Parallel Segregated Schur Complement Methods for Fluid Density Functional Theories

Michael A. Heroux
Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

Outline

- Background on DFTs.
- Observations on generated linear systems.
- Basic solver strategy.
- Specific algorithms:
  - Hard-sphere models.
  - Polymer models.
- Properties of solvers.
- Results.
- Hint of parallel implementation.
- Conclusions.

Sandia National Laboratories

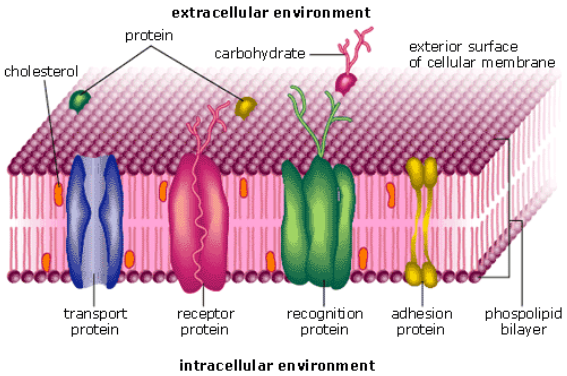
Collaborators

- Laurie Frink:
  - Primary developer of Tramonto.
  - Expertise: computational modeling of inhomogeneous fluids.
- Andy Salinger:
  - Other primary Tramonto developer.
  - Expertise: discretization methods and parallel application design.

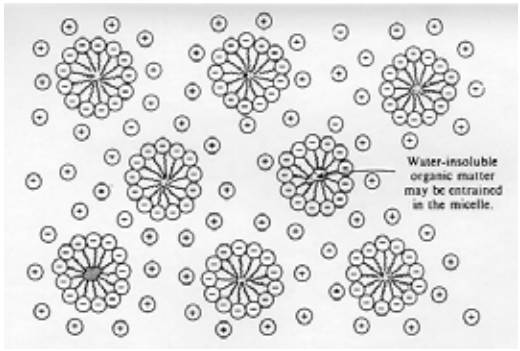
- My Role:
  - Solver algorithms.
  - Parallel implementation.

Sandia National Laboratories

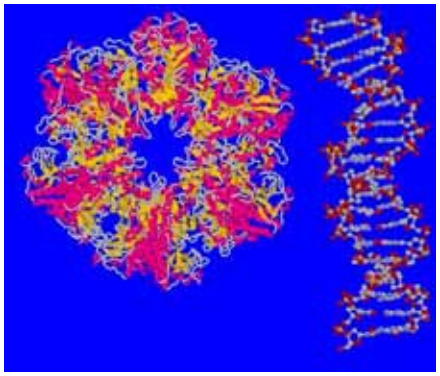
Complex fluid systems…



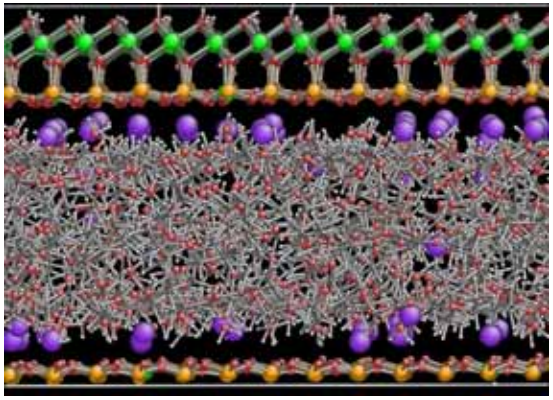
Cell membranes



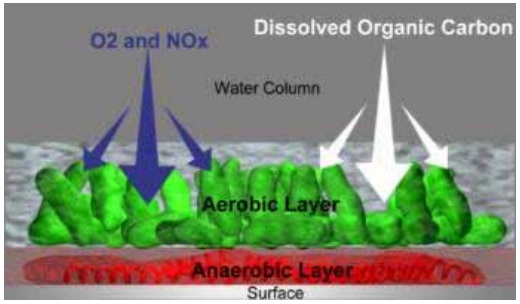
Colloidal/Amphiphilic systems
(www.science.duq.edu)



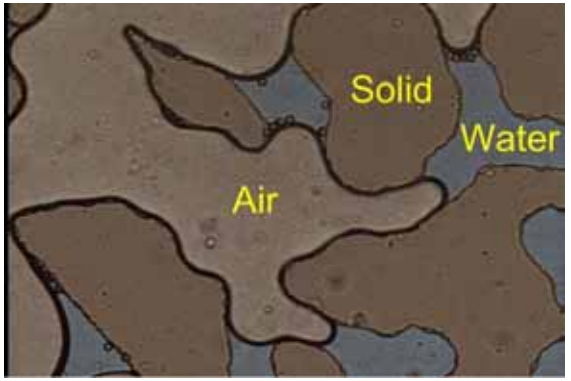
Biological Macromolecules
(www.hmi.de/people/kroy/rota.html)



Clay-polymer nanocomposites
(Univ. College London exclaim.org.uk)



Biofilms
(www.zetacorp.com)



Porous Media
(www2.bren.ucsb.edu/~keller/micromodels.html)

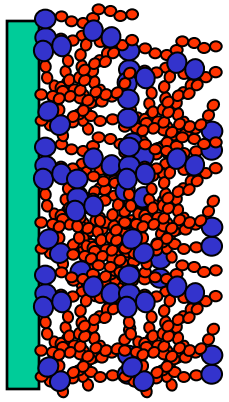
Problem characteristics
Interfacial fluids
Multiple length scales
Phase complexity

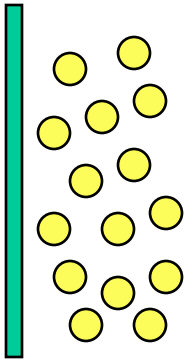
Sandia National Laboratories

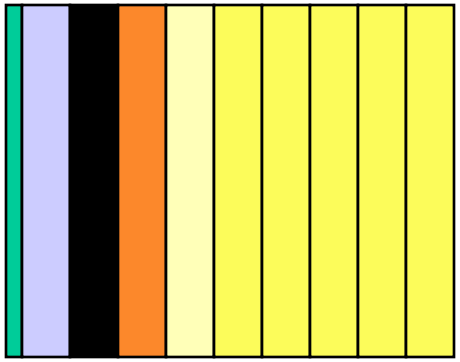
DFT Acronym

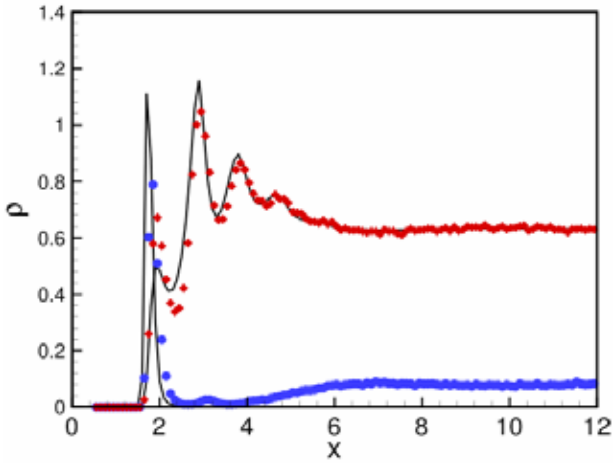
- Quantum mechanical DFTs (QM-DFTs):
  - aka electronic DFTs.
  - Related but not discussed today.
- Fluid DFTs (F-DFTs):
  - aka classical DFTs.
  - We focus on this.
- Discrete Fourier Transforms (DFTs for F-DFTs):
  - Possible to use Fourier Transforms to work in frequency space.
  - FastTram: A version of Fourier Transform version of Tramonto (Mark Sears).
  - Restricted applicability: BCs, preconditioning.
- Real-space approach: Use spatial variables.
  - We focus on this.
  - From this point on: DFT means real-space F-DFTs.

