

Proceedings of the
Multigrid Tutorial, with Applications to
Molecular Dynamics

October 10-12, 1995. Weizmann Institute, Rehovot, Israel

* **Workshop sponsor**

The workshop was sponsored by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science, Rehovot, Israel.

* **Acknowledgement**

We wish to thank the United States Air Force European Office of Aerospace Research and Development for its contribution to the success of this workshop.

19980303 119

DTIC QUALITY INSPECTED 3

Workshop Organizers:

Achi Brandt,
Weizmann Institute of Science
Tamar Schlick,
Courant Institute of Mathematics

Secretary:

Carol Weintraub
Office: 972-8-343545
Fax: 972-8-344122
Email: carol@wisdom.weizmann.ac.il

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 1995	3. REPORT TYPE AND DATES COVERED Conference Proceedings	
4. TITLE AND SUBTITLE Workshop on Multigrid Tutorial, with Applications to Molecular Dynamics			5. FUNDING NUMBERS F6170895W0467	
6. AUTHOR(S) Conference Committee				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science Rehovot Rehovot 76100 Israel			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 802 BOX 14 FPO 09499-0200			10. SPONSORING/MONITORING AGENCY REPORT NUMBER CSP 95-1048	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (Maximum 200 words) The Final Proceedings for Multigrid Tutorial, with Applications to Molecular Dynamics, 10 October 1995 - 12 October 1995				
14. SUBJECT TERMS Nil			15. NUMBER OF PAGES 233	
			16. PRICE CODE N/A	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Table of Contents

	<u>Page</u>
Program	2
List of Participants	4
Lecture Transparencies:	
A. Brandt: Introduction: task and types of multiscale computation	7
A. Brandt: Decomposition of forces and fast force summation	27
K. Stueben: Elements of linear multigrid (MG) for solving elliptic PDEs	37
A. Brandt: Multigrid Monte-Carlo and stochastic coarsening	69
A. Brandt: Global and discrete-state optimization: multiscale annealing	89
K. Stueben: Full multigrid (FMG) and nonlinear multigrid	105
E. Barth, M. Mandziuk, T. Schlick & G. Zhang: Molecular dynamics at large time-steps	115
C. Lambert: Efficient dense Hessian computations in molecular minimization	135
R. Skeel, M. Nelson, T. Bishop & K. Schulten: Efficient use of fast electrostatics in molecular dynamics	151
O.M. Becker: Reduced variable molecular dynamics	165
K. Stueben: Algebraic multigrid (AMG)	187
J. Ruge: AMG applied to functional minimization problems	195
A. Brandt & D. Bai: Multiscale methods in molecular dynamics	205

WORKSHOP PROGRAM -- Multigrid and Molecular Dynamics

October 10-12, 1995

Dept. of Applied Mathematics & Computer Science
The Weizmann Institute of Science, Ziskind Building,
Lecture Room 1 (basement level)

Note: The program may change even during the course itself according to the interest of participants.

Tuesday, October 10th

8:30 - 9:15 Registration, organization, coffee
9:00 - 9:15 Introductory remarks
9:15 - 10:30 Introduction: tasks and types of multiscale
computation (Brandt)
10:30 - 11:00 Coffee
11:00 - 12:15 Multiscale decomposition of forces and fast
force summation (Brandt)
12:15 - 13:30 Lunch
13:30 - 15:00 Elements of linear multigrid for PDEs (Stueben)
15:00 - 15:30 Coffee
15:30 - 17:00 Poster session, including:
-Gomathi Ramachandran: Buckling transitions in
superhelical DNA: dependence on the elastic
constants and DNA size
-Bimalendu Mishra: Error analysis in Langevin dynamics
simulations
-Margaret Mandziuk: Resonance in the dynamics of
chemical systems
-Eric Barth: LIN and a family of related methods for
molecular dynamics simulation
-Hongmei Jian: Computer simulation of DNA using a
discrete worm-like chain model
-Gerd Winter: BEMOLPA project
19:30 Mini-vans depart for dinner at Pearl of the Sea
restaurant, in Rishon L'Tzion

Wednesday, October 11th

8:45 - 9:00 Coffee
9:00 - 10:30 Multigrid Monte-Carlo and stochastic coarsening
(Brandt; Mack?)
10:30 - 11:00 Coffee
11:00 - 12:15 Global and discrete-state optimization: multiscale
annealing (Brandt)
12:15 - 14:00 Lunch
14:00 - 15:00 Full multigrid and nonlinear multigrid (Stueben)
15:00 - 15:30 Eric Barth and Margaret Mandziuk: Molecular dynamics
simulation at large timesteps
15:30 - 16:00 Coffee
16:00 - 16:30 Bob Skeel: Efficient use of fast electrostatics in
molecular dynamics
16:30 - 17:00 Chris Lambert: Efficient dense Hessian computation in
molecular minimization
17:00 - 17:30 Oren Becker: Reduced variable molecular dynamics
17:30 - 18:00 Dexuan Xie: A remark on algebraic multigrid analysis
19:00 Dinner in town at one's choice

Thursday, October 12th

8:45 - 9:00 Coffee
9:00 - 10:00 Algebraic multigrid (Stueben, Ruge)
10:00 - 10:30 Coffee
10:30 - 12:00 Multiscale methods in molecular dynamics (Bai, Brandt)
12:00 - 14:00 Discussion on future directions; box lunch
16:00 Bus departs for dinner in Yaffo, and evening at Tel Aviv
seaside promenade

List of Participants

Dov Bai
Cornell University
Ithaca, NY 14853-7501
USA
bai@msiadmin.cit.cornell.edu

Eric Barth
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
barthe@cims.nyu.edu
(and Howard Hughes Medical Institute)

Oren Becker
School of Chemistry
Tel Aviv University
Ramat Aviv, Tel Aviv 69578
Israel
becker@sapphire.tau.ac.il

Achi Brandt
Dept. of Appl. Math. & Comp. Sci.
Weizmann Institute of Science
Rehovot 76100
Israel
mabrandt@weizmann.weizmann.ac.il

Nathan Dinar
Dept. of Applied Mathematics
Israel Institute of Biological Research
Israel
natan@math.wxiiibr.co.il

Boris Diskin
Dept. of Appl. Math. & Comp. Sci.
Weizmann Institute of Science
Rehovot 76100
Israel
diskin@wisdom.weizmann.ac.il

Patrick Fischer
Dept. of Computer Science
School of Mathematical Sciences
Tel Aviv University
Ramat Aviv 69978
Israel
fischer@math.tau.ac.il

Meirav Galun
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
meirav@wisdom.weizmann.ac.il

Hongmei Jian
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
jianh@acf2.nyu.edu

Jacob Katzenelson
Dept. of Elec. Engr.
Technion
Haifa 32000
Israel
jacob@ee.technion.ac.il

Arnold Krechel
Schloss Birlinghoven
D-53757 St. Augustin
Germany
krechel@gmd.de

Christophe Lambert
Dept. of Computer Science
Duke University
Room D327, LSRC
Durham, NC 27708
USA
lambert@cs.duke.edu

Oren Livne
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
livneo@wisdom.weizmann.ac.il

Ton Lubrecht
INSA de Lyon
LMC Bâtiment 113-20
20 Avenue Albert Einsein
69621 Villeurbanne Cedex
Lyon, France
Ton.Lubrecht@lmc.insa-lyon.fr

Margaret Mandzuik
Howard Hughes Medical Institute
New York University
New York, NY 10021
USA
mandzk@watson.biomath.nyu.edu

Tamar Naeh
Dept. of Mathematics
Tel Aviv University
Ramat Aviv
Tel Aviv
Israel
naeh@math.tau.ac.il

Ruth Pachter (& Meir Pachter)
Materials Directorate
WL/MLPJ
Wright Patterson AFB
Ohio 45433-6533
USA
pachterr@ml.wpafb.af.mil

Gomathi Ramachandran
Howard Hughes Medical Institute
New York University
New York, NY 10021
USA
ramachan@franklin.biomath.nyu.edu

John Ruge
Dept. of Mathematics
Univ. of Colorado
Boulder, CO
USA
jruge@newton.colorado.edu

David Saada
Dept. of Mathematics
Technion
Haifa 32000
Israel
phr76ds@phjoan2.technion.ac.il

Tamar Schlick
Courant Institute of Mathematics
New York University
251 Mercer Street
New York, NY 10021
USA
schlick@acf4.nyu.edu

Schwichtenberg, Horst
Schloss Birlinghoven
D-53757 St. Augustin
Germany
schwichtenberg@gmd.de

Robert D. Skeel
Dept. of Computer Science, UIUC
1304 West Springfield Avenue
Urbana, Illinois 61801-2987
USA
skeel@cs.uiuc.edu

Klaus Stueben
Gesellschaft für Mathematik und Datenver
Inst. für Methodische Grundlagen
Postfach 1316
W-5205 Sankt Augustin 1
Germany
gmap02@f1klaus.gmd.de

Alex Volynski
c/o Dept. of Geology & Mineralogy
Ben Gurion University
P.O.B. 653
Beer Sheva 84105
Israel

Holger Wallmeier
Hoechst AG
Zentralforschung
Scientific Computing
G 865 A
65926 Frankfurt am Main
Germany
wallmeier@msmwia.hoechst.hoechst-ag.d400.de

Winter, Gerd
Schloss Birlinghoven
D-53757 St. Augustin
Germany
Gerd.Winter@gmd.de

Dexuan Xie
Science & Research I, 221c
High Performance Computing Centers
University of Houston
Houston, TX 77204-5502
USA
xie@sina.tcamc.uh.edu

Wei Xu
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
weixu@franklin.biomath.nyu.edu

Leonid Zaslavsky
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
zasl@wisdom.weizmann.ac.il

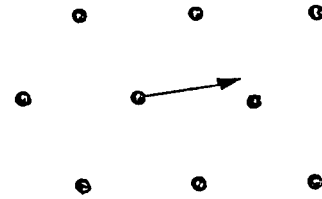
A. Brandt

Introduction: Task and Types
of Multiscale Computation

- n particles : $n \sim 10^3, 10^{10}, \dots$

- Calculating $O(n^2)$ forces
at each time step

- Extremely short steps



- Wrong attraction basins

- Thermal statistical fluctuations :
Need represent many probable u

All addressed by

multi-scale methods

multigrid multi-level multi-resolution

FFT, Renormalization Group,

Wavelets, Multipole, Fractals

Hardware:

Ever smaller and faster

Past: 10 times faster
each ~ 6 years

Future: Parallel processing

Algorithms

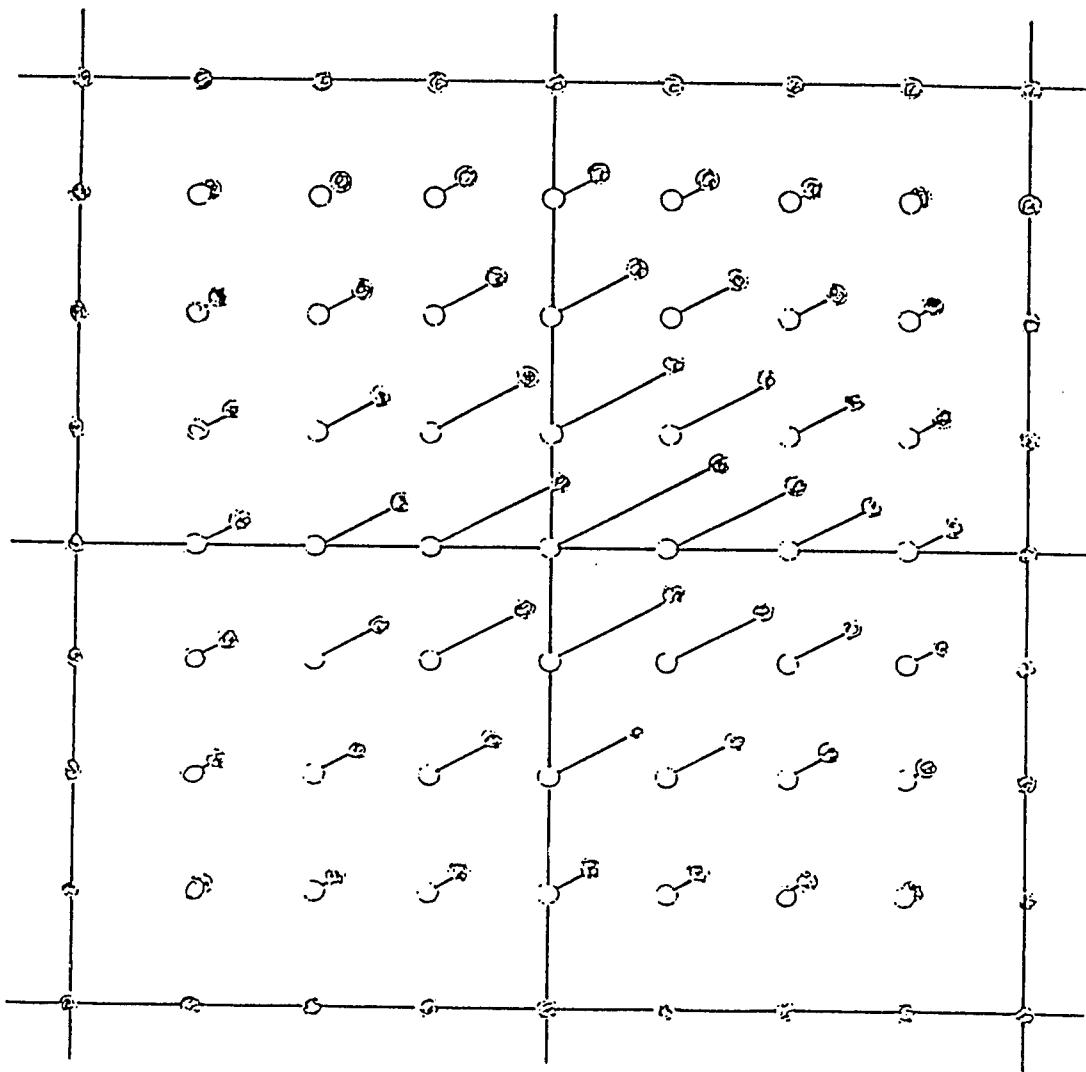
Required amount of computations
rise slower with
increased problem size

Past: comparable contribution

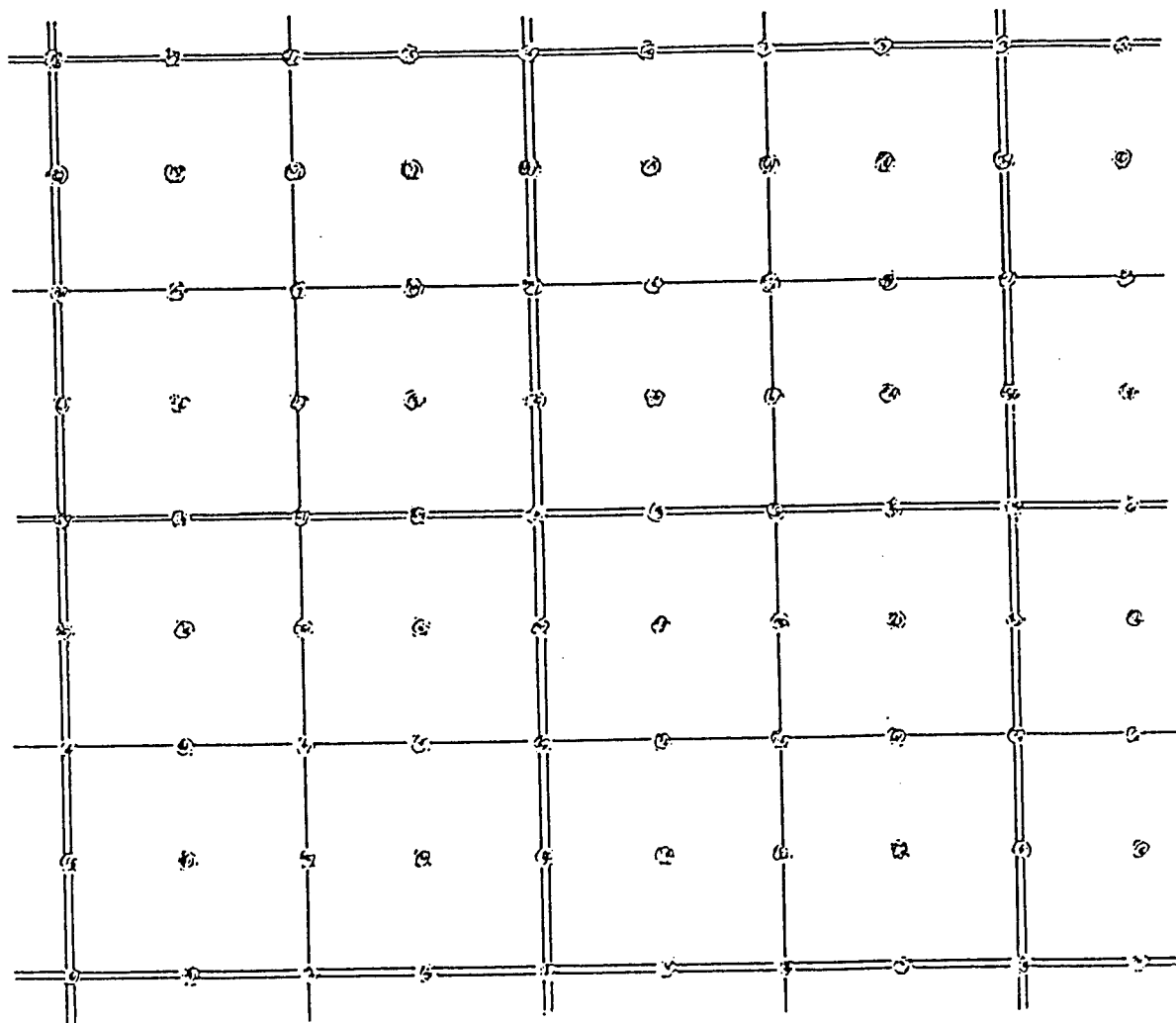
Future: Multiscale methods

Highly compatible

Excel at large problems



- Similarly: for non-uniform atom distribution.
- Collective force: via the finer-grid forces
Energy (grid)



$\leftarrow a \rightarrow$

$\leftarrow h \rightarrow$

$\leftarrow 2h \rightarrow$

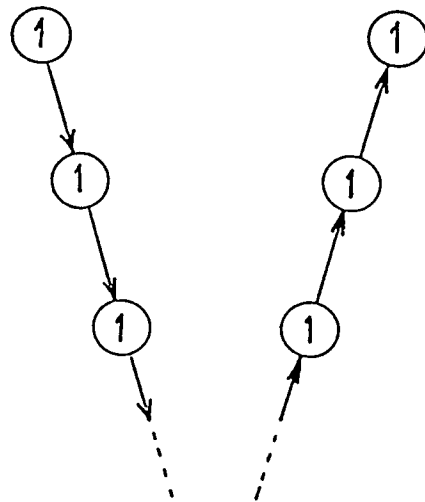
Multigrid cycle

sweeps on

scale a (particles)

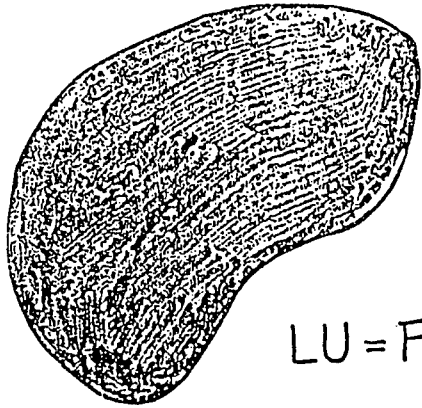
grid h

grid $2h$

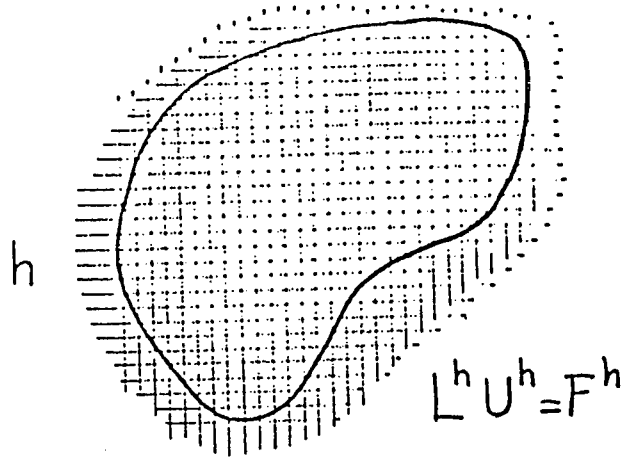


Very large scale changes in one cycle

Negligible work on coarse levels

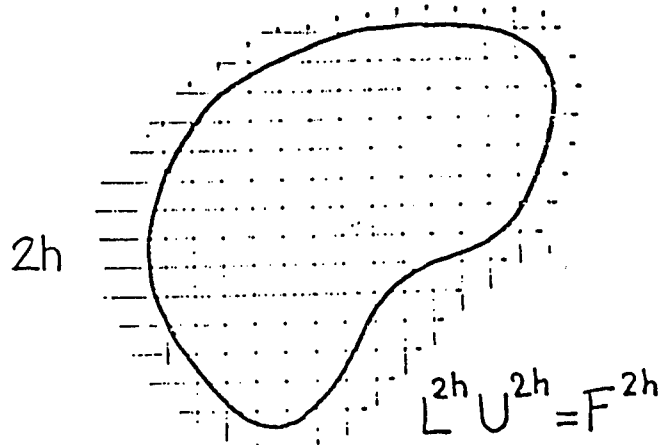


$$LU = F$$



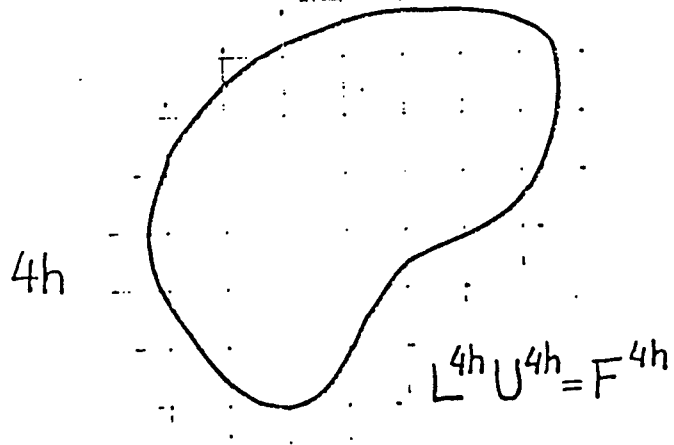
h

$$L^h U^h = F^h$$



$2h$

$$L^{2h} U^{2h} = F^{2h}$$

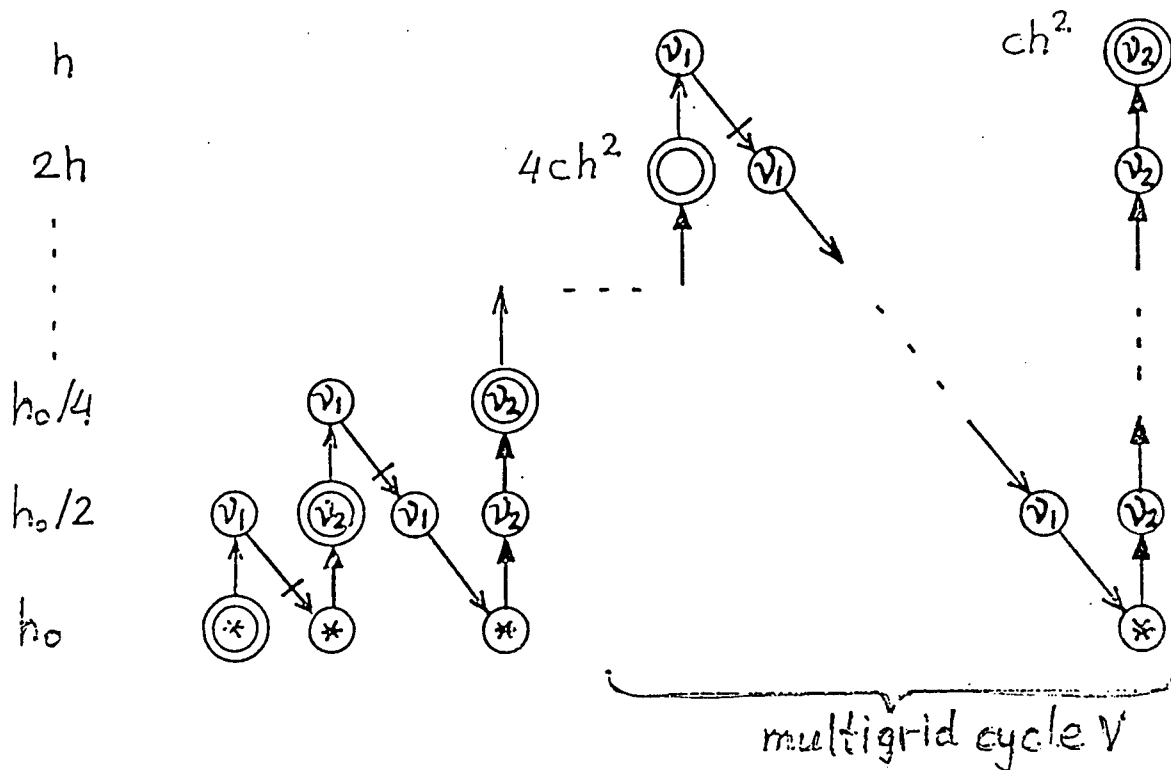


$4h$

$$L^{4h} U^{4h} = F^{4h}$$

⋮

FULL MULTIGRID (FMG) ALGORITHM



- ↑ interpolation (order $l+p$) to a new grid.
- ↑ interpolation (order m) of corrections:
- ⊙ ∇ relaxation sweeps
- ⊙ algebraic error $<$ truncation error
- ↘ residual transfer
- ⊙* enough sweeps or direct solver

Fully efficient ~ 8 minimal work units

Multigrid Solvers

- Linear scalar elliptic equations (1970)*
- Nonlinear
- Adaptive grids FAS
- General boundaries, BC *
- Discontinuous coefficients $\nabla(a\nabla u)=f$
- Disordered coefficients AMG
- Non scalar PDE systems* elasticity, NS, ...
- Non-ellipticity: convection high Re
- Indefinite: waves \longleftrightarrow rays
- Inverse problems $a=?$ DA
- Time dependent space + time
- Dense $n \times n$ matrices $O(n)$
- Crucial features invisible to coarse grids
- Gauge freedom
- Topologies Dirac eqs.
- Near zero modes

* Rigorous

Compressible Navier-Stokes: 2D

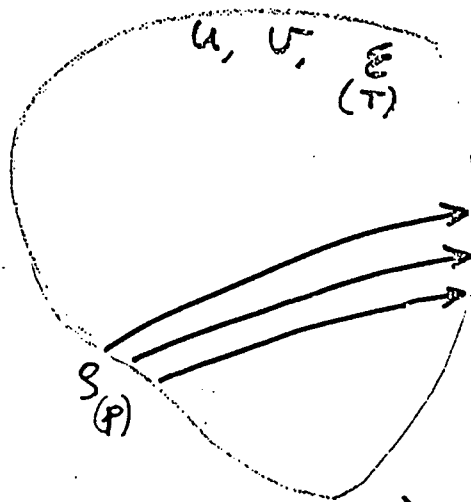
$$\begin{aligned}
 -\mu \Delta u + \rho u u_x + \rho v u_y - (\lambda + \mu)(u_x + v_y)_x + p_x &= 0 \\
 -\mu \Delta v + \rho u v_x + \rho v v_y - (\lambda + \mu)(u_x + v_y)_y + p_y &= 0 \\
 -\kappa \Delta \epsilon + \rho u \epsilon_x + \rho v \epsilon_y + p(u_x + v_y) \\
 -\mu(v_x + u_y)^2 - \lambda(u_x + v_y)^2 - 2\mu(u_x^2 + v_y^2) &= 0
 \end{aligned}$$

$$(\rho u)_x + (\rho v)_y - \nu \Delta \rho = 0$$

$$p = p(\epsilon, \rho)$$

elliptic BVP.
unknown u, v, ϵ, ρ, p

B.C.:



Inviscid case: $\lambda, \mu, \kappa \ll \rho l \max(|u|, |v|)$

$l =$ length at which (u, v, ϵ) change. \Rightarrow Euler eq.

Usually there are viscous layers.

Incompressible:

$$\mathcal{Q} = -\frac{1}{Re} \Delta + u \partial_x + v \partial_y. \quad \begin{pmatrix} \mathcal{Q} & 0 & \partial_x \\ 0 & \mathcal{Q} & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0$$

$$\det \uparrow = (-\Delta) \mathcal{Q} \quad \text{elliptic.}$$

h-principal $\mathbf{L}u = f$

det L

Cauchy - Riemann $\begin{pmatrix} \partial_x & \partial_y \\ \partial_y & -\partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0$ $-\Delta$

2D Stokes $\begin{pmatrix} -\Delta & 0 & \partial_x \\ 0 & -\Delta & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0$ Δ^2

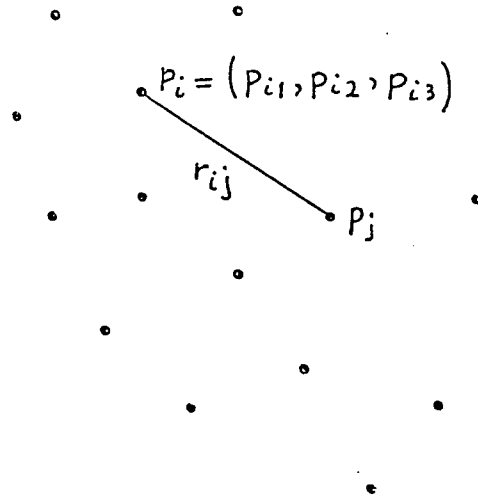
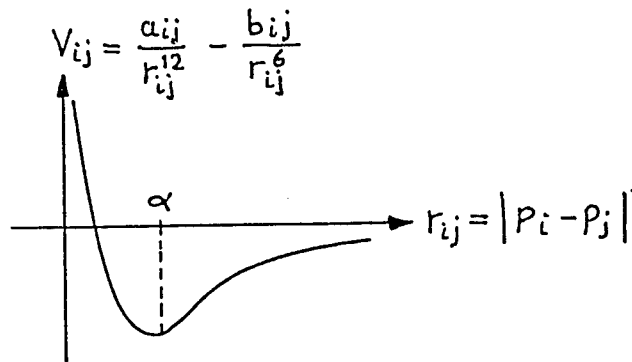
2D Incompressible Navier - Stokes $\begin{pmatrix} Q & 0 & \partial_x \\ 0 & Q & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix}$ $-\eta\Delta$
 $Q = -\frac{1}{R} \Delta + \underline{u} \cdot \underline{\nabla}$

2D Euler $\begin{pmatrix} \rho \underline{u} \cdot \underline{\nabla} & 0 & 0 & 0 & \partial_x \\ 0 & \rho \underline{u} \cdot \underline{\nabla} & 0 & 0 & \partial_x \\ \rho \partial_x & \rho \partial_y & \underline{u} \cdot \underline{\nabla} & 0 & 0 \\ \rho \partial_x & \rho \partial_y & 0 & \rho \underline{u} \cdot \underline{\nabla} & 0 \\ 0 & 0 & -P_\rho & -P_\epsilon & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ \rho \\ \epsilon \\ p \end{pmatrix}$ $\rho^3 (\underline{u} \cdot \underline{\nabla})^2$
 $[\underline{u} \cdot \underline{\nabla}]^2 - \alpha^2 \Delta$
 $\alpha^2 = P_\rho + \frac{P}{\rho^2} P_\epsilon$

Compressible Navier - Stokes (on the viscous scale) $\kappa \mu (2\mu + \lambda) \Delta^3 (\underline{u} \cdot \underline{\nabla})$

Central Cauchy - Riemann Δ^{2h}

Central (Navier -) Stokes $Q^h \Delta^{2h}$



$$E^{\circ}(p) = E^{\circ}(p_1, p_2, \dots) = \sum_{i \neq j} V_{ij}(|p_i - p_j|)$$

$$E^{\circ}(p^{\circ}) = \min_p E^{\circ}(p)$$

With external forces: $E(p) = E^{\circ}(p) - \sum_i f_i \cdot p_i$

Find p^* such that $E(p^*) = \min E(p)$

Particle-by-particle minimization

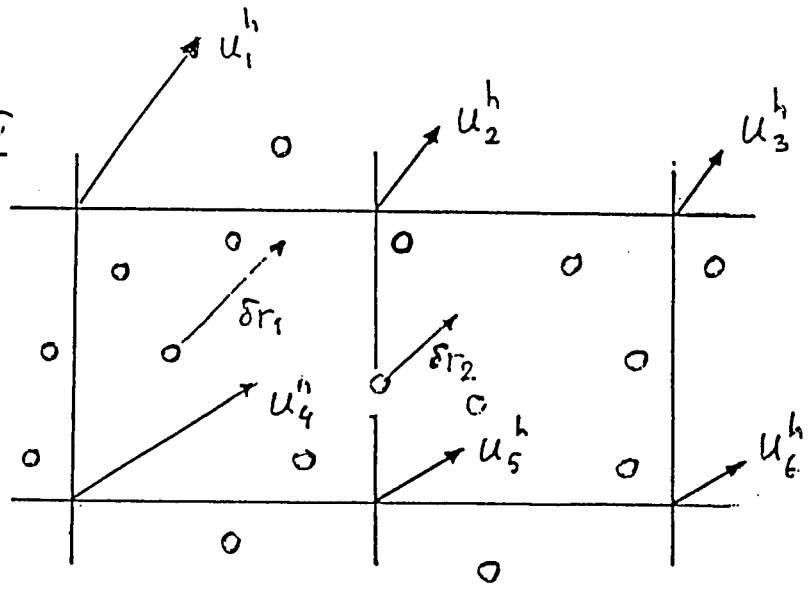
\Rightarrow Extreme slow down

Needed: collective moves

Displacement

field

u^h



$\delta r_i =$ Interpolation from $u^h = (u_1^h, u_2^h, \dots)$

$\Rightarrow E(r + \delta r) = E^h(u^h)$ explicitly
(for smooth u^h)

• Minimize $E^h(u^h)$, only then displace particles

$$E(r) = -\sum f_i^i r_i + \sum_{ij} V_{ij} (|r_i - r_j|) + \sum g^i r_i^2$$

$$\delta r_i = \sum_k \lambda_i^k u_k, \quad \sum_k \lambda_i^k = 1, \quad u = u^h$$

$$\delta r_j = \sum_k \lambda_j^k u_k, \quad \sum_k \lambda_j^k = 1$$

$$\begin{aligned} \delta(r_i - r_j) &= \sum_k (\lambda_i^k - \lambda_j^k) u_k & \sum_k (\lambda_i^k - \lambda_j^k) &= 0 \\ &= \sum_{k,\ell} \mu_{ij}^{k\ell} (u_k - u_\ell) \end{aligned}$$

$$\begin{aligned} V_{ij}(|r_i - r_j|) &= V_{ij}(|r_i^o - r_j^o + \sum \mu_{ij}^{k\ell} (u_k - u_\ell)|) \\ &\approx V_{ij}(|r_i^o - r_j^o|) + a_{ij}^1 \sum \mu_{ij}^{k\ell} (u_k - u_\ell) \\ &\quad + a_{ij}^2 \left[\sum \mu_{ij}^{k\ell} (u_k - u_\ell) \right]^2 + \dots \end{aligned}$$

for smooth u^h : $|u_k^h - u_\ell^h| \leq \epsilon_{k\ell}$ for k near ℓ

$$E^h(u^h) = \sum f_k^h u_k^h + \sum b_{k\ell}^h u_k^h u_\ell^h + \dots$$

⇒ Similarly on coarser grids

⇒ Elasticity - Plasticity - Strain limits

Minimization of $E^h(u^h)$

Point-by-point minimization
 → slow down!

