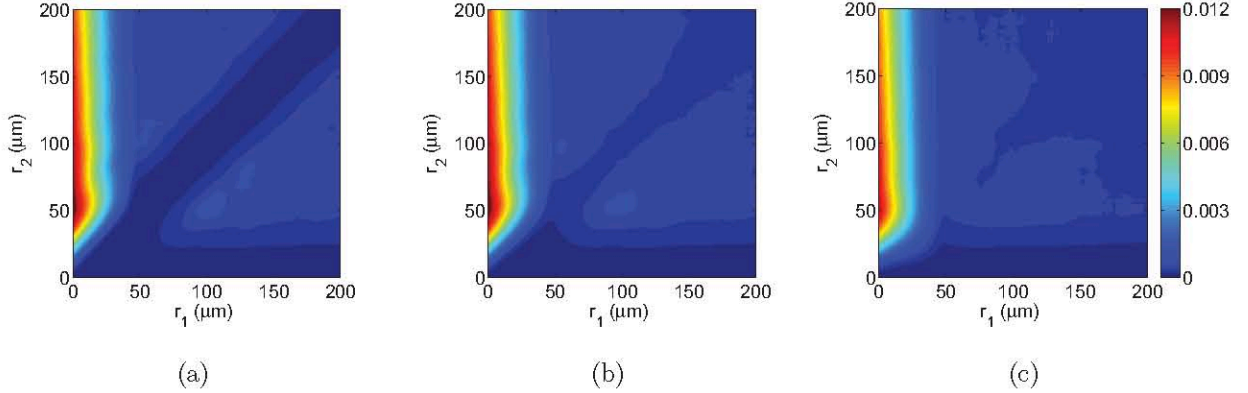


Figure 10: (Color online) Three-point probability function S_{775} of the pack depicted in Figure 5 (a). a) S_{775} for $\theta = 0^\circ$. Note the non-symmetric degeneracy of $S_{775} \rightarrow S_{75}$ on r_2 -axis for $\theta = 0^\circ$. b) S_{775} for $\theta = 72^\circ$. c) S_{775} for $\theta = 180^\circ$.

those of the pack.

4.1 Numerical method

First, let us recall the two-point probability functions in Figure 7. The two-point probability functions depend only on the distance since the material satisfies both the statistical isotropy and homogeneity assumptions. We also observe that the functions start to reach the asymptotic values at $200 \mu\text{m}$ which is a good candidate for the size of the unit cell (h_0).

With this initial guess on the length scale of the cell, we minimize two objective functions which reflect the differences of the one- and two-point probability functions between the pack and the cell. For $0 \leq h \leq h_0$, the first objective function is defined as

$$\mathcal{F}_1(h) = \sqrt{\sum_{i=1}^{N_p} (c_i^p - c_i^c)^2} = \sqrt{\sum_{i=1}^{N_p} \left(c_i^p - \frac{4\pi r_i^3 n_i}{3h^3} \right)^2}, \quad (11)$$

where c_i^p , c_i^c , r_i and n_i are the volume fraction of the pack, the volume fraction of the cell, the particle radius and the total number of particles in phase i , respectively. By finding the local minimum of the first objective function, we can determine the optimal length scale (h_c) of the cell as

$$\frac{d\mathcal{F}_1(h)}{dh} = 0 \Rightarrow h_c = \left(\frac{4\pi \sum_{i=1}^{N_p} n_i^2 r_i^6}{3 \sum_{i=1}^{N_p} c_i N_i r_i^3} \right)^{1/3}. \quad (12)$$

To construct the second objective function, we first consider a functional which represents the difference of the two-point probability functions between the pack and the cell,

$$\mathcal{F}_2(\mathbf{x}_n) = \sum_{r=1}^N \sum_{s=1}^N \|S_{rs}^p - S_{rs}^c\|_{L_2^D}, \quad (13)$$

where $\mathbf{x}_n$ is the position vector of all particle centers $\mathbf{x}_i$. To avoid the overlap between particles, we impose a constraint on Equation (13):

$$\sqrt{(\mathbf{x}_i - \mathbf{x}_j)^2} \geq r_i + r_j, \quad \forall i = 1, \dots, n-1, \quad \forall j = i+1, \dots, n, \quad (14)$$

where n is the total number of particles in the cell ($n = \sum_{i=1}^{N_p} n_i$), and r_i and r_j are the radii of i -th and j -th particles, respectively. The number of particles in the cell is calculated based on the ratio of the volume of the pack and that of the cell from the initial guess. Note that only a whole particle, not its fraction, can be added into a particular particle mode in the cell.