DFT for fluid structure









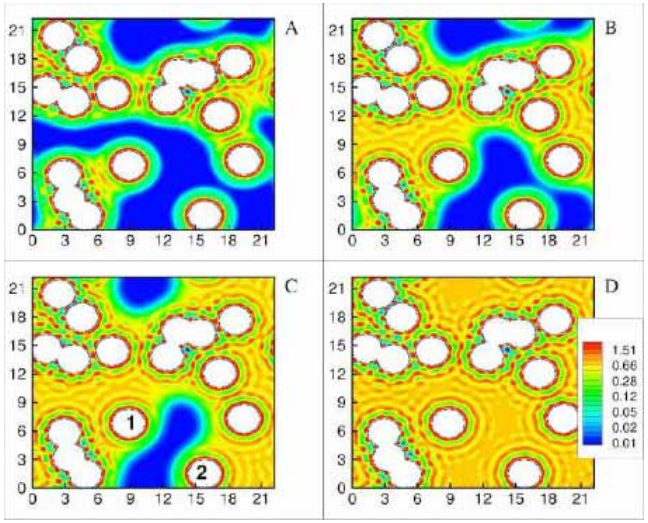
Sandia National Laboratories

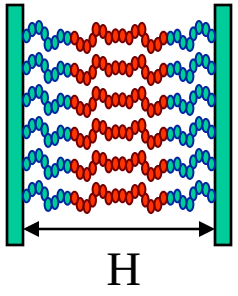
Applications at different scales

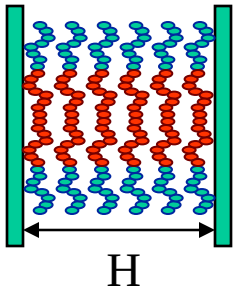
Electronic structure…$(1nm)^3$
Fluid Structure…$(10\ nm)^3$
Colloids, polymers, proteins…
$(10\text{-}1000nm\ )^3$

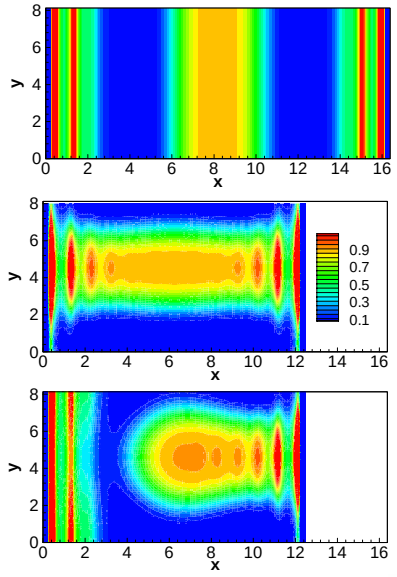
Some apps…

- Self Assembly
- Corrosion
- Surface forces
- Adsorption in porous media
- Ion channel proteins
- Lipid bilayer membranes
- Protein solvation
- Protein-protein interactions
- Protein structure









Sandia National Laboratories

DFT for fluids

The free energy functional … (a) The theory is exact, but the precise nature of the equations often cannot be derived. (b) Approximate functionals have been developed often as perturbations to a hard sphere reference system.

Coulomb interactions

Associations (H-bonding)

[Applied field]

$$\Omega[\rho(r)] = F_{id} + F_{hs} + F_{vdW} + F_c + F_{assoc} + \int \rho(r)[V(r) - \mu]$$

Ideal gas

Hard sphere

Dispersion attractions

Legendre Transform from Canonical to Grand canonical ensemble

We seek the the stationary states of the free energy functional with the understanding that the thermodynamically relevant state should be found at the global free energy minimum.

$$\frac{\delta\Omega}{\delta\rho(r)}_{\mu,T} = 0$$

Sandia National Laboratories

Properties of *F-DFT systems*

DFT - Integral equations of finite range
(matrix density is system size dependent)

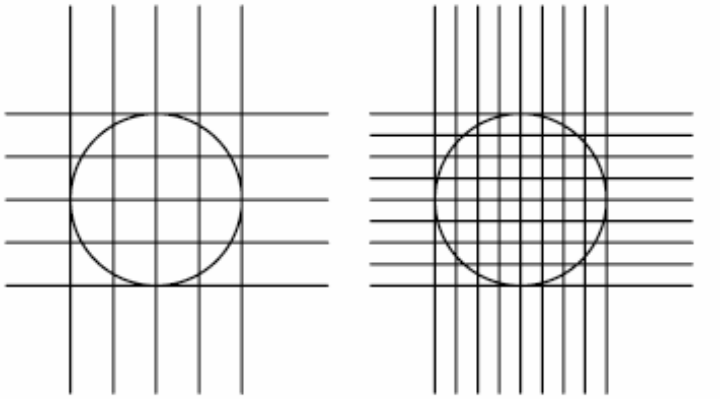
PDE - matrix density independent of system size.

DFT- Inter-physics coupling dominates

PDE - Inter-nodal coupling dominates

DFT - Stencils based on physical constants

PDE - Stencils based on nearest neighbors



DFT - May have large numbers of DOF per node
- HS (3D) 10+
- Polymer (20 beads) 42+
- Most DOFs are “constraints” on densities.

PDE - Usually a few DOFs per node

Sandia National Laboratories

General Segregation Strategy

- Observations:
  - Internodal coupling is weak.
  - Some DOFs have simple “one-way” dependence.
- Idea:
  - Organize DOFs physics-first.
  - Reorder blocks of DOFs to expose one-way dependences.
  - Apply 2-by-2 block partitioning such that one-way dependencies are in A_{11} block.
  - Use Schur complement on A_{22} block.

General Strategy

$$\left(\begin{array}{ccc|ccc} A_{11}^{11} & \cdots & A_{11}^{1j} & A_{12}^{1,j+1} & \cdots & A_{12}^{1k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{11}^{j1} & \cdots & A_{11}^{jj} & A_{11}^{j1} & \cdots & A_{11}^{jj} \\ \hline A_{21}^{j+1,1} & \cdots & A_{21}^{j+1,j} & A_{22}^{j+1,j+1} & \cdots & A_{22}^{j+1,k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{21}^{k1} & \cdots & A_{21}^{kj} & A_{22}^{k,j+1} & \cdots & A_{22}^{kk} \end{array}\right) \left(\begin{array}{c} x_1^1 \\ \vdots \\ x_1^j \\ \hline x_2^{j+1} \\ \vdots \\ x_2^k \end{array}\right) = \left(\begin{array}{c} b_1^1 \\ \vdots \\ b_1^j \\ \hline b_2^{j+1} \\ \vdots \\ b_2^k \end{array}\right)$$

- Identify and order DOFs in A_{11} block so that A_{11}^{-1} easy to apply.
- Implicitly (or explicitly in some instances) form Schur complement system:

$$Sx_2 = (A_{22} - A_{21}A_{11}^{-1}A_{12})x_2 = b_2 - A_{21}A_{11}^{-1}b_1$$

- Solve Schur system via preconditioned GMRES.
- Solve finally for x_1 .

Sandia National Laboratories

Hard Sphere Problems

Sandia National Laboratories

Hard spheres: Important observation

Frink and Salinger, J. Comp. Phys., 2000

There are two ways to form the matrix problem…

(1) implicit representation of auxillary variables, $n^{(\gamma)}$ leads to a "second order" matrix problem.

$$J(r,r') = \frac{\delta(r,r')}{\rho(r)} + \int \sum_{\varepsilon} \sum_{\gamma} \frac{\partial^2 \Phi}{\partial n^{(\varepsilon)} \partial n^{(\gamma)}} (r'') \omega^{(\gamma)}(r - r'') \omega^{(\varepsilon)}(r' - r'') dr''$$

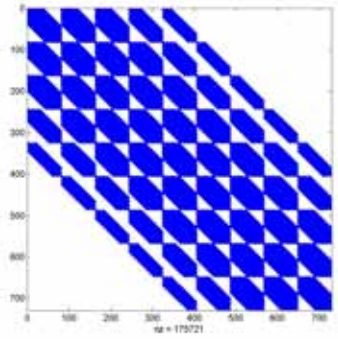
$$[A_{11}][\Delta\rho] = [b]$$

(2) explicit representation of auxillary variables, $n^{(\gamma)}$ leads to a "first order" matrix problem.