• Transfer to grid $2h$

$\delta u_i^h =$ Interpolation from u^{2h}

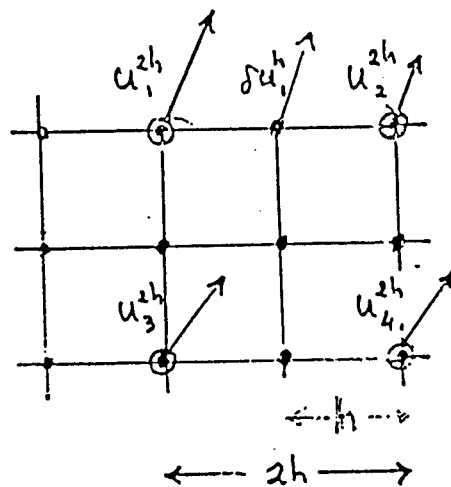
Construct explicitly

$$E^{2h}(u^{2h}) = E^h(u^h + \delta u^h(u^{2h}))$$

• Etc. $E^{4h}(u^{4h}), E^{8h}(u^{8h}), \dots$

Each u^{2H} describes smooth change of u^H

• Similarly if the original problem is
 a grid problem



Fine scale

Uniformly elliptic eq.

Wave eq.

$$\Delta u + k^2 u = 0$$

Particles

Lennard-Jones
energy
minimization

Coarse scale

Same eq.

"Rays" eq.

Continuum eqs.

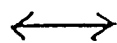
Elasticity
eqs.

Derivation of macroscopic "equations"

(Simpler than renormalization group)

Stochastic

Deterministic



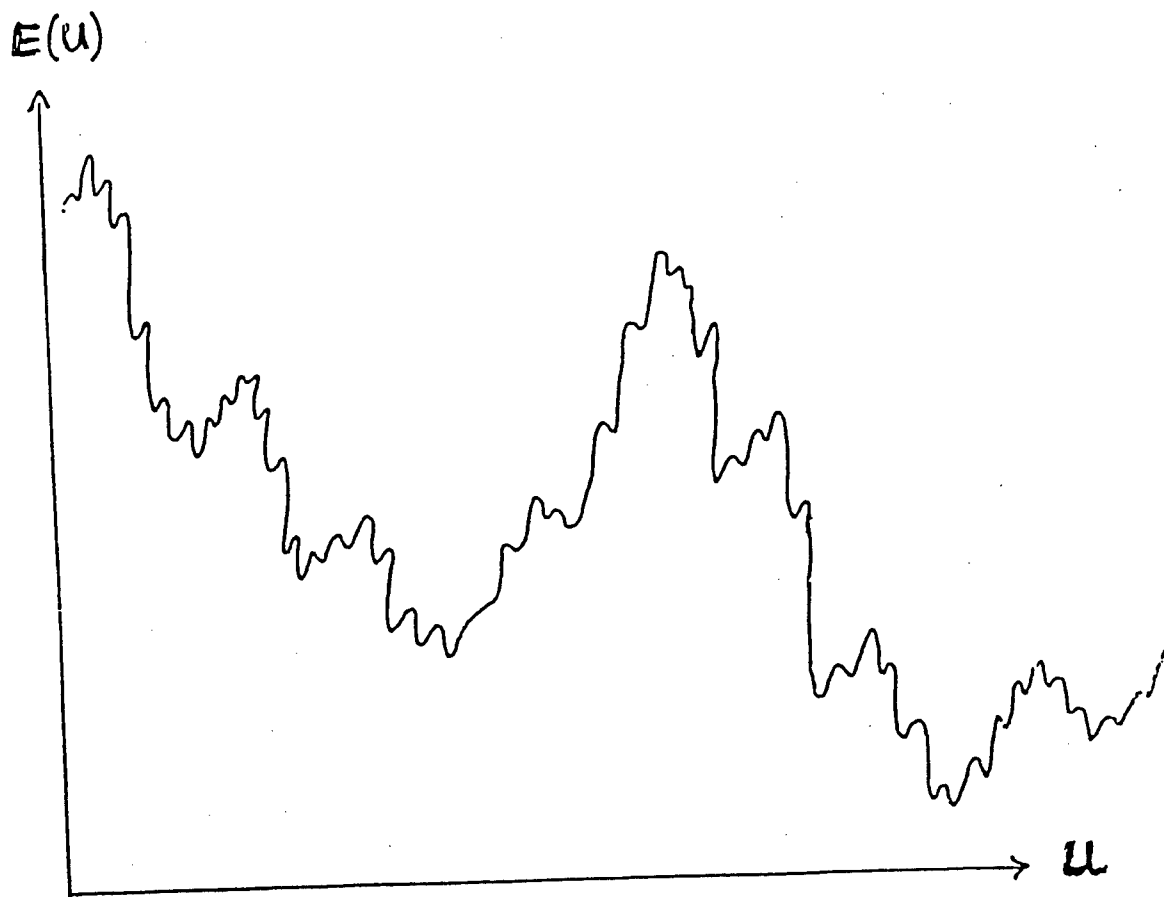
Deterministic

Stochastic

Optimization

$\min E(u)$

- Fast convergence near optimum
- Global optimization
Escape false attraction basins



multi-scale attraction basins

$\min E(u)$

False attraction basins

Particle by particle minimization
trapped in local attraction basins.

Simulated annealing $P(u) \sim e^{-E(u)/T}$
 $T \rightarrow 0$

trapped in large-scale attraction basins.

Multi-level annealing

Collective moves, at all scales
converges (fast, in probability).

- A large-scale move is decided only after optimizing around it at all finer scales.
- Fast annealing at each level.
- Recombinations at each level.

Thermal Statistical Fluctuations

1. Need to produce (stochastically, in a correct probability distribution) independent configurations
2. Need to produce very many of them (to average out deviations)

Multigrid Monte-Carlo

1. Collective moves, at all scales
2. Producing many samples on coarse grids (little work per sample, averaging out large-scale fluctuations; local fluctuations are self-averaging)

Optimal:

Thermodynamic limit $\pm \epsilon$
in $O(\epsilon^{-2})$ computer operations

- General non-linear PDE

steady state: Elliptic, non-elliptic

time dependent

inverse problems. Optimal control

waves \longleftrightarrow rays. Many-eigenfunction.

topological singularities + disorder (Dirac eqs.)

- Integral transform, equations

- n-body interactions

- Molecular mechanics : Ground states
Equilibrium
Dynamics

- Global optimization

- Linear Programming

- Monte Carlo

- Determinant

- Multiple-fermion path integrals

- Derivation of macroscopic equations

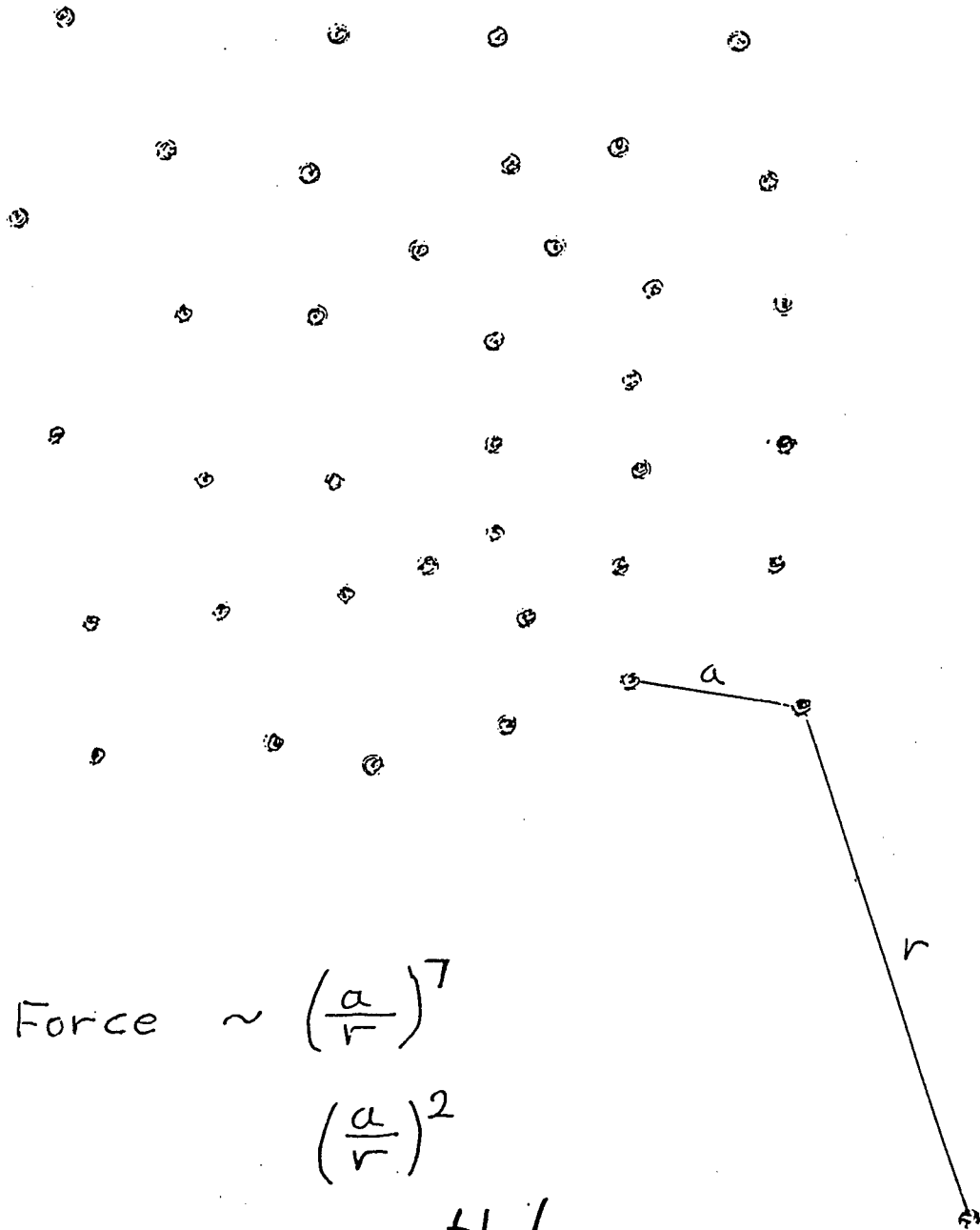
- Image processing

- Tomography X rays, NMR, PET, ... radar

A. Brandt

Decomposition of Forces
and Fast Force Summation

Charges



Integral Transforms

$$V(x) = \int G(x,y) u(y) dy$$

$$x = (x_1, x_2, x_3), \quad y = (y_1, y_2, y_3)$$

Asymptotically **smooth kernels**

$$G(x,y) = e^{i \cdot x \cdot y} \quad \text{Fourier}$$

$$e^{-x \cdot y} \quad \text{Laplace}$$

$$e^{-(x-y)^2 / a^2} \quad \text{Gauss}$$

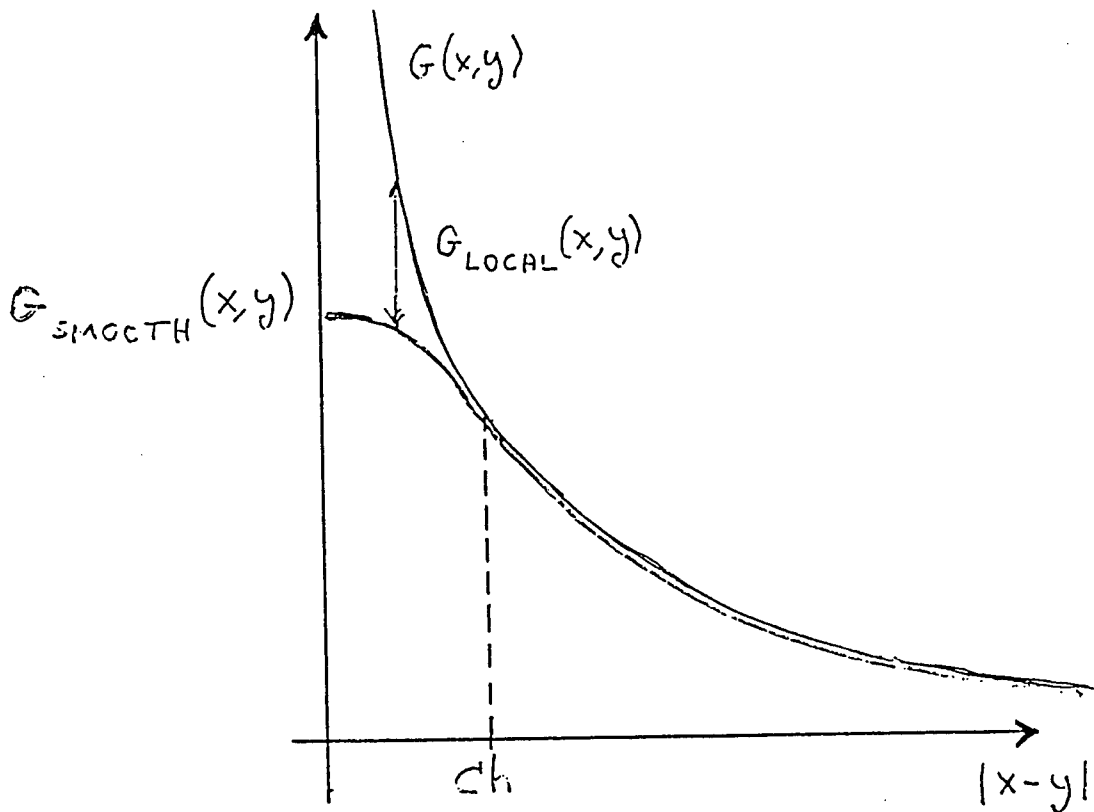
$$|x-y|^{-k}, \log |x-y| \quad \text{Potential}$$

Oscillatory kernels

$$\tilde{G}(x,y) e^{ik|x-y|}$$

Integro-differential equations

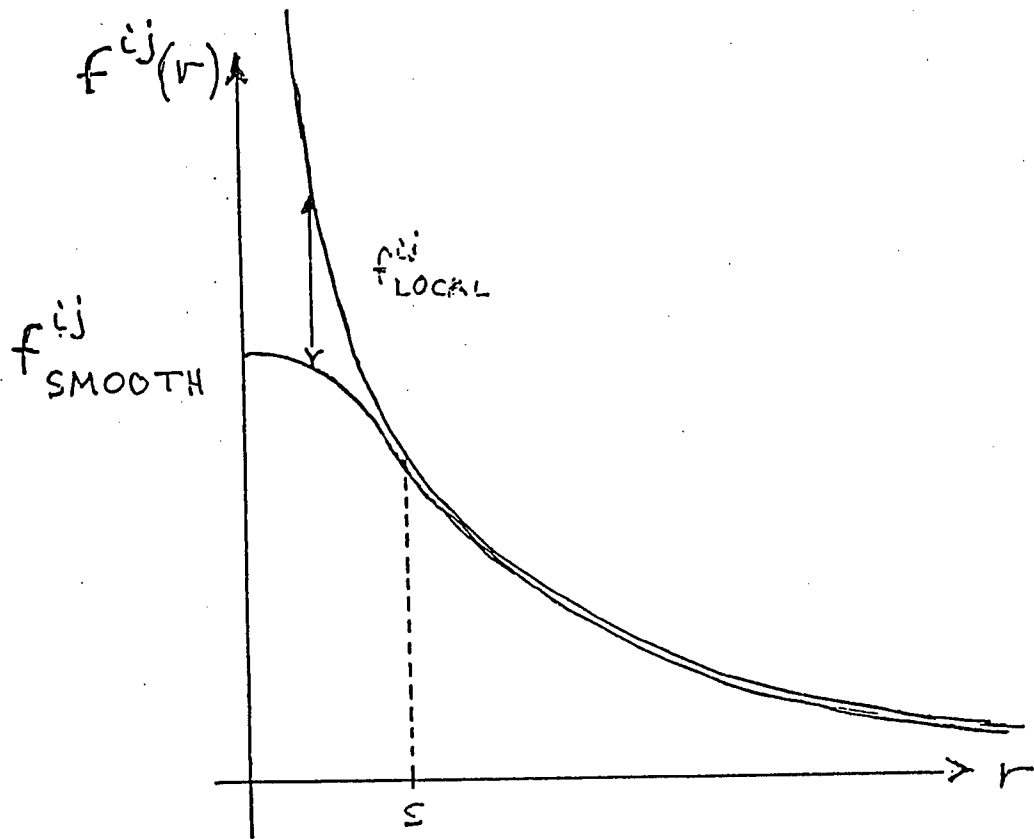
$$V(x_i) = \sum_{y_j \in \Omega^h} G(x_i, y_j) u(y_j) \quad x_i \in \tilde{\Omega}^h$$



$$G(x,y) = G_{LOCAL}(x,y) + G_{SMOOTH}(x,y)$$

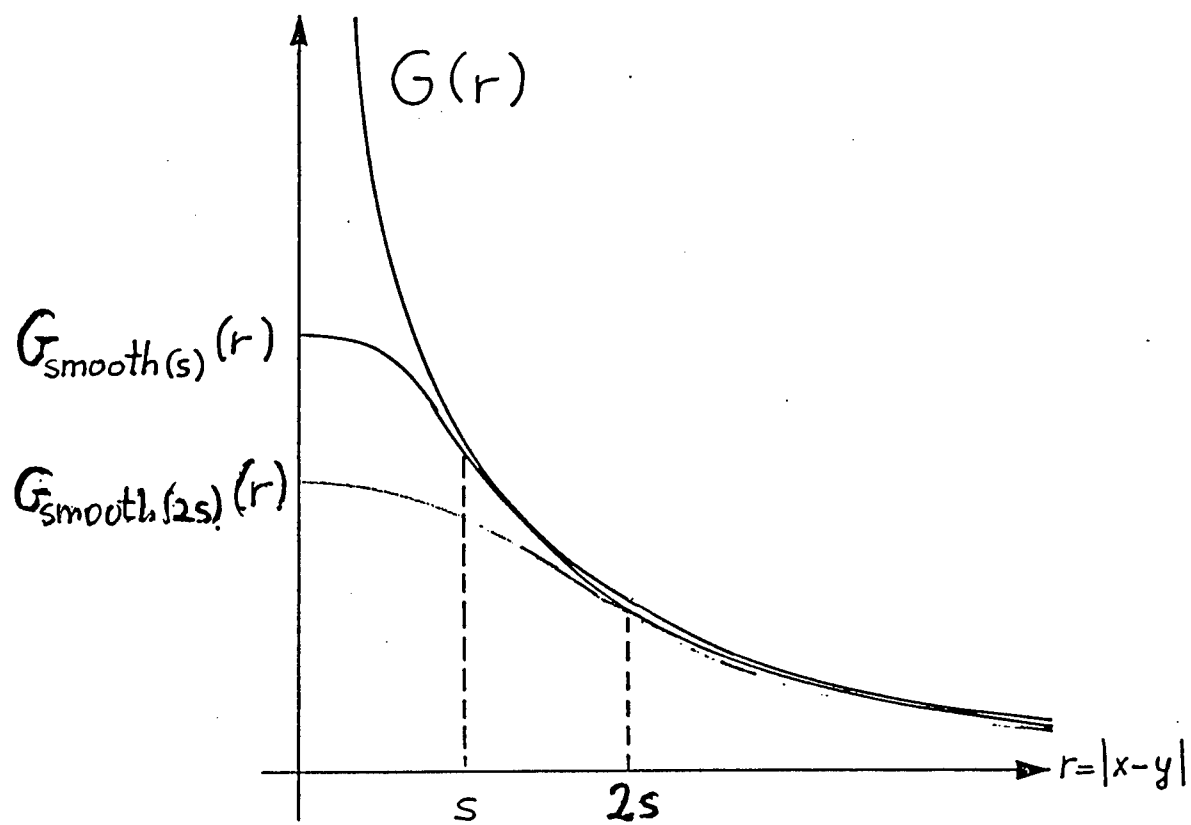
directly

coarser grid



$$f^{ij}_{LOCAL}(r) = 0 \quad \text{for } r \geq s \quad \text{na}$$

$$\left| \frac{\partial^k}{\partial r^k} f^{ij}_{SMOOTH}(r) \right| \leq c s^{-k} \left| f^{ij}_{SMOOTH}(r) \right|$$



$$\left| \frac{\partial^k}{\partial r^k} G_{\text{smooth}(s)}(r) \right| \leq \frac{C(s)}{s^k}$$

$$v(x_i) = \sum_{j=1}^m G(x_i, y_j) u(y_j), \quad (i=1, \dots, n)$$

x_1, \dots, x_n
 y_1, \dots, y_m } particles or gridpoints

X_1, \dots, X_N
 Y_1, \dots, Y_M } (coarser) uniform grids
 meshsize h

p -order interpolation

$$f(x_i) = \sum_k w_{ik} f(X_k) + O((\gamma h)^p f^{(p)})$$

$$\tilde{f}(y_j) = \sum_\ell \tilde{w}_{j\ell} \tilde{f}(Y_\ell) + O((\gamma h)^p \tilde{f}^{(p)})$$

$$v(x_i) \approx \sum_{j,\ell} \tilde{w}_{j\ell} G(x_i, Y_\ell) u(y_j)$$

$$= \sum_\ell G(x_i, Y_\ell) U(Y_\ell) \quad U(Y_\ell) = \sum_j \tilde{w}_{j\ell} u(y_j)$$

$$\approx \sum_{\ell,k} w_{ik} G(X_k, Y_\ell) U(Y_\ell) \quad \textcircled{1} \text{ interpolation}$$

$$= \sum_k w_{ik} V(X_k) \quad V(X_k) = \sum_\ell G(X_k, Y_\ell) U(Y_\ell)$$

$$\textcircled{3} \text{ interpolation}$$

$$\textcircled{2} \text{ recursion}$$

$$\textcircled{4} \text{ correction: } v(x_i) = v(x_i) + \sum G_{\text{local}}(x_i, y_j) u(y_j)$$

n-particle forces

Calculated directly: $9n^2$ operations

Multiscale calculation: Cn operations

C (dimension, accuracy (def.), uniformity)

Only the local part of the forces
should be used/updated in local motions

Smooth part of the force
- in larger-scale (collective) motions.

operations: $O(Ns^{d+1})$

N gridpoints

$O(h^s)$ accuracy

d dimension

Coefficient $\ll 1$

Harmonic kernels: $\sim 6dsN$

via multigrid Poisson solver

Integro-Differential Equation

$$Lu(x) = \int G(x,y) u(y) dy$$

↑
differential

Multigrid Solver

distributive relaxation:

1st order	$+ \delta$	$- \delta$	
2nd order	$+ \delta$	$- 2\delta$	$+ \delta$

Solution cost \approx one fast transform
(one fast evaluation of the
discretized integral transform)

Elements of linear multigrid (MG)
for solving elliptic PDEs

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

Contents:

1. General remarks, history
2. Literature
3. Terminology
4. Basic principles
 - Smoothing
 - Two-grid method
 - Multigrid method
 - Computational work
5. Simple examples

MG1-1

General remarks

What is multigrid NOT?

A particular solver

What IS multigrid?

A general strategy for constructing solvers

This strategy exploits the fact that a problem can be approximated on different scales of resolution ("grids", "levels").

General idea

Compute only those "parts" of a solution on a fine fine "grid" which REQUIRE a fine resolution.

Compute other "parts" on coarser "grids"!

Features of multigrid

- (1) Provides "optimal" methods (i.e.: $O(N)$)
- (2) General applicability
- (3) Involves only "local processes"
(Parallel computer architectures!)

MG1-2

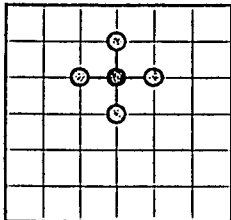
Classical methods

increasing efficiency increasing generality

Gauß-Elimination
 Gauß-Seidel
 SOR
 ADI
 Buneman



Comparison with MG



Poisson equation:

symmetric differencing (2nd order)

unit square: $h=1/n$

unknowns: $N = n^2$

Solution up to discretization error $\epsilon = O(h^2)$

method	# operations	time, $h=1/256$
Gauss-Elimin.	$O(N^2)$	approx. 1 day
SOR	$O(N^{3/2}) \log \epsilon$	approx. 30 min
ADI	$O(N \log N) \log \epsilon$	4 min
Buneman	$O(N \log N)$	15.5 sec
MG (iterative)	$O(N) \log \epsilon$	18.5 sec
MG (FMG)	$O(N)$	7.6 sec

$\log \epsilon = O(\log N)$

MG1-3

Range of MG applicability

General domains
General boundary conditions
Variable coefficients
Linear / non-linear problems
Scalar problems / systems of PDEs
Singular / nearly singular problems
Eigenvalue problems

Local refinements
Local coordinates

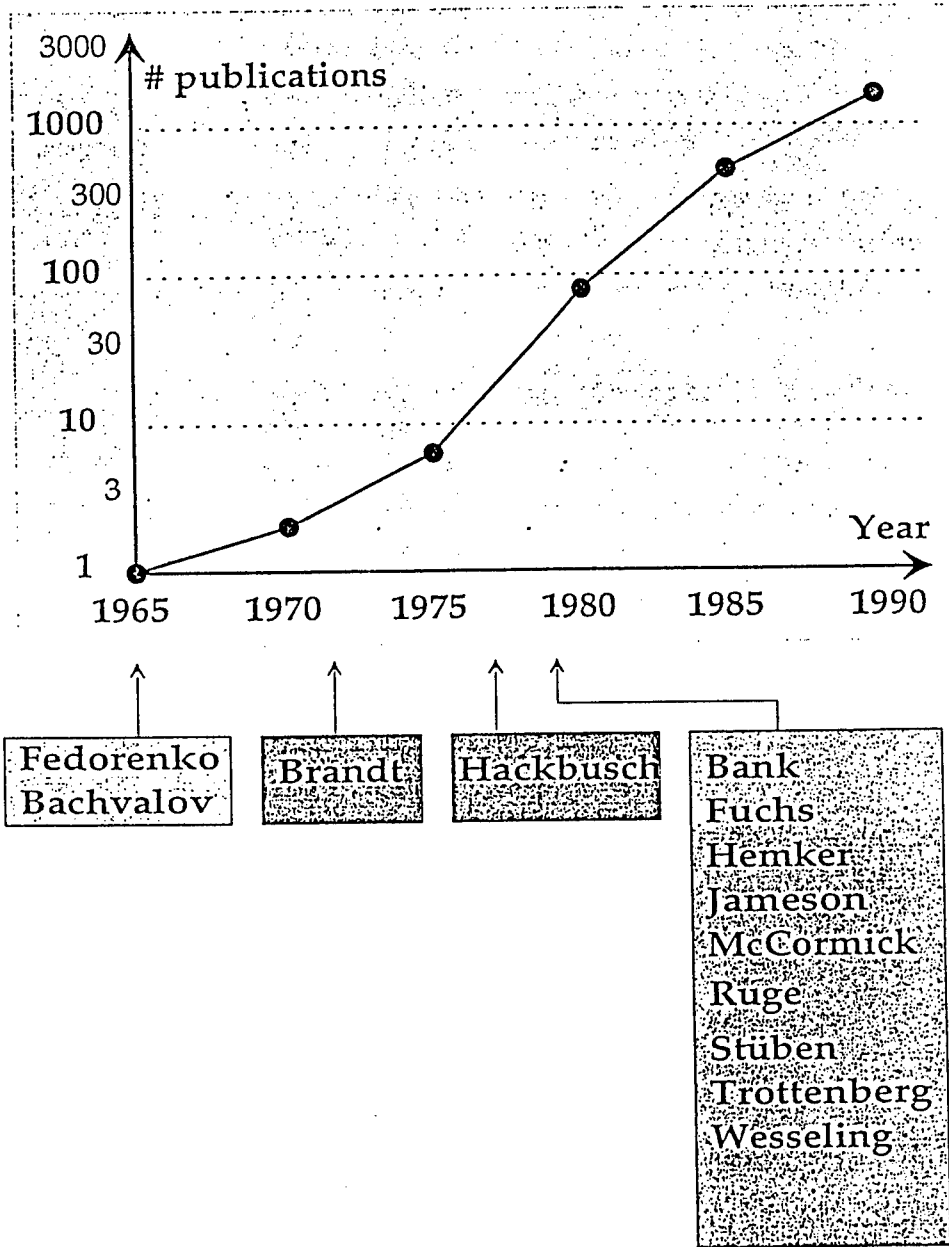
Finite differences
Finite volumes
Finite elements

However,

... the development of concrete multigrid methods
for complex problems may be non-trivial:

The "components" of a multigrid method have to be
adjusted to particular situations:

Historical development



MG1-5

Pioneering papers

Discovery of multigrid (theoretical)

Fedorenko, R.P.: *The speed of convergence of an iterative process*,
USSR Comput. Math. and Math. Phys. 4,3 (1964)

Bakhvalov, N.S.: *On the convergence of a relaxation method with
natural constraints on the elliptic operator*,
USSR Comput. Math. and Math. Phys. 6,5 (1966)

Papers marking the "beginning" of multigrid

Brandt, A.: *Multi-level adaptive technique (MLAT) for fast
numerical solution to boundary value problems*,
Lecture Notes in Physics 18, Springer (1973)

Brandt, A.: *Multi-level adaptive solutions to boundary-value
problems*, Math. Comp. 31 (1977)

Re-discovery of multigrid (theoretical)

Hackbusch, W.: *On the multigrid method applied to difference
equations*, Computing 20 (1978)

Introductions to MG

- Stüben, K.; Trottenberg, U.: *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, Lecture Notes in Mathematics 960, Springer (1982)
- Brandt, A.: *Multigrid techniques: 1984 Guide with applications to fluid dynamics*, GMD-Studie No. 85 (1984)
- Hackbusch, W.: *Multigrid methods and applications*, Springer Series in Comp. Math. 4, Springer (1985)
- McCormick, S. (ed.): *Multigrid methods*, Frontiers in Applied Mathematics, Vol. 5, SIAM, Philadelphia (1987)
- Briggs, W.: *A multigrid tutorial*, SIAM, Philadelphia (1987)
- Wesseling, P.: *An introduction to multigrid methods*, Pure and Applied Mathematics series, John Wiley and Sons (1992)
- Joppich, W.; Mijalkovic, S.: *Multigrid Methods for Process Simulation*, Computational Microelectronics 17, Selberherr (ed.), Springer Verlag, 1993

Proceedings

Copper Mountain MG Conference

- 1st: Appl. Math. Comp. 13 (1983)
- 2nd: Appl. Math. Comp. 19 (1986)
- 3rd: Lecture Notes in Pure and Appl. Math. 110,
Marcel Dekkar (1988)
- 4th: SIAM, Philadelphia (1989)
- 5th: Communications in Applied Numerical Methods;
Special Issue on MG, Vol 8, No 9&10,
John Wiley & Sons, New York (1992)
- 6th: NASA Conference Publication 3224, Parts 1&2,
NASA, Hampton, Virginia (1993)
- 7th: ??

European MG Conference

- 1st: Lecture Notes in Mathematics 960, Springer (1982)
- 2nd: Lecture Notes in Mathematics 1228, Springer (1986)
- 3rd: International Series of Numerical Mathematics, Vol. 98,
Birkhäuser (1991)
- 4th: International Series of Numerical Mathematics, Vol. 116,
Birkhäuser (1994)

MG1-8

The three basic principles of multigrid

Relaxation
Coarse-grid correction
Nested iteration

None of these principles yields an efficient method by itself!

First step:

Multigrid iteration ("cycling")

Relaxation + Coarse-grid correction

Second step:

Full multigrid (FMG)

Multigrid cycling + Nested iteration

Relaxation as a solver

Basic principle:

Given: $A X = b$ Splitting: $A = A_1 + A_2$

One iteration step $x \rightarrow \bar{x}$:

$$A_1 \bar{x} + A_2 x = b$$

$$\rightarrow \bar{x} = -A_1^{-1} A_2 x + A_1^{-1} b$$

$\rho(-A_1^{-1} A_2) =$ asympt. conv. factor (spectral radius)

Typical PDE situation

$$A = \begin{array}{|c|} \hline & U \\ \hline L & D \\ \hline \end{array}$$

Jacobi relaxation:

$$A_1 = D, \quad A_2 = L + U$$

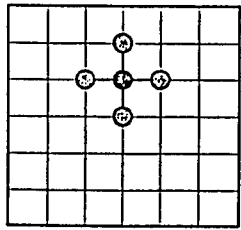
Gauss-Seidel relaxation:

$$A_1 = D + L, \quad A_2 = U$$

$$\rho = 1 - O(h^2) \quad \underline{h - dependent!}$$

**Relaxation methods are
very inefficient solvers!**

Relaxation as a smoother



Poisson equation

$$\frac{1}{h^2} \begin{bmatrix} & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & & & \\ & & & & \\ & & & & \end{bmatrix}_h U^h = f^h$$

Pointwise Gauss-Seidel relaxation

For all grid points do

$$u_{ij}^h = \frac{1}{4} \left[h^2 f_{ij}^h + u_{i-1,j}^h + u_{i+1,j}^h + u_{i,j-1}^h + u_{i,j+1}^h \right]$$

Effect on *error* $v^h = U^h - u^h$

$$v_{ij}^h = \frac{1}{4} \left[v_{i-1,j}^h + v_{i+1,j}^h + v_{i,j-1}^h + v_{i,j+1}^h \right]$$

→ averaging process

The *error* gets *smooth* very quickly!

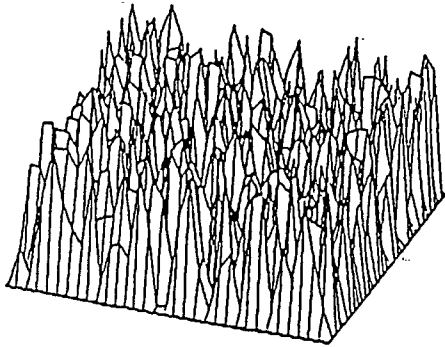
Relaxation processes are local:

Locally, information is spread quickly

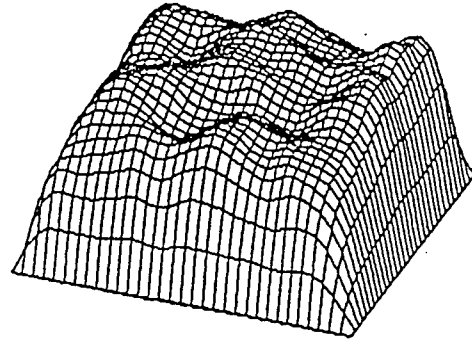
Globally, information is spread slowly
(in particular, boundary values!)