The raw objective function and the constraint can be combined to construct an overall objective function or a fitness function, $\mathcal{F}$, that yields

$$\mathcal{F} = \frac{\sum_{r=1}^N \sum_{s=1}^N \|S_{rs}^p - S_{rs}^c\|_{L_2^D}}{\max_{\mathcal{P}_0} \sum_{r=1}^N \sum_{s=1}^N \|S_{rs}^p - S_{rs}^c\|_{L_2^D}} + p. \quad (15)$$

The denominator in the first term of Equation (15) represents the value of the objective function $\mathcal{F}_2$ of the worst sample pack in the initial population $\mathcal{P}_0$ and p is a penalty function in the range $[0, 1]$ that is used to enforce the constraint (14). We also use the mass-spring dynamic mutation operator to eliminate the overlap as discussed in [12]. This fitness function is multi-modal with multiple local minima. To minimize this function, we use the augmented simulated annealing technique, a stochastic optimization method based on the principle of evolution, such as genetic algorithms (GA) combined with simulated annealing (SA), introduced in [33]. The genetic algorithm is briefly described in Algorithm 1.

```

1   $g = 0$ 
2  generate and evaluate population  $\mathcal{P}_g$  of size  $N_i$ 
3  while (not termination-condition) {
4      select  $m$  individuals to  $\mathcal{M}_g$  from  $\mathcal{P}_g$       (apply sampling mechanism)
5      alter  $\mathcal{M}_g$                                 (apply genetic operators)
6      create and evaluate  $\mathcal{P}_{g+1}$  from  $\mathcal{M}_g$       (insert  $m$  new individuals into  $\mathcal{P}_{g+1}$ )
7       $g = g + 1$ 
8  }
```

Algorithm 1: Principle of genetic algorithm.

For our optimization problem, the population $\mathcal{P}_g$ in the Algorithm 1 becomes a family of possible cell configurations with particles of fixed diameters, that are distributed according to the re-binned pdf (Figure 3), placed inside the cell. The unknowns are positions of the particle centers $\mathbf{x}_i$ (see Eq. (13)) in the Algorithm 1. These methods were previously implemented into our code, *Recon3D*, that reconstructs the unit cell in parallel, preserving the end-to-end parallelism and allowing for the optimal computational domain to be generated quickly [12]. See [12, 24] for more details on the optimization process.

4.2 Comparison of statistical properties between pack and cell

To establish the robustness of the proposed reconstruction method, we run *Recon3D* on 2048 CPUs and reconstruct five different statistically optimal cells. We note now that periodic conditions are enforced on the cell boundary, i.e., all particles intersected by the bounding box have periodic replicas (two if a particle is on a face, four if a particles in on an edge and eight if a particle sits in a corner). Periodic boundary conditions are extensively used in computational studies due to their simplicity, reasonable physical relevance and deep mathematical foundation [26]. Thus, a non-periodic voxel medium is mapped to a statistically equivalent periodic RUC. All the runs consist of a population of $N_i = 512$ individuals and there are $m = 16$ individuals in the mating pool, $\mathcal{M}$ (pool where genetic operators are applied, see Algorithm 1). During the initialization of the genetic algorithm, one-half of the individuals in the population is generated randomly, i.e., all particles with fixed diameters are placed in the cell of dimensions h_c without overlap considerations. The other half is generated using ballistic deposition, i.e., a randomly deposited particle is checked for overlap and this process is repeated until all particles are placed successfully, allowing only a certain degree of overlap. Each reconstruction

Pack #	$\mathcal{F}_2$ before GA			$\mathcal{F}_2$ after GA			# Iteration	Overlap
	Min	Mean	Max	Min	Mean	Max		
Run 1	4.035	7.253	10.963	0.520	0.606	0.839	578	0.000 %
Run 2	4.202	7.600	11.348	0.581	0.774	1.163	942	0.001 %
Run 3	3.993	7.537	11.179	0.590	0.773	1.253	807	0.002 %
Run 4	3.989	7.335	10.801	0.587	0.652	0.793	1452	0.000 %
Run 5	4.075	7.420	11.128	0.535	0.574	0.699	1726	0.000 %

Table 1: Comparison of the raw fitness $\mathcal{F}_2$ in Equation (13) before and after performing the genetic algorithm for the five reconstructed cells. The number of GA iterations for convergence and the overall particle overlap are also shown.

simulation is stopped when the maximum number of iterations (10,000) is reached or when the best individual set is not improved for 300 generations. In Table 1, we compare the raw fitness $\mathcal{F}_2$ before and after running the genetic algorithm. As one can see, a large improvement in the fitness after optimization has been obtained. The fitness after the genetic algorithm was improved by 91 % on average. The final number of iterations needed for convergence and the overall particle overlap are also shown.

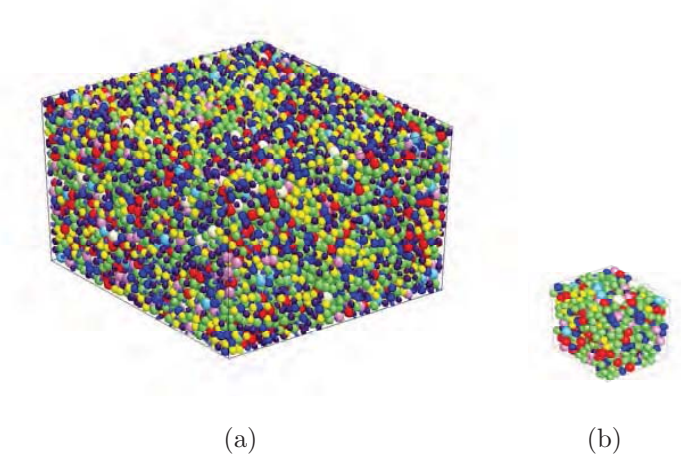


Figure 11: (Color online) a) The pack contains 19,123 particles in $1445.372 \times 1287.892 \times 789.106 \mu\text{m}^3$. b) The reconstructed cell is composed of 1,082 particles in $399.632 \times 399.632 \times 399.632 \mu\text{m}^3$.