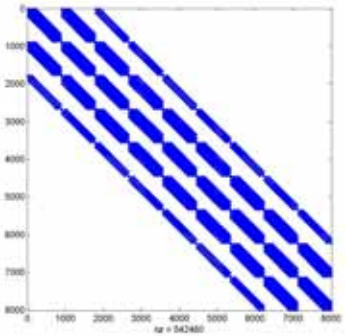
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} \Delta n^{(\gamma)}(r) \\ \Delta\rho(r) \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Sandia National Laboratories

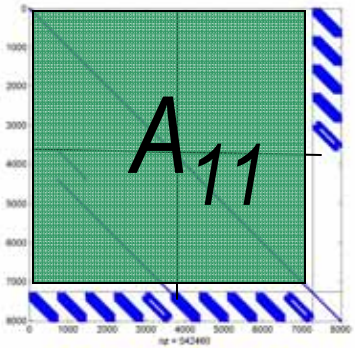
Hard Sphere Formulations



Second Order formulation → Expensive O(N^2) fill



First order Formulation With nodal ordering → Expensive solve
- matrix size
- matrix conditioning

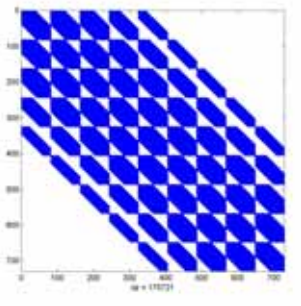


First order Formulation With physics Based blocking → Structure is natural for a Schur complement based approach. (e.g. easily inverted blocks) Fast fill / Fast solve

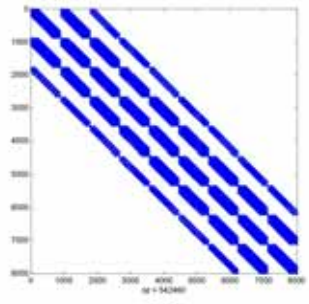
$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

Sandia National Laboratories

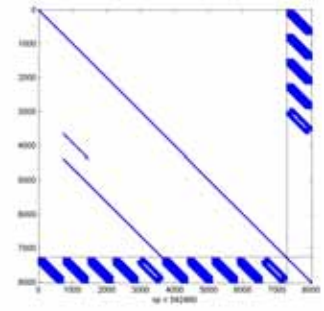
Results - comparison of formulations



EXP(ρ): GMRES: A



IMP1(ρ,n): GMRES: A



IMP2(ρ,n): Schur/GMRES A_{22}

Stats \Form Phase Cost \	EXP	IMP1	IMP2	IMP2 has cost of
System Dimension	4,913	54,043	4,913	EXP
Matrix nonzero cnt	5M	15M	15M	IMP1
Condition Number	6E2	1E4	6E2	EXP
Matrix Preprocessing	105.4	10.8	10.8	IMP1
Matrix Fill	118.0	4.0	4.0	IMP1
Solve Time	12.2	145.9	38.4	Both
Linear Solve Iterations	46	124	45	EXP
Total Time (4 Newton Steps)	628.3	616.9	186.9	Both

**note no preconditioner was used for these studies.*

Sandia National Laboratories

Details of Hard Sphere Schur Complement

$$A_{11} = \begin{bmatrix} I & 0 \\ X & I \end{bmatrix}$$

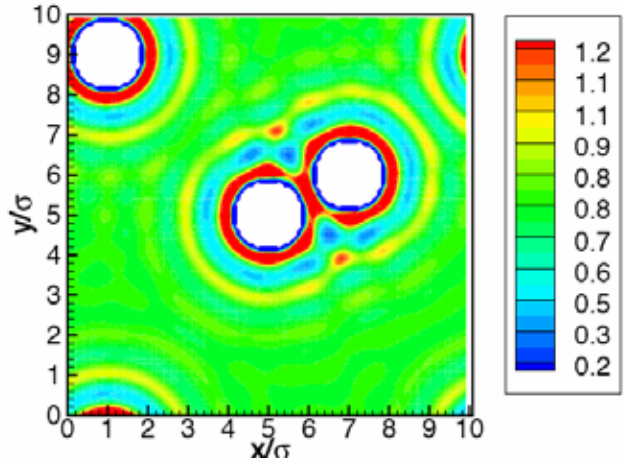
$$inv(A_{11}) = \begin{bmatrix} I & 0 \\ -X & I \end{bmatrix}$$

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

- *S can be applied with just matvec's.*
- Precondition *S* with A_{22}. (Diagonal scaling)
- Provide option to explicitly compute *S*.

Sandia National Laboratories

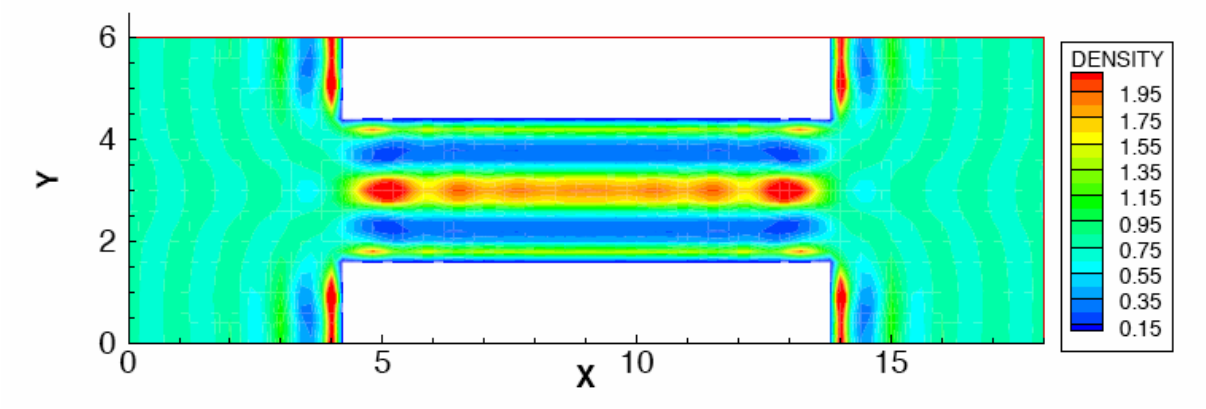
Results - parallel scaling (2D)



#Procs	<# Lin iters>		Time/N_{iter}		T_{1Proc}/T	T_{old}/T_{new}
	Old	New	Old	New	(New)	
1	8	22	495.1	14.7	1	33.8
2	10	22	148.7	8.5	1.7	17.5
4	11	22	47.4	4.2	3.5	11.3
8	12	22	16.1	2.1	6.9	7.6
16	13	22	6.3	1.1	13.3	5.7
32	15	22	2.1	0.6	24.8	3.6
64	18	22	1.1	0.3	43.1	3.2

Sandia National Laboratories

Results - parallel scaling (3D)



Nanopore in a membrane: (One slice in a 3D domain)

#Procs	N_{iter}	<# Lin iters>		Time/N_{iter}		T_{4Proc}/T (New)	T_{old}/T_{new}
		Old	New	Old	New		
4	11	-	-	-	117.4	1	-
8	11	-	76	-	61.0	1.9	-
16	11	53*	76	544*(6)	29.8	3.9	18.*
32	11	73	76	154.5	16.7	7.0	9.3
64	11	80	76	55.4	8.3	14.1	6.7
128	11	89	76	19.9	4.7	24.8	4.2

Sandia National Laboratories

3D Hard Sphere Results

- **New solver (NS): 8-4X fewer processors compared to old solver (OS).**
- **NS: Fixed linear solve cost.**
  OS: Iteration creap.
- **NS: Roughly linear speedup.**
  OS: (Very) superlinear speedup.
- **NS: Storage cost of preconditioner negligible.**
  OS: Prec storage 3-7 times nnz(global matrix).
- **NS: Far more efficient for mesh refinement.**
  OS: Memory costs constrain problem size.