Influence of (pointwise) Gauss-Seidel relaxation on the error

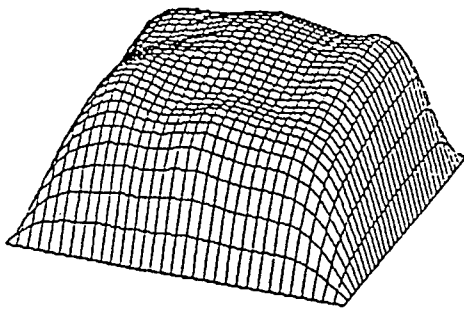
Poisson equation, *uniform grid*



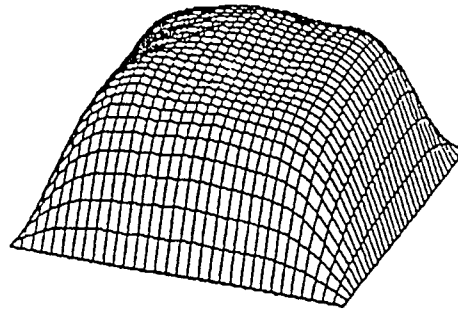
Error of initial guess



Error after 5 relaxations



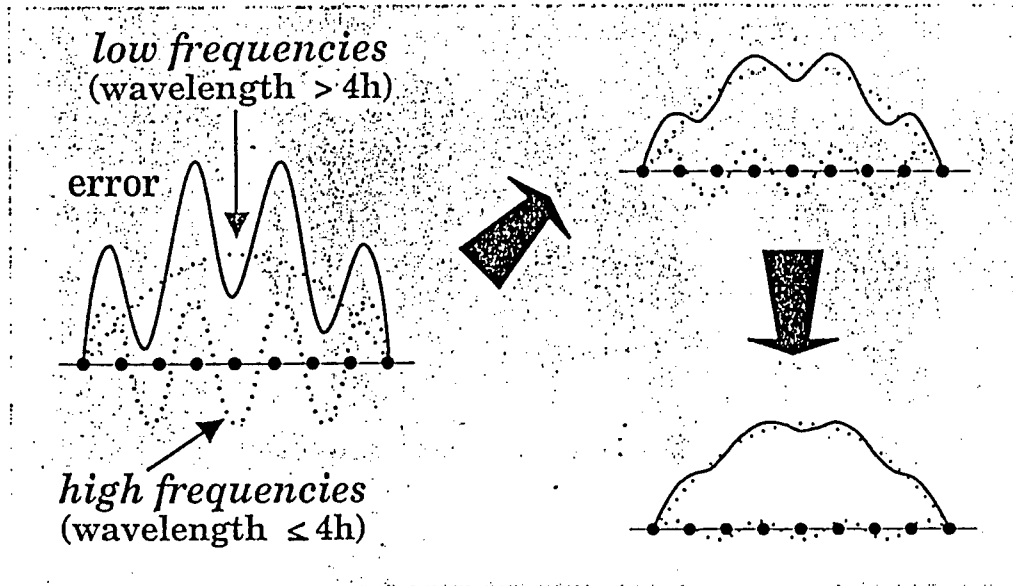
Error after 10 relaxations



Error after 15 relaxations

Smoothing analysis: Fourier decomposition

Fourier decomposition of *error* (1D)



High frequency reduction factor per sweep
("Smoothing factor" μ)

Independent of h !

Analysis by (local) Fourier analysis

Poisson equation: GS: $\mu = 0.5$, RB: $\mu = 0.25$

Efficient smoothers exist for all "sufficiently elliptic"
PDEs (or systems of PDEs)

Typical reduction: 1 order of magnitude in 2-4 sweeps

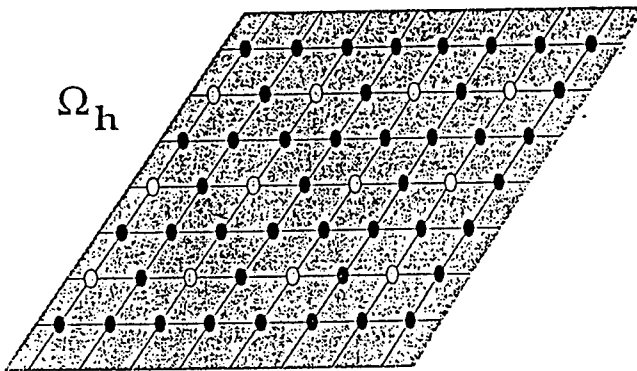
MG3-4

Hierarchy of discretizations

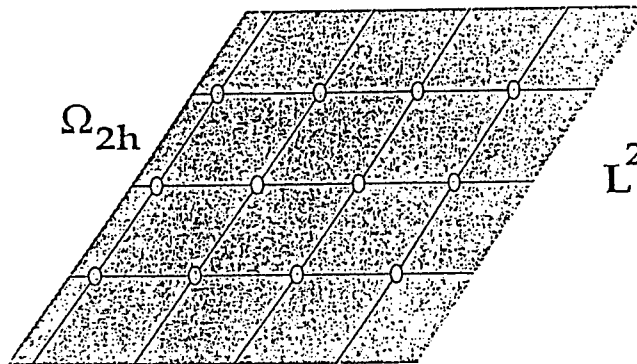
fine grid: Ω_h coarser grid: Ω_H

$$\begin{aligned} L^h U^h &= f^h && \text{difference equations on } \Omega_h \\ L^H U^H &= f^H && \text{difference equations on } \Omega_H \end{aligned}$$

Standard coarsening ($H=2h$):



$$L^h = \frac{1}{h^2} \begin{bmatrix} & & -1 & & \\ & & -1 & 4 & -1 \\ & & & & \\ -1 & 4 & -1 & & \\ & & & & \end{bmatrix}_h$$



$$L^{2h} = \frac{1}{4h^2} \begin{bmatrix} & & -1 & & \\ & & -1 & 4 & -1 \\ & & & & \\ -1 & 4 & -1 & & \\ & & & & \end{bmatrix}_{2h}$$

MG2-2

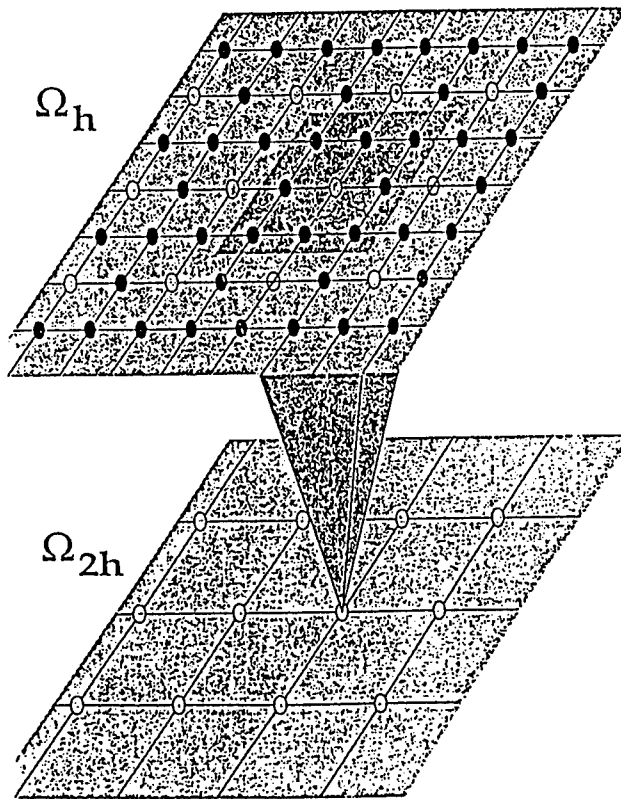
Data transfer between different grids

fine grid: Ω_h coarser grid: Ω_H

I_h^H transfer from Ω_h to Ω_H : *restriction*

I_H^h transfer from Ω_H to Ω_h : *prolongation*

Standard coarsening ($H=2h$):



Restriction:
("full weighting")

$$I_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Prolongation:
(linear interpolation)

$$I_{2h}^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Coarse-grid correction

Given: $L^h U^h = f^h$ approximate solution: u^h

Correction equation:

$$L^h V^h = d^h \quad \text{with} \quad d^h := f^h - L^h u^h \quad (= \text{residual})$$

$$\longrightarrow U^h = u^h + V^h$$

Given in addition: Ω_H and L^H

Coarse-grid correction:

$$d^h := f^h - L^h u^h$$

$$u^h \leftarrow u^h + v^h$$

$$d^H := I_h^H d^h$$

(restriction)

$$v^h := I_H^h V^H$$

(prolongation)

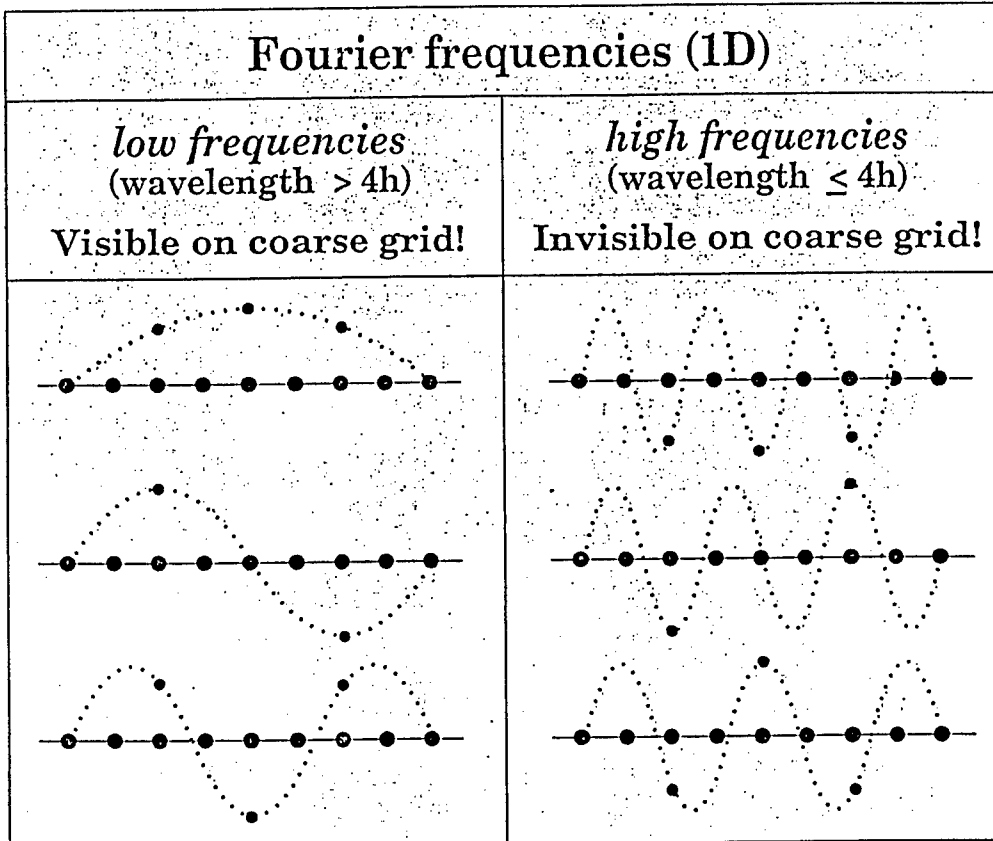
$$\text{Solve: } L^H V^H = d^H$$

This process, applied iteratively,
CANNOT converge!

$$I_h^H d^h = 0 \longrightarrow \text{no correction!}$$

Coarse-grid aliasing of Fourier components

Assume: Fourier decomposition of *error*

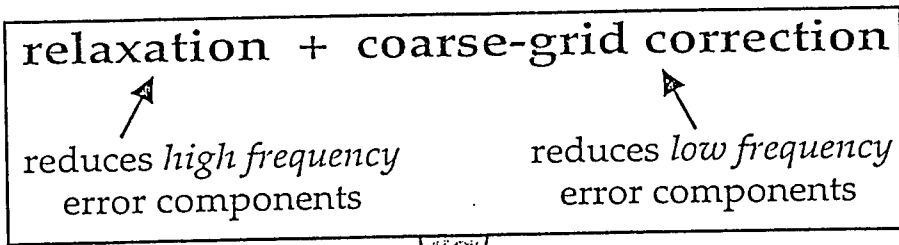


High frequency Fourier components
CANNOT be corrected by coarse grid!

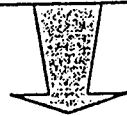
However:

Coarse-grid correction makes sense, if
low frequencies are dominating in the *error*!

The (iterative) two-grid cycle

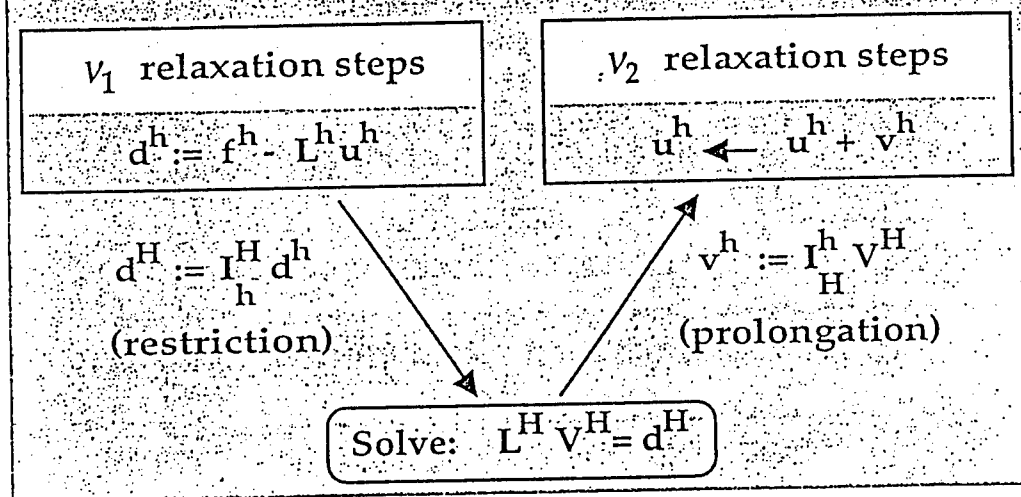


v_1 : pre-smoothing



v_2 : post-smoothing

One two-grid iteration step (cycle):



Convergence independent of h !

v_1, v_2 small (typically 1 or 2)

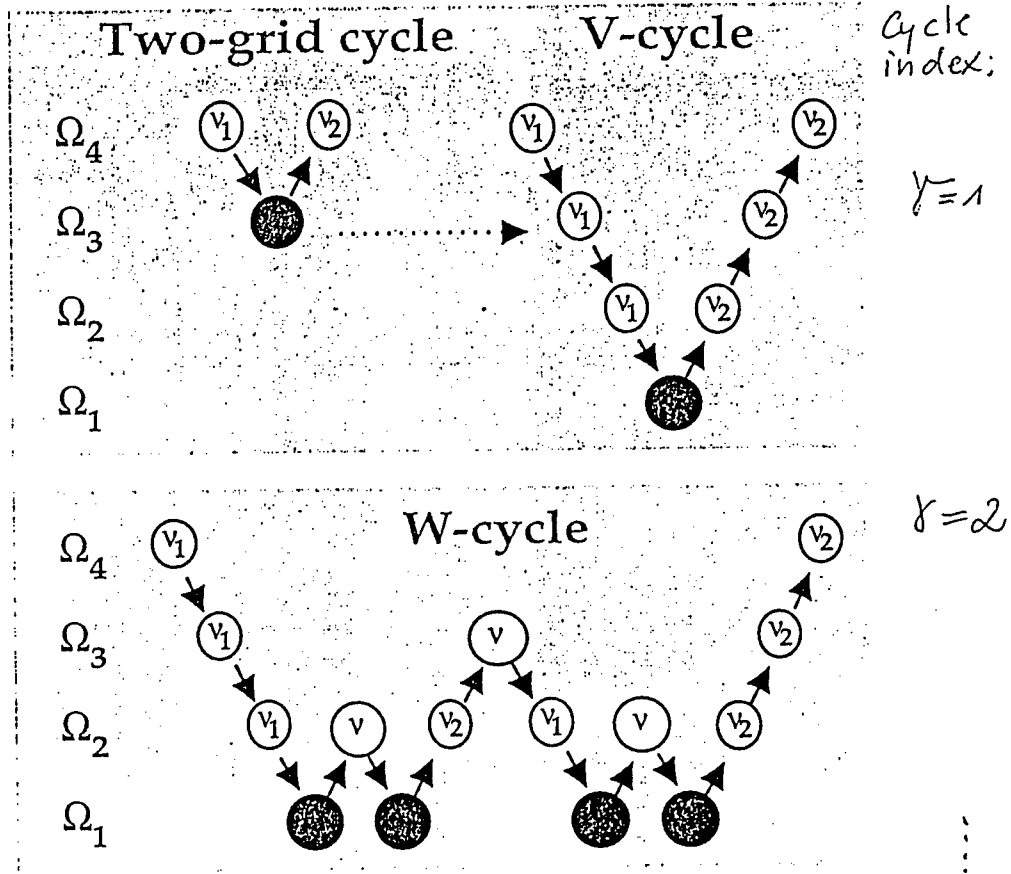
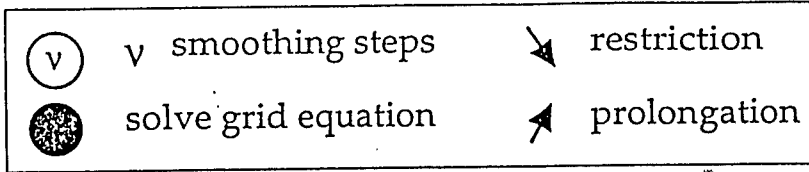
Rough convergence prediction: smoothing factor

Analysis by (local) Fourier analysis

However: Two-grid method is not yet efficient!

Typical multigrid cycles

Notation:



Notation: $V(v_1, v_2)$ - cycle, $W(v_1, v_2)$ - cycle

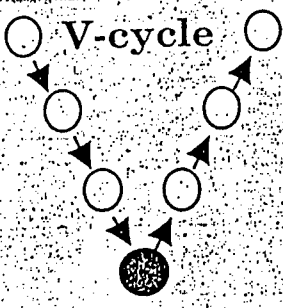
Compromise: F-cycle

MG4-4

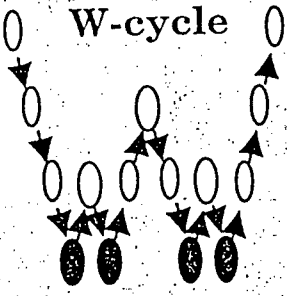
Work count

$\mathcal{W} := 1$ relaxation sweep on finest grid

$v = v_1 + v_2, \quad \text{standard coarsening}$

	2D	3D
 <p style="text-align: center;">V-cycle</p>	$v \mathcal{W}$ $v \mathcal{W} / 4$ $v \mathcal{W} / 16$ \vdots <hr style="width: 80%; margin: 0 auto;"/> $4/3 v \mathcal{W}$	$v \mathcal{W}$ $v \mathcal{W} / 8$ $v \mathcal{W} / 64$ \vdots <hr style="width: 80%; margin: 0 auto;"/> $8/7 v \mathcal{W}$

$v = 3: \quad 4 \mathcal{W} \quad 3.5 \mathcal{W}$

 <p style="text-align: center;">W-cycle</p>	$v \mathcal{W}$ $v \mathcal{W} / 2$ $v \mathcal{W} / 4$ \vdots <hr style="width: 80%; margin: 0 auto;"/> $2 v \mathcal{W}$	$v \mathcal{W}$ $v \mathcal{W} / 4$ $v \mathcal{W} / 16$ \vdots <hr style="width: 80%; margin: 0 auto;"/> $4/3 v \mathcal{W}$
---	---	--

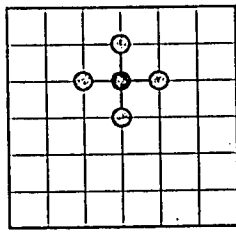
$v = 3: \quad 6 \mathcal{W} \quad 4 \mathcal{W}$

**Typical: Error reduction by 1 order of magnitude
with work equivalent to 5-10 \mathcal{W}**

Solution up to discretization error ϵ

$$O(N) \log \epsilon = O(N) \log N$$

Standard MG for Poisson's equation



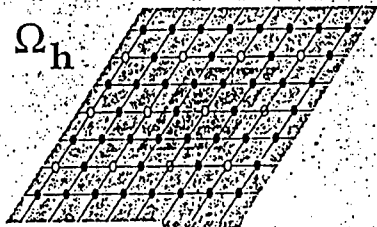
h

Poisson equation

$$\frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} U^h = f^h$$

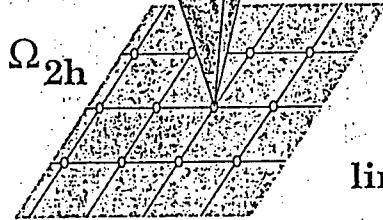
Relaxation: GS

L^h
standard (h)



I_h^{2h}
full weighting

L^{2h}
standard (2h)



I_{2h}^h
linear interpolation

Convergence factors ($h=1/128$, 7 grid levels)

v_1, v_2	μ^v	W-cycle	V-cycle
1, 0	0.500	0.395	0.400
1, 1	0.250	0.190	0.190
2, 1	0.125	0.117	0.120

Summary on MG components

<i>Discretization</i>	MG straightforward if "sufficiently elliptic"
<i>Relaxation</i>	Gauss-Seidel type (point-, line-, blockwise); collective and distributive relaxations; ILU-iterations
<i>Coarsening strategy</i>	Standard coarsening (h,2h,4h,...); Semi- or adaptive coarsening; Multi-coarsening; Variational coarsening
<i>Coarse-grid operators</i>	Analogous on all grids; Galerkin operators: $L^H = I_h^H L^h I_H^h$
<i>Prolongation</i>	Linear interpolation; "Adaptive" interpolation (e.g. discrete eqns.)
<i>Restriction</i>	Local averaging (e.g. full weighting); Transpose of prolongation; "Adaptive" weighting
<i>Cycle-Typ</i>	V-, F-, W-cycle; "Adaptive" cycles
ν_1, ν_2	Typically 1 or 2, usually $\nu_1 \geq \nu_2$; dynamically chosen

Relevant for non-PDE problems:
Variational coarsening

Optimized MG for Poisson's equation

GS-relaxation + straight injection

$$I_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \approx \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix}$$

Convergence factors

v_1, v_2	μ^v	W-cycle	V-cycle
1, 0	0.500	0.440	<i>div</i>
1, 1	0.250	0.197	0.199
2, 1	0.125	0.088	0.089

Residuals do not change much locally

RB-relaxation + half injection

$$I_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \approx \frac{1}{16} \begin{bmatrix} 0 & 2 & 0 \\ 2 & 8 & 2 \\ 0 & 2 & 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \end{bmatrix}$$

Convergence factors

v_1, v_2	μ^v	W-cycle	V-cycle
1, 0	0.250	0.487	0.809
1, 1	0.063	0.118	0.173
2, 1	0.016	0.034	0.059

Residuals vanish at black points

Residuals do not change much locally at the red points

Variational Coarsening

$$\Leftrightarrow AU = b \quad \text{on grid } h, \quad A \text{ symm. pos. def.}$$

$$\frac{1}{2} U^T A U - U^T b = \min!$$

$I_{2h}^h V^{2h}$ coarse-grid correction
to u : $U \approx u + I_{2h}^h V^{2h}$

↓

minimize

$$\frac{1}{2} (u + I_{2h}^h V^{2h})^T A (u + I_{2h}^h V^{2h}) - (u + I_{2h}^h V^{2h})^T b$$

$$\frac{1}{2} V^{2h T} \underbrace{I_{2h}^h T A I_{2h}^h}_{A^{2h}} V^{2h} - V^{2h T} \underbrace{I_{2h}^h T (b - Au)}_{R^{2h}}$$

⇔

$$A^{2h} V^{2h} = R^{2h} \quad \text{where}$$

$$A^{2h} = I_{2h}^h T A I_{2h}^h, \quad R^{2h} = I_{2h}^h T (b - Au)$$

Symmetric, Conservative

Full Weighting

Variational coarsening

Given: L^h, Ω_h, Ω_H and I_H^h

Define: $I_h^H := (I_H^h)^T$ and $L^H := I_h^H L^h I_H^h$

Galerkin operator

L^h symmetric $\longrightarrow L^H$ symmetric

L^h pos. definite $\longrightarrow L^H$ pos. definite

Variational principle (L^h pos. definite)

$v^h := U^h - u^h$ error *before* coarse-grid correction

$\bar{v}^h := U^h - \bar{u}^h$ error *after* coarse-grid correction

Then: $\|\bar{v}^h\| = \min_{\delta^h \in \mathcal{R}} \|v^h + \delta^h\|$

$\|\cdot\|$ energy norm $\mathcal{R} := \text{range}(I_H^h)$

Remarks:

Applicable to non-PDE problems

Only applicable for linear problems

The computation of L^H is relatively expensive

Origin: finite elements

Proof of the variational principle

$$\langle u^h, w^h \rangle_h := (L^h u^h, w^h) \quad \text{energy inner product}$$

$$\mathcal{R} := \text{range}(I_H^h), \quad \mathcal{R}^\perp := \text{orthogonal complement}$$

Two-level
correction operator:
$$K^h = I^h - I_H^h (L^H)^{-1} I_h^H L^h$$

Properties:

K^h is symmetric w.r.t. $\langle \cdot, \cdot \rangle$

$$v^h \in \mathcal{R} \quad \longrightarrow \quad K^h v^h = 0$$

$$v^h \in \mathcal{R}^\perp \quad \longrightarrow \quad K^h v^h = v^h$$

$$\begin{aligned} \longrightarrow \quad & K^h \text{ is orthogonal projector onto } \mathcal{R}^\perp \\ & I^h - K^h \text{ is orthogonal projector onto } \mathcal{R} \end{aligned}$$

$$\begin{aligned} \longrightarrow \quad & \|\bar{v}^h\| = \|K^h v^h\| \\ & = \|v^h - (I^h - K^h)v^h\| \\ & = \min_{\delta^h \in \mathcal{R}} \|v^h + \delta^h\| \end{aligned}$$

Model problems

The performance of a concrete multigrid method is not sensitive to modifications in the problem, unless its "nature" changes essentially.

Some typical model problems

Poisson equation $-u_{xx} - u_{yy} = f$

Anisotropic equations $-\varepsilon u_{xx} - u_{yy} = f$
 $0 < \varepsilon \ll 1$ or $\varepsilon \gg 1$

Discontinuous coefficients $-(au_x)_x - (bu_y)_y = f$

Singular perturbations $-\varepsilon \Delta u + au_x + bu_y = f$
 $0 < \varepsilon \ll 1$

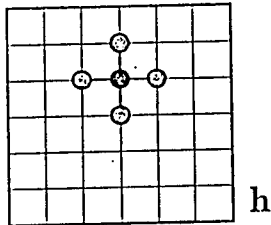
Indefinite problems $-\Delta u - cu = f$

Nearly-singular problems $c > 0$

Eigenvalue problems

Higher order equations $\Delta \Delta u = f$

Anisotropic problems (2D)



$$-\varepsilon U_{xx} - U_{yy} = f(x,y)$$

$$\mathbf{L}^h = \frac{1}{h^2} \begin{bmatrix} & & -1 & & \\ & & & & \\ -\varepsilon & & 2(1+\varepsilon) & & -\varepsilon \\ & & & & \\ & & & & -1 \end{bmatrix}_h$$

Gauss-Seidel relaxation

For all grid points do

$$u_{ij}^h = \frac{1}{2(1+\varepsilon)} \left[h^2 f_{ij}^h + \varepsilon u_{i-1,j}^h + \varepsilon u_{i+1,j}^h + u_{ij-1}^h + u_{ij+1}^h \right]$$

Smoothing analysis (standard coarsening):

$$\mu \rightarrow 1 \quad \text{if } \varepsilon \rightarrow 0 \quad \text{or } \varepsilon \rightarrow \infty$$

Behavior of error $v^h = U^h - u^h$ during relaxation:

$$v_{ij}^h = \frac{1}{2(1+\varepsilon)} \left[\varepsilon v_{i-1,j}^h + \varepsilon v_{i+1,j}^h + v_{ij-1}^h + v_{ij+1}^h \right]$$

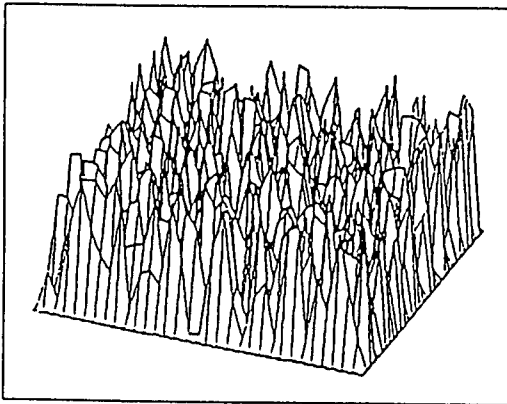
If $0 < \varepsilon \ll 1$: Smoothing only in y-direction!

If $\varepsilon \gg 1$: Smoothing only in x-direction!

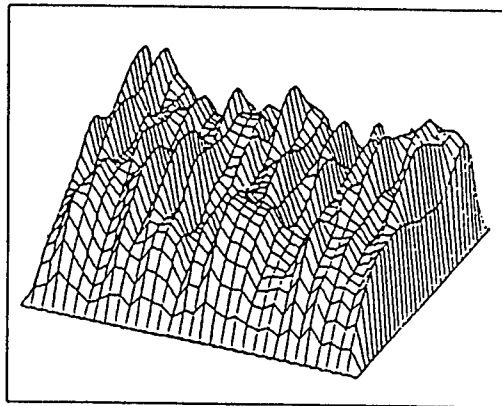
Influence of (pointwise) Gauss-Seidel relaxation on the error

$$-\varepsilon U_{xx} - U_{yy} = f, \quad 0 < \varepsilon \ll 1$$

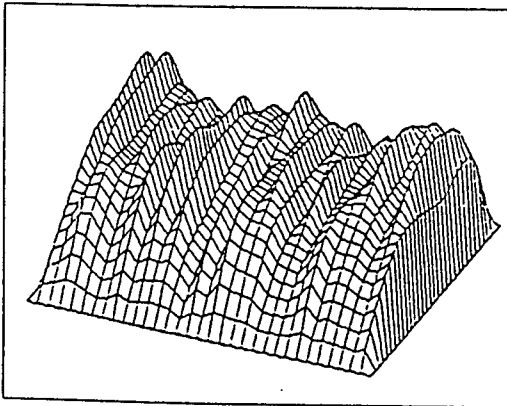
uniform mesh



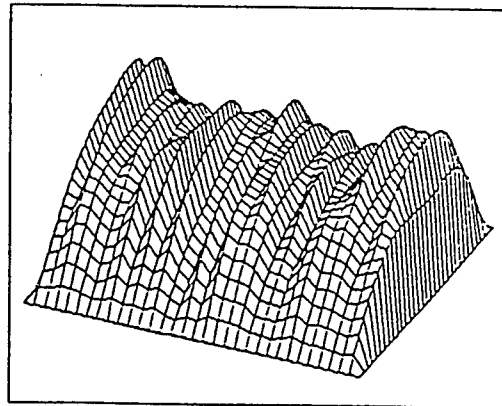
Error of initial guess



Error after 5 relaxations



Error after 10 relaxations



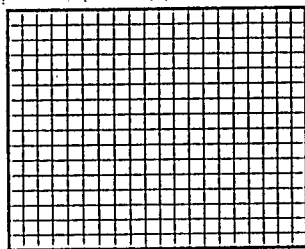
Error after 15 relaxations

Anisot-3
H 663-7

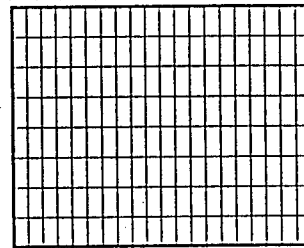
Remedy 1: Semi-coarsening

Use Gauss-Seidel (pointwise) relaxation BUT
 coarsen only in the direction of error smoothness
 (i.e. in the direction of *strong couplings*).

For example: $0 < \varepsilon \ll 1$



→
y-line
 coarsening



$$\frac{1}{h^2} \begin{bmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}$$

$$\frac{1}{4h^2} \begin{bmatrix} & -1 & \\ -4\varepsilon & 2(1+4\varepsilon) & -4\varepsilon \\ & -1 & \end{bmatrix}$$

The anisotropy *gets weaker* on coarser grids!

What to do for *arbitrarily varying* $\varepsilon = \varepsilon(x,y)$?

- **Adaptive coarsening**
 Technically rather complicated (cf. AMG)
- **"Multiple" coarsening**
 Use more than one coarse grid on the same level,
 and combine the corresponding corrections properly.

Remedy 2: Line relaxation

Use standard coarsening BUT

relax all strongly coupled points simultaneously
(line relaxation in the direction of strong couplings)

If $0 < \varepsilon \ll 1$: Gauss-Seidel y-line relaxation

If $\varepsilon \gg 1$: Gauss-Seidel x-line relaxation

What to do for *arbitrarily varying* $\varepsilon = \varepsilon(x,y)$?

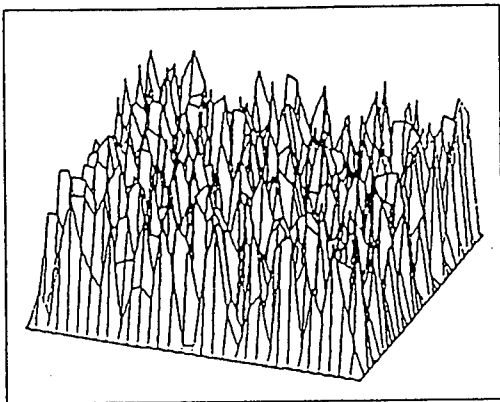
Alternating line relaxation

One step of x-line relaxation followed by one
one step of y-line relaxation.

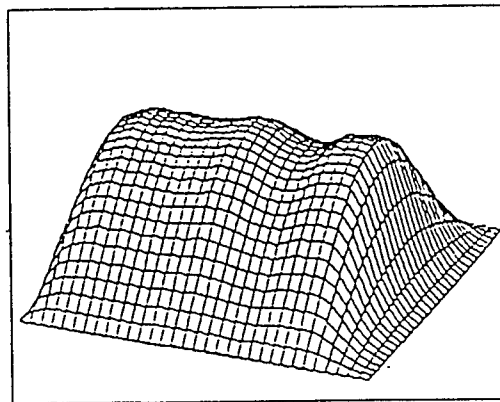
Influence of Gauss-Seidel y-line relaxation on the error

$$-\varepsilon U_{xx} - U_{yy} = f, \quad 0 < \varepsilon \ll 1$$

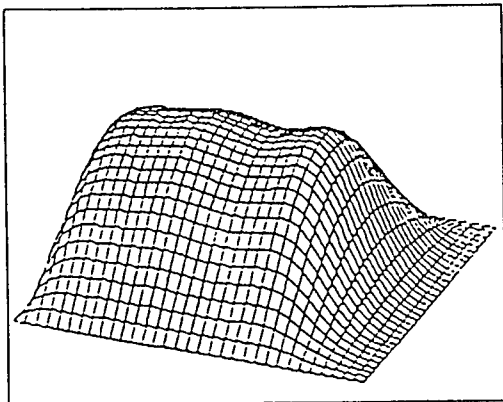
uniform mesh



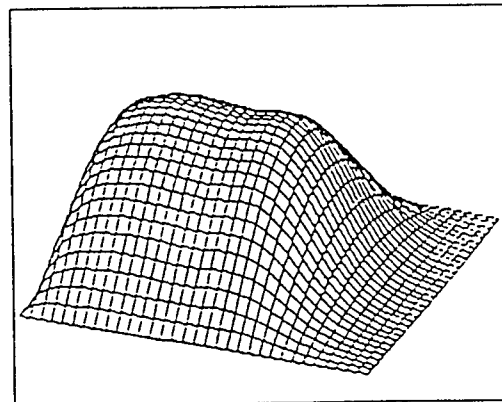
Error of initial guess



Error after 5 relaxations



Error after 10 relaxations



Error after 15 relaxations

Aniso1.6
17663-6

A. Brandt

Multigrid Monte-Carlo and
Stochastic Coarsening

OBJECTIVES

MS

Energy minimization

- Near-minimum start
- Far start
- Homogenization

MG, AMG
ms annealing

Equilibrium statistics

- $O(n)$ per sample
- $O(1)$ per sample
- ↓
- Homogenization

ms
Monte-Carlo

Dynamics

- Large Δt
- Very large Δt

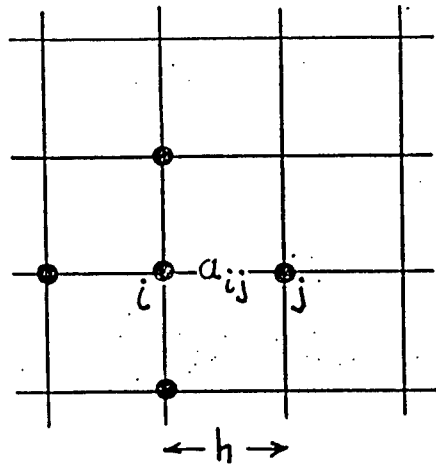
MG, AMG
ms annealing

Stochastic dynamics

ms Monte-Carlo at each time step



$$E(u) = \frac{1}{2} \sum_{\langle i,j \rangle} a_{ij} \left(\frac{u_i - u_j}{h} \right)^2 - \sum_i f_i u_i$$



$$E(U) = \min_u E(u)$$

$$0 = \frac{\partial E}{\partial u_i} \Big|_{u=U} = \frac{1}{h^2} \sum_{\langle j,i \rangle} a_{ij} (u_i - u_j) - f_i$$

$$\frac{1}{h^2} \sum_{\langle j,i \rangle} a_{ij} (U_i - U_j) = f_i$$

$$\rightarrow \frac{\partial}{\partial x} \left(a \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(a \frac{\partial U}{\partial y} \right) = f$$

$$\Rightarrow E(U) = \min E(u)$$

$$E(u) = \iint \left[\frac{a}{2} (u_x^2 + u_y^2) - f u \right] dx dy$$

Physical distribution

$$P_E(u) = \frac{1}{Z} e^{-\beta E(u)}, \quad \beta = \frac{1}{k_B T}$$

$$Z = \sum_u e^{-\beta E(u)}$$

Observables

$$E(u)$$

$$M(u) = \sum_i u_i, \quad M(u)^2$$

Averages

$$\langle M \rangle = \sum_u P_E(u) M(u)$$

$$\langle M^2 \rangle, \quad \langle E \rangle$$

How to compute?