Figure 11 shows the comparison of the dimensions of the pack and one particular RUC. The pack includes 19,123 particles in $1445.372 \times 1287.892 \times 789.106 \mu\text{m}^3$ as mentioned previously, while the unit cell has 1,082 particles in $399.632 \times 399.632 \times 399.632 \mu\text{m}^3$. These numbers highlight the substantial reduction of the computational space.

We now assess the quality of the reconstruction for both the first- and the second-order

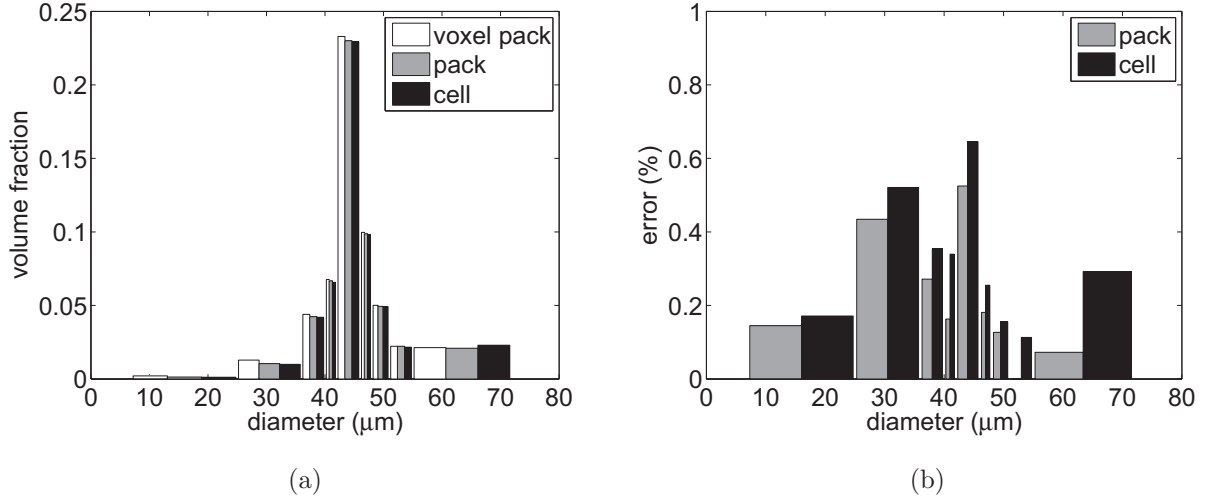


Figure 12: a) Volume fractions of each bin for the voxel pack, the pack and the cell. The voxel pack values are computed directly from voxels, whereas the pack and cell values are computed by statistical sampling. b) Relative error in volume fraction for each bin as given by Eq. (16). Errors in volume fractions between the voxel pack and the pack, which are the consequence of mapping voxel based inclusions to spheres, are smaller than those between the voxel pack and the cell, since the cell is geometrically smallest possible object for a given finite number of whole particles.

probability functions. First, we average the one-point probability functions over the five cells. The averaged one-point probabilities of the five cells are compared with those of the voxel pack and the pack in Figure 12(a). The maximum standard deviation for the sampled volume fractions is less than 0.003 (bin 9) and is not displayed. The corresponding relative error (ε_i^q) of the volume fraction in the i th-bin between the voxel pack and the pack and/or the cell is calculated by

$$\varepsilon_i^q(\%) = \frac{|c_i^v - c_i^q|}{\sum_{i=1}^{n_{pm}} c_i^v} \times 100, \quad (16)$$

where c_i^v is the i -th bin's volume fraction of the voxel pack and q is either p (pack) or c (cell). As seen in Figure 12(b), the error is less than 0.8% for both the pack and the cell.

Next, we focus on the second-order statistics. One of the error measures is provided directly by the objective function $\mathcal{F}_2$ itself (Equation (13)). Note that we optimized the full probability spectrum composed of one-hundred second-order probability functions for this example. The comparison of the isotropic two-point probability functions between the pack and the cell is presented in Figure 13. The error bars in Figure 13 represent the standard deviations of the two-point probability functions computed on the five cells. Note that the error bars contain both the numerical (sampling) and the physical (statistics of five cells) error components. To assess the relative error, the difference of the two-point probability functions between the pack