						T_{16Proc}/T		T_{OLD}/T
					New	Old	New	Ratio
					291	-	0.25	-
					70.9	-	0.48	-
					28.3	1*	1	17.3*
					83.2	3.3	1.8	9.3
64	11	80	76	608.9	91.4	9.3	3.6	6.7
128	11	89	76	219.2	51.9	25.9	6.3	4.2

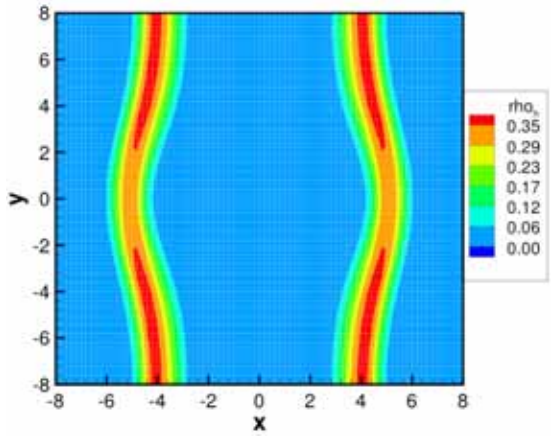
Δx	Niter	<# Lin iters>		Solve Time/Niter		$T/T_{\Delta x=0.2}$		T_{OLD}/T
		Old	New	Old	New	Old	New	Ratio
$\sigma/5$	7	45	44	17.16	4.14	1	1	4.14
$\sigma/7$	8	51	49	150.51	18.87	8.8	4.55	8.0
$\sigma/10$	9	-	51	-	121.11	-	29.25	-

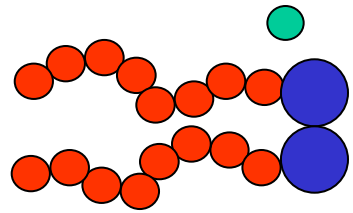
TABLE 6.2

Results for a 3D Hard sphere test problem. The second column now contains the number of nonlinear iterations needed to solve the problem. For a description of all other columns see the table 1 caption. All data in the upper part of the table were generated with a mesh spacing of $\Delta x = \sigma/5$*, and with a bulk density of* $\rho\sigma^3 = 0.75$*. All data in the lower part of the table were generated on 128 processors with a bulk density of* $\rho\sigma^3 = 0.6$*.*

Sandia National Laboratories

Second class of Problems: Self-assembly of lipid bilayers





8-2-8 Chain

Sandia National Laboratories

A 2nd Case…CMS-DFT / polymers

- developed for polymers
- chains are flexible
- 2nd order density expansion

Chandler, McCoy, Singer (1986);
McCoy et al. (1990s)

$$\rho_\alpha(r) = \frac{\rho_\alpha^b}{N_\alpha} \sum_{s=1}^{N_\alpha} \frac{G_s(r) G_s^i(r)}{e^{-\beta U_\alpha(r)}}$$

Chain density distribution

$$U_\alpha(r) = V_{ext}(r) - \sum_\gamma \int c_{\alpha\gamma}(r - r')[\rho_\gamma(r') - \rho_\gamma^b]dr'$$

Mean field

$$c(r) = c_{rep}(r) - u_{att}(r)$$

PRISM Theory **RPM Approx**

$$G_s(r) = e^{-\beta U_{\alpha,s}} \int w(r - r') G_{s-1}(r') dr'$$

$$G_s^i(r) = e^{-\beta U_{\alpha,s}} \int w(r - r') G_{s+1}^i(r') dr'$$

$$G_1 = G_N^i = e^{-\beta U(r)}$$

$$w(r) = \frac{1}{4\pi\sigma^2} \delta(|r| - \sigma)$$

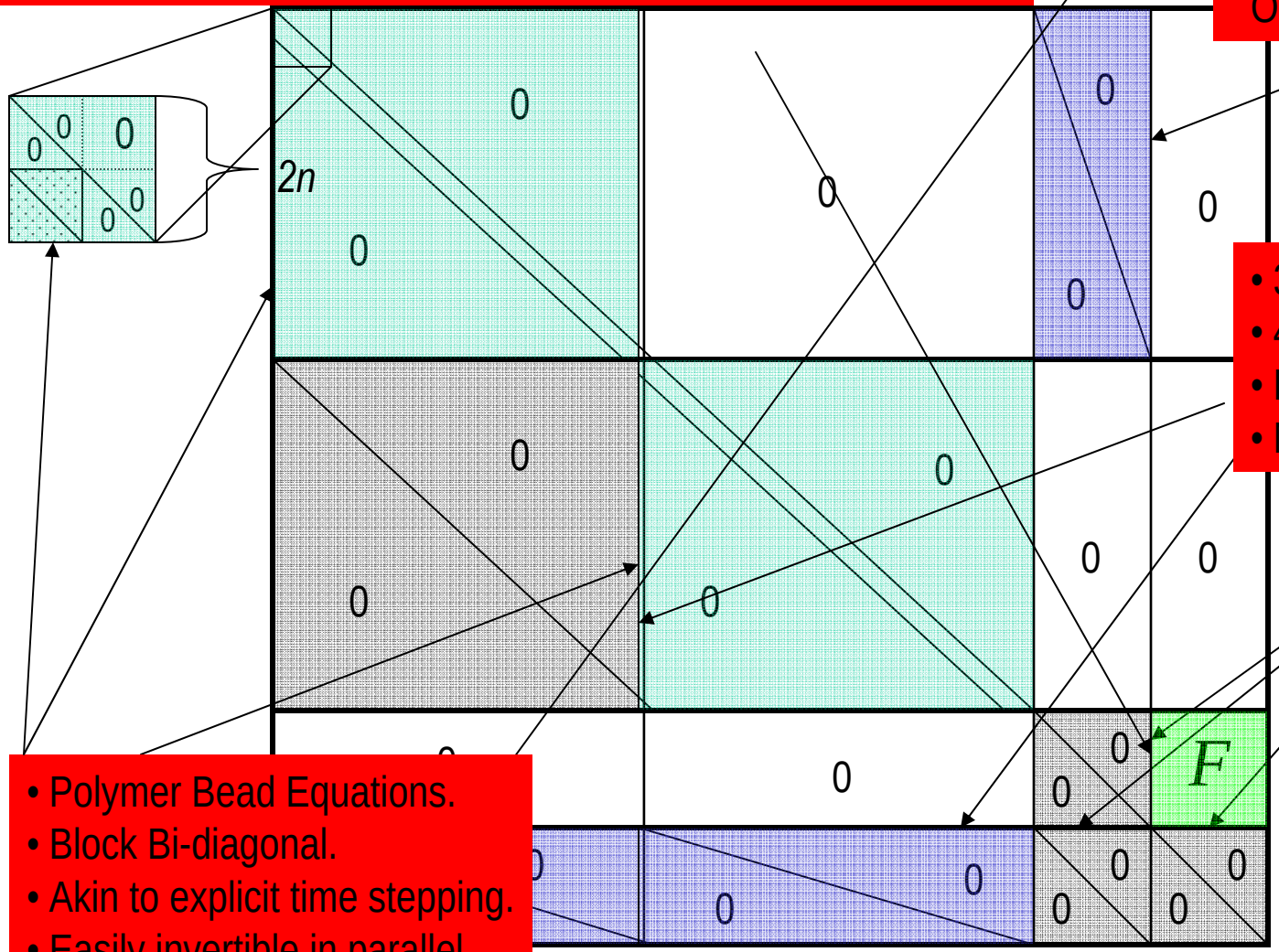
Chain Architecture (freely-jointed chains)

Sandia National Laboratories

Proble

- There is only ONE interesting block in this whole matrix.
- *F* describes CMS field dependence on primitive densities.
- 2.5 σradius integral at each grid node (mesh independent).
- Not sparse, nor dense. Constant coefficient.

- Diagonal-like.
- One non-zero per row/col in long dimension.
- Like Prolongation/restriction Operators?



- 3rd block: CMS Field
- 4th block: Prim Densities
- Diagonal matrices.
- No spatial coupling.