Monte-Carlo Markov chains (MC)

$$u^{(0)}, u^{(1)}, \dots, u^{(n)}, \dots$$

$$P_n(u) = P(u^{(n)} = u)$$

$$= \sum_{u'} P_{n-1}(u') \underline{P_{n-1}^n(u' \rightarrow u)}$$

$P_0(u)$ given (e.g., random)

Detailed balance (DB)

$$\frac{P_{n-1}^n(u' \rightarrow u)}{P_{n-1}^n(u \rightarrow u')} = \frac{P_E(u)}{P_E(u')} \quad \text{or} \quad \frac{0}{0}$$

$$\Rightarrow \lim_{n \rightarrow \infty} P_n(u) = P_E(u) \quad \forall u$$

$$\langle M \rangle = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{n=1}^k M(u^{(n)})$$

Metropolis

current $u \leftrightarrow$ one candidate u'

$$P(u \rightarrow u') = \min\left(1, e^{\beta[E(u) - E(u')]} \right)$$

\Rightarrow Detailed balance

Point-by-point Metropolis

extremely slow to produce a new independent sample

A. Simple Energy Basins real u_i

Slow to equilibrate
and be sampled: $I_H^h V^H$

Coarse Monte-Carlo
Hamiltonian: $E^h(\tilde{u}^h + I_H^h V^H)$
 $\equiv E^H(V^H)$

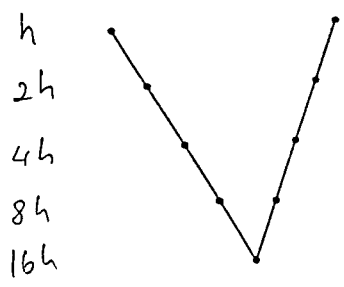
Multigrid cycle:

- on each level:
1. Monte-Carlo passes
 2. γ^* cycles on coarser
 3. Monte-Carlo passes

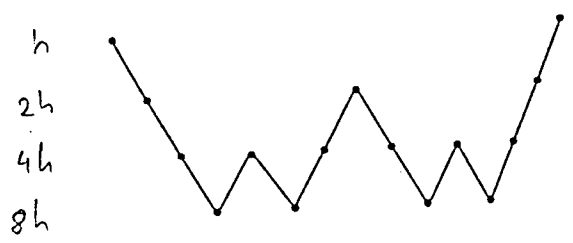
Near equilibration and decorrelation
in one cycle ($\gamma^* = 2$).

Small work on coarser grids ($\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$)

$$\gamma = 1$$



$$\gamma = 2$$

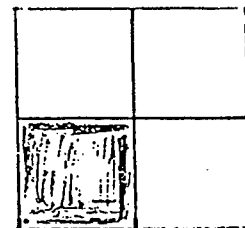


Independent sampling
on coarser levels



1. Large-scale statistics:
accumulated on coarser levels
in $\delta = 2^d$ cycle

2. Vast domains:
only on coarser levels
by domain replication
(instead of $\delta = 2^d$)



3. Macroscopic dynamics:
cheaply, directly

Hamiltonian Simplification

$$H(u) = H'(u) + \Delta(u), \quad \tilde{u} \xrightarrow{H'} u$$

1. Apriori lottery

Delete Δ in probability $P_{\Delta}(\tilde{u}) \sim e^{-\Delta(\tilde{u})/T}$
Freeze Δ in probability $1 - P_{\Delta}(\tilde{u})$

Local.

Allows coarse dynamics.

Freezing should correspond to I_H^h

2. Aposteriori lottery

$$P_{\text{accept}}(\tilde{u} \rightarrow u) = \begin{cases} 1 & \text{if } \Delta(\tilde{u}) > \Delta(u) \\ e^{\Delta(\tilde{u}) - \Delta(u)} & \end{cases}$$

Global, convenient

Requires small $\Delta(u)$ for smooth $u - \tilde{u}$

Prohibits coarse dynamics?

B. Ising Spins

$$u_i = \pm 1$$

$$E(u) = - \sum_{\langle i, j \rangle} J_{ij} u_i u_j$$

$$P(u) \sim e^{-E(u)/T}$$

- Coarse moves = block flips.
- Standard blocks are improbable.
- Blocks should have uniform sign.
- Basing blocks on current u disrupts statistical fidelity

$$\frac{P(u_1 \rightarrow u_2)}{P(u_2 \rightarrow u_1)} = \frac{P(u_2)}{P(u_1)} \quad (\text{detailed balance}).$$

Stochastic coarsening at \tilde{u}

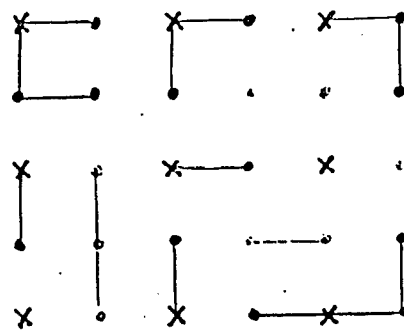
- Delete $V_{ij}(u) = -J_{ij} u_i u_j$ from E
in probability $p_{ij} = q_{ij} e^{-V_{ij}(\tilde{u})/T}$
- Freeze V_{ij} (block i with j) if undeleted.

→ detailed balance. [Swendsen-Wang]

Process all interactions.

Also between blocks.

Except between any
two "coarse spins".



► Coarse Hamiltonian $E^1 = -\sum J_{ij}^1 U_i U_j$
+ "independent blocks".

Repeat: Hamiltonians $E^0, E^1, E^2, E^3, \dots$
On meshsizes $1, 2, 4, 8, \dots$
+ accumulating list of independents.

μ -cycle (V, W, \dots)

► Multi-scale island dynamics.

Fast sampling. But no independence
of coarse samplings.

Detailed Balance

$$E(u) = -V(u) + E'(u)$$

Probability of deleting V given \tilde{u} :

$$p_V(\tilde{u}) = q_V e^{-V(\tilde{u})/T} = q_V \frac{z P_{E'}(\tilde{u})}{z P_E(\tilde{u})}$$

otherwise: freeze V

If $V(\tilde{u}) \neq V(u)$:




$$\begin{aligned} \frac{P(\tilde{u} \rightarrow u)}{P(u \rightarrow \tilde{u})} &= \frac{p_V(\tilde{u}) P_{E'}(\tilde{u} \rightarrow u)}{p_V(u) P_{E'}(u \rightarrow \tilde{u})} = \frac{p_V(\tilde{u})}{p_V(u)} \frac{P_{E'}(u)}{P_{E'}(\tilde{u})} \\ &= \frac{P_E(u)}{P_E(\tilde{u})} \end{aligned}$$

If $V(\tilde{u}) = V(u)$:

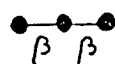
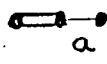
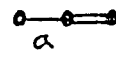

$$p_V(\tilde{u}) = p_V(u) = p$$

$$\begin{aligned} \frac{P(\tilde{u} \rightarrow u)}{P(u \rightarrow \tilde{u})} &= \frac{p P_{E'}(\tilde{u} \rightarrow u) + (1-p) \hat{P}_E(\tilde{u} \rightarrow u)}{p P_{E'}(u \rightarrow \tilde{u}) + (1-p) \hat{P}_E(u \rightarrow \tilde{u})} \\ &= \frac{P_{E'}(u)}{P_{E'}(\tilde{u})} = \frac{P_E(u)}{P_E(\tilde{u})} \end{aligned}$$

2-spin coarsening (SW)

2-spin configuration	given H	alternative		transition pr.	
		H ₁	H ₂	P ₁	P ₂
++	$-\beta$	∞	0	$1 - e^{-2\beta}$	$e^{-2\beta}$
+-	β	0	0	0	1
					

3-spin coarsening (example)

3-spin configur.	given H	alternative			transition probab.		
		H ₁	H ₂	H ₃	P ₁	P ₂	P ₃
+++	-2β	$-a$	$-a$	$-b$	$\frac{1 - e^{-4\beta}}{2}$	$\frac{1 - e^{-4\beta}}{2}$	$e^{-4\beta}$
++-	2β	∞	∞	$-b$	0	0	1
+-+	0	a	∞	b	$1 - p_*$	0	p_*
---	0	∞	a	b	0	$1 - p_*$	p_*
							

detail balance $\Leftarrow e^{2a} = \frac{e^{2\beta} - e^{-2\beta}}{2(1 - p_*)}$, $e^{2b} = \frac{e^{-2\beta}}{p_*}$

GRID	fine level	$\langle M^2 \rangle$ coarse-level deviation	
		2-spin (sw) coarsening	3-spin ($p_* = .15$) coarsening
4 x 4	12.2	± 1.8	$\pm .7$
8 x 8	41.4	± 7.2	± 1.5
16 x 16	139.5	± 25.6	± 4.0
32 x 32	470.2	± 81.6	± 10.6

25% sites treated

Stochastic Dynamics

At time t^n : position vector x^n
velocity vector V^n
potential energy $E(x^n)$.

Deterministic implicit time step :

$$-\nabla E(x^{n+1}) = p^{n+1}$$

$$p_k^{n+1} = m_k \frac{\delta V_k}{\delta t}, \quad \delta V_k = V_k^{n+1} - V_k^n, \quad V_k^{n+1} = \frac{x_k^{n+1} - x_k^n}{\delta t}$$

$\Leftrightarrow x^{n+1}$ minimizes $H_n(x^{n+1})$

$$H_n(x^{n+1}) = \frac{1}{2} \sum m_k (\delta V_k)^2 + E(x^{n+1})$$

Stochastic time step :

$$P(x^{n+1}) \sim e^{-\beta H_n(x^{n+1})}$$

$$\beta = \frac{1}{k_B T}$$

Results of the stochastic step:

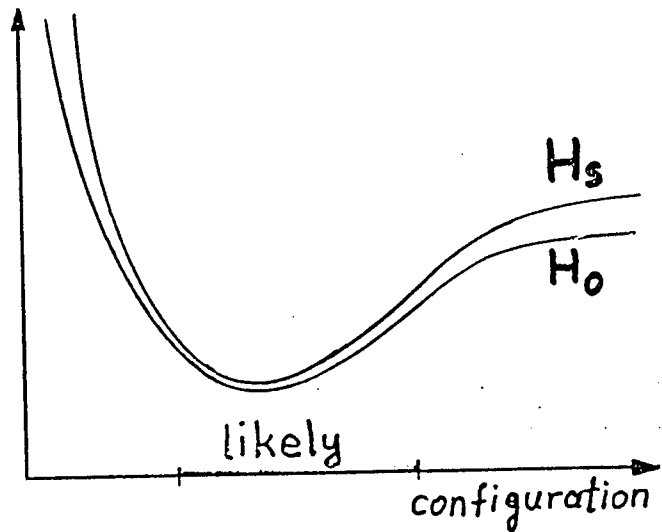
- $\langle p^{n+1} \rangle = -\nabla E(x^{n+1})$
- $\langle (p_k^{n+1})^2 \rangle - \langle p_k^n \rangle^2 = m_k / \beta$
- Local components, with
oscillation period $\ll \delta t$
are fully thermalized:
$$P(x^{n+1}) \sim e^{-E(x^{n+1})/k_B T}$$

- Large scale components, with
oscillation period $\gg \delta t$
satisfy Newton law.

- Easier multiscaling

Stochastic / Hamiltonian Simplification
coarse-

STOCHASTIC HAMILTONIAN SIMPLIFICATION



\mathcal{H}_0 = Original Hamiltonian
 obtained from fine grid \implies complicated

\mathcal{H}_s = Simplified Hamiltonian
 FAS: original form + polynomial terms

$$\mathcal{H}_s \geq \mathcal{H}_0, \mathcal{H}_s \approx \mathcal{H}_0$$

Detailed $P(H_0 \rightarrow H_s) = e^{\mathcal{H}_0 - \mathcal{H}_s} \Big|_{\text{current}}$

Balance: $P(\text{freeze } H_s - H_0) = 1 - P(H_0 \rightarrow H_s)$

Simpler: Approximate detailed balance

Thermodynamic limit

Statistical average of
an infinite system: $N \rightarrow \infty$

Calculation to accuracy $\epsilon \sigma$

\Rightarrow Large $N = n^d$ points

$\times n^2$ Monte-Carlo sweeps
to produce a sample

$\times O(\epsilon^{-2})$ samples

[\times calculation of $\delta \log \det A$]

$$= O(n^{2+d} \epsilon^{-2}) [\times \dots]$$

MG calculation in $O(\epsilon^{-2})$

Statistical Fields

$$\text{Probability}(u) \sim e^{-\text{Energy}(u)/T}$$

Monte Carlo: particle-by-particle
(or point-by-point) simulation

Slow large-scale sampling:

1. Small changes per sweep
2. Few samples per configuration

Multigrid Monte Carlo:

1. Moves on all scales
2. Much sampling on coarse levels

E.g., Gaussian model on $L \times \dots \times L = L^d$ lattice:

$O(1)$ instead of $O(L^{d+2})$ operations per sample

\Rightarrow Homogenization

A. Brandt

Global and Discrete-State Optimization:
Multiscale Annealing

$\min E(u)$

False attraction basins

Particle by particle minimization
trapped in local attraction basins.

Simulated annealing $P(u) \sim e^{-E(u)/T}$
 $T \rightarrow 0$

trapped in large-scale attraction basins.

Multi-level annealing

collective moves, at all scales

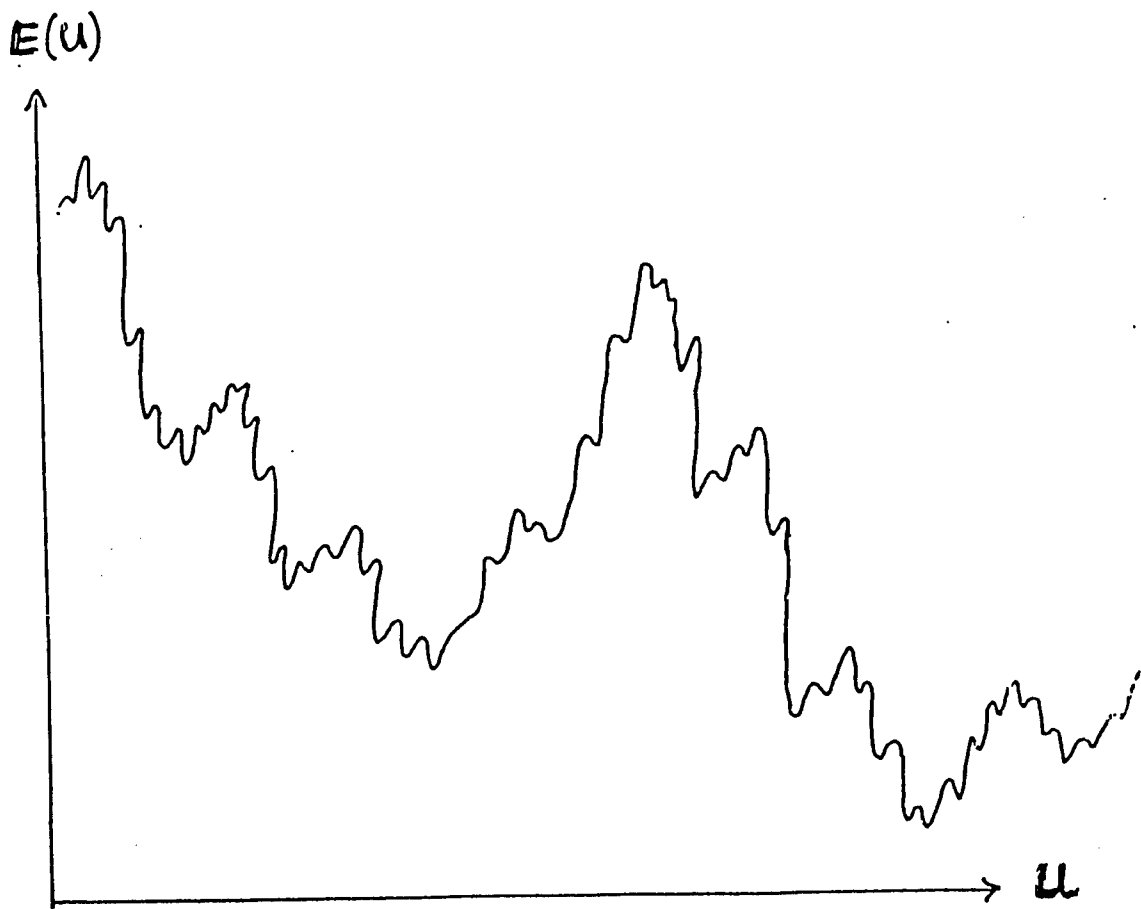
converges (fast, in probability).

- A large-scale move is decided only after optimizing around it at all finer scales.
- Fast annealing at each level.
- Recombinations at each level.

Optimization

$\min E(u)$

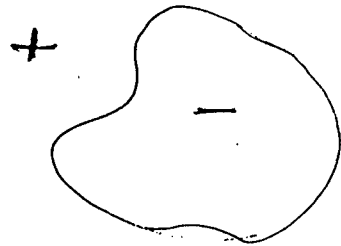
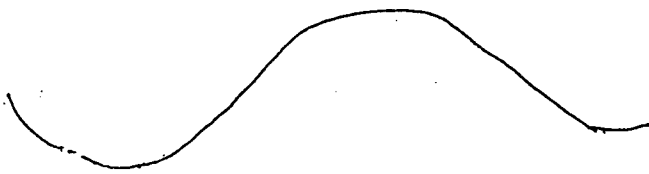
- Fast convergence near optimum
- Global optimization
Escape false attraction basins



multi-scale attraction basins

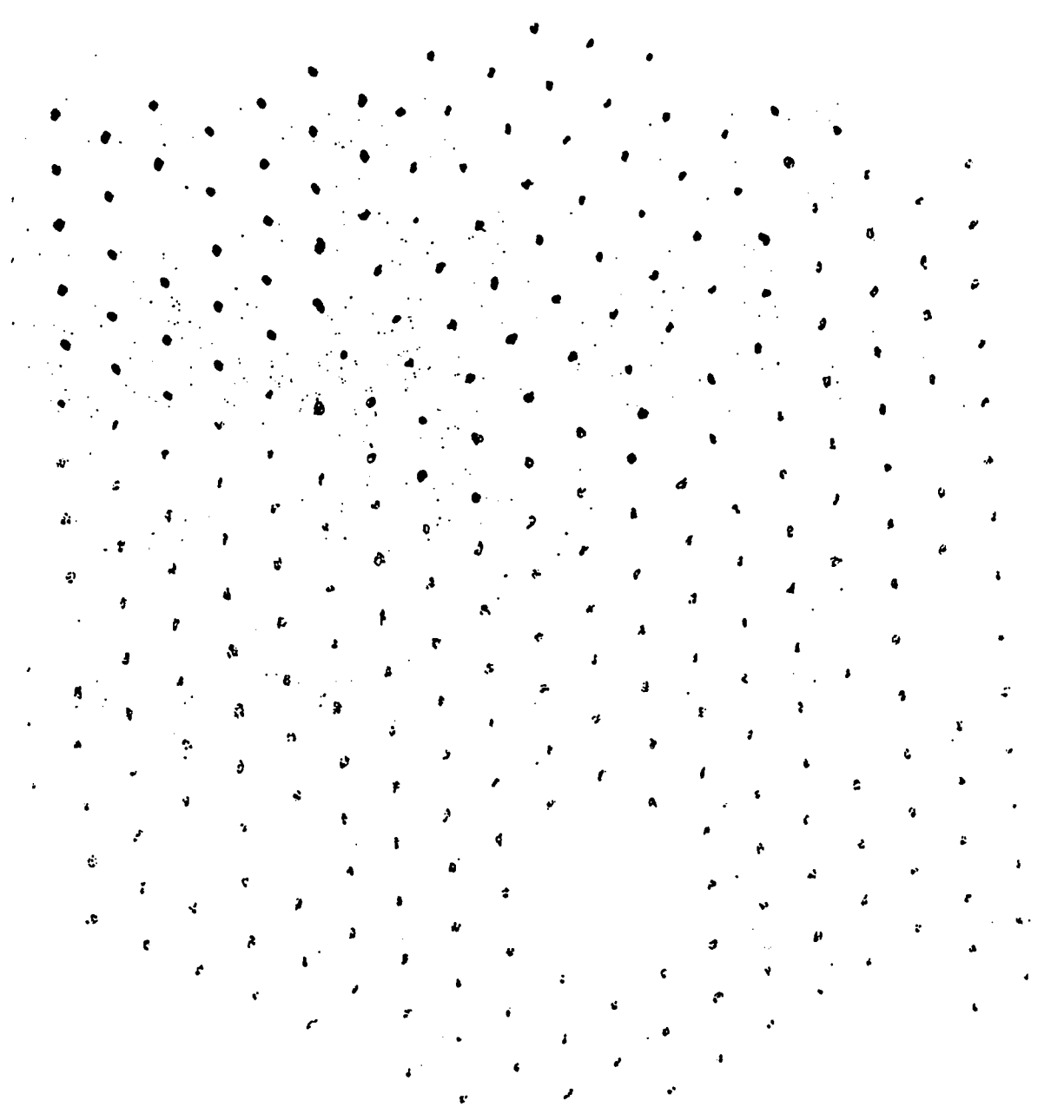
Multi-level stochastic Optimization (minimization)

Large basins \Leftrightarrow large-scale
spatial features



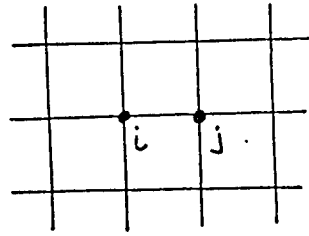
Large scale changes
decided (stochastically)
only after
optimizing around them
at all finer scales.

Recursion



$$P(u) = \frac{1}{Z} e^{-E(u)/T}$$

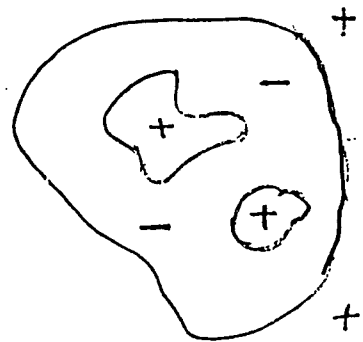
$$E(u) = \sum_{\langle i,j \rangle} (u_i - u_j)^2 - \sum_i f_i u_i$$



Real u_i : $f_i = 0 \Rightarrow u_i \approx u_j$

O_1 : Ising spins $u_i = \pm 1$

Islands within islands
Hierarchy of basins



O_2, O_3, \dots spins: vectors $|u_i| = 1$

Many Local Minima *)

Ising spins: $\min_{S_i = \pm 1, BC} E(s)$
Spin glasses

$$E(s) = - \sum_{\langle i, j \rangle} J_{ij} s_i s_j - \sum_i h_i s_i$$

Relaxation point-by-point minimization:

Flipping a spin if the energy decreases

Artificial temperature $\frac{1}{\beta}$ $P(s) \sim e^{-\beta E(s)}$

Point-by-point stochastic relaxation

Annealing: Increase β gradually

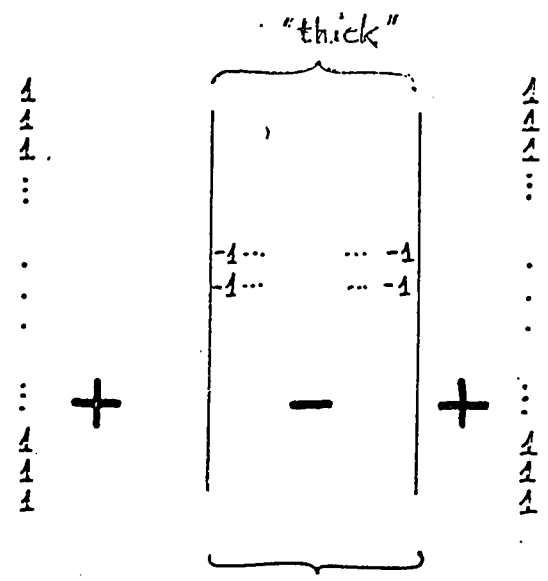
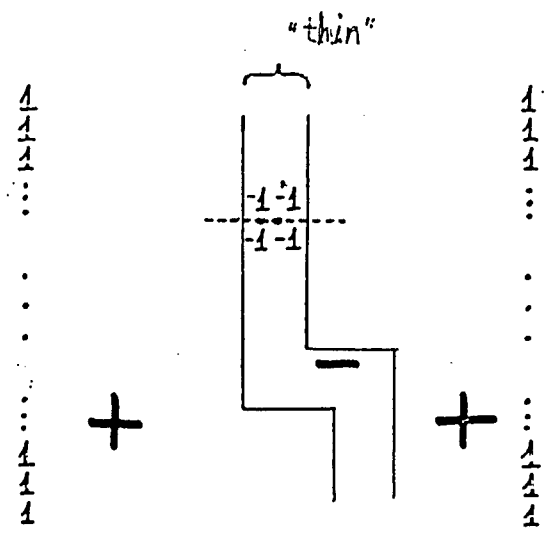
*) with Dorit Ron (1985)

Ising Model

$$E(S) = - \sum_{\langle i,j \rangle} S_i S_j$$

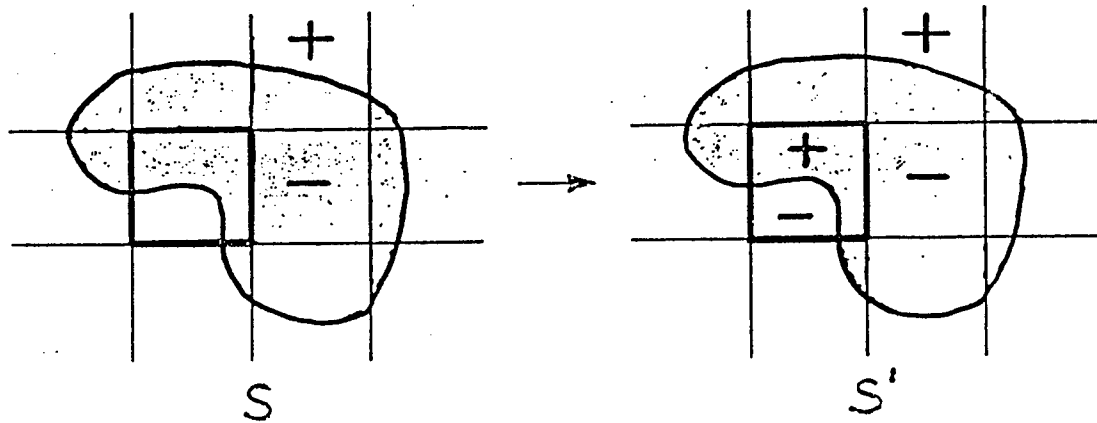
• Ground-State : $S_i \equiv 1$

$$\begin{array}{c} 1 \\ 1 \\ 1 \\ \vdots \\ \cdot \\ \cdot \\ \cdot \\ \vdots \\ 1 \\ 1 \\ 1 \end{array} + \begin{array}{c} | \\ 1 \quad -1 \\ 1 \quad -1 \\ 1 \quad -1 \\ | \end{array} - \begin{array}{c} | \\ 1 \\ 1 \\ 1 \\ | \end{array} + \begin{array}{c} 1 \\ 1 \\ 1 \\ \vdots \\ \cdot \\ \cdot \\ \cdot \\ \vdots \\ 1 \\ 1 \\ 1 \end{array}$$



- External Magnetic Field : $H < 0$
- Ground-State remains : $S_i \equiv 1$

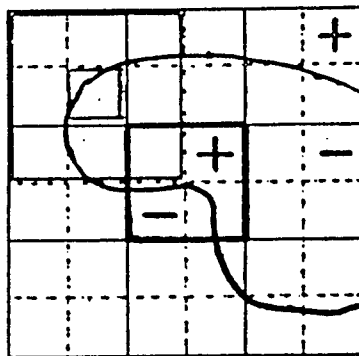
Block-by-block Relaxation



$$E(S) < E(S')$$

\Rightarrow Local optimization around S'

Working Area:



Block revision process \Rightarrow revised block

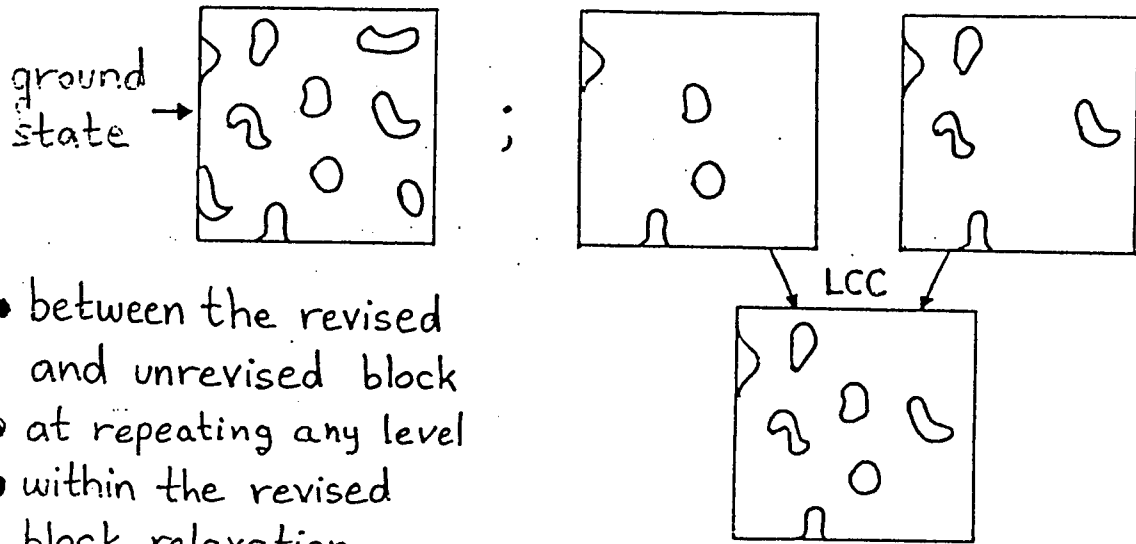
PRINCIPLES of DISCRETE-STATE
and highly non-quadratic OPTIMIZATION

1. Hierarchy of change scales
2. Large-scale change is decided only after calculating its effects (optimizing around it) at all finer scales, starting from the finest.
3. At each scale employ stochasticity just large enough to escape local minima on that scale, then strict minimization.
4. Repeat (the finer - the more) with LCC
5. Recursion

- Acceleration
- Hyper parallel
- Escaping local minima with large attraction basins

Supplementary Minimization Techniques

1. Lowest Common Configuration - LCC



- between the revised and unrevised block
- at repeating any level
- within the revised block relaxation

2. Lower starting β

- increases the probability of breaking long blocks

3. Adaptive relaxation

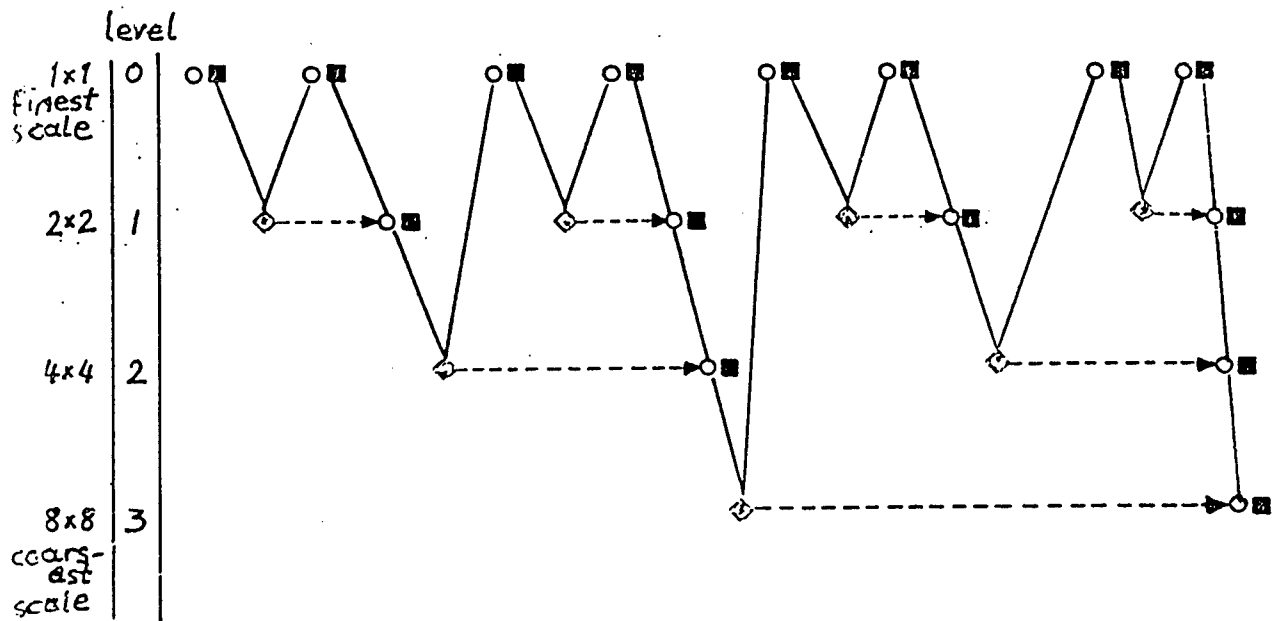
- saves work
- relaxation may expand beyond original work areas

4. Position shifts

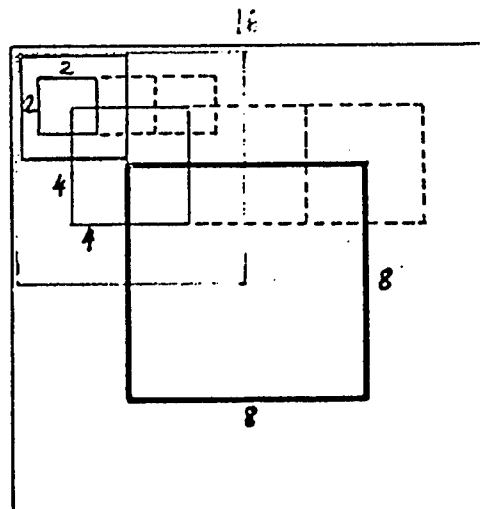
- half-size of the blocks

5. New type of blocks : broken along weak bonds

EXAMPLE: A complete multilevel cycle



- ◆ flipping a square of spins
- stochastic relaxation
- minimization relaxation



Complexity: $O(N^{3/2})$; never exceeds $O(N^2)$

Linear Programming Transportation Problem

Lumping 2 neighboring
destinations, origins, ...
blocks, super-blocks

Coarse costs
= fine marginals

FMG-like solver

Total work
 \approx 1 simplex (MODI) step

Role of relaxation [Kaminsky]

MULTI-LEVEL OPTIMIZATION TECHNIQUES

Equations + Optimization

1-FMG : Parameter optimization on coarse
Local optimization at relaxation

Interactive design: Re-solving mostly on coarse

optimal
control

Functional extremization

Quadratic functional (\Rightarrow linear equations)

Constrained quadratic: FAS $u_j \geq 0$
 $|u_j| \leq 1$

Non quadratic (few extrema): FAS

Highly non-quadratic } (many extrema): multi
Discrete states $u_j = \pm 1$ } level
approx. min!