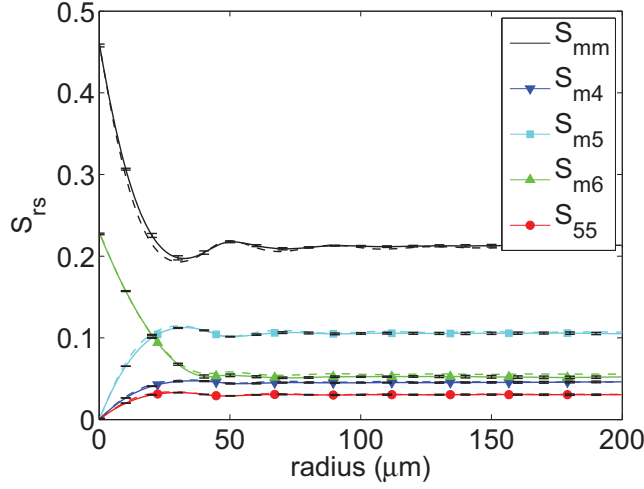


Figure 13: (Color online) Comparison of the averaged two-point probability functions between the pack (dotted lines) and the cell (solid lines). Average values are computed from five optimization runs as given in Table 1. Note the very small error bars. Only selected probability functions are plotted.

and the cell is calculated by

$$\mathcal{E}(\%) = \frac{\sum_{r=1}^N \sum_{s=1}^N \|S_{rs}^p - S_{rs}^c\|_{L_2^D}}{\sum_{r=1}^N \sum_{s=1}^N (\|S_{rs}^p\|_{L_2^D} + \|S_{rs}^c\|_{L_2^D}) / 2} \times 100, \quad (17)$$

where S_{rs}^c is the averaged isotropic two-point probability function over the five cells. In the present case we have excellent agreement with only 0.0387% error. Assuming the uniform distribution of the error for all one hundred second-order probability functions, the error measure (13) yields a very small inaccuracy of ~ 0.006 between S_{rs}^p of the pack and S_{rs}^c of the cell over the length of ~ 200 microns. This is a remarkable resolution for the full second-order probability spectrum, manifested by almost coincident lines in Figure 13, and this would hold for all the other second-order probability functions as well.

Even though the unit cell is reconstructed based on the one- and two-point probability functions, we also investigate the three-point probability functions. The three-point probability function of the cell, S_{rsq}^c , is compared with that of the pack, S_{rsq}^p , by calculating the difference, $|S_{355}^p - S_{355}^c|$ for $(r, s, q) = (3, 5, 5), (7, 7, 5)$ corresponding to Figure 9 and Figure 10. Figure 14 and Figure 15 show the differences at the three different angles $0^\circ, 72^\circ, 180^\circ$ depicted in Figure 8 for these two cases, respectively. The difference of the three-point probability between the pack and the cell is noticeable in comparison to the good agreement of the lower-order probability functions, which indicates that optimization of the three-point probability functions would be desirable.

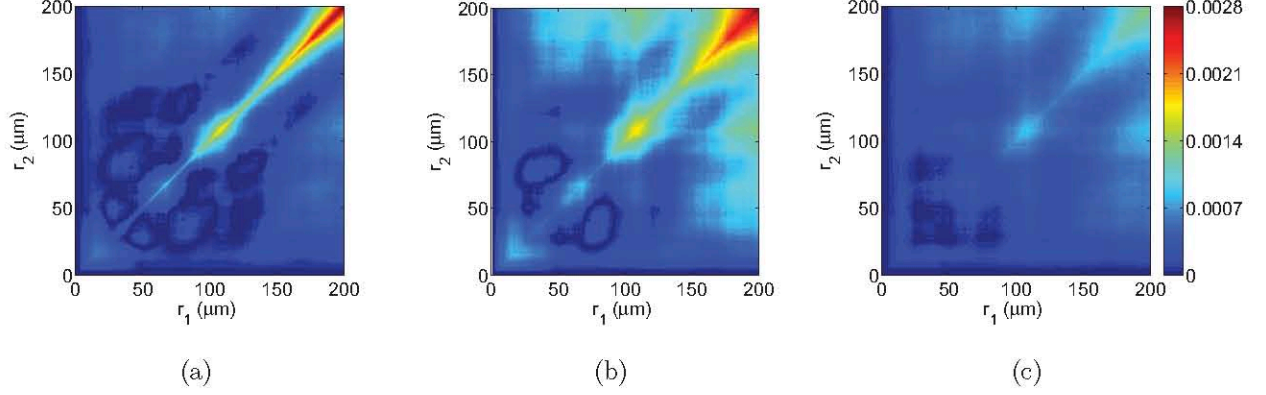


Figure 14: (Color online) The difference, $|S_{355}^p - S_{355}^c|$, between the pack and the cell for several angles θ . a) The difference for $\theta = 0^\circ$. b) The difference for $\theta = 72^\circ$. c) The difference for $\theta = 180^\circ$. Note that the saturation value for S_{355} is ~ 0.01 on $r_1 = r_2$ ray at $\theta = 0^\circ$, and thus, the errors depicted above are of $\sim 28\%$. Also note that the largest errors are obtained at the unit cell boundaries for $r_1 \rightarrow h_c$ and $r_2 \rightarrow h_c$, where h_c is the size of the RUC. The degenerate case for S_{355} is depicted in Figure 16.