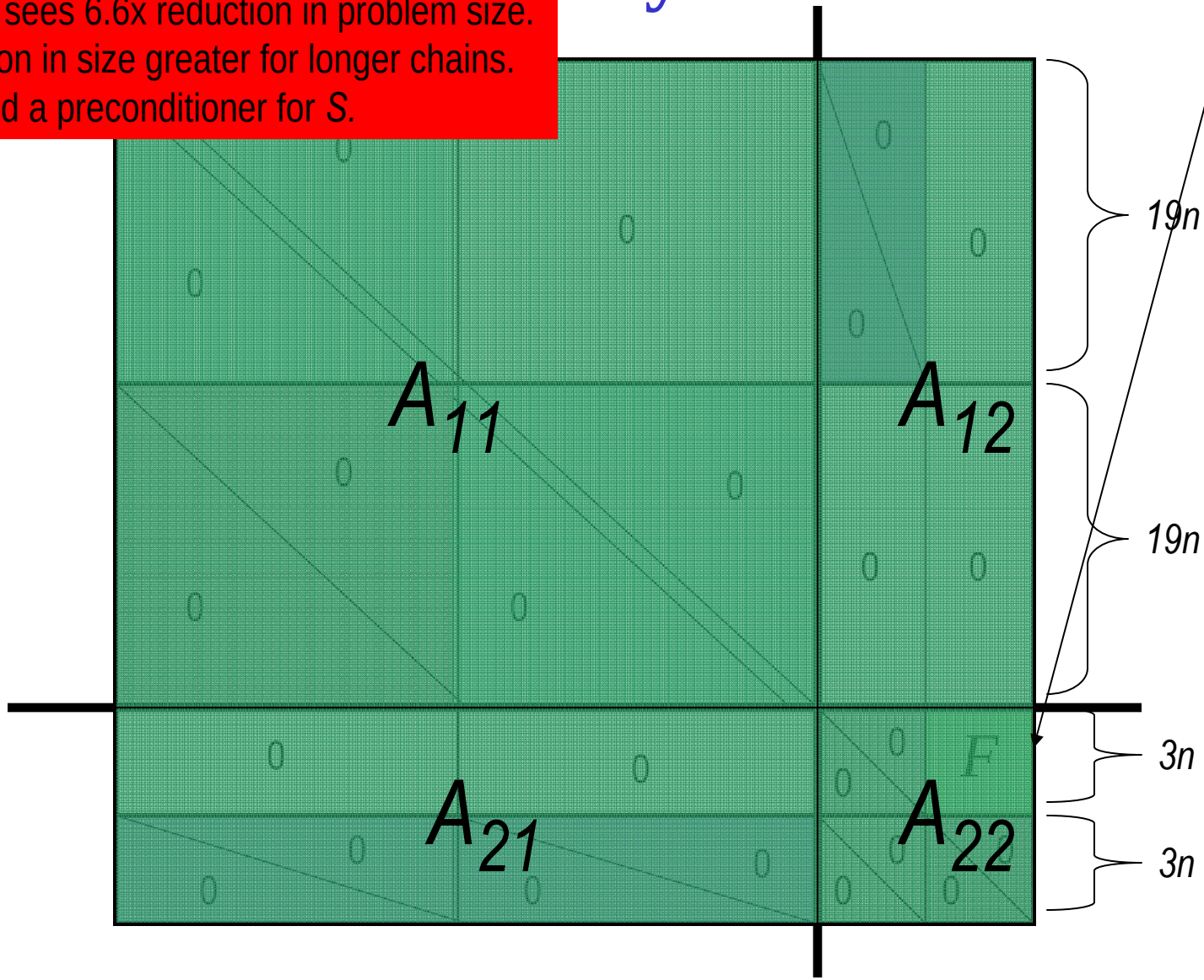
- Polymer Bead Equations.
- Block Bi-diagonal.
- Akin to explicit time stepping.
- Easily invertible in parallel.

Sandia National Laboratories

Layer Problem

- Last layer of structure: 2-by-2 partitioning.
- A_{11} solve easily applied in parallel.
- Apply GMRES to $S = A_{22} - A_{21}*inv(A_{11})*A_{12}$
- GMRES sees 6.6x reduction in problem size.
- Reduction in size greater for longer chains.
- Still need a preconditioner for S.

- F has strong overlap:
  Distribute separate from rest of problem.



Sandia National Laboratories

Preconditioner for *S*

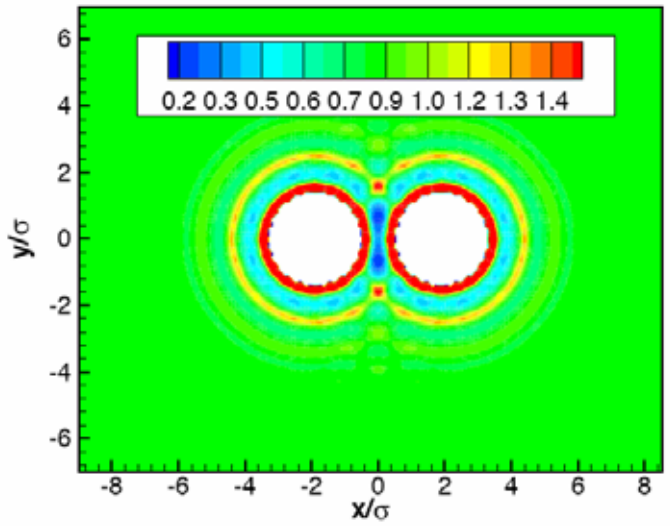
$$A_{22} = \begin{bmatrix} D_{11} & F \\ D_{21} & D_{21} \end{bmatrix}$$

$$A_{22} \approx \tilde{A}_{22} = \begin{bmatrix} D_{11} & F \\ 0 & D_{21} \end{bmatrix}$$

- $D_{11}, D_{22} = O(1), \; D_{21} = O(1e\text{-}10)$
- Ignore D_{21} for preconditioning.
- *P(S)* requires
  - 2 diagonal scalings,
  - matvec with F.
- All distributed operations.

Sandia National Laboratories

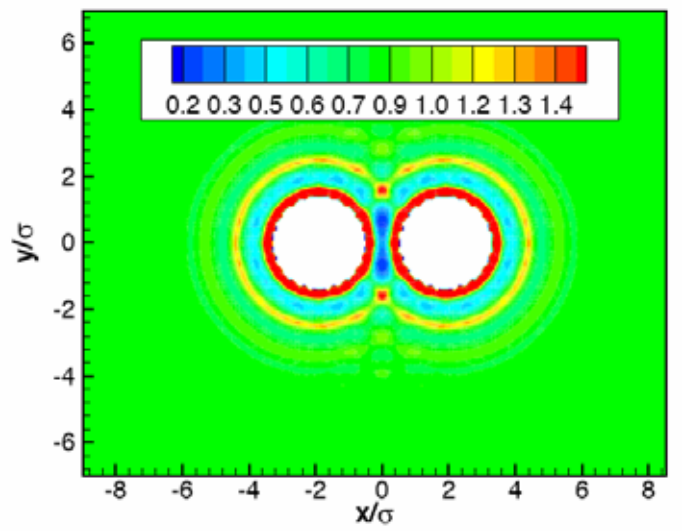
Parallel Scaling - CMS-DFT

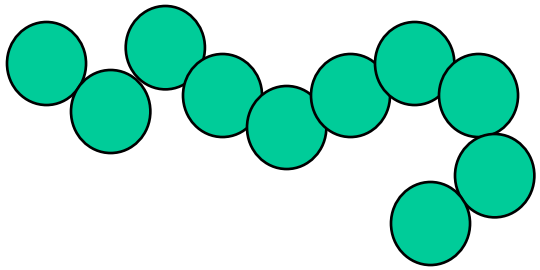


#Procs	N_{iter}	<# Lin iters>		Time/N_{iter}		T_{1Proc}/T	T_{FullA}/T_{SA22}
		FullA	*SA22*	*FullA*	*SA22*	(*SA22*)	
1	9		70	-	88	1	-
2	9		70	-	45	2.0	-
4	9	25	70	290*(8)	26.2	3.4	11.0
8	9	28	70	77	12	7.3	6.4
16	10	33	70	23	6.7	13.1	3.4
32	8	40	70	7.1	3.8	23.2	1.9
64	9	46	69	2.8	2.2	40.0	1.3
128	9	58	69	1.4	1.8	49.0	0.8

Sandia National Laboratories

Scaling with chain length

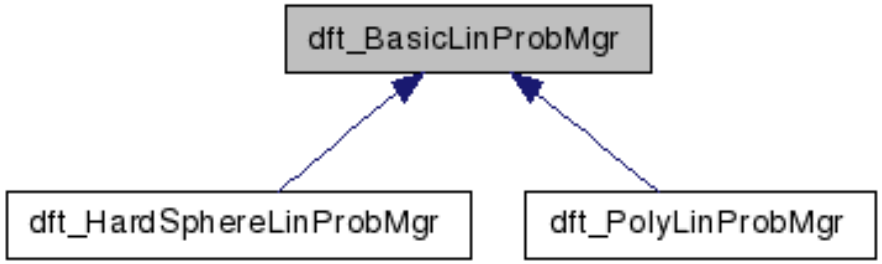




N_{seg}	N_{iter}	<# Lin iters>		Time/N_{iter}		$T/T_{N_{seg}=10}$	T_{FullA}/T_{SA22}
		FullA	*SA22*	*FullA*	*SA22*	(*SA22*)	
10	8	40	70	7.1	3.7	1	1.9
20	8	48	69	21.7	8.9	2.4	2.4
40	8	62	70	58.7	15.1	4.1	3.9
80	7	93	68	257.4	30.9	8.4	8.3