Linear programming transportation problem

Kaminsky

Full multigrid (FMG)
and
nonlinear multigrid

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

Goal of FMG:
 $O(N) \log N \rightarrow O(N)$

MGG1-1

Full multigrid (FMG)

Problems with iterative methods:
 First guess? Stopping criterion?

First guess?

Use coarse-grid approximations as first guesses for iterations on finer grids.

→ *nested iteration*

Stopping criterion?

→ $\|U - u^h\| \approx \|U - U^h\|$ (*)

approximation error
discretization error
 (of current approximation)

Generally, it makes no sense to iterate the *algebraic error* such that

$$\|U^h - u^h\| \ll \|U - U^h\|$$

algebraic error
discretization error

FMG is, formally, just a combination of
 MG cycling + nested iteration

However,

Since cycle convergence is *h-independent*:

(*) is naturally obtainable also;

Total FMG work: $O(N)$

Full multigrid algorithm

Sequence of m increasingly finer grids

$$\Omega_1, \Omega_2, \dots, \Omega_m (= \Omega_h)$$

For $k=1,2,\dots,m$ do

 If $k=1$ then

$$\text{ Solve } L^k U^k = f^k$$

 else

$$\text{ } u^k = \Pi_{k-1}^k u^{k-1}$$

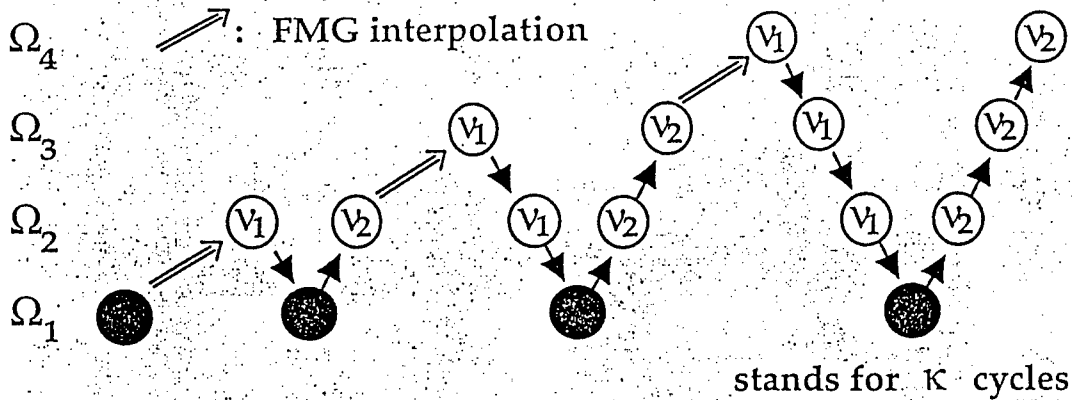
$$\text{ } u^k \leftarrow \text{MG}^{\kappa}(u^k; L^k, f^k)$$

 endif

enddo

Π_{k-1}^k : denotes higher order interpolation (FMG interpolation).

$\kappa > 0$: denotes the number of MG-cycles to be performed.



Order of the FMG interpolation:
at least p (= order of discretization)

Standard coarsening, 2nd order discretizations:
cubic interpolation

MGG1-3

FMG performance

$\rho :=$ MG-cycle convergence factor

$$E_k := \|U - U^k\|$$

discretization error

$$e_k := \|U - u^k\|$$

current approximation error

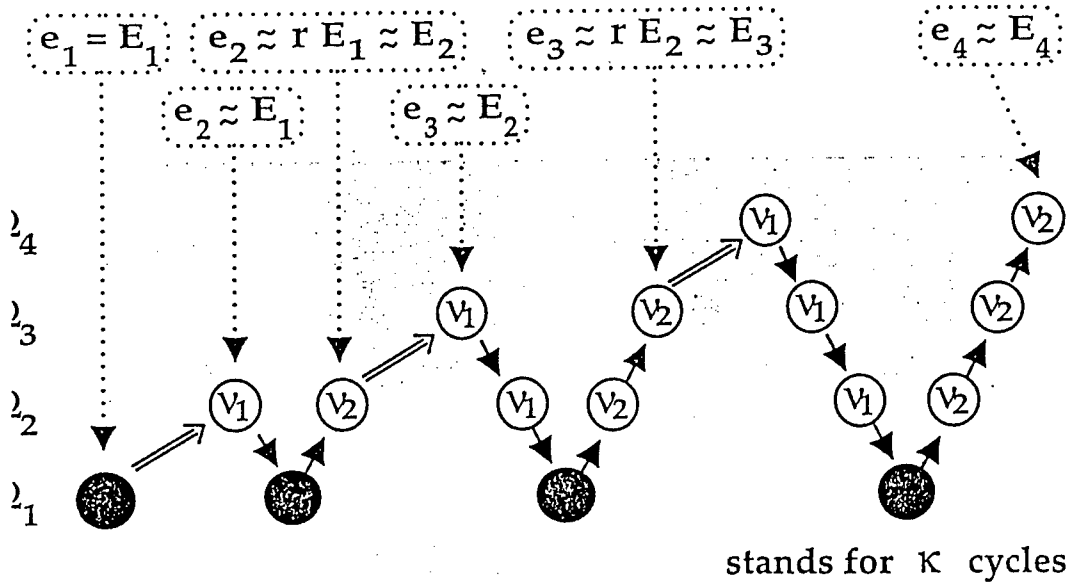
Goal: $e_k \approx E_k$

Assumption: $E_k \approx r E_{k-1}$

Roughly:

For each k , the final approximation u^k is accurate up to the level of the discretization error, i.e. $e_k \approx E_k$, if

$$\rho^k < r$$



MGG1-4

Work count in 2D, standard coarsening

$\mathcal{W}_{\text{cycle}} :=$ work per cycle *on finest level*

$$\mathcal{W}_{\text{FMG}} = \kappa \mathcal{W}_{\text{cycle}} (1 + 1/4 + 1/16 + \dots) = 4/3 \cdot \kappa \mathcal{W}_{\text{cycle}}$$

ignored: FMG interpolation

FMG is true $O(N)$ method

Poisson equation:

V(2,1)-cycle, $\rho \approx 0.1$, $r = 0.25 \rightarrow \kappa = 1$

$\mathcal{W}_{\text{FMG}} \approx 4/3 \mathcal{W}_{\text{cycle}} \approx 7$ relax on finest grid

Solving Poisson-like problems with FMG
costs approximately the equivalent of
7 relaxations on the finest grid!

Remarks

$L^k U^k = f^k$ should "make sense" on the coarser grids

Otherwise, start FMG on a higher level or
use "averaged equations" on the lower levels.

Nonlinear multigrid (FAS)

Nonlinear $L^h U^h = f^h$, approximate solution: u^h

$L^h V^h = d^h$ makes no sense any more

"Naive" approach

Global linearization (outer iteration) + linear MG (inner iteration)

- Linearization (e.g. Newton) may be complex
- The inner + outer iterations have to be matched

"Direct" approach

- No global linearization!
(Local) linearization only in relaxation
 - No matching of different iterations required
 - Same cycle structure as for linear problems
 - Sometimes easier to converge
- Multigrid performance as for linear problems
- Easy incorporation of "special techniques"
(adaptive multigrid, τ -extrapolation,)

Nonlinear correction equation

Instead of $L^h V^h = d^h$

$$L^h(V^h + u^h) - L^h u^h = d^h \quad \text{with} \quad d^h := f^h - L^h u^h$$

$$\rightarrow U^h = u^h + V^h$$

Approximation w.r.t. Ω_H and L^H ?

$$L^H(V^H + I_h^{\vee H} u^h) - L^H(I_h^{\vee H} u^h) = I_h^H d^h$$

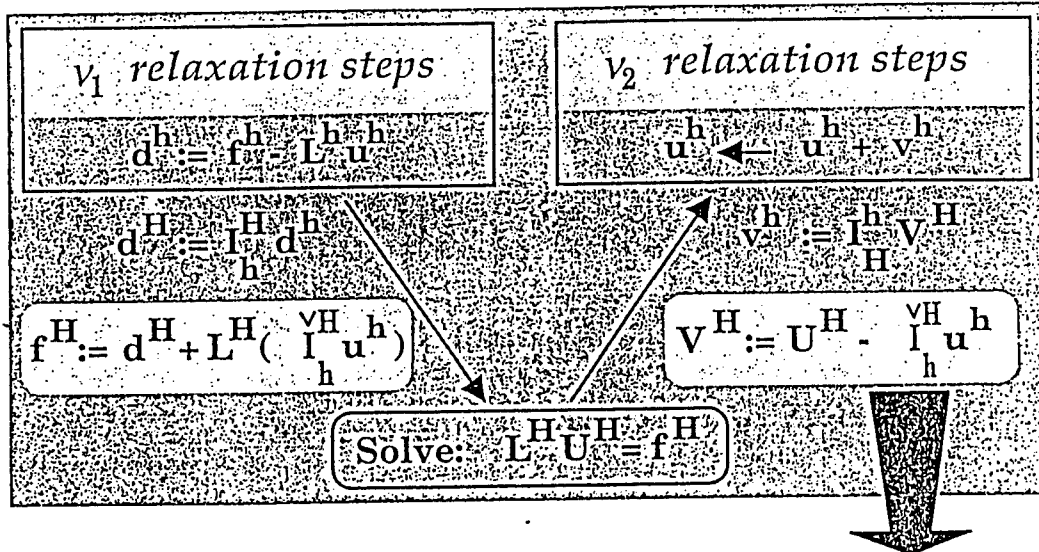
$$\rightarrow \underbrace{L^H(V^H + I_h^{\vee H} u^h)}_{=: U^H} = \underbrace{I_h^H d^h + L^H(I_h^{\vee H} u^h)}_{=: f^H}$$

$$\rightarrow L^H U^H = f^H$$

$$V^H := U^H - I_h^{\vee H} u^h$$

- $I_h^{\vee H}$ is a transfer operator which may differ from I_h^H
- Note that U^H and f^H are defined above. They are NOT the solutions and right hand side, resp., of the H-discretized continuous problem!

Nonlinear two-grid cycle



Only *smooth quantities* (i.e. errors) must be interpolated back to the fine grid!! Do not interpolate U^H itself!

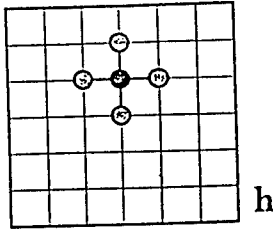
- Smoothing requires a *nonlinear relaxation scheme*
- Extension to "real MG" as in the linear case
- This nonlinear cycle *coincides with the linear one* if the problem is linear
- At convergence:

$$u^h = U^h \quad \text{and} \quad U^H = I_h^H U^h$$

Standard choice: $I_h^{2h} = \text{straight injection}$

At convergence, all coarse-grid solutions are identical to the finest-grid solution

Nonlinear Example



$$-\Delta U + c(x,y,U) = 0$$

$$\text{Assumption: } c_u(x,y,U) \geq 0$$

Nonlinear Gauss-Seidel relaxation

For all grid points do: compute $\bar{u}_{i,j}^h$ such that

$$\left[4\bar{u}_{i,j}^h - u_{i-1,j}^h - u_{i+1,j}^h - u_{i,j-1}^h - u_{i,j+1}^h \right] + h^2 c(x,y, \bar{u}_{i,j}^h) = 0$$

$$\text{and } u_{i,j}^h \leftarrow \bar{u}_{i,j}^h$$

Picard linearization (only good if $c_u(x,y, U_{i,j}^h)$ "small"):

$$c(x,y, \bar{u}_{i,j}^h) \rightarrow c(x,y, u_{i,j}^h)$$

Newton linearization:

$$c(x,y, \bar{u}_{i,j}^h) \rightarrow c(x,y, u_{i,j}^h) + c_u(x,y, u_{i,j}^h) (\bar{u}_{i,j}^h - u_{i,j}^h)$$

→ Same efficiency as Poisson-solver!

Further remarks

$$h\text{-truncation error: } \tau_h := L^h U - LU$$

$$H\text{-truncation error: } \tau_H := L^H U - LU$$

Coarse-grid equation

$$\begin{aligned} L^H U^H = f^H &= I_h^H f^h + \boxed{L^H \begin{pmatrix} \sqrt{H} \\ I_h^H \end{pmatrix} u^h - I_h^H L^h u^h} \\ &=: \tau_h^H = (h,H)\text{-relative truncation error} \end{aligned}$$

Under certain assumptions

$$\tau_h^H \approx \tau_H - \tau_h = L^H U - L^h U$$

Applications

τ -estimation

$$\text{For example: } \tau_H \approx c \tau_h \longrightarrow \tau_h \approx \tau_h^H / (c-1)$$

$$\text{Stopping criterion: } \|L^h u^h - f\| < \|\tau_h\|$$

τ -extrapolation

$$\tau_h^H \longleftarrow \frac{c}{c-1} \tau_h^H \approx \tau_H$$

Adaptive multigrid (MLAT)

Molecular Dynamics at Large Time-Steps

Eric Barth
Margaret Mandziuk
Tamar Schlick
Guihua Zhang

Courant Institute of Mathematical Sciences

Department of Chemistry

New York University

and

Howard Hughes Medical Institute

In molecular dynamics, calculating the N -body forces of interaction dominates computational effort.

$$V = \sum_{\text{bonds}} k_l \Delta l + \sum_{\text{angles}} k_\theta \Delta \theta + \sum_{\text{dihedrals}} k_\phi (1 + \cos(n\phi + \delta)) + \sum_{\text{atom pairs } ij} \left(\frac{q_i q_j}{R_{ij}} - \frac{\alpha_{ij}}{R_{ij}^6} + \frac{\beta_{ij}}{R_{ij}^{12}} \right)$$

- Standard explicit integrators require one force evaluation per time-step.
- Fastest motions necessitate small steps.

Unfeasible number of force evaluations needed to simulate slow motions over time intervals of interest.

Improved efficiency has been sought by:

- Constraining fast interactions

SHAKE

reduced variable treatments—Rice & Brunger, Moldyn, Oren Becker

- Dividing the forces by time scales

multiple time-step methods—Humphries, Byrne, Watanabe, Karplus

- Super-stable implicit integration schemes

LI—Schlick

- Reference systems for the fast motions.

LIN and related methods—Zhang & Schlick

$$\begin{aligned} V = & \sum_{\text{bonds}} k_l \Delta l \\ & + \sum_{\text{angles}} k_\theta \Delta \theta \\ & + \sum_{\text{dihedrals}} k_\phi (1 + \cos(n\phi + \delta)) \\ & + \sum_{\text{atom pairs } ij} \left(\frac{q_i q_j}{R_{ij}} - \frac{\alpha_{ij}}{R_{ij}^6} + \frac{\beta_{ij}}{R_{ij}^{12}} \right) \end{aligned}$$

The Equations of Motion

The Hamiltonian system

$$\dot{q} = M^{-1}p$$

$$\dot{p} = -\nabla V(q)$$

Langevin Dynamics

$$\dot{q} = M^{-1}p$$

$$\dot{p} = -\nabla V(q) - \gamma p + r$$

Overdamped Langevin Dynamics

(Brownian Dynamics)

$$\dot{q} = M^{-1}p$$

$$0 = -\nabla V(q) - \gamma p + r$$

Traditional Molecular Dynamics

The Hamiltonian system

$$\dot{q} = M^{-1}p$$

$$\dot{p} = -\nabla V(q)$$

Discretization: Velocity Verlet Method

$$p_{n+1/2} = p_n - \frac{h}{2}\nabla V(q_n)$$

$$q_{n+1} = q_n + hM^{-1}p_{n+1/2}$$

$$p_{n+1} = p_{n+1/2} - \frac{h}{2}\nabla V(q_{n+1})$$

Verlet is:

- Explicit
- Second order in time
- Symplectic
- Time reversible
- The gold standard in MD

Constrained Dynamics

The Hamiltonian system with constraints.

$$\begin{aligned}\dot{q} &= M^{-1}p \\ \dot{p} &= -\nabla V(q) + g'(q)^t \lambda \\ 0 &= g(q)\end{aligned}$$

Discretization: Verlet with SHAKE

$$\begin{aligned}p_{n+1/2} &= p_n - \frac{h}{2}\nabla V(q_n) + \frac{h}{2}g_n'^t \lambda_n \\ q_{n+1} &= q_n + hM^{-1}p_{n+1/2} \\ 0 &= g(q_{n+1}) \\ p_{n+1} &= p_{n+1/2} - \frac{h}{2}\nabla V(q_{n+1}) + \frac{h}{2}g_{n+1}'^t \lambda_{n+1}\end{aligned}$$

A typical bond constraint between atoms i and j :

$$g_\alpha(q) = \frac{1}{2} \left(r_{ij}^2 - L_\alpha^2 \right) = 0.$$

Torsion dynamics, multibody treatments, and other variable reduction techniques use generalized coordinates which rules out Verlet integration

Multiple Time-Step Dynamics

The split Hamiltonian system.

$$\dot{q} = M^{-1}p$$

$$\dot{p} = -\nabla V_{\text{fast}}(q) - \nabla V_{\text{slow}}(q)$$

Discretization: reversible-RESPA

$$q_0 = q_n$$

$$p_0 = p_n - \frac{kh}{2} \nabla V_{\text{slow}}(q_n)$$

for $i = 0 : k - 1$,

$$p_{i+1/2} = p_i - \frac{h}{2} \nabla V_{\text{fast}}(q_i)$$

$$q_{i+1} = q_i + hM^{-1}p_{i+1/2}$$

$$p_{i+1} = p_{i+1/2} - \frac{h}{2} \nabla V_{\text{fast}}(q_{i+1})$$

end

$$q_{n+1} = q_k$$

$$p_{n+1} = p_k - \frac{kh}{2} \nabla V_{\text{slow}}(q_{n+1})$$

LIN:

Langevin Dynamics with Taylor series forces.

$$\dot{q} = M^{-1}p$$

$$\dot{p} = [-\nabla V(q_0) - H(q - q_0) - \dots] - \gamma p + r$$

Discretization:

LIN family of methods

solve the linearized system over $[0, h]$,

with initial conditions $q_{\text{ref}}^0 = q_n, p_{\text{ref}}^0 = p_n$

$$\dot{q}_{\text{ref}} = M^{-1}p_{\text{ref}}$$

$$\dot{p}_{\text{ref}} = -\nabla V(q_0) - H(q_{\text{ref}} - q_0) - \gamma p_{\text{ref}} + r$$

integrate the residual with timestep h

$$(q - q_{\text{ref}}) = M^{-1}(p - p_{\text{ref}})$$

$$(p - p_{\text{ref}}) = -\nabla V(q) + \nabla V(q_0) + H(q_{\text{ref}} - q_0) - \gamma(p - p_{\text{ref}})$$

VERLET - type integrators
dominate

MACROMOLECULAR (BIO-) DYNAMICS

- * one-step integrator
(one evaluation of forces per timestep)
- * relatively high order
("feels" the curvature)
- * time-reversible, symplectic
(good long-time behavior)

"STABLE INTEGRATION"
in biomolecular dynamics

$$1) \quad \Delta E = \frac{1}{N} \sum_i \left| \frac{E_{\text{initial}} - E_i}{E_{\text{initial}}} \right|$$

$$\Delta E \leq 0.001 - 0.003$$

$$2) \quad R = \frac{\Delta E_{\text{rms}}}{\Delta KE_{\text{rms}}}$$

$$R \leq 0.01$$

Rule of thumb: $\Delta t = T/20$

SHAKE

(Ryckaert, Ciccolti, Berendsen,
J. Comp. Phys., 23, 327 (1977))

* BOND LENGTH CONSTRAINTS - small effect
on long-time dynamics

* ANGLE CONSTRAINTS - reduce mobility
of the system

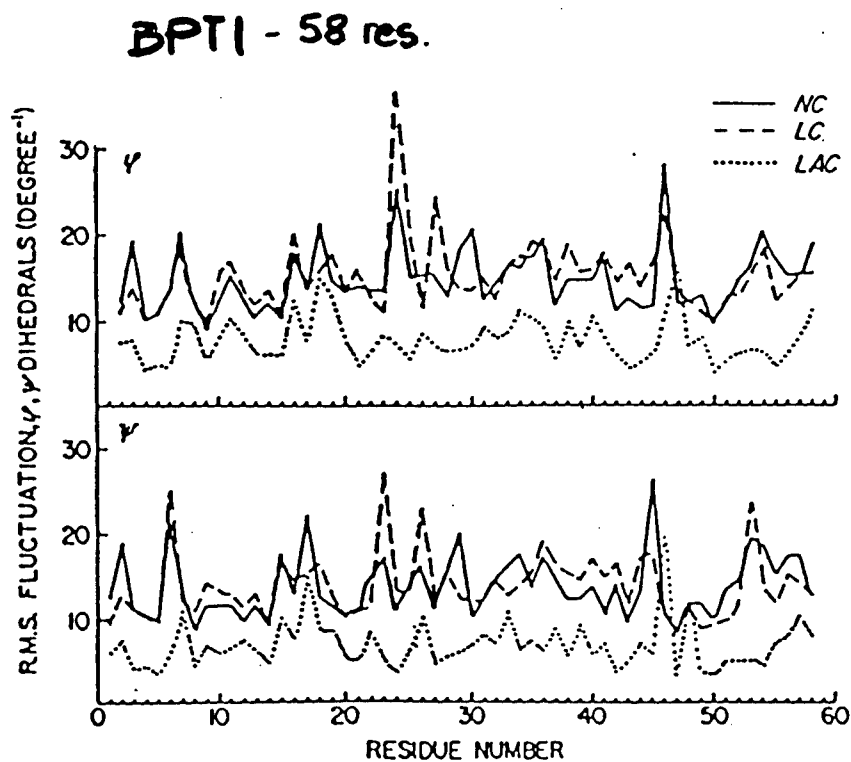


Figure 7. Root-mean-square fluctuations of the dihedrals angles ϕ, ψ over 25 "ps": (—) NC run; (---) LC run; (...) LAC run.

(van Gunsteren, Karplus, Macromolecules 15, 1528 (1982))

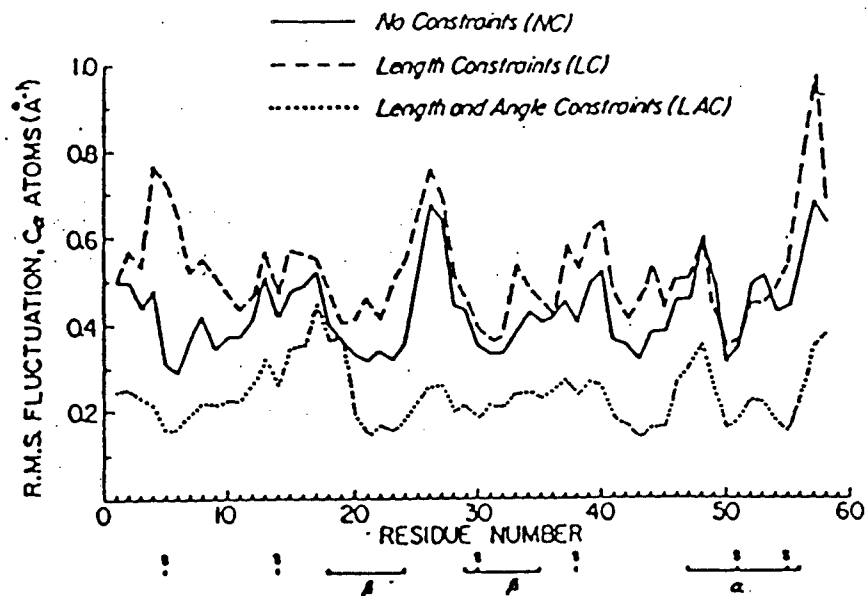


Figure 3. Root-mean-square fluctuations of C_{α} atoms over 25 "ps": (—) NC run; (---) LC run; (···) LAC run.

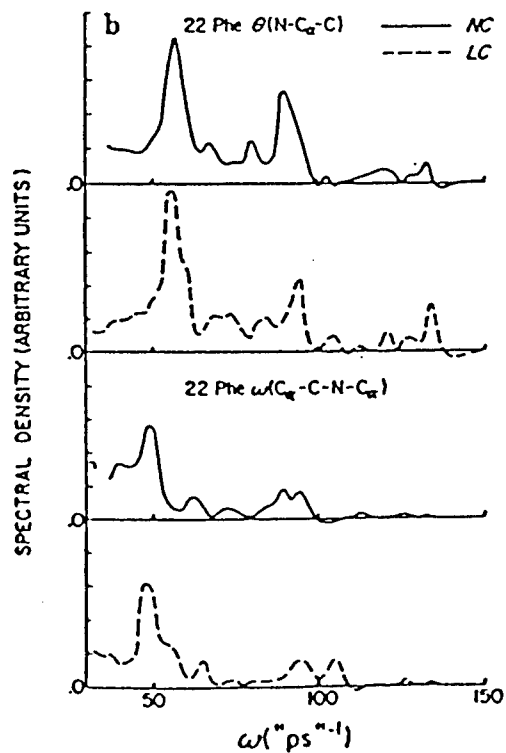


Figure 6. Bond-angle and dihedral angle autocorrelation and spectral density functions: (—) NC run; (---) LC run. (a) Autocorrelation function; (b) spectral density.

TABLE 3: Combined Stiff/Soft, Internal/External, and Short/Long-Range Nonbonded Force Decomposition. Comparison of Energy Conservation and Associated CPU Times Spent in the Various Force Routines for Velocity Verlet ($n_1 = n_2 = n_3 = 1$) and r-RESPA Using the Propagator Given by eq 38 and eq 43^a

Δt	$\delta\tau_1$	n_1	n_2	n_3	$\log(\Delta E)$	R	$T_{stretch}$	T_{bend}	$T_{torsion}$	$T_{nonbonded}$
0.25	0.25	1	1	1	-3.7073	0.0022	30.7	186.0	399.7	10085.9
0.50	0.50	1	1	1	-3.0123	0.0087	15.5	93.3	201.2	5073.4
1.00	1.00	1	1	1	-2.2388	0.0398	7.7	47.9	102.8	2579.6
2.00	2.00	1	1	1	-1.0475	0.1981	4.2	23.6	52.5	1326.7
3.00	0.25	1	6	2	-2.9880	0.0102	31.9	189.1	407.6	1090.0
3.00	0.25	2	3	2	-2.9995	0.0102	32.1	92.3	198.7	1062.9
3.00	0.50	1	3	2	-2.7944	0.0127	15.6	93.6	201.0	1078.2

^a $\log(\Delta E)$ and R are given by eq 48 and eq 49. Δt and $\delta\tau_1$ are in femtoseconds. Here $r_c = 6.0 \text{ \AA}$ and $\Delta r = 2.0 \text{ \AA}$ for all r-RESPA cases considered. $T_{stretch}$, T_{bend} , $T_{torsion}$, and $T_{nonbonded}$ are the CPU times in seconds spent in the stretch, bend, torsion, and nonbonded force routines, respectively. CPU time spent calculating r-RESPA neighbor lists is included in $T_{nonbonded}$.

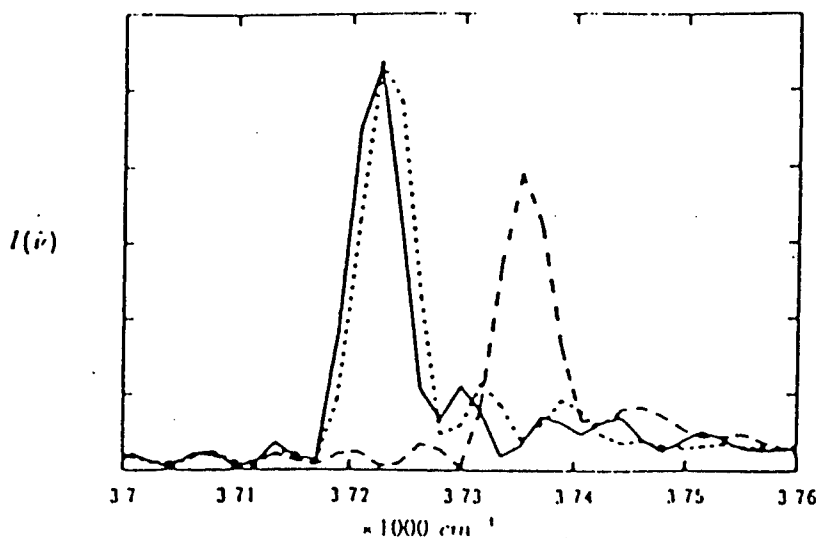


Figure 6. Detail of spectral density $I(\nu)$ of eq 52 as a function of wavenumber for "Exact" (i.e. velocity Verlet $\Delta t = 0.25$ fs) (solid line), velocity Verlet $\Delta t = 0.5$ fs (dashed line), and r-RESPA using the propagator given by eq 38 and eq 43 with $n_1 = 1$, $n_2 = 6$, $n_3 = 2$, and $\delta\tau_1 = 0.25$ fs (dotted line). Intensities are in arbitrary units.

r-RESPA (Tuckerman, Berne, Martyna, JCP 97, 1990 (1992))

EFFICIENCY of the method depends on
the optimal split of the forces

(Humphreys, Friesner, Berne, JPC 98, 6885 (1994))
crambin - 46 res.

$$V = V_b + V_\theta + V_\varphi + V_{nb}$$

↓ ↓

$$\underbrace{\quad\quad\quad}_{\tau_1} \quad \underbrace{\quad\quad\quad}_{n_1 \tau_1} \quad \underbrace{V_{nb(1)}}_{n_1 n_2 \tau_1} \quad \underbrace{V_{nb(2)}}_{n_1 n_2 n_3 \tau_1 = \Delta t}$$

Watanabe, Karplus, J. Phys. Chem. 99, 5680 (1995)

Single r-RESPA

$$V = \underbrace{V_b + V_\theta}_{} + \underbrace{V_\varphi + V_{nb}}_{}.$$

Double r-RESPA

$$V = \underbrace{V_b + V_\theta + V_\varphi}_{} + \underbrace{V_{nb}}_{\downarrow} + \underbrace{V_{nb}}_{\text{rest}}$$

$V_{nb}(H)$

BPTI

TABLE 10: Average Total Energies, Fluctuations of Total and Kinetic Energies, and CPU Ratios for the BPTI Molecule

model	$\langle E_{total} \rangle$ (kcal/mol)	ΔE_{total} (kcal/mol)	ΔE_{kin} (kcal/mol)	$\log(\Delta E_{total})^a$	CPU ratio ^b	
Standard						
1	1134.55	0.127	15.40	-2.08	1.00	0.5 fs
2	1134.48	0.545	15.76	-1.46	0.52	1.0 fs
SHAKE						
	1061.12	0.844	15.81	-1.26	0.28	2.0 fs
Single r-RESPA						
1	1134.71	0.290	15.40	-1.73	0.28	0.5fs - 1.0fs
2	1135.21	0.498	15.38	-1.49	0.22	0.5fs - 2.5fs
3 ^c					0.21	B, A rest
Double r-RESPA						
1	1134.41	0.562	15.20	-1.43	0.26	0.5 - 2.0 - 4.0
2	1135.36	0.662	15.48	-1.37	0.22	0.5 - 2.5 - 5.0
3 ^c					0.19	internal nb _H nb

^a See eq 18. All values calculated over 10 ps of production run.

^b Each CPU ratio was calculated with CPU time (method)/CPU time (standard 1). Total energies of sr-RESPA 3 and dr-RESPA 5 are not conserved.