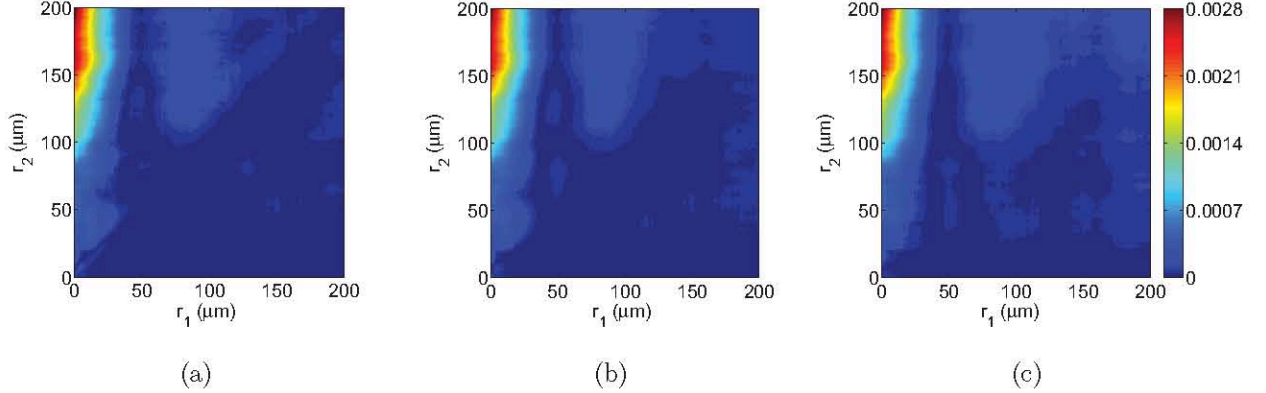


Figure 15: (Color Online) The difference, $|S_{775}^p - S_{775}^c|$, between the pack and the cell for several angles θ . a) The difference for $\theta = 0^\circ$. b) The difference for $\theta = 72^\circ$. c) The difference for $\theta = 180^\circ$. Note that the saturation value for S_{775} is ~ 0.012 on $r_1 = 0$ ray at $\theta = 0^\circ$, and thus, the errors depicted above are of $\sim 23\%$. Also note that the largest errors are obtained at the unit cell boundaries for $r_1 = 0$ and $r_2 \rightarrow h_c$, where h_c is the size of the RUC. The degenerate case for S_{775} is depicted in Figure 17.

In Figure 16 and Figure 17, we further compare the three-point probability functions, S_{355} and S_{775} , with the two-point probability, functions S_{35} and S_{75} . We recall the convergence of the three-point probability function at the second limit case in Equation (6), which states that the third-order function degenerates to the second-order one at the angle 0° , where $r_1 = r_2$ for

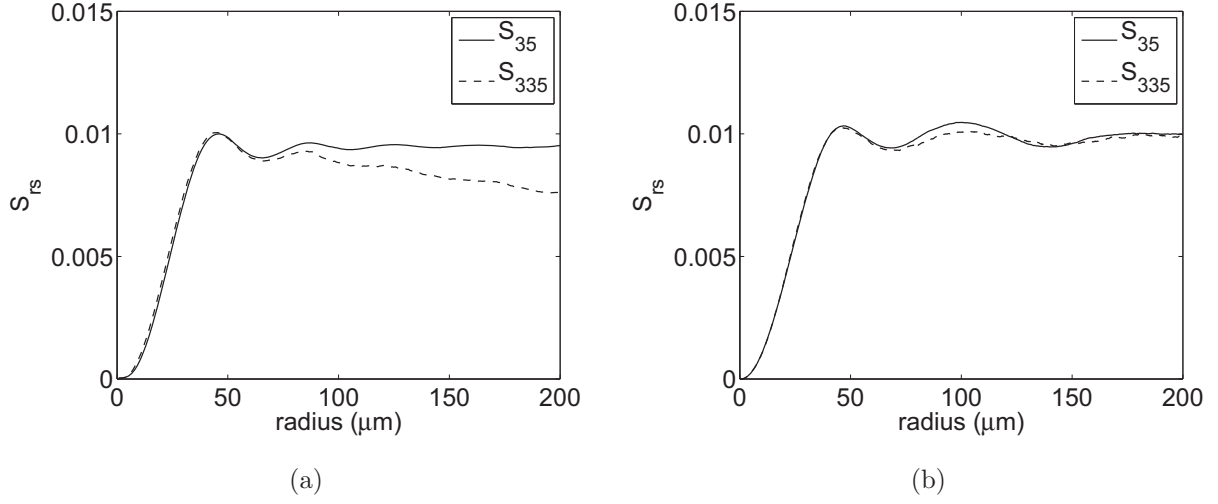


Figure 16: Comparison between the two-point probability function S_{35} (solid line) and the degenerated S_{335} (dashed line) on $r_1 = r_2$ ray at $\theta = 0^\circ$. a) Comparison computed for the pack. b) Comparison computed for the cell. Note that S_{35} is computed to be statistically isotropic, whereas S_{335} is not. Thus, for the pack (a) we observe an anisotropy in the third-order, whereas the RUC (b) is fairly isotropic.