Sandia National Laboratories

Tramonto Solver API



int **setNodalRowMap** (int numOwnedNodes, int *GIDs)

Define global to local index row mappings.

int **setNodalColMap** (int numBoxNodes, int *GIDs)

Define global to local index column mappings, the rectangular box containing all ghost nodes and owned nodes.

virtual int **setCoarsenedNodesList** (int numCoarsenedNodes, int *GIDs)

Define the nodes on this processor that will be mesh-coarsened, must be nodes set as part of setNodalRowMap().

virtual int **finalizeBlockStructure** ()

Method that must be called once, when all row and column maps are set.

virtual int **initializeProblemValues** ()

Method that must be called each time prior to starting matrix, lhs

virtual int **insertRhsValue** (int ownedPhysicsID, int ownedNode,

Insert rhs value based on ownedNode and ownedPhysicsID.

virtual int **insertMatrixValue** (int ownedPhysicsID, int ownedNo

Insert single matrix coefficient into system.

virtual int **insertMatrixValues** (int ownedPhysicsID, int ownedN
numEntries)

Insert matrix coefficients for a given row, where columns are all from the same physics type at different nodes.

virtual int **insertMatrixValues** (int ownedPhysicsID, int ownedNode, int *boxPhysicsIDList, int boxNode, double *values, int numEntries)

Insert matrix coefficients for a given row, where columns are from different physics types at the same node.

virtual int **finalizeProblemValues** ()

Method that must be called each time matrix value insertion is complete (usually once per nonlinear iteration).

virtual int **setupSolver** ()

Setup up the solver for solving the current linear problem.

virtual int **solve** ()

Solve the current linear problem.

- API uses terms/structures of Tramonto.
- Independent of any particular solver.
- Specializations of BasicLinProbMgr selectively reimplement methods.
- New methods added as solvers evolve.

Sandia National Laboratories

Coulomb Effects

$$\left(\begin{array}{ccc|ccc} A_{11}^{11} & \cdots & A_{11}^{1j} & A_{12}^{1,j+1} & \cdots & A_{12}^{1k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{11}^{j1} & \cdots & A_{11}^{jj} & A_{11}^{j1} & \cdots & A_{11}^{jj} \\ \hline A_{21}^{j+1,1} & \cdots & A_{21}^{j+1,j} & A_{22}^{j+1,j+1} & \cdots & A_{22}^{j+1,k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{21}^{k1} & \cdots & A_{21}^{kj} & A_{22}^{k,j+1} & \cdots & A_{22}^{kk} \end{array}\right) \left(\begin{array}{c} x_1^1 \\ \vdots \\ x_1^j \\ \hline x_2^{j+1} \\ \vdots \\ x_2^k \end{array}\right) = \left(\begin{array}{c} b_1^1 \\ \vdots \\ b_1^j \\ \hline b_2^{j+1} \\ \vdots \\ b_2^k \end{array}\right)$$

- Coulomb Effects: Poisson operator (only true spatial coupling operator).
- No interaction with G or G^i: Can place in A_{11}.
- Answer:
  - If direct solver feasible (1-2D on few processors): Put in A_{11}.
  - If ML (3D, many processors): Put in A_{22}.
- General experience with API:
  - Abstractions support unintended situations: e.g., coarsening.
  - Provides flexibility going forward.

Sandia National Laboratories

Polymer A_{11} Block: Some implementation details

$$A_{11} = \begin{bmatrix} A_{11}^{11} & 0 & \cdots & 0 \\ A_{11}^{21} B^2 & A_{11}^{22} & & \vdots \\ A_{11}^{31} B^3 & A_{11}^{32} & \ddots & 0 \\ A_{11}^{41} & A_{11}^{42} B^4 & A_{11}^{43} & A_{11}^{44} \end{bmatrix}$$

$$A_{11}^{ii} = diag()$$

- To Apply A_{11} inverse:
  - Form rectangular matrices $\boldsymbol{B^i}$ in lower triangle.
  - Apply matvecs using $\boldsymbol{B^i}$, followed by diagonal scaling using $\boldsymbol{A_{11}^{ii}}$.

Sandia National Laboratories

Glimpse of Data Mapping using Epetra

- A_{11}
- 2-node mesh
- 4 DOFs/node

$$\begin{bmatrix}
a_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & a_{33} & 0 & 0 & 0 & 0 & 0 \\
a_{41} & a_{42} & 0 & a_{44} & 0 & 0 & 0 & 0 \\
a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & 0 & 0 & 0 \\
a_{61} & a_{62} & a_{63} & a_{64} & 0 & a_{66} & 0 & 0 \\
a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & 0 \\
a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & 0 & a_{88}
\end{bmatrix}$$

B^2 B^3 B^4

- 2 PEs, full matrix maps:
  - On PE 0: DomainMap = RangeMap = {0, 2, 4, 6}
  - On PE 1: DomainMap = RangeMap = {1, 3, 5, 7}
- When forming submatrices:
  - Store each block B^i as individual Epetra_CrsMatrix.
  - For each submatrix:
    - Use the global ID space of full matrix.
    - Use domain/range maps of full matrix.
  - Easy to form many individual submatrices since global index space the same.