LIN (Zhang, Schlick, J. Comp. Chem. 14, 1212 (1993))

Langevin dynamics - total energy fluctuate

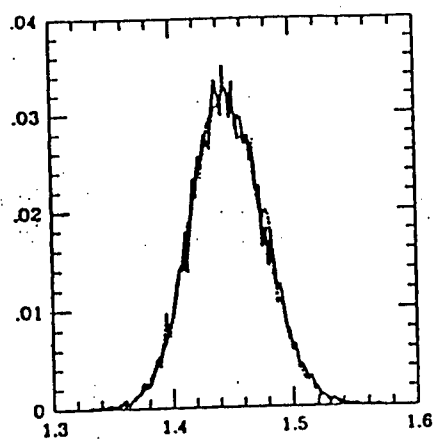
Averages for Alanine Dipeptide, LIN ($\Delta t = 30$ fs)
vs. Explicit ($\Delta t = 0.5$ fs) Langevin Trajectories 2 ns

	mean, ex.	var., ex.	mean, LIN	var., LIN
E^a	35.0	4.8	36.8	5.4
E_k	19.7	3.4	20.6	3.6
E_p	15.4	3.3	16.6	3.8
ϕ^b	-116.0	32.9	-113.2	33.0
ψ	64.6	90.2	66.9	86.2
r_{CN}^c	1.344	.028	1.345	.028
r_{NC_a}	1.446	.030	1.447	.030
r_{C_aC}	1.520	.034	1.520	.034
θ_{CNC_a}	123.4	3.3	123.2	3.3
θ_{NC_aC}	109.3	4.4	109.5	4.5
θ_{C_aCN}	117.0	2.8	117.0	2.9

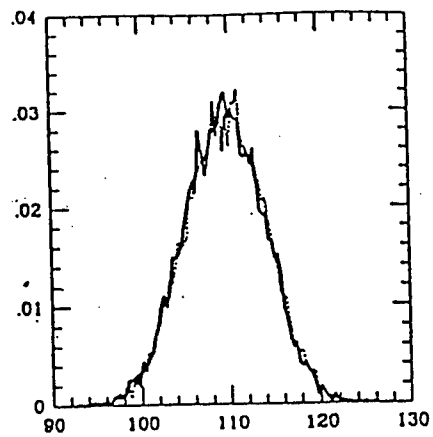
^aEnergy is given in kcal/mol

^bAngles are measured in degrees

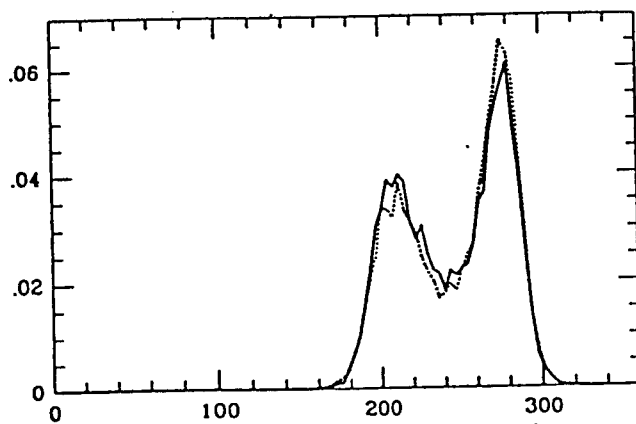
^cBond lengths are given in Å



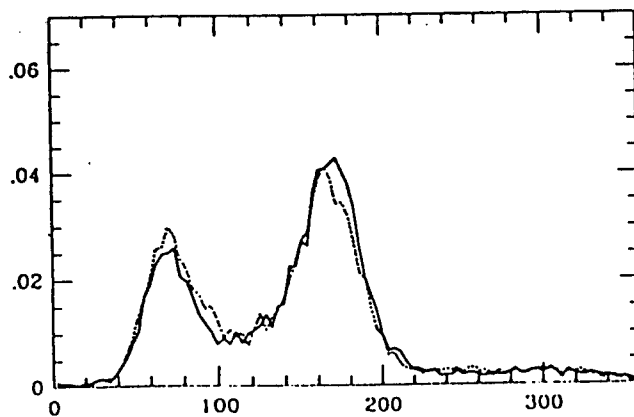
Γ_{N-C_α}



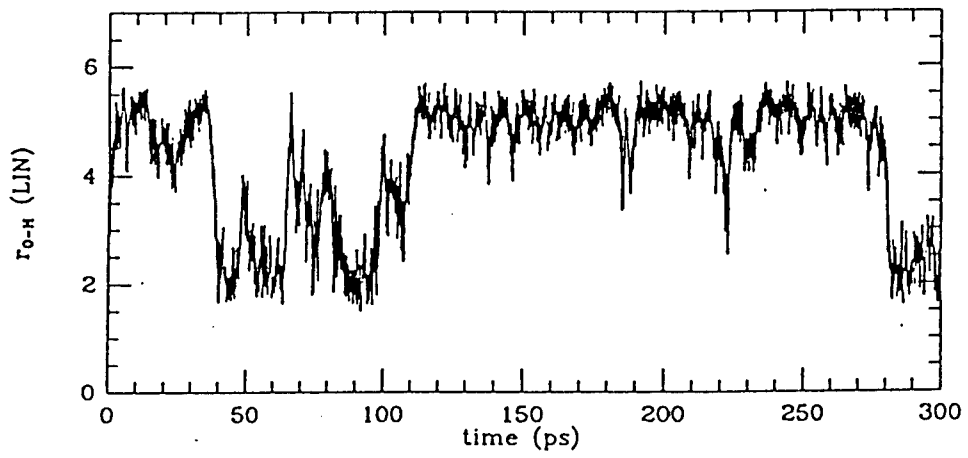
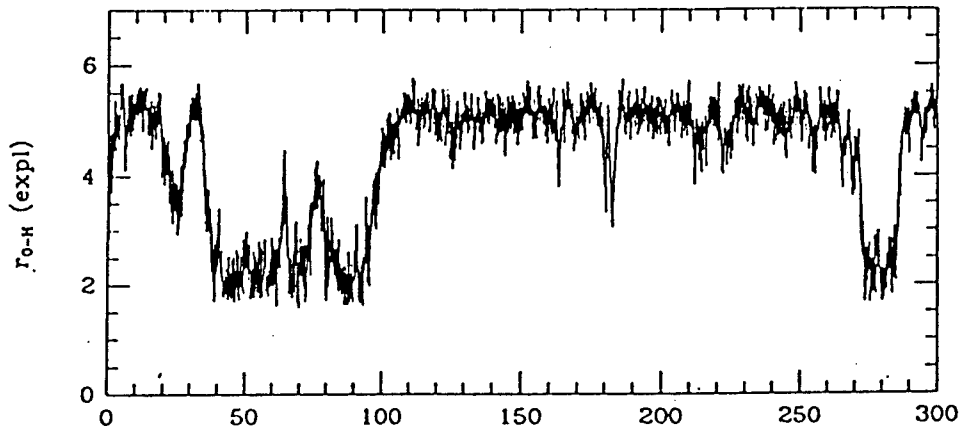
$\theta_{N-C_\alpha-C}$



φ



ψ



COMPARISON OF LONG TIMESTEP APPROACHES

METHOD	Approximation	Δt (fs)
VERLET		0.5
SHAKE	constrains	2.0
REDUCED VARIABLE CONF. SAMPL.		2.0
SUBSTRUCTURING		
RESPA	multiple timestep	3.0 5.0
LIN	separating framework	29.0

BROWNIAN DYNAMICS	constrains substructuring	10^3 40.0
ESSENTIAL SUBSPACE	normal modes local modes $\langle \Delta r_i, \Delta r_j \rangle$	16.7

Efficient Dense Hessian Computations
in
Molecular Minimization

Christophe G. Lambert

Duke University
Department of Computer Science
lambert@cs.duke.edu

Overview of Presentation

- Introduction and Motivation
- Newton-Based Minimization and the Hessian
- Need for Hessian-Vector Product in Iterative Matrix Solve
- Goal: Describe Product in Terms of N-body Problem
- The Hessian Matrix Structure
- Decompose Product into Off-Diagonal and Diagonal Parts
- Discussion of Complete Computation
- Ongoing Work

Introduction and Motivation

- Diverse applications of Potential Energy Minimization
 - NMR and X-Ray structure refinement
 - Molecular modeling
 - Long time step dynamics (Schlick)
- Newton Methods
 - Quadratic convergence near minimum
 - Large computational overhead
 - * $O(n^2)$ space
 - * $O(n^3)$ time with direct matrix solve
 - * $O(n^2)$ time per iteration within iterative matrix solve
- Goal is to reduce overhead of Newton method
 - $O(n)$ space
 - $O(n)$ time per iteration within iterative matrix solve
 - Include all electrostatic interactions (no cutoffs)

Newton-Based Minimization

Goal: minimize $E(z)$, $E : \mathfrak{R}^n \rightarrow \mathfrak{R}$.

Expand the function $E(z + z^*)$ about z^* as:

$$E(z + z^*) = E(z^*) + \sum_{i=1}^n \frac{\partial E(z^*)}{\partial z_i} z_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 E(z^*)}{\partial z_i \partial z_j} z_i z_j + O(\|z^*\|^3).$$

Set gradient of quadratic approximation to zero:

$$\frac{\partial E(z + z^*)}{\partial z_k} = \frac{\partial E(z^*)}{\partial z_k} + \sum_{j=1}^n \frac{\partial^2 E(z^*)}{\partial z_k \partial z_j} z_j = 0, \forall k = 1 \dots n.$$

Express this in matrix form:

$$\begin{bmatrix} \frac{\partial E(z^*)}{\partial z_1} \\ \frac{\partial E(z^*)}{\partial z_2} \\ \vdots \\ \frac{\partial E(z^*)}{\partial z_n} \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 E(z^*)}{\partial z_1 z_1} & \frac{\partial^2 E(z^*)}{\partial z_1 z_2} & \dots & \frac{\partial^2 E(z^*)}{\partial z_1 z_n} \\ \frac{\partial^2 E(z^*)}{\partial z_2 z_1} & \frac{\partial^2 E(z^*)}{\partial z_2 z_2} & \dots & \frac{\partial^2 E(z^*)}{\partial z_2 z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(z^*)}{\partial z_n z_1} & \frac{\partial^2 E(z^*)}{\partial z_n z_2} & \dots & \frac{\partial^2 E(z^*)}{\partial z_n z_n} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = 0,$$

OR

$$H(z^*)z = -g(z^*).$$

Newton Minimization Algorithm

1. Start with initial guess for the minimum, z^* .
2. Compute the gradient g and Hessian H at z^* .
3. Solve the equation $H z = -g$.
4. Set z^* to $z + z^*$. If not converged, goto step 2.

Iterative Matrix Solve

- Preconditioned Conjugate Gradients
- Need Fast Hessian-Vector Product
- Goal is $O(n)$ Product

$\exists O(n)$ Algorithm to Solve
N-Body Problem:

$$\begin{aligned}
 -\vec{F}_x &= \begin{bmatrix} q_1 \frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \\
 &= \text{diag} \left(q_i \frac{\partial}{x_i} \right) \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix}, \dots
 \end{aligned}$$

where

$$\Lambda_{i,j} = \frac{1}{r} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

$$\Phi_i = \sum_{j \neq i} q_j \Lambda_{i,j}.$$

Goal: Describe Hessian Matrix-Vector Product in Terms of N-Body Problem

Off-Diagonal Components of Product

Consider off-diagonal parts of upper left subproduct:

$$\begin{bmatrix} 0 & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & 0 & \cdots & \frac{\partial^2 E}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & \frac{\partial^2 E}{\partial x_n \partial x_2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{bmatrix}$$

Let:

$$\Lambda_{i,j} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

We can rewrite the off-diagonal part of this matrix as:

$$\begin{bmatrix} q_1 \frac{\partial}{\partial x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{\partial x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \bar{x}_1 \frac{\partial}{\partial x_1} \\ q_2 \bar{x}_2 \frac{\partial}{\partial x_2} \\ \vdots \\ q_n \bar{x}_n \frac{\partial}{\partial x_n} \end{bmatrix}$$

Let:

$$Q_{x_i} = q_i \bar{x}_i$$

$$\Lambda_{x_i,j} = \frac{\partial}{\partial x_j} \Lambda_{i,j} = \frac{x_i - x_j}{((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)^{\frac{3}{2}}}$$

Rewrite product as:

$$\begin{bmatrix} q_1 \frac{\partial}{\partial x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{\partial x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{x_1,2} & \cdots & \Lambda_{x_1,n} \\ \Lambda_{x_2,1} & 0 & \cdots & \Lambda_{x_2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{x_n,1} & \Lambda_{x_n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} Q_{x_1} \\ Q_{x_2} \\ \vdots \\ Q_{x_n} \end{bmatrix}$$

$\exists O(n)$ Algorithm to Solve
N-Body Problem:

$$\begin{aligned}
 -\vec{F}_x &= \begin{bmatrix} q_1 \frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \\
 &= \text{diag} \left(q_i \frac{\partial}{x_i} \right) \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix}, \quad .
 \end{aligned}$$

where

$$\Lambda_{i,j} = \frac{1}{r} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

$$\Phi_i = \sum_{j \neq i} q_j \Lambda_{i,j}.$$

Diagonal Components of Product

Diagonal part of upper left sub-product:

$$\begin{bmatrix} \frac{\partial^2 E}{\partial x_1 \partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial^2 E}{\partial x_2 \partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2 E}{\partial x_n \partial x_n} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{bmatrix}$$

is equivalent to:

$$\begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} \sum_{j \neq 1} q_j \Lambda_{1,j} & \cdots & 0 & \cdots & 0 \\ 0 & \frac{\partial^2}{\partial x_2^2} \sum_{j \neq 2} q_j \Lambda_{2,j} & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2}{\partial x_n^2} \sum_{j \neq n} q_j \Lambda_{n,j} & \cdots \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{bmatrix},$$

which can be rewritten as:

$$\begin{bmatrix} q_1 \bar{x}_1 \frac{\partial^2}{\partial x_1^2} & 0 & \cdots & 0 \\ 0 & q_2 \bar{x}_2 \frac{\partial^2}{\partial x_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \bar{x}_n \frac{\partial^2}{\partial x_n^2} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix},$$

or

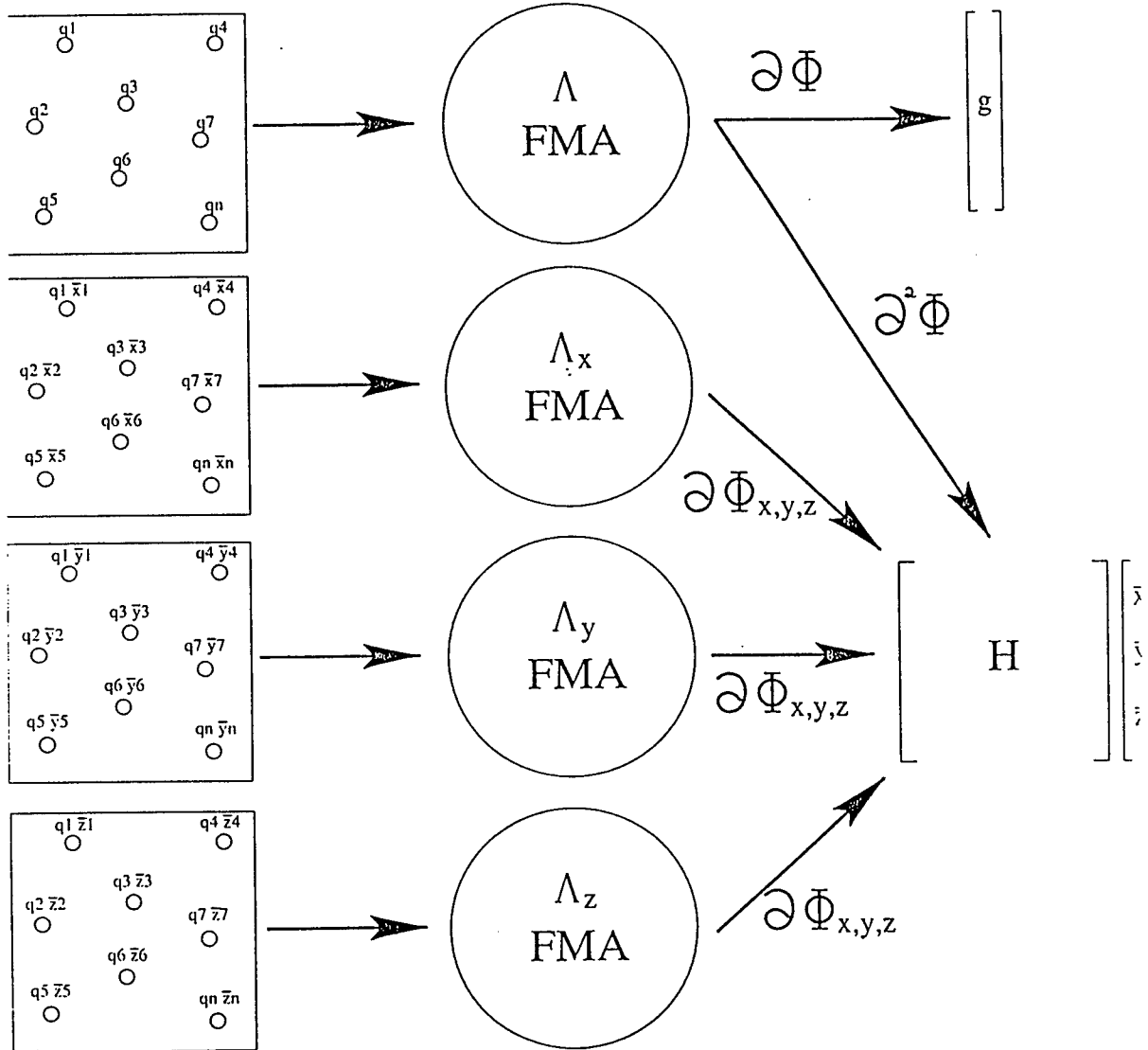
$$\begin{bmatrix} q_1 \bar{x}_1 \frac{\partial^2}{\partial x_1^2} & 0 & \cdots & 0 \\ 0 & q_2 \bar{x}_2 \frac{\partial^2}{\partial x_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \bar{x}_n \frac{\partial^2}{\partial x_n^2} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix}.$$

More Compactly...

$$\begin{bmatrix} \left[q \frac{\partial}{\partial x} \right] & \left[q \frac{\partial}{\partial x} \right] & \left[q \frac{\partial}{\partial x} \right] \\ \left[q \frac{\partial}{\partial y} \right] & \left[q \frac{\partial}{\partial y} \right] & \left[q \frac{\partial}{\partial y} \right] \\ \left[q \frac{\partial}{\partial z} \right] & \left[q \frac{\partial}{\partial z} \right] & \left[q \frac{\partial}{\partial z} \right] \end{bmatrix} \begin{bmatrix} [\Lambda_x] & [\Lambda_y] & [\Lambda_z] \\ [\Lambda_x] & [\Lambda_y] & [\Lambda_z] \\ [\Lambda_x] & [\Lambda_y] & [\Lambda_z] \end{bmatrix} \begin{bmatrix} [q\bar{x}] \\ [q\bar{y}] \\ [q\bar{z}] \end{bmatrix} +$$

$$\begin{bmatrix} \left[q\bar{x} \frac{\partial^2}{\partial x \partial x} \right] & \left[q\bar{y} \frac{\partial^2}{\partial x \partial y} \right] & \left[q\bar{z} \frac{\partial^2}{\partial x \partial z} \right] \\ \left[q\bar{x} \frac{\partial^2}{\partial y \partial x} \right] & \left[q\bar{y} \frac{\partial^2}{\partial y \partial y} \right] & \left[q\bar{z} \frac{\partial^2}{\partial y \partial z} \right] \\ \left[q\bar{x} \frac{\partial^2}{\partial z \partial x} \right] & \left[q\bar{y} \frac{\partial^2}{\partial z \partial y} \right] & \left[q\bar{z} \frac{\partial^2}{\partial z \partial z} \right] \end{bmatrix} \begin{bmatrix} [\Phi] \\ [\Phi] \\ [\Phi] \end{bmatrix}$$

Pictorially...



Ongoing Work

- Implement rudimentary $O(n \log n)$ tree-code without multipole translation operations
- Derive multipole translation operations for $\Lambda_x, \Lambda_y, \Lambda_z$ “potentials” for full-blown FMA
- Incorporate code into MD efforts at Schlick lab
- Parallel implementation

Acknowledgements

- Tamar Schlick (New York University)
- Xiaobai Sun (Duke University)
- John Board (Duke University)

Efficient Use of Fast Electrostatics in Molecular Dynamics

R. Skeel, M. Nelson, T. Bishop, K. Schulten

Theoretical Biophysics Group
Beckman Institute
University of Illinois

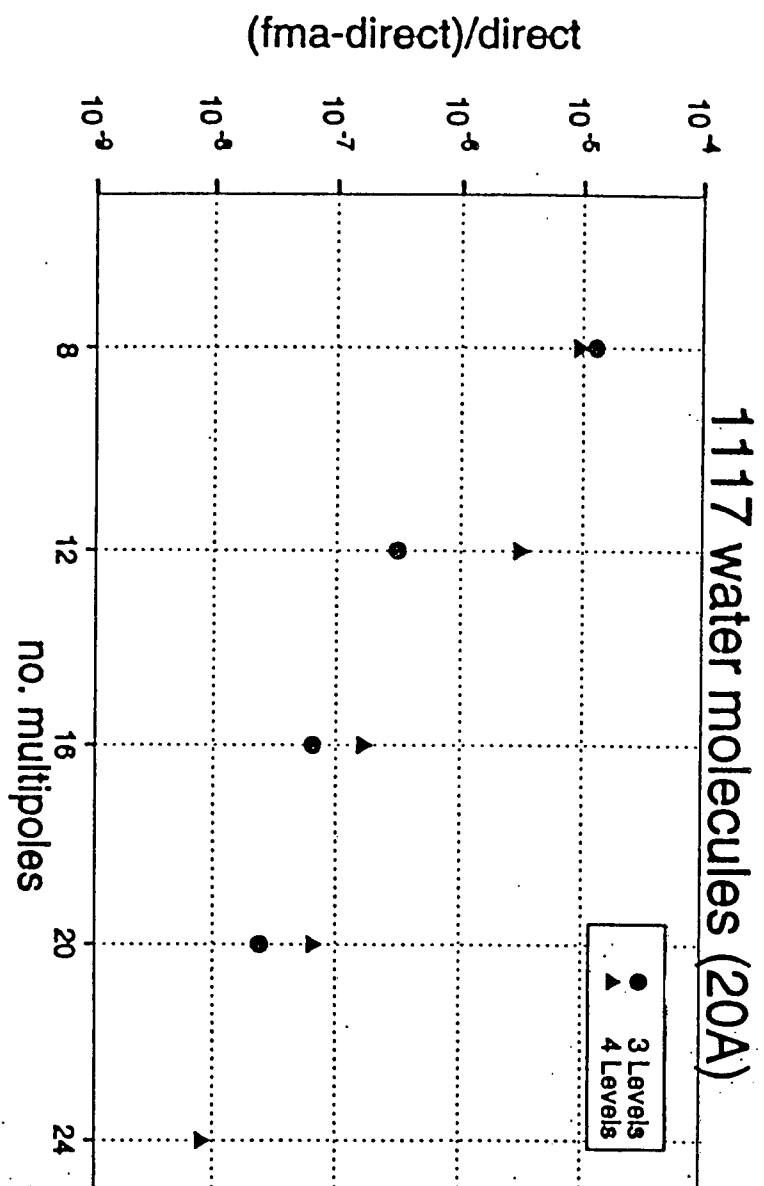
long-range electrostatics must often
be included

e.g. protein-DNA, 30000 atoms

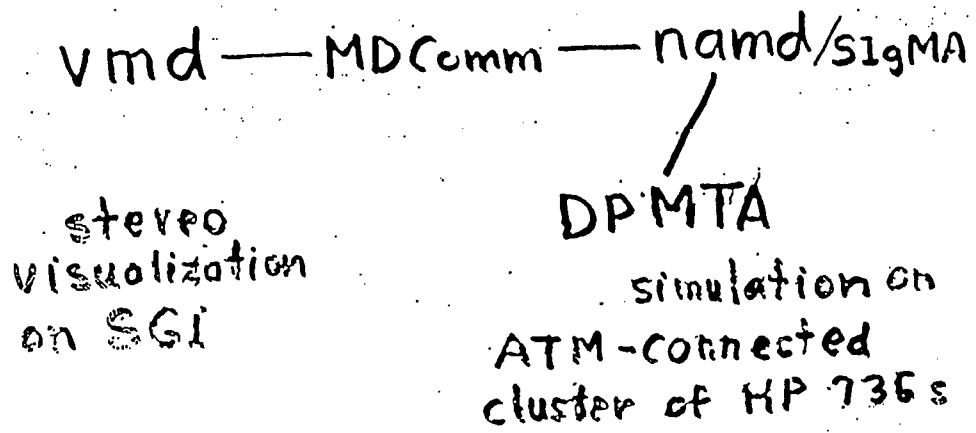
direct method N^2

FMM to s. prec. $3000N$

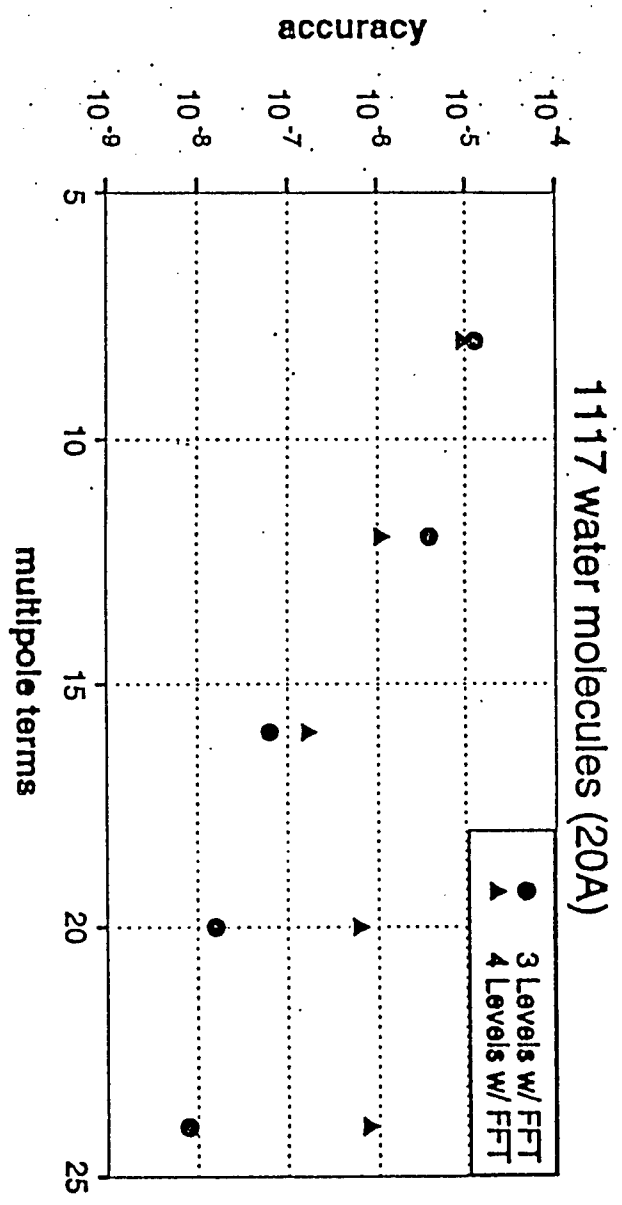
10 times the cost of using cutoffs



MDScope

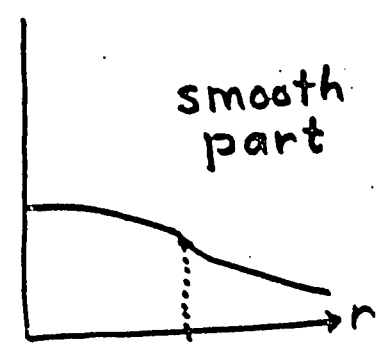
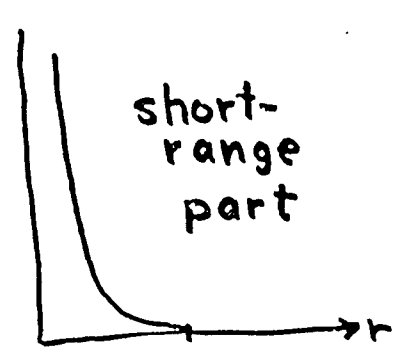
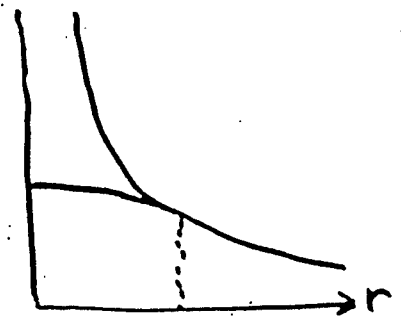


a courtesy of DPMTA



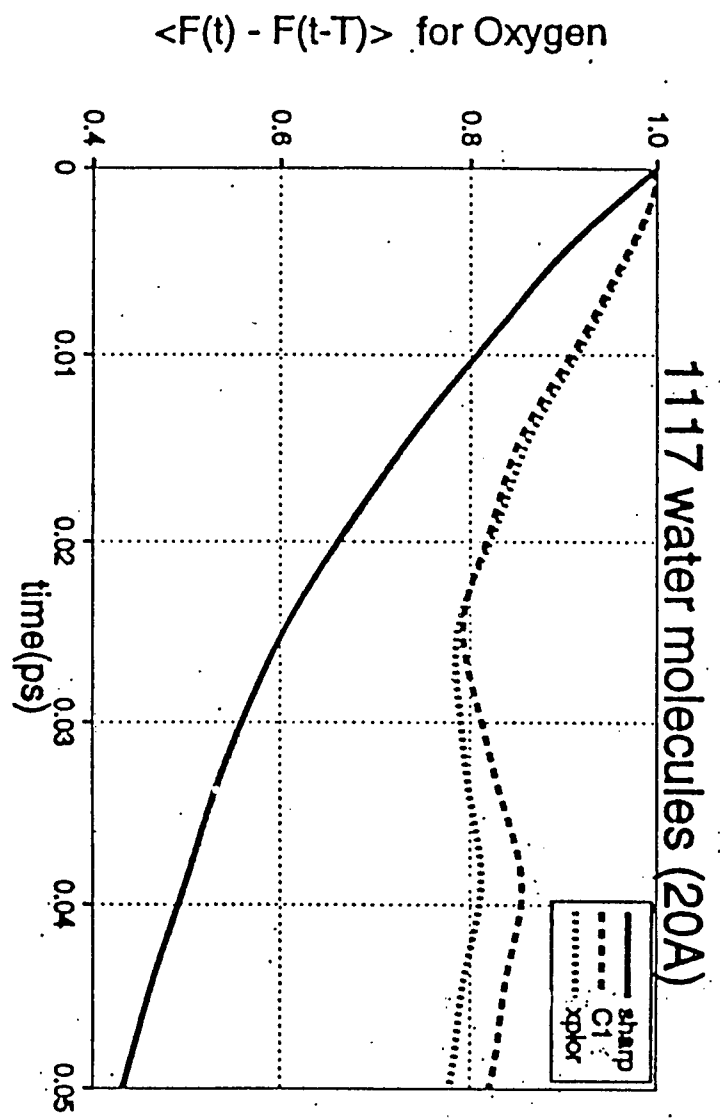
idea (partly A. Windemuth)

#1 artificial splitting of electrostatics



#2 use long timestep for smooth part
e.g. 20 fsecs

⇒ 50% premium for full electrostatics



long-timestep methods 7
 x positions, p momenta, M masses

$$\frac{d}{dt} x = M^{-1} p$$

F short-range forces, G smooth forces

$$\frac{d}{dt} p = F(x) + G(x)$$

idea

- (i) sample G infrequently
- (ii) solve "reduced problem"
involving F and G -samples

worthwhile if either

- (i) $\text{cost}(G) \gg \text{cost}(F)$ and/or
- (ii) long timesteps w/ F are economical

example 1. *LIN*

example 2. impulse method
aka Verlet-I, RESPA

$$\bar{P}^n = P^n + \frac{\Delta t}{2} G(X^n);$$

$(X^{n+1}, \bar{P}^{n+1}) = \Delta t$ -flow of

$$\frac{d}{dt} x = M^{-1} p, \quad \frac{d}{dt} p = F(x)$$

applied to (X^n, \bar{P}^n) ;

$$P^{n+1} = \bar{P}^{n+1} + \frac{\Delta t}{2} G(X^{n+1})$$

symplectic, reversible
if $F+G \dots$

e.g. 2-timestep method

$O(\Delta t)$ impulse

example 3. Verlet-X method

9

$$G^n = G(x^n);$$

$(x^{n+1}, \bar{p}^{n+1}) = \Delta t$ -flow of

$$\frac{d}{dt} x = M^{-1} p, \quad \frac{d}{dt} p = F(x) + G^n$$

applied to (x^n, p^n) ;

$$p^{n+1} = \bar{p}^{n+1} + \frac{\Delta t}{2} (G(x^{n+1}) - G^n)$$

$O(\Delta t^2)$ impulse

example 4.

Verlet-II method

$$G^n = G(x^n)$$

$(x^{n+1}, p^{n+1}) = \Delta t$ -flow of

$$\frac{d}{dt}x = M^{-1}p, \quad \frac{d}{dt}p = F(x) + G^n + 3\frac{t^{n+\frac{1}{2}}}{\Delta t}(G^n - G^{n-1})$$

applied to (x^n, p^n)

numerical experiments w/ water
in progress

Multiple timesteps $\frac{d}{dt}x = M^{-1}p \quad \frac{d}{dt}p = F_0(x)$ //

$$\begin{aligned}
 F_0 &= G_0 + F_1 \\
 &= G_0 + G_1 + F_2 \\
 &\vdots \\
 &= G_0 + G_1 + \dots + G_{n-1} + F_n \\
 &\quad \Delta t \quad 2^{-1}\Delta t \quad \quad 2^{1-n}\Delta t \quad 2^{-n}\Delta t
 \end{aligned}$$

j -th level timestep (impulse method)

step($2^{-j}\Delta t, F_j$) =

if $j < n$ then

$$p := p + 2^{-j-1}\Delta t G_j(x);$$

$$\text{step}(2^{-j-1}\Delta t, F_{j+1});$$

$$\text{step}(2^{-j-1}\Delta t, F_{j+1});$$

$$p := p + 2^{-j-1}\Delta t G_j(x)$$

else

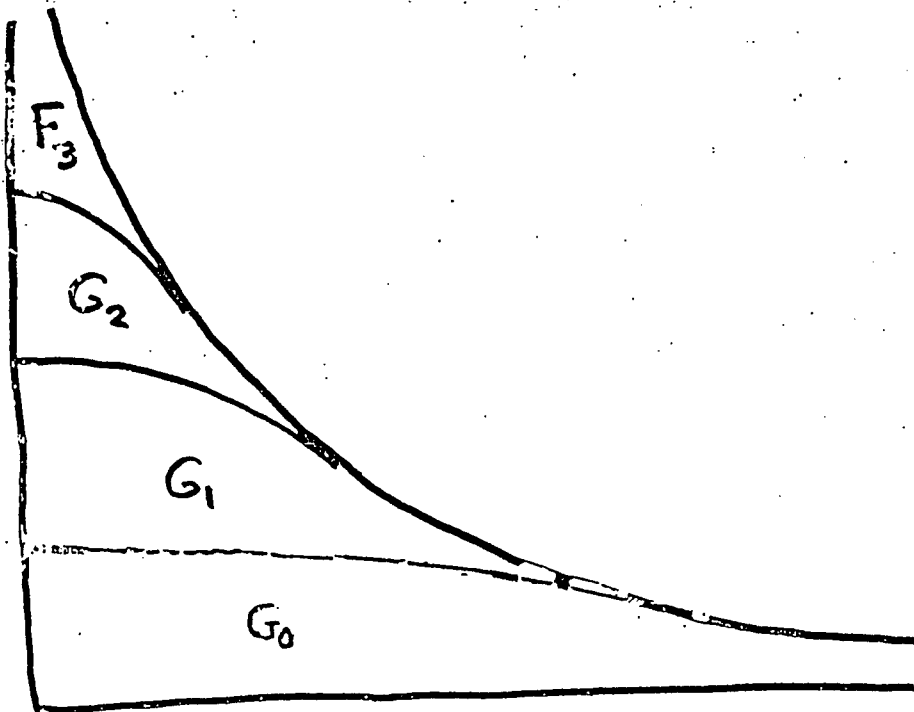
$$p := p + 2^{-n-1}\Delta t F_n(x);$$

$$x := x + 2^{-n}\Delta t \cdot M^{-1}p; \leftarrow$$

$$p := p + 2^{-n-1}\Delta t F_n(x)$$

end if

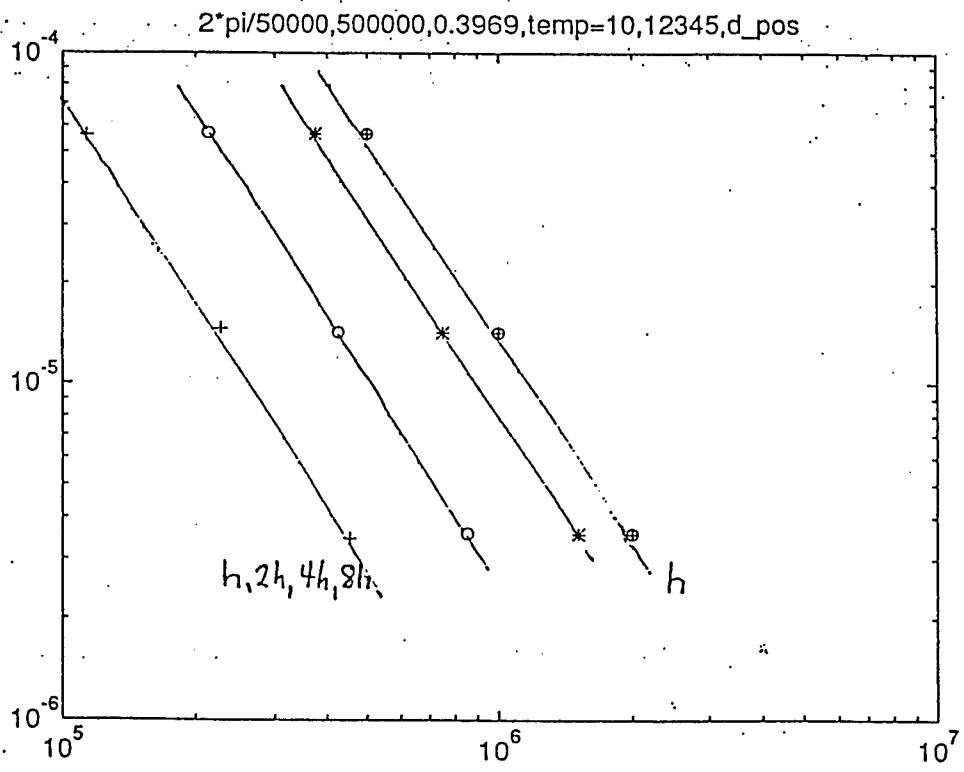
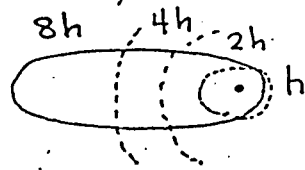
flow of $\frac{d}{dt}x = M^{-1}p \quad \frac{d}{dt}p = 0$



Z19A

first an example

Kepler problem, eccentricity 0.9



Reduced Variable Molecular Dynamics

Oren M. Becker, Tel-Aviv U.