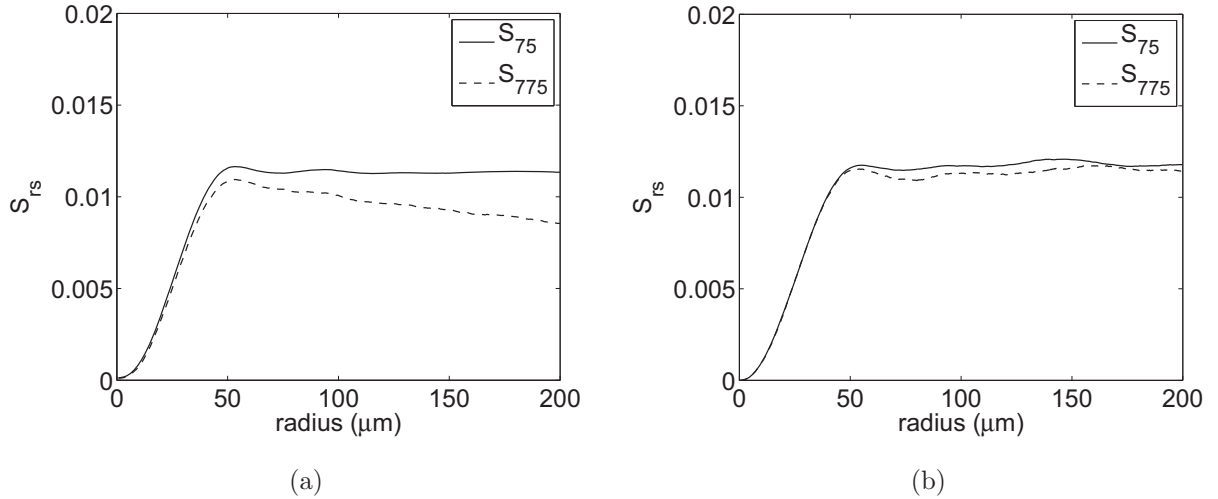


Figure 17: Comparison between the two-point probability function S_{75} (solid line) and the degenerated S_{775} (dashed line) on r_2 -axis at $\theta = 0^\circ$. a) Comparison computed for the pack. b) Comparison computed for the cell. Note that S_{75} is computed to be statistically isotropic, whereas S_{775} is not. Thus, for the pack (a) we observe an anisotropy in the third-order, whereas the RUC (b) is fairly isotropic.

S_{355} and $r_1 = 0$ for S_{775} .

The two- and the three-point probability results produced for the pack do not come to agreement, whereas those for the cell do. This discrepancy can be attributed to the periodicity imposed on the cell boundary and lack thereof in the case of the pack, but also to the strict enforcement of the statistical isotropy, given by Equation (13), in the cell reconstruction, which is not enforced for the pack statistics. To investigate this discrepancy further, let us recall that we compute S_{rsq} by assuming the semi-isotropic assumption leaving θ as an extra variable to account for anisotropy, while S_{rs} is averaged over all possible orientations in the cell reconstruction. Figures 16(a) and 17(a) indicate that the pack is anisotropic (see also Figures 9 and 10) in third-order statistics, since the degenerate cases are not recovered for the numerically converged solution. On the other hand, the degenerate cases for the cell, Figures 16(b) and 17(b), are well established. This finding suggests that the cell structure is less ordered than that of the pack, which has some third-order directional variance (anisotropy) (see also Figures 9 and 10). The reasons for more randomness in the cell morphology as contrasted to the pack one, are our objective function, Equation (13), that strictly enforces the statistical isotropy (directional invariance), and the periodic boundary conditions that mimic an infinite medium as opposed to a closed pack with wall effects. Note that the characteristic material length scale, h_c , established in this work is of the second-order only, and that the third-order bound would possibly require a larger window of observation.

5 Conclusions

The paper describes a reconstruction procedure for statistically optimal representative unit cells from rich three-dimensional tomographic data. The particulate composite under investigation consists of glass beads packed in a cylindrical container. High resolution microtomography is employed to gather the material data, and the image recognition software *Amira* is used for data processing. The first-, second- and third-order probability functions are used to characterize a polydisperse particulate medium. Error measures are established to assess the quality of the statistical characterization. A fully represented probability spectrum is optimized without distortion of the particle shape and with a constraint on the particle overlap, furnishing the resulting minimization problem highly complex with several local minima. Therefore, the parallel augmented simulated annealing technique is employed to solve the optimization problem on massively parallel computers. Presented results show good repeatability of the reconstruction procedure. Excellent agreement is obtained for statistics of the voxel based pack and statistics of the reconstructed unit cell.

Investigation of the higher order probability functions reveals disagreement in the third-order probabilities between the pack and the cell, even though the first- and second-order functions are well optimized. Thus, potential extension of this work is in expansion of the fitness function for

the third-order statistics. Also, optimization of polydisperse composites with different inclusion shapes, such as ellipsoids, rhombi, etc., is of interest.

It is important to note that the reconstructed unit cells are only representative from a geometrical statistics point of view and that the representativity of the unit cell must also account for the physical processes of interest, such as mechanical or transport properties. However, the construction of a geometrically equivalent representative unit cell is an important first step in describing behavior of complex heterogeneous materials, and both computational and experimental evidence suggests that a statistical approach adopted in this work accounts for the most important interactions [27, 42]. Moreover, advances in parallel computing are making fully resolved simulations of complex physical phenomena, such as combustion [29], on cells presented in this work a reality.