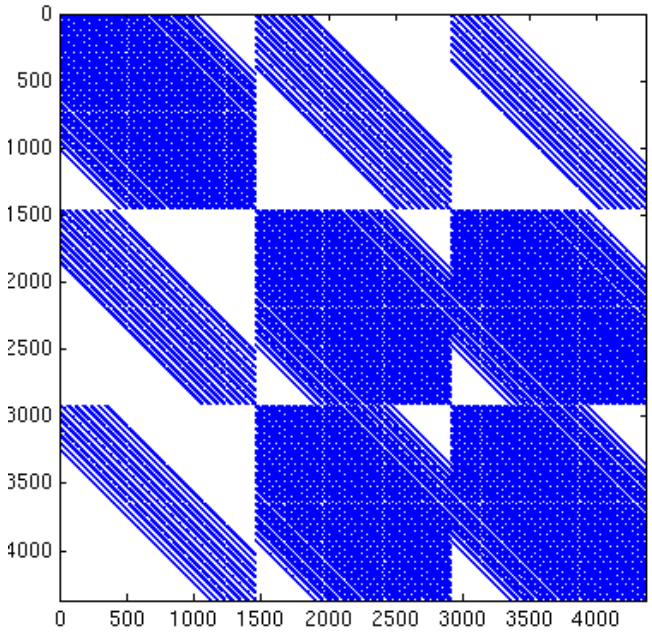
Sandia National Laboratories

A_{11} Stats for Polymer A_{11} block

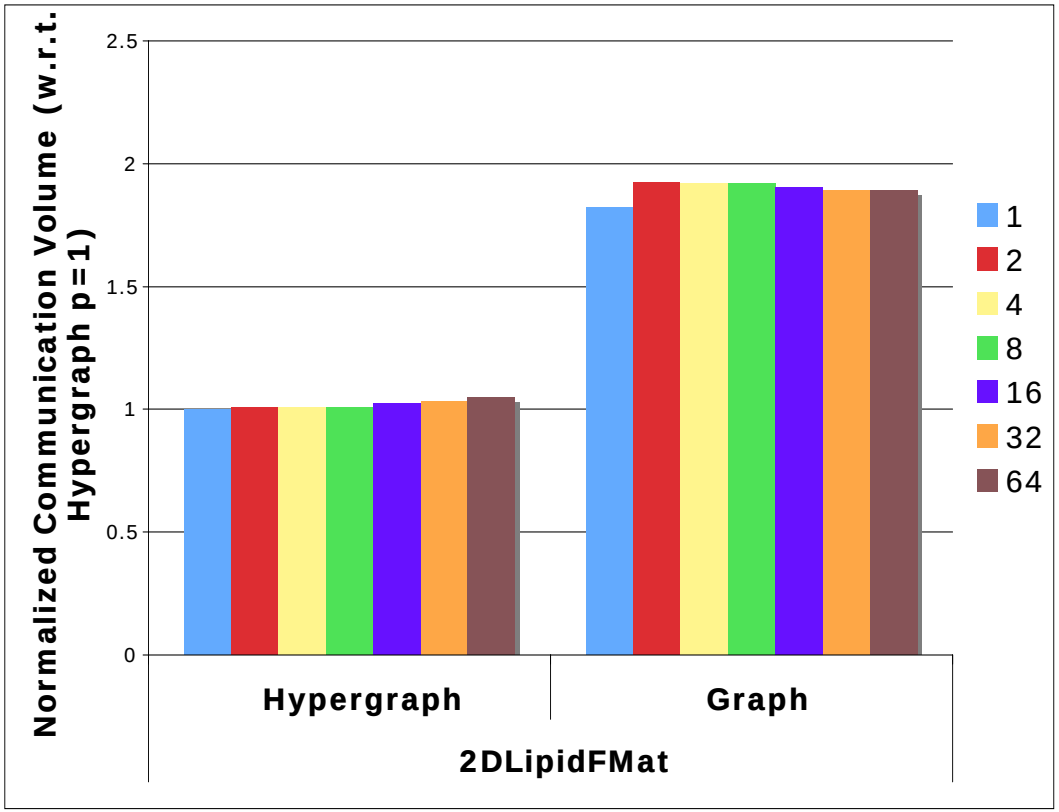
- 38 DOFs in A_{11} : 37 Epetra_CrsMatrix objects.
- ApplyInverse call requires 38 parallel vector updates interleaved with 37 matvecs.
- This is for 18-length polymer chain.
- 100-length chain: 199 Epetra_CrsMatrices.

Partitioning of *F* Block: One Case study

- *Hypergraph partitioning produces partitions with lower communication volume than graph partitioning.*



Tramonto 2DLipidFMat

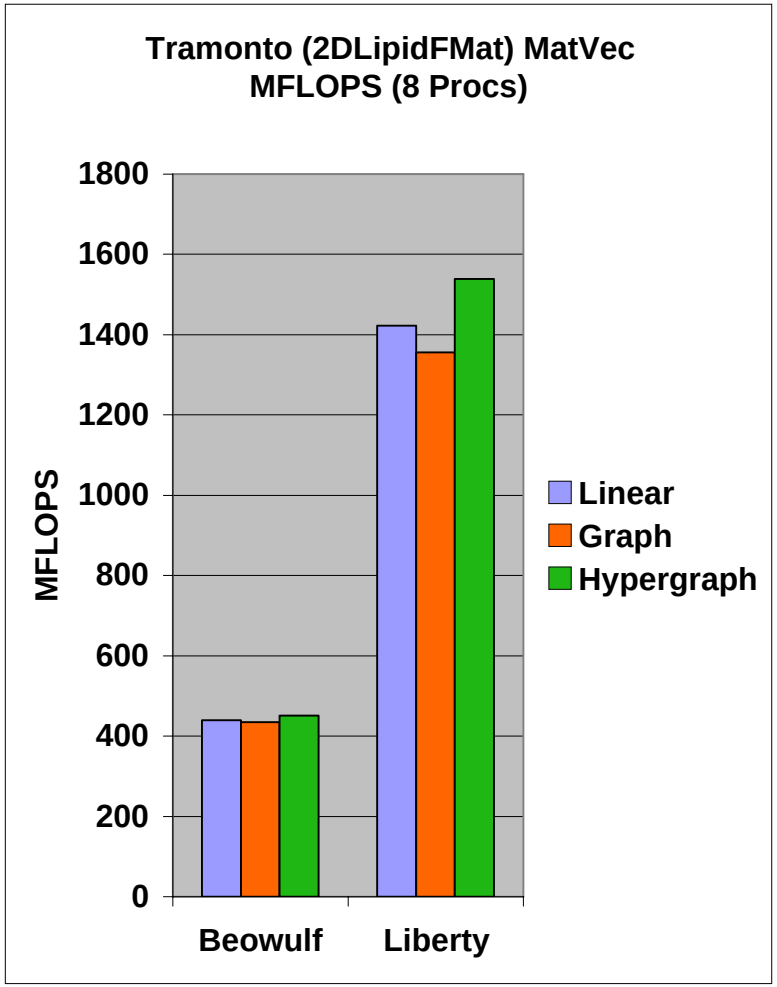


Courtesy Karen Devine

Sandia National Laboratories

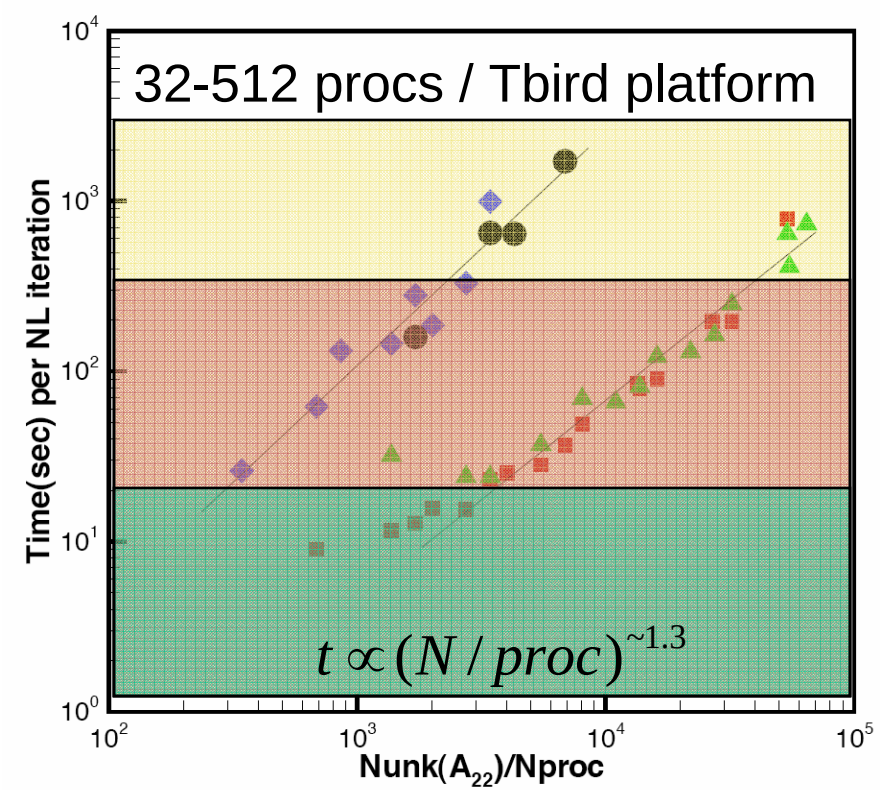
Performance

- *Hypergraph partitioning yields more MFLOPS in linear solve than graph partitioning.*



Sandia National Laboratories

Quantifying the computing challenge



HS: 1 comp.
HS+Poisson: 3 comp.
LJ Att: 1comp.
5-mer bonded hard chain

"Modeling" Calculations
O(1-100) solutions

"Design" Calculations
O(1000) solutions

"Embedded" Calculations
O(10000+) solutions

Sandia National Laboratories

Summary

Emphasis on algorithms has impacted applications work in a significant way.

Many complex 3D systems can be studied **now.**

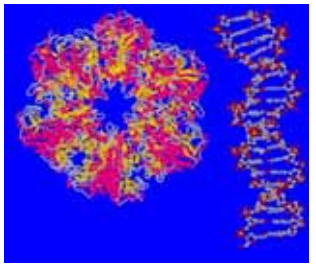
Much more work to be done

- Parallel Partitioning
- DFTs with greater complexity
- Optimization of preconditioners
- Solution complexity and physical phases
- Design applications
- Coupled (multiscale) methods
- Other better approaches

Sandia National Laboratories

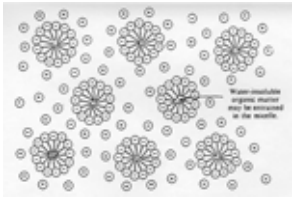
Summary, cont.