Harvard U.

Moldyn Inc.

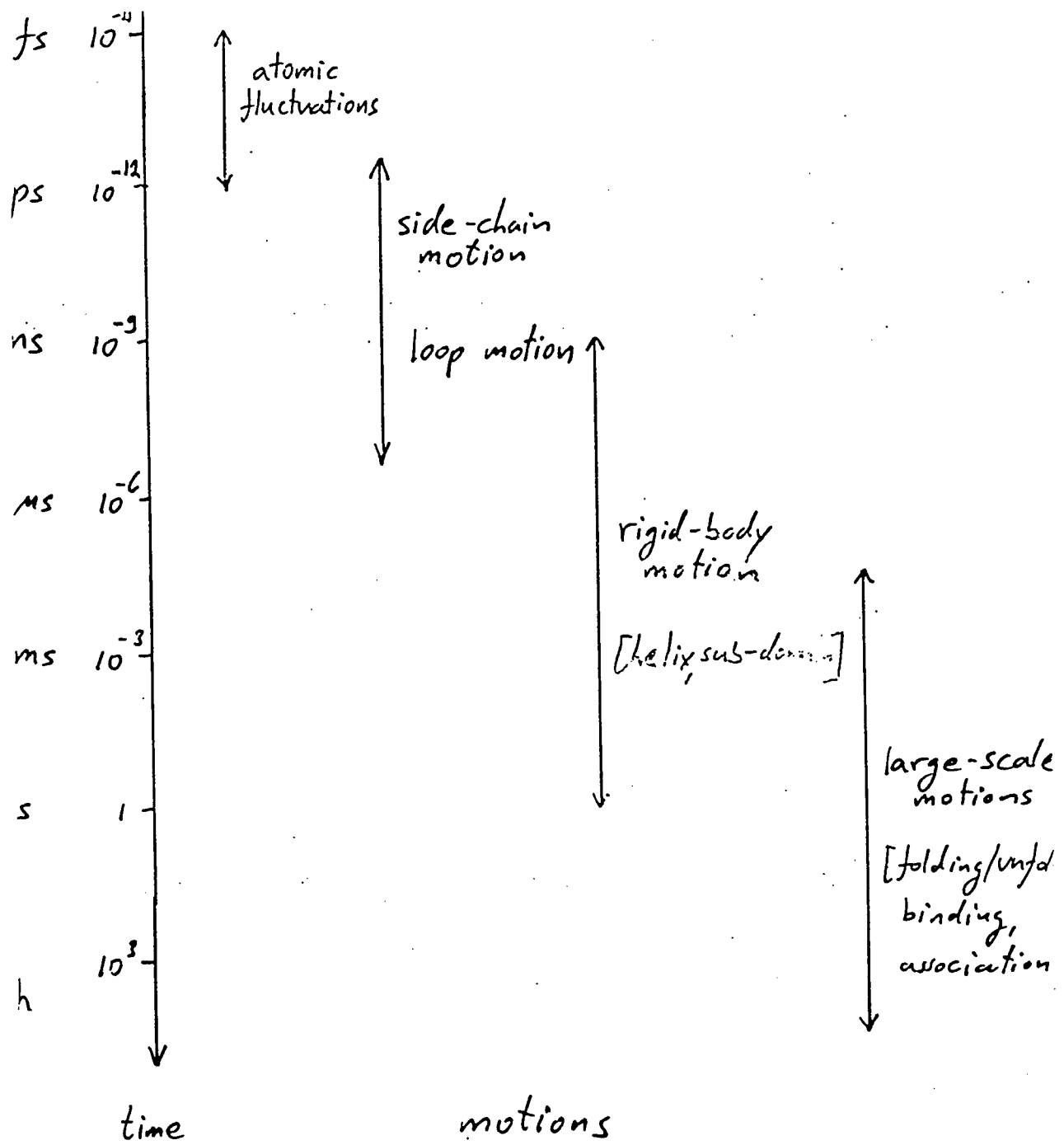
becker@sapphire.tau.ac.il

Proteins

Large system

Condensed

3D structure



Protein Dynamics

Goal:

Study Dynamics of Large Biomolecules
on "biological" time scales.

- ⇒ (1) Thermodynamic properties.
- (2) Biological function & activity.
- (3) Design new drugs or proteins.

Molecular Dynamics

1. Atom-based models

2. Empirical force-field:

$$E = \sum \text{bonds} + \sum \text{angles} + \sum \text{dihedrals} + \\ \sum \text{electrostatic} + \sum \text{vdW} + \dots$$

3. Numerical integration of
Newton's equation of motion:

$$m_i \ddot{r}_i = -\nabla_i V(r_1, \dots, r_N)$$

in Cartesian coordinates

i.e.,
$$r_i = x_i \hat{n}_x + y_i \hat{n}_y + z_i \hat{n}_z$$

Limited Length of Simulation

1. Small integration time-step

$$\bar{\nu}_{\max} \approx 3000 \text{ cm}^{-1} \Rightarrow \tau_{\min} \approx 10 \text{ fs}$$

$$\frac{\tau_{\min}/2}{\Delta t} \approx 10 \Rightarrow \Delta t = 0.5 \text{ fs}$$

2. N^2 non-bonded interactions

($>90\%$ of computation time)

3. Treat all atoms on equal basis,
but often one is interested only
in a smaller sub-system (eg, active site).

Methods to extend simulation time

1. Constraints to eliminate high frequency motions.

If eliminating X-H bonds $\Rightarrow \bar{\nu}_{\max} \approx 1500 \text{ cm}^{-1}$

$$\Rightarrow \Delta t = 1 \text{ fs}$$

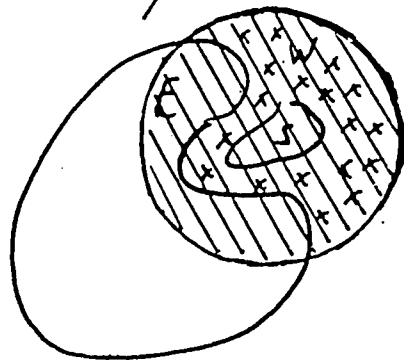
e.g. SHAKE iterative $s_x = \frac{(r_{ij}^2 - d_{ij}^2)}{d_{ij}^2} < \epsilon$

2. Cut off on non-bonded interactions
(today \rightarrow multipole expansion electrostatics).

3. Reduce no. of DOF (approximations)

(i) Vacuum simulations

(ii) Stochastic boundary



Today simulations of a 1500 atom system
(~ 100 a.a.) are typically taken to 1 ns.

Larger systems and/or longer times
only with Approximate Methods:

(I) Stochastic Dynamics

propagate Langevine EOM:

$$m_i \ddot{\mathbf{r}}_i = -\nabla_i V(\mathbf{r}_1, \dots, \mathbf{r}_N) - \alpha_i \dot{\mathbf{v}}_i + R(t)$$

(II) Harmonic Dynamics

propagate along normal-modes.
(very limited)

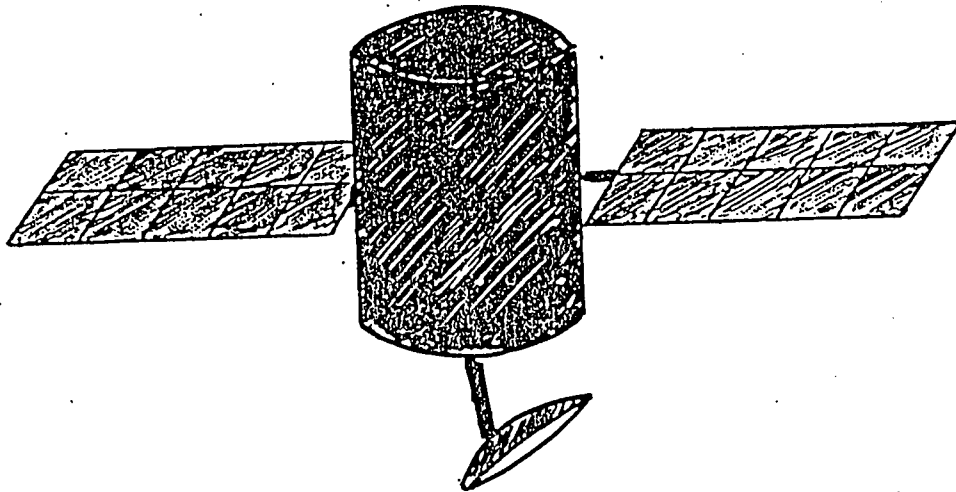
(III) Rigid body simulations

Get to longer times by
focusing on biologically interesting
motions, while retaining a
correct but cheap description
of faster motions.

Variable Reduction Techniques

Generate EOM of mechanical systems with rigid and flexible components.

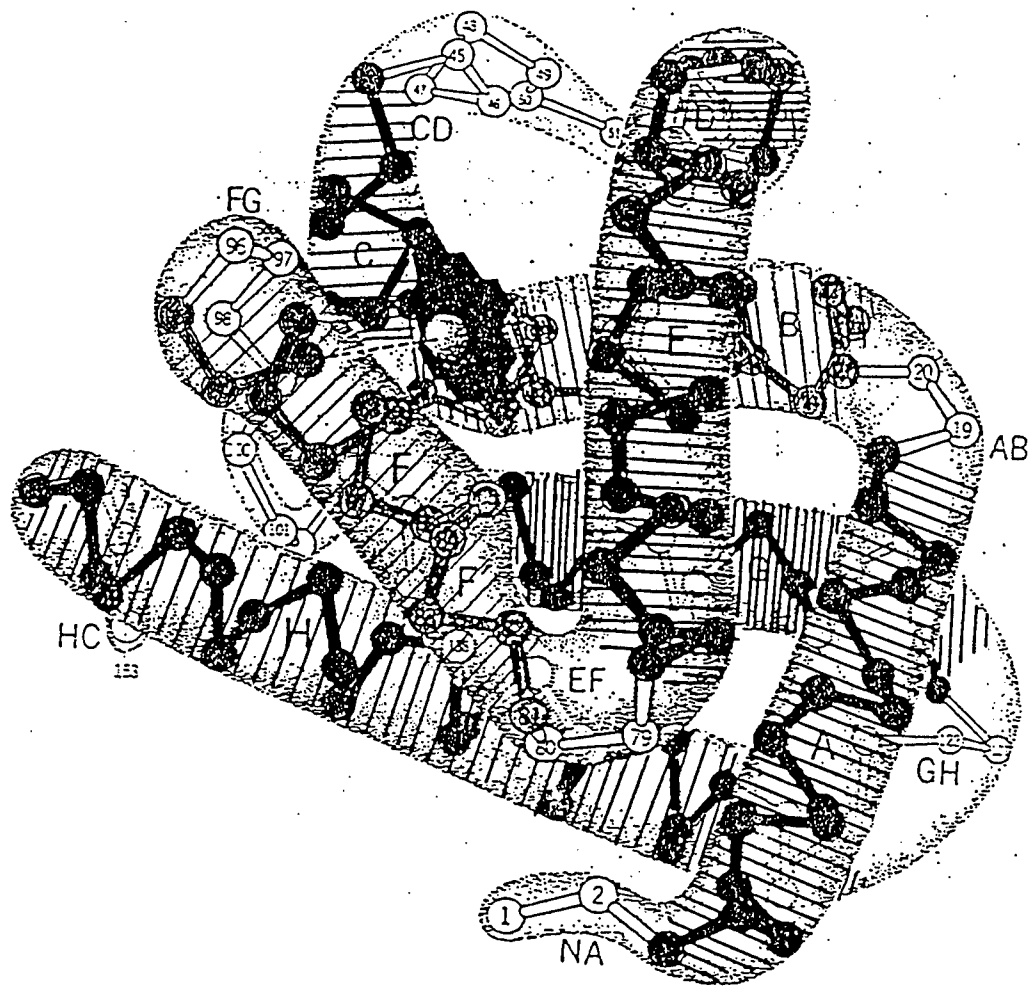
Aerospace dynamics



Bodies connected by hinges (constraints).

Rigid bodies - Euler-Newton EOM
(rotation & translation)

flexibility - low frequency normal-modes.



Deoxy Myoglobin

Reduced Variable Molecular Dynamics

1. Substructuring

Group atoms together to define 'Bodies'.

Criterion: Relative displacements small compared to inter-body displacements.

e.g. α -helix

but not loop-regions, terminal arms.

⇒ (*) Mixture of 'bodies' and 'particles'.

(*) All bodies must be flexible.

2. Equations of Motion

Particles - Newton EOM (r)

Bodies - Euler-Newton EOM (r, θ)

Hinge constraints

3. Flexibility

Modeled by a reduced set of "component" normal-modes (low frequency) remove high freq., i.e., local vibrations.

For each body generate modes in the field of the rest of the molecule.

'Body-modes' are coupled to large-scale motion through rigid-body kinematics.

Separation of variables:

$$U(R; t) = \sum_{k=1}^{N_m} \underbrace{\Phi_k(R)}_{\text{eigenvectors}} \underbrace{\xi_k(t)}_{\text{time dependent amplitude}}$$

Component mode synthesis:

$$m_A \ddot{X}_A = f_{AA}(X_A) + f_{AB}(X_A, X_B)$$

solve for $X_A = \Phi(R) \xi(t)$ to get

$$(\Phi^T m_A \Phi) \ddot{\xi}_A = - \underbrace{(\Phi^T K_{AA} \Phi)}_{\text{linear}} \xi_A + \underbrace{\Phi^T f_{AB}(\xi_A, \xi_B)}_{\text{non-linear}}$$

Application

First tests AMBER-MBO(N)D

J.D. Turner et al. 1993

Extensive interface CHARMM-MBO(N)D

O.M. Becker et al.

Expected speed-up

- (I) Large reduction in n° of DOF.
- (II) Simplify energy evaluation.
- (III) Larger integration time-step. (MTS)
- (IV) Efficient constraint algorithm.

expected 10-100-fold speed-up.

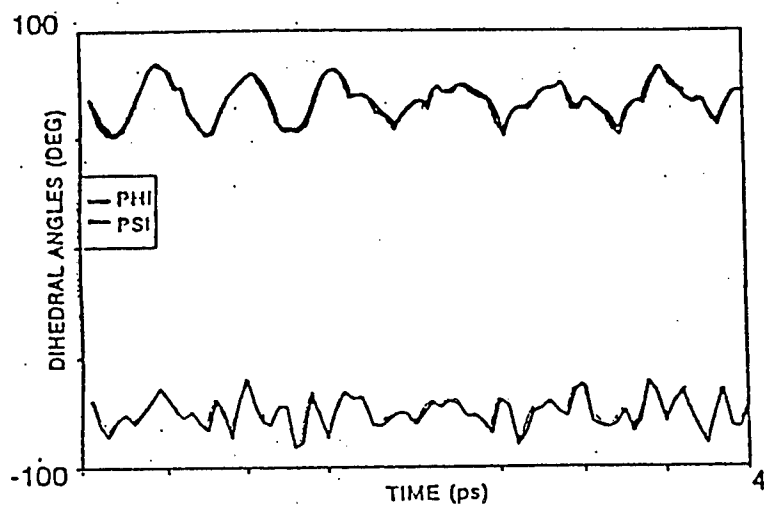
1.0	3.33 (0.123)	-12.34 (1.50)	00.11 (0.24)
0.2	3.48 (0.0144)	-12.20 (1.00)	00.01 (8.23)
<u>ZHAKK</u>			
5.0	3.38 (0.031)	-12.31 (1.55)	00.01 (8.20)
1.0	3.40 (0.0015)	-12.32 (0.28)	00.00 (8.80)
<u>WBO(M)D</u>			
Δt (ls)	Лотшл ешсггλ	φ	λ

ερσςιηε ριβεεηιηε - .4 βε

Glycine dipeptide

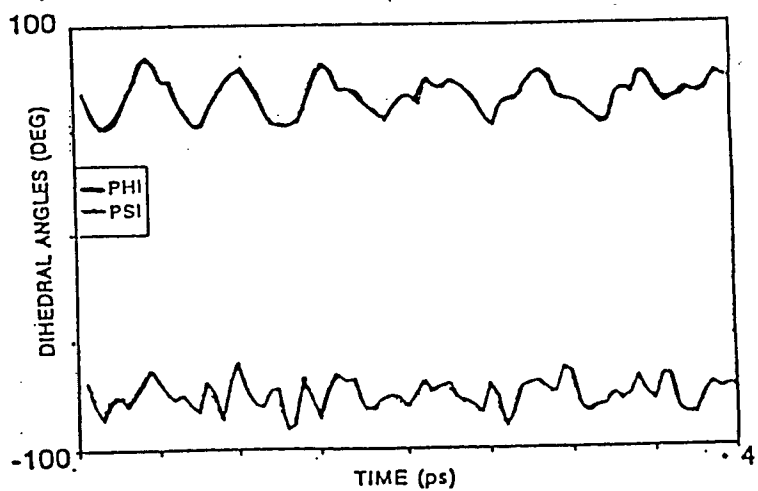
SHAKE

Verlet, TIME = 4ps, $\Delta T = 0.5fs$



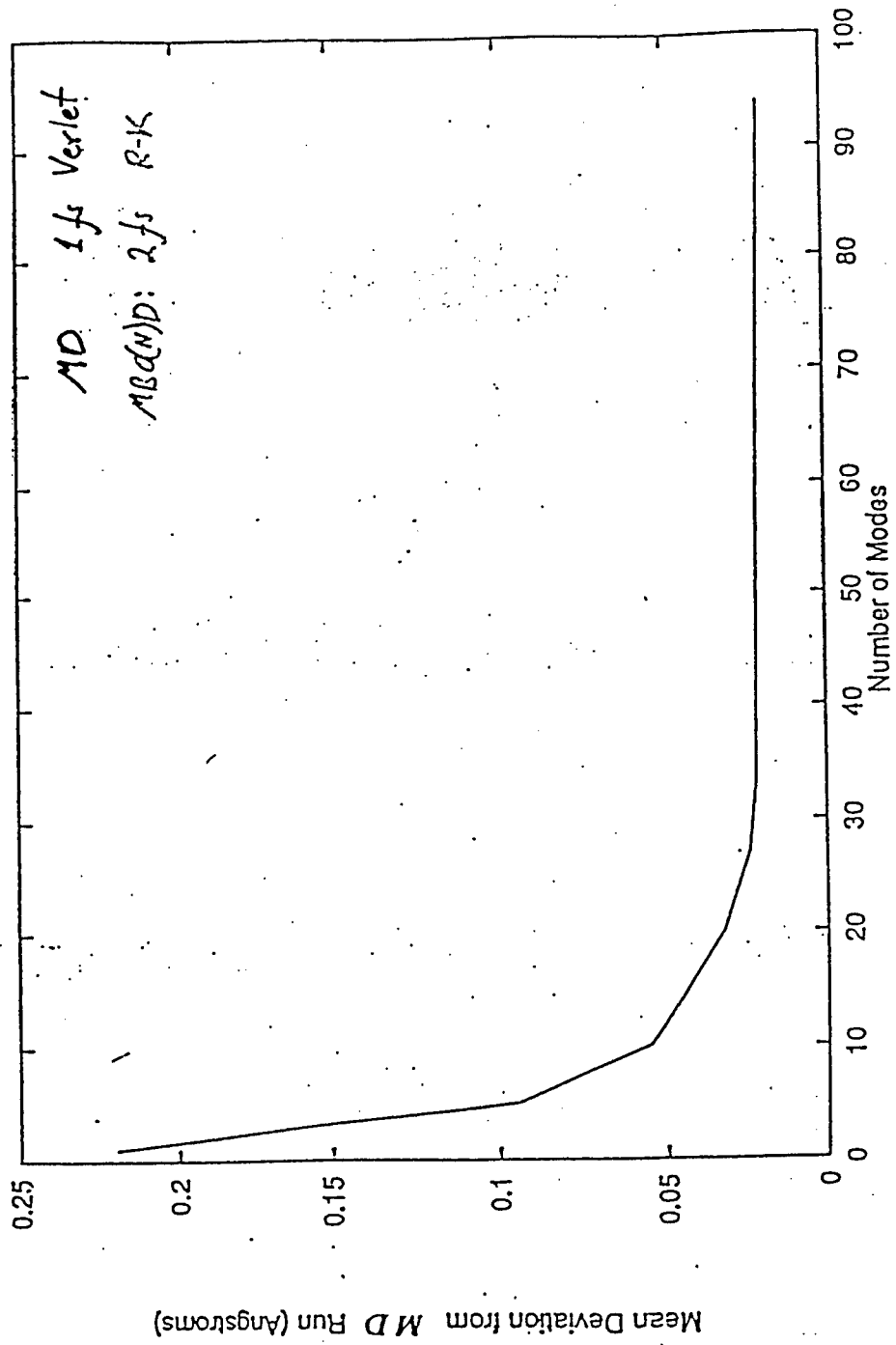
MBO(N)D

RK4, TIME = 4ps, $\Delta T = 1fs$

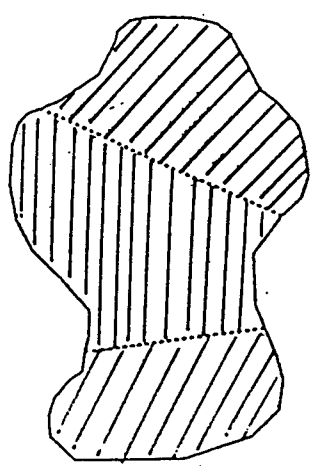


Deca-Gly, 20 ps

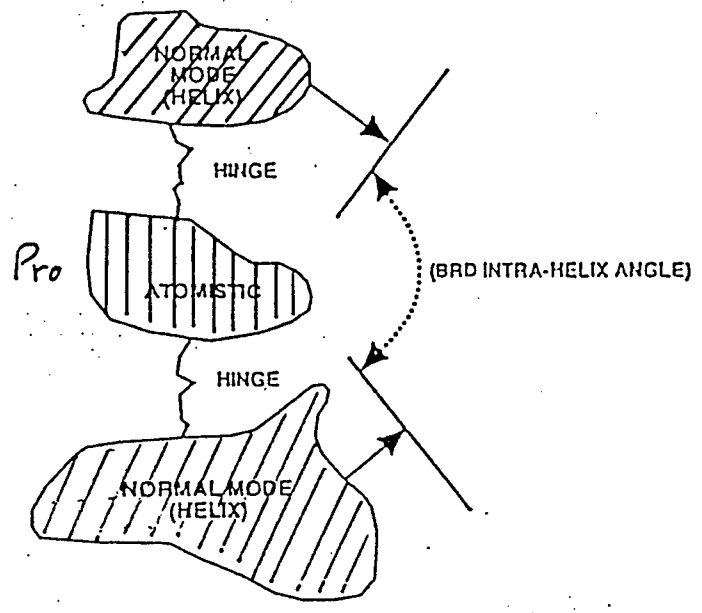
α -helix, 1.5 μ s



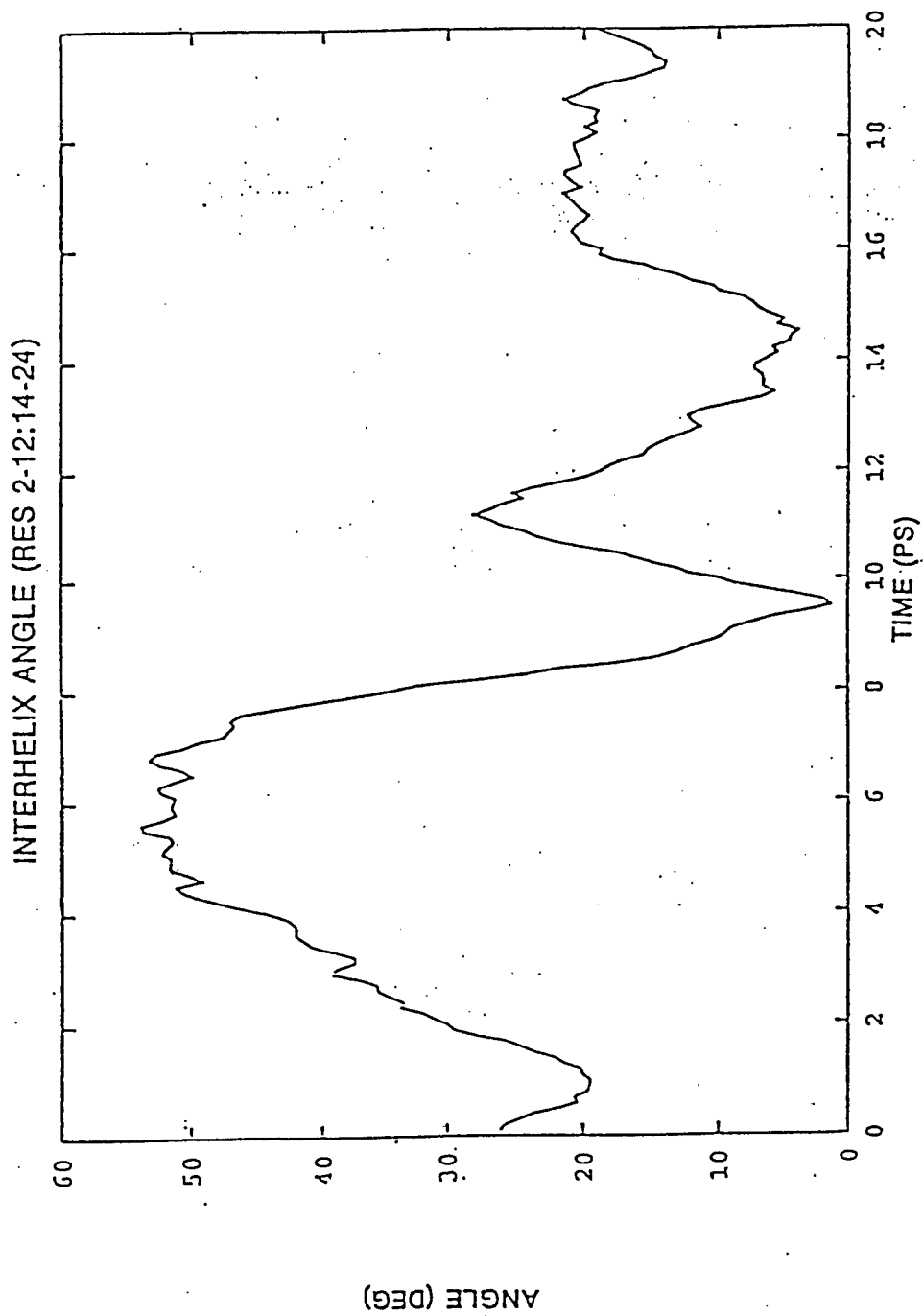
25 aa from bacteriorhodopsin (237 atoms)



GLOBAL BEHAVIOR



SUBSTRUCTURE MODELS



25-aa. from Bacteriorhodopsin

Type of run	R.m.s. fluctuation (degrees)
MD	<u>17.55</u>
One flex body	8.05
Three rigid	1.5
<u>Flex (8) flex (4) flex (8)</u>	10.79
<u>Flex (2) particle flex (2)</u>	12.01
<u>Flex (8) particle flex (8)</u>	<u>18.18</u>

Summary

Reduced Variable MD - new possibility for long simulations of large biomolecules.

Open Questions

- partitioning strategies.
- mode generation & update.
- accuracy vs. efficiency.

Present applications

4-helical bundle (rop)

8-helical bundle (myoglobin)

Advanced applications

(1) Enzyme hinge-bending motion

e.g. Liver Alcohol Dehydrogenase (2×3000 heavy atoms)

(2) Membrane dynamics.

Oren M. Becker	(TAV)
Leo S.D. Caves	(York)
Robert Nagle	(Harvard)
Herman Vlijman	(Harvard)
Martin Karplus	(Harvard)
Hon M. Chun	(Moldyn inc.)

Algebraic Multigrid (AMG)

*Multi-level strategy for solving
matrix problems $AU = f$*

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

A sparse and (approx.) positive type:

$$a_{ij} \leq 0 \quad (i \neq j) \quad \text{and} \quad \sum_j a_{ij} \geq 0$$

References on algebraic multigrid:

Brandt, A.: *Algebraic multigrid theory: the symmetric case*,
Appl. Math. Comp. 19 (1986)

Ruge, J.; Stüben, K.: *Algebraic Multigrid*. In: "Multigrid Methods",
McCormick, S. (ed.), *Frontiers in Applied Mathematics*, Vol. 5, SIAM,
Philadelphia (1987)

AMG-1

Algebraic characterization of *smoothness*

GS-relaxation for i -th unknown: $u_i \rightarrow \bar{u}_i$ where

$$\bar{u}_i = \frac{1}{a_{ii}} (f_i - \sum_{j \neq i} a_{ij} u_j) = u_i + \frac{d_i}{a_{ii}}$$

$d_i := f_i - \sum_j a_{ij} u_j$ = residual *before* relaxing

$$\rightarrow \bar{e}_i = e_i - \frac{d_i}{a_{ii}}$$

The error e is *smooth* \iff relaxation is stalling
 $\iff \bar{e}_i \approx e_i$ for all i
 $\iff |d_i| \ll a_{ii} |e_i|$ for all i
 i.e., the residual is much smaller than the error!

Since $d_i = a_{ii} e_i + \sum_{j \neq i} a_{ij} e_j$

smooth error e satisfies approx.

$$a_{ii} e_i = - \sum_{j \neq i} a_{ij} e_j \quad \text{for all } i$$

This relation is the basis both for coarsening and interpolating

Remember (variational principle!):

Smooth error has to be "close to" $\mathcal{R} := \text{range}(\text{interpol})$

Note that, generally, (algebraically) smooth error is NOT necessarily (geometrically) smooth and vice versa!

AMG-3

General approach

Fix GS as smoothing method

Construct subset of unknowns ("coarser levels")
and "interpolation" such that *

smooth error is interpolated well, i.e.

smooth error "close to" $\mathcal{R} := \text{range}(I_H^h)$

coarsening is "efficient", i.e.

fast reduction of # unknowns

no serious fill-in on coarser levels

Two-level method by variational approach

Multilevel method by recursive application

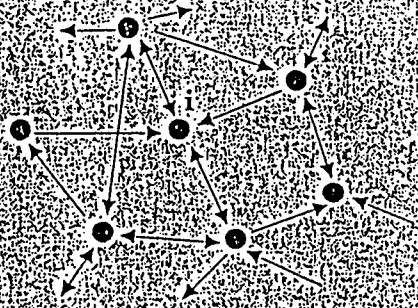
*) This construction is part of an AMG algorithm!

Graph representation of matrix A

With each unknown, i ,
associate a "point":

$\Omega := \{i\} = \text{fine "grid"}$

$N_i := \{j \neq i : a_{ij} \neq 0\}$
neighborhood of i



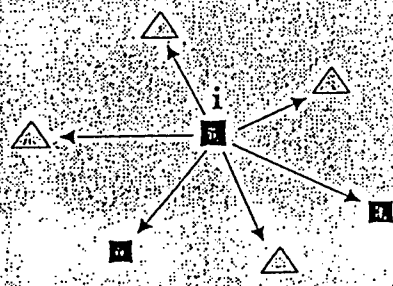

$i \rightarrow j$: i is connected to j : $a_{ij} \neq 0$
 $j \rightarrow i$: j is connected to i : $a_{ji} \neq 0$

Requested: splitting $\Omega = F \cup C$
(C-points: coarser "grid")

How to interpolate *smooth* error e
at F-points from those at C-points?

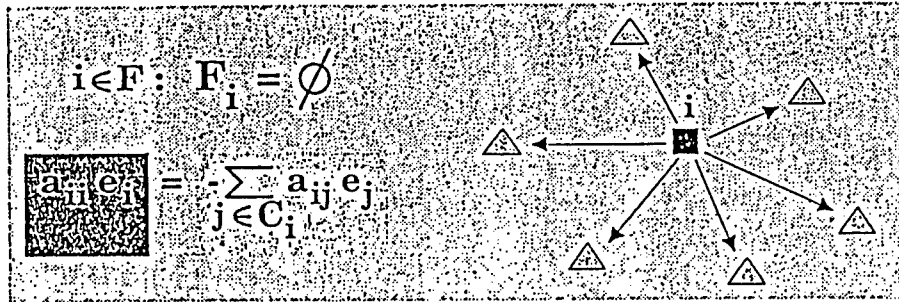
\blacksquare : F-points \triangle : C-points
 $F_i := F \cap N_i$ $C_i := C \cap N_i$

smooth error satisfies:

$$a_{ii} e_i = - \sum_{j \in C_i} a_{ij} e_j - \underbrace{\sum_{j \in F_i} a_{ij} e_j}_{\text{has to be "eliminated"}}$$



A simple but inefficient coarsening strategy

Choose C such that for all $i \in F$: $N_i \subset C$
 i.e., F -points are *totally decoupled!*



Two-level method:

GS-relaxation with C - F -ordering
 Above error relation holds *exactly* at F -points!

Coarse-grid correction

$$\begin{aligned}
 \Omega_h &:= \Omega & \Omega_H &:= C \\
 (I_H^h e^h)_i &:= \begin{cases} e_i^H & (i \in C) \\ - \sum_{j \in C_i} a_{ij} e_j^H / a_{ii} & (i \in F) \end{cases}
 \end{aligned}$$

Is a *direct solver*, i.e. residual=0 after one cycle!
 (After relaxation of F -points: $e^h \in \mathcal{R} := \text{range}(I_H^h)$)

However, recursive application extremely inefficient:
Drastic fill-in + slow coarsening!

A practical coarsening strategy

Idea: distinguish *strong* and *weak* connections:

i is *strongly* connected to j if

$$-a_{ij} \geq \alpha \max_{k \neq i} (-a_{ik}) \quad (0 < \alpha \leq 1)$$

otherwise, we call it *weakly* connected.



Typical value in practice is $\alpha = 0.25$

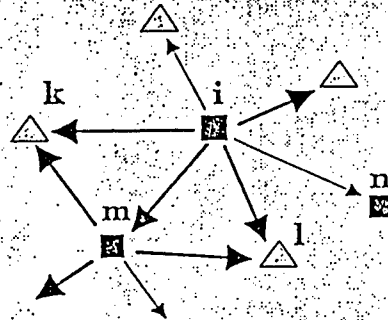
Representative situation:

$$a_{ii} e_i = - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in \bar{C}_i} a_{ij} e_j$$

"Elimination" of n and m :

$$e_n \rightarrow e_i \quad (\text{weak connection!})$$

$$e_m \rightarrow \frac{a_{mk} e_k + a_{ml} e_l}{a_{mk} + a_{ml}}$$



General requirements:

- 1) For all $i \in F$: all *strong* neighbors should be
 - either C-points used in interpolation (like k, l)
 - or strongly connected to such points (like m)
- 2) The # of C-points should be *as small as possible*

These are the basic objectives which allow for an automatic and purely algebraic coarsening process!