Acknowledgments

The authors would like to gratefully acknowledge the support from the Center for Simulation of Advanced Rockets (CSAR) at the University of Illinois under the contract number B523819 by the U.S. Department of Energy as a part of its Advanced Simulation and Computing (ASC) program. K. Matouš and H. Lee would like to also acknowledge the support from Buckmaster Research - DoD STTR program, AFOSR: Dr. J. Buckmaster (Buckmaster Research) and Dr. A. Nachman (AFOSR) program managers. Moreover, the authors thank Michael Campbell for running the reconstruction code on Red Storm computer located at Sandia National Laboratories, NM, and to Sergei Poliakov for running the statistics code on the Turing cluster. Finally, the authors gratefully acknowledge the use of the Turing cluster maintained and operated by the Computational Science and Engineering Program at the University of Illinois.

References

- [1] J.E. Andrade and X. Tu. Multiscale framework for behavior prediction in granular media. *Mechanics of Materials*, 41(6):652–669, 2009.
- [2] P. Areias and K. Matouš. Finite element formulation for modeling nonlinear viscoelastic elastomers. *Computer Methods in Applied Mechanics and Engineering*, 197:4702–4717, 2008.
- [3] C.H. Arns, M.A. Knackstedt, and K.R. Mecke. Reconstructing complex materials via effective grain shapes. *Physical Review Letters*, 91(21):215506–1–215506–4, 2003.
- [4] T. Aste, M. Saadatfar, and T.J. Senden. Geometrical structure of disordered sphere packings. *Physical Review E*, 71(061302):061302–1–061302–15, 2005.

- [5] M.J. Beran. *Statistical Continuum Theories*. Interscience Publishers, 1968.
- [6] J.D. Bernal. A geometric approach to the structure of liquids. *Nature*, 183:141–147, 1959.
- [7] B. Bochenek and R. Pyrz. Reconstruction of random microstructures—a stochastic optimization problem. *Computational Materials Science*, 31:93–112, 2004.
- [8] D. Bouvard, J.M. Chaix, R. Dendievel, A. Fazekas, J.M. Ltang, G. Peix, and D. Que-nard. Characterization and simulation of microstructure and properties of eps lightweight concrete. *Cement and Concrete Research*, 37:1666–1673, 2007.
- [9] P.M. Chaikin and T.C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, New York, 2000.
- [10] B.C. Collins. Reconstruction of statistically optimal periodic unit cells of multimodal particulate composites. Master’s thesis, University of Illinois at Urbana-Champaign, 2008.
- [11] B.C. Collins, F. Maggi, K. Matouš, T.L. Jackson, and J. Buckmaster. Using tomography to characterize heterogeneous propellants. In *46th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA, 2008. Paper No: AIAA 2008-941.
- [12] B.C. Collins, K. Matouš, and D. Rypl. Three-dimensional reconstruction of statistically optimal unit cells of multimodal particulate composites. *International Journal for Multiscale Computational Engineering*, accepted for publication, 2009.
- [13] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices, and Groups*. Springer, New York, 1987.
- [14] K.J. Dong, R.Y. Yang, R.P. Zou, and A.B. Yu. Role of interparticle forces in the formation of random loose packing. *Physical Review Letter*, 96(145505):145505–1–145505–4, 2006.
- [15] S. Rolland du Roscoat, M. Decain, X. Thibault, C. Geindreau, and J.-F. Bloch. Estimation of microstructural properties from synchrotron x-ray microtomography and determination of the rev in paper materials. *Acta Materialia*, 55:2841–2850, 2007.
- [16] D.T. Fullwood, S.R. Niezgoda, and S.R. Kalidindi. Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. *Acta Materialia*, 56:942–948, 2008.
- [17] S. Gallier and F. Hiernard. Microstructure of composite propellants using simulated packings and x-ray tomography. *Journal of Propulsion and Power*, 24(1):147–150, 2008.
- [18] M. Jerkins, M. Schroter, and H.L. Swinney. Onset of mechanical stability in random packings of frictional spheres. *Physical Review Letter*, 101(018301):018301–1–018301–4, 2008.
- [19] X. Jia and R.A. Williams. A packing algorithm for particles of arbitrary shapes. *Powder Technology*, 120:175–186, 2001.

- [20] Y. Jiao, F. H. Stillinger, and S. Torquato. Modeling heterogeneous materials via two-point correlation functions: Basic principles. *Physical Review E*, 76(031110):031110–1–031110–15, 2008.
- [21] Y. Jiao, F. H. Stillinger, and S. Torquato. Modeling heterogeneous materials via two-point correlation functions. ii. algorithmic details and applications. *Physical Review E*, 77(031135):031135–1–031135–15, 2008.
- [22] Y. Jiao, F.H. Stillinger, and S. Torquato. Optimal packings of superballs. *Physical Review E*, 79(041309):041309–1–041309–11, 2009.
- [23] S.V. Kochevets, T.L. Jackson J. Buckmaster, and A. Hegab. Random packs and their use in the modeling of heterogeneous solid propellant combustion. *Journal of Propellant and Power*, 17(4):883–891, 2001.
- [24] N.C. Kumar, K. Matouš, and P.H. Geubelle. Reconstruction of periodic unit cells of multi-modal random particulate composites using genetic algorithms. *Computational Materials Science*, 42:352–367, 2008.
- [25] J. Liang and K.A. Dill. Are proteins well-packed? *Biophysical Journal*, 81(2):751–766, 2001.
- [26] J.L. Lions. *Some methods in the mathematical analysis of systems and their control*. New York, Gordon and Breach, Science Publisher, Inc., 1981.
- [27] C. Liu. On the minimum size of representative volume element: An experimental investigation. *Experimental Mechanics*, 45(3):238–243, 2005.
- [28] F. Maggi, S. Stafford, T. L. Jackson, and J. Buckmaster. Nature of packs used in propellant modeling. *Physical Review E*, 77(046107):046107–1–046107–17, 2008.
- [29] L. Massa, T.L. Jackson, and J. Buckmaster. New kinetics for a model of heterogeneous propellant combustion. *Journal of Propulsion and Power*, 21(5):914–924, 2005.
- [30] K. Matouš and P.H. Geubelle. Multiscale modelling of particle debonding in reinforced elastomers subjected to finite deformations. *International Journal for Numerical Methods in Engineering*, 65:190–223, 2006.
- [31] K. Matouš, H.M. Inglis, X. Gu, D. Rypl, T.L. Jackson, and P.H. Geubelle. Multiscale modeling of solid propellants: From particle packing to failure. *Composites Science and Technology*, 67:1694–1708, 2007.
- [32] K.R. Mecke and C.H. Arns. Fluids in porous media: a morphometric approach. *J. Phys.: Condens. Matter.*, 17:S503S534, 2005.
- [33] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution programs*. Springer, Berlin, 1992.

- [34] E.R. Nowak, J.B. Knight, E. Ben-Naim, H.M. Jaeger, and S.R. Nagel. Density fluctuations in vibrated granular materials. *Physical Review E*, 57(2):1971–1982, 1998.
- [35] G.L. Povirk. Incorporation of microstructural information into materials. *Acta Metallurgical Materials*, 43:3199–3206, 1995.
- [36] J.A. Quintanilla and W. Max Jones. Using convex quadratic programming to model random media with gaussian random fields. *Physical Review E*, 75(046709):046709–1–046709–9, 2007.
- [37] J. Schmalzing and K.M. Gorski. Minkowski functionals used in the morphological analysis of cosmic microwave background anisotropy maps. *Mon. Not. Roy. Astron. Soc.*, 297:355365, 1998.
- [38] G.T. Seidler, G. Martinez, L.H. Seeley, K.H. Kim, E.A. Behne, S. Zaraneek, B.D. Chapman, S.M. Heald, and D.L. Brewster. Granule-by-granule reconstruction of a sandpile from x-ray microtomography data. *Physical Review E*, 62(6):8175–8181, 2000.
- [39] C.E. Shannon. *Bell Labs Technical Journal*, 27:379, 1948.
- [40] P. Smith and S. Torquato. Computer simulation results for the two-point probability function of composite media. *Journal of Computational Physics*, 76:176–191, 1988.
- [41] V. Sundararaghavan and N. Zabaras. Classification and reconstruction of three-dimensional microstructures using support vector machines. *Computational Materials Science*, 32:223–239, 2005.
- [42] S. Swaminathan and S. Ghosh. Statistically equivalent representative volume elements for unidirectional composite microstructures: Part ii - with interfacial debonding. *Journal of Composite Materials*, 40(7):605–621, 2006.
- [43] I. Taguchi, M. Kirashige, and K. Imai. Effects of cubic container’s wall or floor on random packing structures of spherical particles. *JSME International Journal*, 49(2A):265–272, 2006.
- [44] D.R.S. Talbot and J.R. Willis. Variational principles for inhomogeneous nonlinear media. *IMP Journal of Applied Mathematics*, 35:39–54, 1985.
- [45] S. Torquato. Statistical description of microstructure. *Annual Review of Materials Research*, 32:77–111, 2002.
- [46] M.B. Weissman. Low-frequency noise as a tool to study disordered materials. *Annual Review of Materials Research*, 26:395–429, 1996.
- [47] P. Wochner, C. Gutt, T. Autenrieth, T. Demmer, V. Bugaev, A.D. Ortiz, A. Duri, F. Zontone, G. Grüel, and H. Dosch. A geometric approach to the structure of liquids. *Proceedings of the National Academy of Sciences*, 106(28):11511–11514, 2009.

- [48] C.L.Y. Yeong and S. Torquato. Reconstructing random media. *Physical Review E*, 57:495–506, 1998.
- [49] C.L.Y. Yeong and S. Torquato. Reconstructing random media. ii. three-dimensional media from two-dimensional cuts. *Physical Review E*, 58:224–233, 1998.
- [50] J. Zeman and M. Šejnoha. From random microstructures to representative volume elements. *Modelling Simul. Mater. Sci. Eng.*, 15:S325–S335, 2007.