- (1) is responsible for good interpolation of smooth error
- (2) is important for low-cost cycles

MG convergence (unit square, uniform grid)

problem		AMG		BOXMG*	
		time/cycle setup	ρ	time/cycle setup	ρ
$-u_{xx} - u_{yy}$		0.30 1.90	0.056	0.33 0.23	0.015
$-\epsilon u_{xx} - u_{yy}$	0.001	0.33 1.53	0.084	0.33 0.24	0.043
	0.01	0.34 1.75	0.093	0.33 0.24	0.107
	0.1	0.38 2.70	0.058	0.33 0.24	0.088
	0.5	0.31 1.90	0.069	0.32 0.24	0.036
	2	0.31 1.91	0.079	0.32 0.23	0.037
	10	0.38 2.65	0.087	0.32 0.24	0.084
	100	0.33 1.74	0.093	0.32 0.24	0.104
	1000	0.32 1.55	0.083	0.32 0.24	0.043
$(100^{x+y-1} u_x)_x - u_{yy}$		0.33 1.95	0.080	0.32 0.24	0.044
$-\nabla(a\nabla u)**$		0.32 1.98	0.069	0.32 0.24	0.044

**

$a :=$	10	100
	1	1000

* BOXMG (J. Dendy) was used in its most robust form (i.e. alternating line relaxation, Galerkin,...)

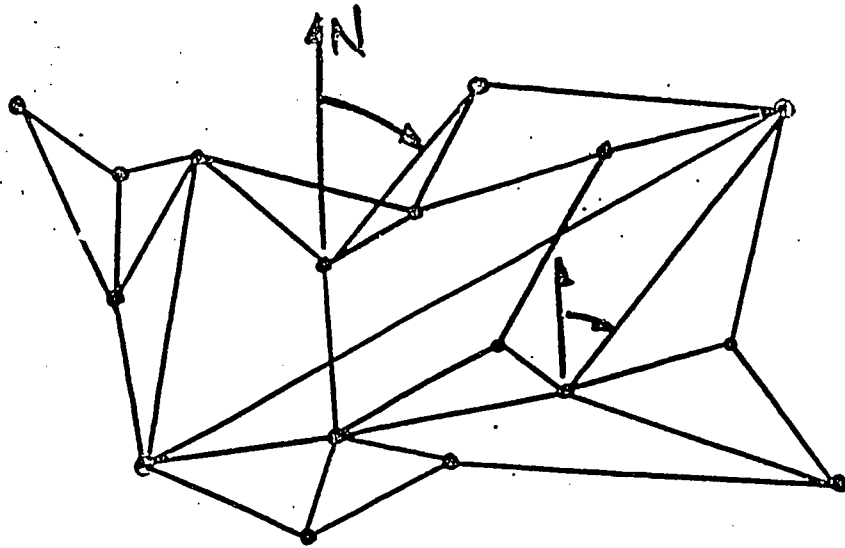
AMG-8

AMG Applied to
Functional Minimization Problems
(Unstructured meshes, vector unknowns)

John Ruge
Achi Brandt
S. McCormick
Zhao Xingjun
etc.

Supported by AFOSR, Veritas Research,...

Geodetic Survey Problems



- Sites with unknown locations (x_i, y_i)
- Measurements (distance) d_{ij}
Varying accuracy (different methods)


 Measurements (angles) θ_{ij}

= Find (x_i, y_i) that give "best" fit to data

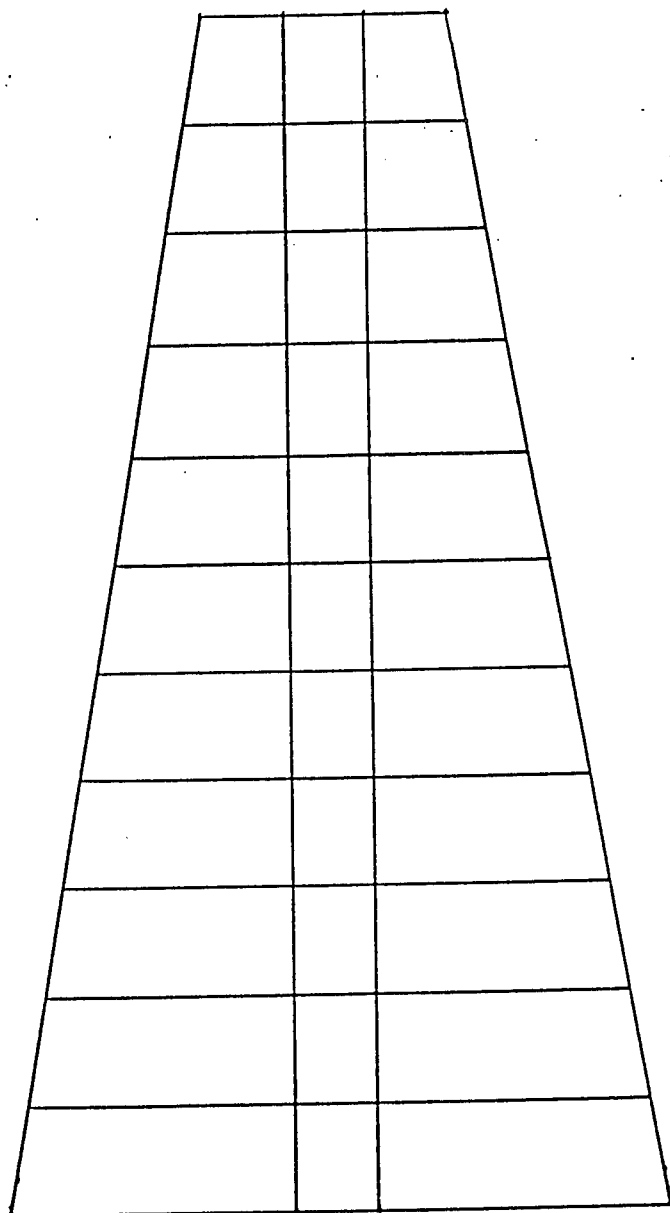
$$\sum_{ij} w_{ij} (|(x_i, y_i) - (x_j, y_j)| - d_{ij})^2 + \sum_{ij} \alpha_{ij} (\tilde{\theta}_{ij} - \theta_{ij})^2$$

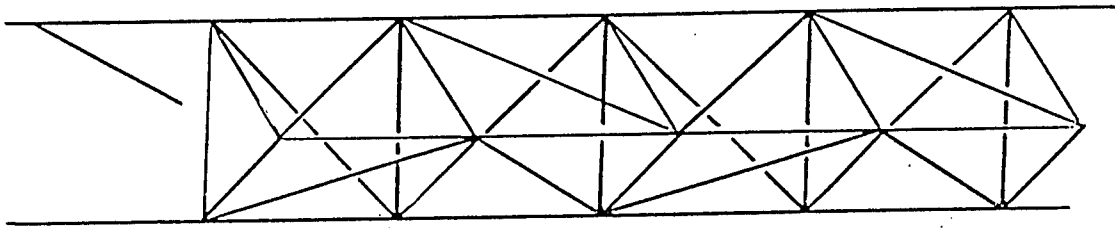
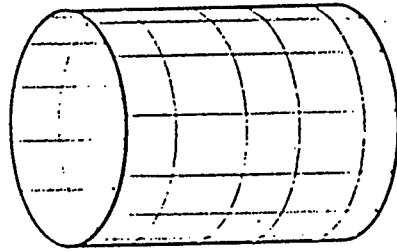
computed angle

Truss (Beam) Structures

- Small displacements -
beams behave like springs
(force = $k \Delta l$)
- Joints hinged or stiff. (gives angle term)
- Formulate energy functional
$$\sum a_{ij} (\Delta l_{ij})^2$$

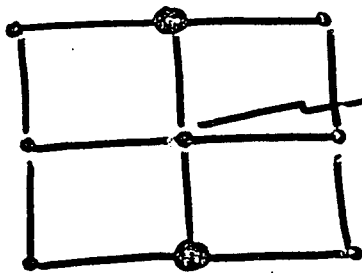
(Or sum forces, set = 0)





Vector unknowns & AMG

- Coarsen "pointwise"
- "Strong connection" now vector
⇒ each component (x_i, y_i, \dots)
must be well-determined by C .



not well-determined by
2 points :

∴ Requires modification of AMG
coarsening process.

For now, ignore CG choice.

Assume functional of the form:

$$\sum_{ij} \alpha_{ij} (\beta_{ij} (x_i - x_j) + \delta_{ij} (y_i - y_j) - \delta_{ij})^2$$

(i.e. linearized)

Minimization gives the problem

$$Au = f$$

$$u = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \end{bmatrix}$$

$$A = \sum A_{ij} \quad A_{ij} \text{ from } ij \text{ term.}$$

More generally, denote terms as A

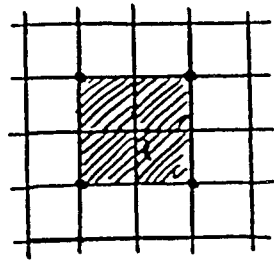
"Element" interpolation

N_i - neighboring points of i .

E_i - elements with i as a node

Instead of minimizing $v^T A v = v^T \sum_{\alpha} A_{\alpha} v$

minimize only $v^T \sum_{\alpha \in E_i} A_{\alpha} v$ to get v_i .



$\cdot \Omega^c$

/// elements used.

- point i interpolates only from points in $N_i \cap \Omega^c$.

To define element interpolation to i :

Consider the following 3 sets

$$G = \{i\} \cup \{j \notin \Omega^c : j \in N_i \mid E_j = E_i\}$$

$$F = \{j \in N_i : j \notin G, j \notin \Omega^c\}$$

$$C = \{j \in N_i : j \in \Omega^c\}$$

Remember the points: $G \ F \ C \ \dots$

$$\sum_{k \in E_i} A_k = \begin{bmatrix} A_{GG} & A_{GF} & A_{GC} & \\ A_{FG} & A_{FF} & A_{FC} & 0 \\ A_{CG} & A_{CF} & A_{CC} & \\ & 0 & & 0 \end{bmatrix}$$

Partition $v = (v_G, v_F, v_C, v_R)^T$

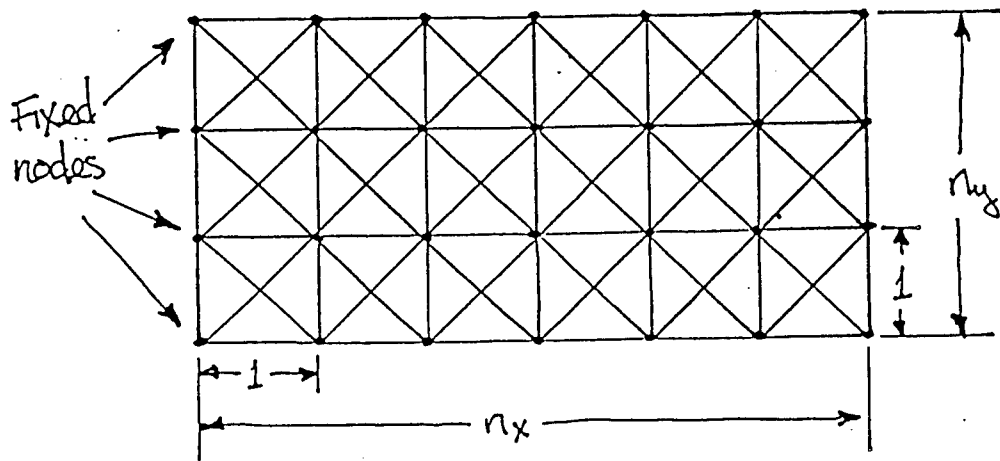
Then, with v_C fixed and $v^T \sum_{k \in E_i} A_k v$

minimal,

$$\underline{v_G = -(A_{GG} - A_{GF} A_{FF}^{-1} A_{FG})^{-1} (A_{GC} - A_{GF} A_{FF}^{-1} A_{FC}) v_C}$$

2-D Beam Structures

- Hinged joints
- Beams subject to compression/extension
- Equations derived from Hooke's Law.



n_x	n_y	V-cycle Rates	
		element int.	linear int.
16	16	.26	.20
32	8	.27	.64
64	4	.28	.97
128	2	.28	.93 +

A. Brandt & D. Bai

Multiscale Methods in
Molecular Dynamics

MULTI-SCALE (MS) RESEARCH METHODOLOGY

Problem size (# atoms) = n

Objective: Total work = $O(n)$

Research stages:

increasingly complicated problem

But at each stage $n \rightarrow \infty$

At each stage add one complication,
insisting on still obtaining $O(n)$

DIMENSIONS

1D, 2D, 3D

TOPOLOGIES

Many non-bonded

- Identical atoms
- Identical small molecules
- Several species

Bonded chains :

- Identical atoms, stretched
- Helics
- Identical amino acids

Chains + solvent

General

OBJECTIVES

MS

Energy minimization

- Near-minimum start
- Far start
- Homogenization

MG, AMG
ms annealing

Equilibrium statistics

- $O(n)$ per sample
- $O(1)$ per sample
- ↓
- Homogenization

ms
Monte-Carlo

Dynamics

- Large Δt
- Very large Δt

MG, AMG
ms annealing

Stochastic dynamics

ms Monte-Carlo at each time step

POTENTIALS

Bonds

- Harmonic (1) constant coupling
(2) strongly variable
- Bond length
- Length + angle
- Length + angle + torsion

Non-bond, local

- Van der waals
- Hydrogen bonding

Non-local

- Electrostatics : (1) constant dielectric
(2) variable dielectric

Combinations

Stochastic Dynamics

At time t^n : position vector x^n
velocity vector V^n
potential energy $E(x^n)$.

Deterministic implicit time step:

$$-\nabla E(x^{n+1}) = p^{n+1}$$

$$p_k^{n+1} = m_k \frac{\delta V_k}{\delta t}, \quad \delta V_k = V_k^{n+1} - V_k^n, \quad V_k^{n+1} = \frac{x_k^{n+1} - x_k^n}{\delta t}$$

$\iff x^{n+1}$ minimizes $H_n(x^{n+1})$

$$H_n(x^{n+1}) = \frac{1}{2} \sum m_k (\delta V_k)^2 + E(x^{n+1})$$

Stochastic time step:

$$P(x^{n+1}) \sim e^{-\beta H_n(x^{n+1})}$$

$$\beta = \frac{1}{k_B T}$$

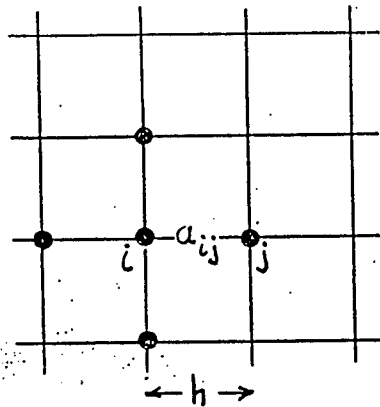
Results of the stochastic step :

- $\langle p^{n+1} \rangle = -\nabla E(x^{n+1})$
- $\langle (p_k^{n+1})^2 \rangle - \langle p_k^n \rangle^2 = m_k / \beta$
- Local components, with
oscillation period $\ll \delta t$
are fully thermalized:
 $P(x^{n+1}) \sim e^{-E(x^{n+1})/k_B T}$
- Large scale components, with
oscillation period $\gg \delta t$
satisfy Newton law.
- Easy multiscale equilibration

Relaxation

- Atom by atom } minimization
 { Monte-Carlo
- Only local forces
 including local part of global forces
- For each atom - use its
 natural (internal) coordinates

$$E(u) = \frac{1}{2} \sum_{\langle i,j \rangle} a_{ij} \left(\frac{u_i - u_j}{h} \right)^2 - \sum_i f_i u_i$$



$$E(U) = \min_u E(u)$$

$$0 = \frac{\partial E}{\partial u_i} \Big|_{u=U} = \frac{1}{h^2} \sum_{\langle j,i \rangle} a_{ij} (u_i - u_j) - f_i$$

$$\frac{1}{h^2} \sum_{\langle j,i \rangle} a_{ij} (U_i - U_j) = f_i$$

$$\rightarrow \frac{\partial}{\partial x} \left(a \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(a \frac{\partial U}{\partial y} \right) = f$$

$$\iff E(U) = \min E(u)$$

$$E(u) = \iint \left[\frac{a}{2} (u_x^2 + u_y^2) - f u \right] dx dy$$

Algebraic Relaxation Theory

$$Ax = b \quad A = \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mm} \end{pmatrix} \quad \text{error } e = x - \tilde{x} = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$$

residuals $r = Ae = b - A\tilde{x}$

Normalized residuals $\bar{r}_i = r_i / |a_{i1}| = a_{i1} e / |a_{i1}|$

Kaczmarz Relaxation: GS for $AA^T y = b$
 x

i.e. $\tilde{x} \leftarrow \tilde{x} - a_i^T (r_i / |a_{i1}|^2)$, ($i = 1, \dots, m$).

Theorem. slow convergence $\Leftrightarrow |\bar{r}| \ll |e|$

$|e|^2$ is reduced per sweep at least by $|\bar{r}|^2 / \left(\max_i \frac{\sum_j |a_{ij}|^2}{|a_{i1}|^2} \right)$

- Slowly converging errors are special
They can be approximated by a smaller system (coarser grid). i.e., by $I_H^h U^H$

Discrete h-elliptic eqs. slow convergence

$$\Leftrightarrow |L^h e^h| \ll |L^h| |e^h| \Leftrightarrow e^h \text{ smooth on scale } \eta$$

A. Simple Energy Basins real u:

Slow to equilibrate
and be sampled:

$$I_H^h \psi^H$$

Coarse Monte-Carlo
Hamiltonian:

$$E^h(\tilde{u}^h + I_H^h \psi^H) \\ \equiv E^H(\psi^H)$$

Multigrid cycle:

- on 1. Monte-Carlo passes
each 2. γ cycles on coarser
level: 3. Monte-Carlo passes

Near equilibration and decorrelation
in one cycle ($\gamma = 2$).

Small work on coarser grids ($\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$)

Coarse-level variables

- Gridpoints
- Subset of atoms.

Interpolation.

- of displacements! (smooth)
- Reflects lowest eigen-modes
• much more efficient than direct representation of all these modes
- Strong couplings are not constant
- Non linear relations retained
(to the extent needed)

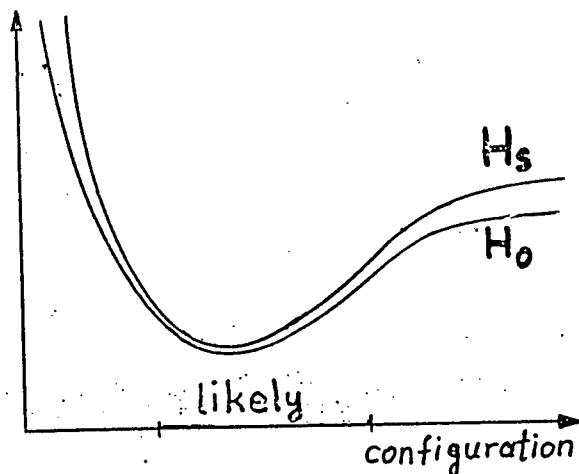
- Interpolation order
- Derivation of interpolation

electrostatics

Non linearity retained:

Smooth part of the forces
(and residuals of the local part)
participate (and change)
in the coarse-level dynamics

STOCHASTIC HAMILTONIAN SIMPLIFICATION



\mathcal{H}_0 = Original Hamiltonian
 obtained from fine grid \implies complicated

\mathcal{H}_s = Simplified Hamiltonian
 FAS: original form + polynomial terms

$$\mathcal{H}_s \geq \mathcal{H}_0, \mathcal{H}_s \approx \mathcal{H}_0$$

Detailed $P(H_0 \rightarrow H_s) = e^{\mathcal{H}_0 - \mathcal{H}_s} |_{\text{current}}$

Balance: $P(\text{freeze } H_s - H_0) = 1 - P(H_0 \rightarrow H_s)$

Simpler: Approximate detailed balance
 A posteriori acceptance check

GOAL:

Fast minimization of
energy functionals

MOTIVATION:

MD implicit schemes
with large timestep

METHODOLOGY:

Minimize potential energy
functionals in 1D, 2D, 3D.

Realistic couplings.

MULTI-SCALING
CONSIDERATIONS

RELAXATION:

Simple, point-by-point

COARSENING:

AMG theory: Strongest
coupling to coarse level

1:2 RATIO

INTERPOLATION:

Precise for infinite
coupling

RECURSIVENESS:

All levels

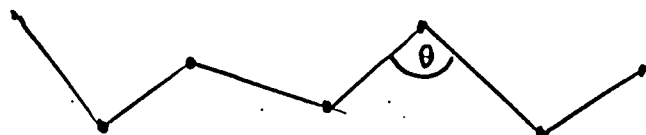
Investigated cases

(1) 1D chain



Nearest neighbor
coupling

(2) 2D chain



bond + angle
interaction

angle - 2nd neighbor

(3) 3D chains

bond + angle + torsion

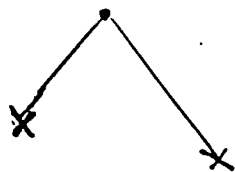
torsion - 3rd neighbor

PROPERTIES OF COARSE-LEVEL REPRESENTATION

Coarse atoms: more freedom

Bond constraints:

2D



x - coarse points

• - fine-only point

3D



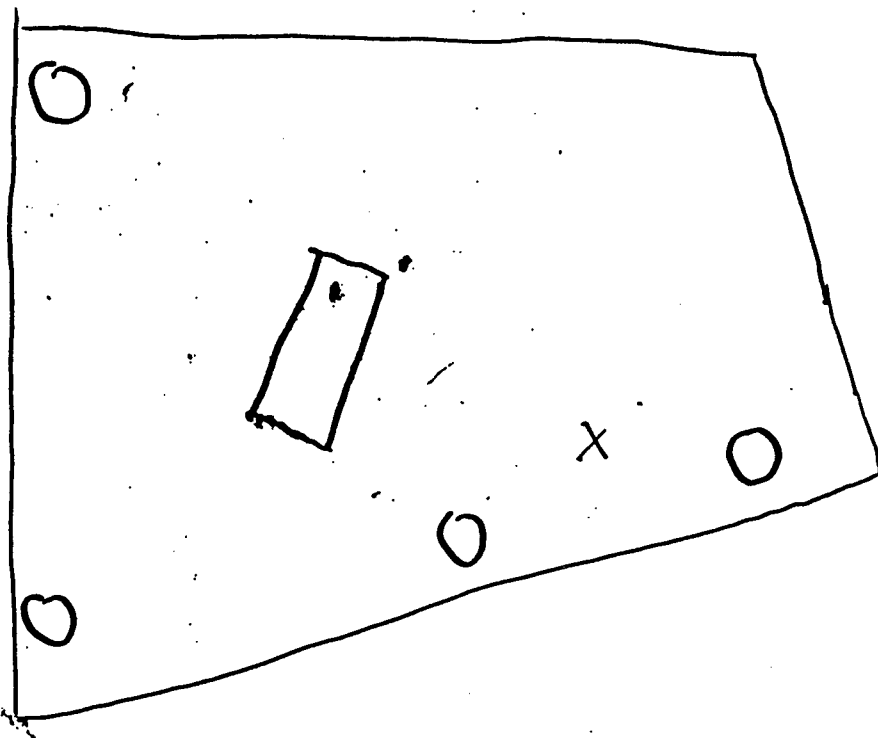
↑ - External field perpendicular to plane of atoms

NON-LINEARITIES


Typical reduction of strength of interaction
on a coarse level in 2D.

	0.01		0.1		0.5		1.0		10.0	
	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$
Fine	1.04	3.0	1.4	3.0	3.0	3.0	5.0	3.0	40.0	3.0
Coarse	0.05	0.09	0.5	0.9	2.5	4.5	5.0	9.0	49.0	90.0

K_x, K_y - Second energy derivatives
in x, y directions



 - center points

 - remote points

$$E_{\triangle} = \begin{pmatrix} x_f^T & x_c^T \end{pmatrix} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} x_f \\ x_c \end{pmatrix} = x_f^T A x_f + 2x_f^T B x_c + x_c^T C x_c$$

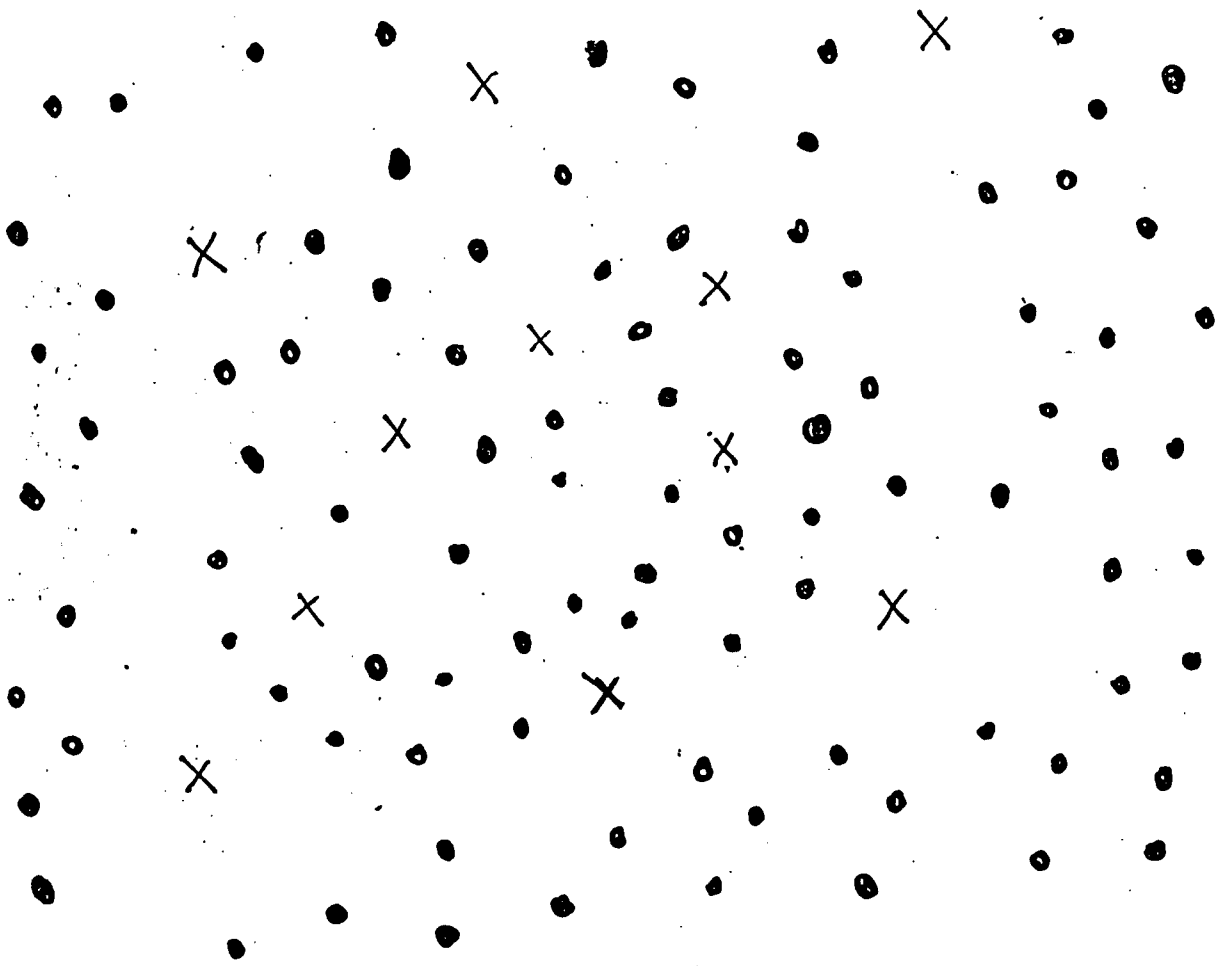
minimum energy

for a given x_c

$$x_f = -A^{-1} B x_c$$

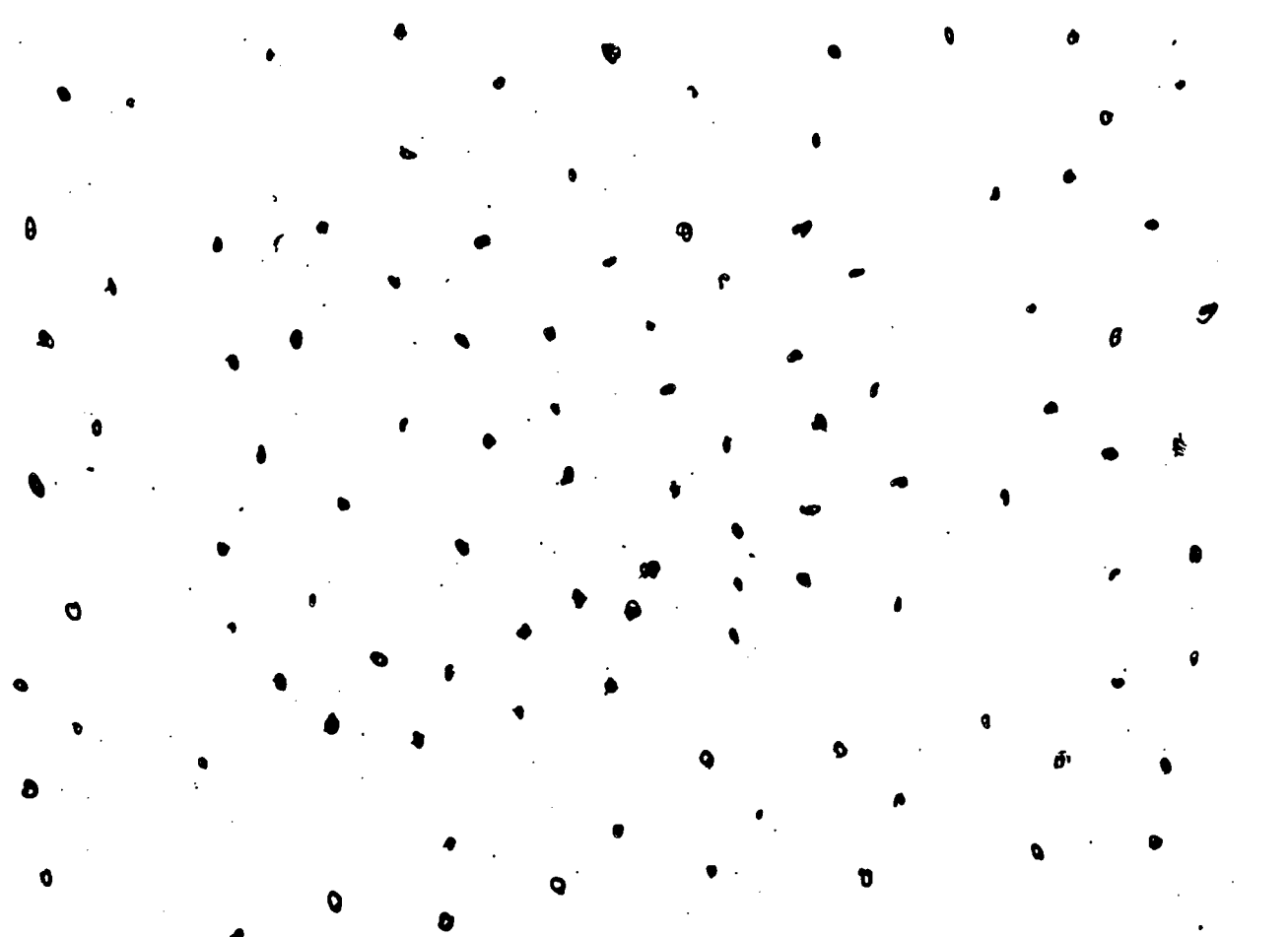
Apply only to





● fine-only points

x coarse points



SLOWNESS OF RELAXATION

Slowed down by:

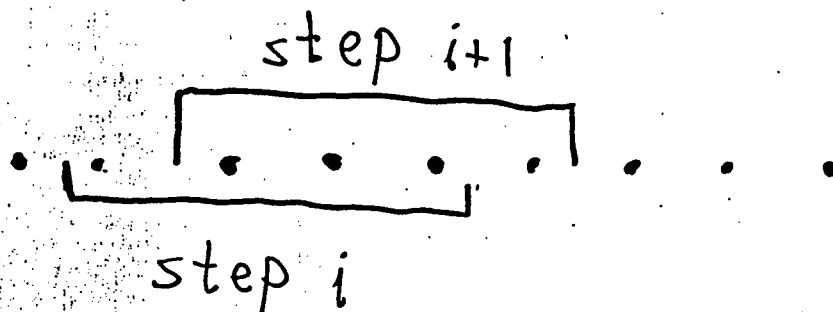
N - number of atoms

q - coupling ratio

$1 - \varepsilon$: convergence rate

$$\varepsilon < O(1/qN^2)$$

simultaneous changes:



Effect of BS on convergence rates

3D generic chain

Energy terms:

$$K_1(r-r_0)^2 - \text{bonding}$$

$$K_2(\theta-\theta_0)^2 - \text{angle interaction}$$

$$K_3(\varphi-\varphi_0)^2 - \text{torsion}$$

$N=26$

$K_1 = 1$

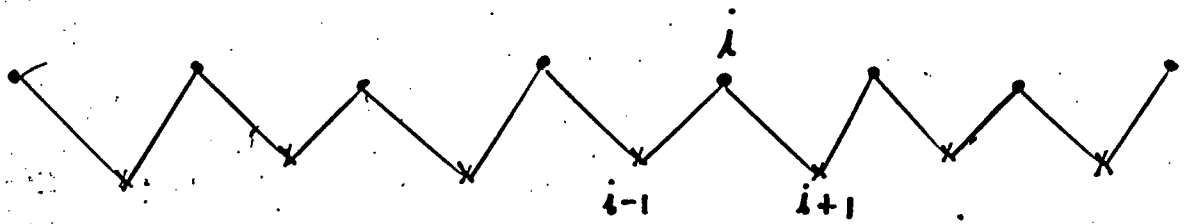
$K_2 = \varphi$

$K_3 = \varphi^2$

BS	φ			
	1.0	0.3	0.09	0.027
1	2.2 (-4)	5.4 (-5)	7.3 (-6)	2.3 (-6)
2	1.2 (-3)	8.3 (-4)	3.7 (-4)	1.4 (-4)
3	1.1 (-3)	2.4 (-3)	7.0 (-4)	3.0 (-4)
4	2.6 (-3)	2.6 (-3)	2.2 (-3)	1.4 (-3)
5	4.6 (-3)	4.6 (-3)	3.8 (-3)	3.3 (-3)
6	8.8 (-3)	9.4 (-3)	9.2 (-3)	8.9 (-3)

ε

INTERPOLATIONS



3D chain

- time only

- x coarse

(x_i, y_i, z_i) - coordinates of point i

$$\begin{pmatrix} \delta x_i \\ \delta y_i \\ \delta z_i \end{pmatrix} = A_{i-1,i} \begin{pmatrix} \delta x_{i-1} \\ \delta y_{i-1} \\ \delta z_{i-1} \end{pmatrix} + A_{i+1,i} \begin{pmatrix} \delta x_{i+1} \\ \delta y_{i+1} \\ \delta z_{i+1} \end{pmatrix} +$$

$$A_{i-3,i} \begin{pmatrix} \delta x_{i-3} \\ \delta y_{i-3} \\ \delta z_{i-3} \end{pmatrix} + A_{i+3,i} \begin{pmatrix} \delta x_{i+3} \\ \delta y_{i+3} \\ \delta z_{i+3} \end{pmatrix} + \dots$$

A - 3×3 coefficient matrix

LOCAL SETS

For deriving interpolation
Coefficient

Minimize energy locally:

local relaxations

direct solution

Local set is a truncated chain

Ignore out-of-set links

Facilitate accurate smooth motions

Size of local set

Larger for strong angle (2D) and
torsion (3D)

Small remotest interpolation matrix

Effect of local set size in 2D

$$k_1 = 1$$

k_2	Local set size	
	$5=3+2$	$9=5+4$
0.01	0.05	0.03
0.1	0.31	0.09
1.0	0.48	0.12
2.0	0.48	0.1
5.0	0.52	0.11
10.0	0.56	0.20

Asymptotic Convergence

Factor of $V(1,1)$ cycles

Energy terms:

$k_1(r-r_0)^2$ - bond interaction

$k_2 r^2 (\theta - \theta_0)^2$ - angle interaction