- New family of scalable solvers for complex fluid systems in Tramonto.
- Properties:
  - No tuning parameters.
  - Robust to processor count increase.
  - 5-20 times memory use reduction over previous approaches.
  - *O(10)-O(100)* reduced implicit problem size.
  - Nearly linear scalability in: processor count, mesh density, polymer chain length.
  - Candidate for petascale class computing.

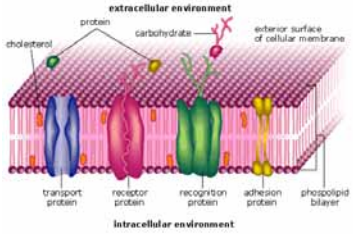


Biological Macromolecules
(www.hmi.de/people/kroy/rota.html)

- Enables:
  - Fundamentally new calculations for important bio problems. Quotes from *Physical Review Letters* referees on computations using these solvers:
    - "This is (to my knowledge) the first time [Fluid] DFT has been used to analyze the important problem of pore structure in biological membranes."
    - "This appears to me to be a highly significant advance in theoretical biophyics, even by the high standards of Physical Review Letters. I suspect that this Sandia group is the only one in the world to have developed classical DFT methods sufficiently sophisticated to deal with such a remarkably complex problem in colloidal physics…"
    - "…I would then recommend at least a footnote that gives some introductory hint as to how they have managed to cope numerically with such complex structures; presumably a 3d finite element method with **all manner of tricks**?
  - The "tricks" are the solvers.
  - *Parallel Segregated Schur Complement Methods for Fluid Density Functional Theories, M. Heroux, L. Frink, A. Salinger* to appear in SIAM SISC.
  - Tramonto first public release this year.



Colloidal/Amphiphilic systems
(www.science.duq.edu)



Cell membranes

http://software.sandia.gov/tramonto

Welcome to the Tramonto home page: Software for Nanostructured Fluids in materials and biology

Project goals

This project is based at Sandia National Laboratories, and is focused on developing molecular theory based computational tools for predicting the structure and properties of fluids at the nanoscale near surfaces and macromolecules. At this length scale fluids are inhomogeneous and common approximations for bulk fluids such as incompressibility do not apply. The specific capabilities of Tramonto and the related FasTram software packages are detailed in the Capability links to the left. In both cases, the molecular theories treated by the codes are fluid density functional theories (F-DFTs). These theories compute fluid structure near surfaces or as a result of self-assembly in contrast to quantum density functional theories (Q-DFTs) which are widely used to compute electronic structure of materials.

Applications of Fluids-DFTs

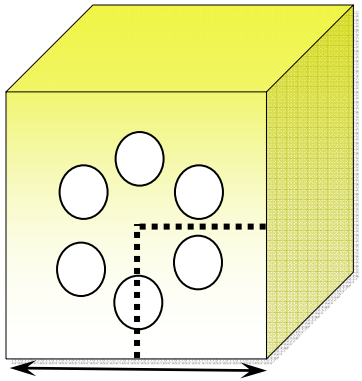
Fluids Density Functional Theory approaches have been used to study a wide range of physical systems. Some examples are: fluids at interfaces, surface forces, colloidal fluids, wetting, porous media, capillary condensation, interfacial phase transitions, nucleation phenomena, freezing, self-assembly, lipid bilayers, ion channel proteins, solvation of surfaces and molecules. The characteristic particle size in F-DFT models ranges from atoms (e.g Argon) to colloidal particles, proteins, or cells. Thus these F-DFT approaches provide a multiscale framework for studying the physcis of many complex fluid systems. Some of these applications are represented in the publication list on the left, others may be found in a very diverse literature. The Tramonto code does not capture all of the F-DFT approaches that have been developed to date, but can be extended to new theories and models.

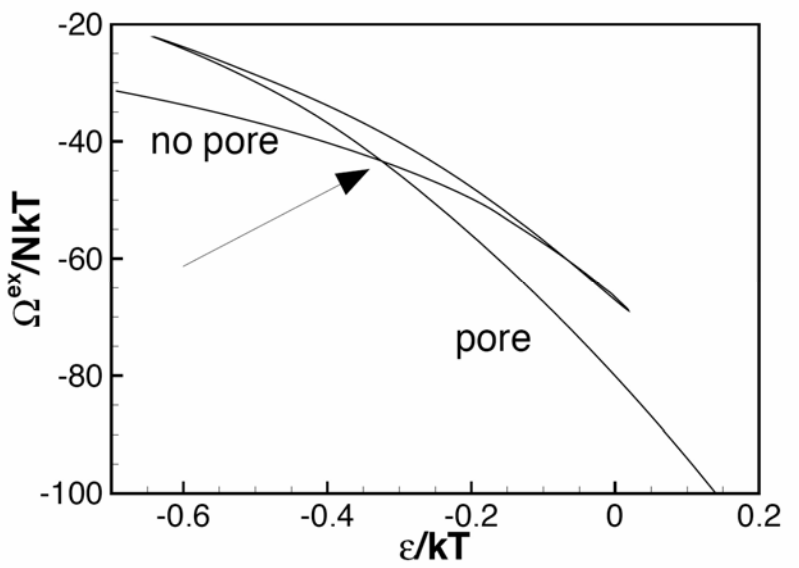
Motivation - a Scientific Computing perspective.

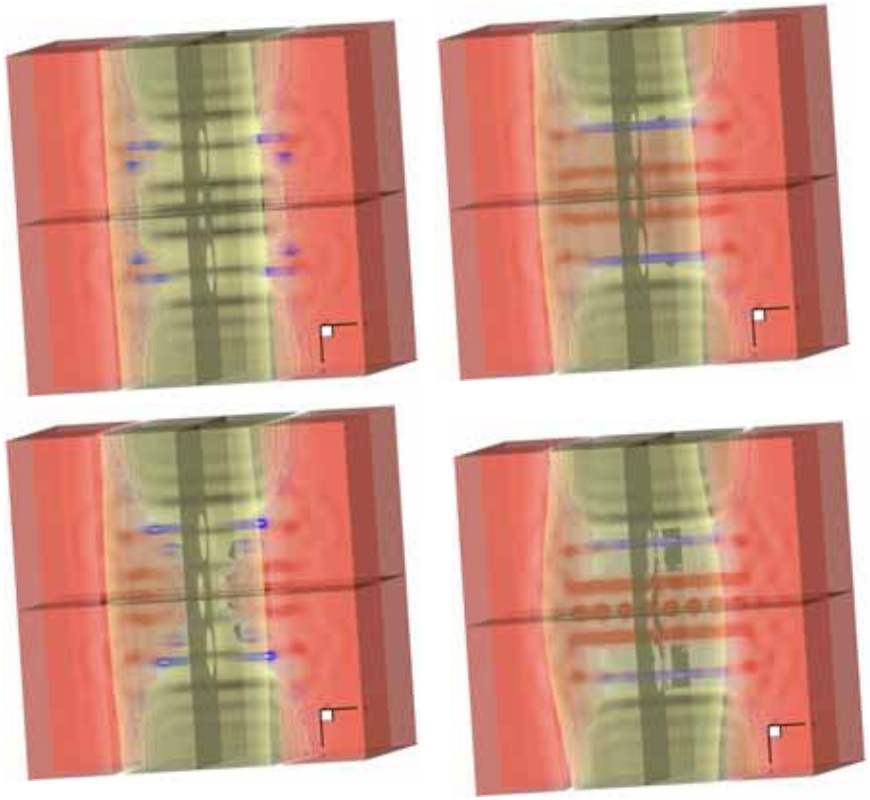
Until recently, application of F-DFTs to problems in inhomogeneous fluids was limited primarily to systems with two dimensions of symmetry allowing for 1-dimensional computations. In that domain, fast calculations can be performed on single processer computers using algorithms of limited sophistication (e.g. Picard iterations).

Two and three dimensional calculations for F-DFTs are much more costly due to the integral nature of the systems of equations. To understand the computational cost, consider the differences between partial differential equations (PDEs) and the integral equations associated with F-DFTs. The nodes in PDEs generally interact only with nearest neighbors or next nearest neighbors often resulting in diagonally dominant sparse matrices. As the mesh is refined the number of interactions remains constant although there are more nodes to process. In the case of DFTs the range of the integration stencils is significantly longer based on the underlying physics included in

3D Studies of Antimicrobial Peptide Assemblies in lipid bilayers with CMS-DFT…



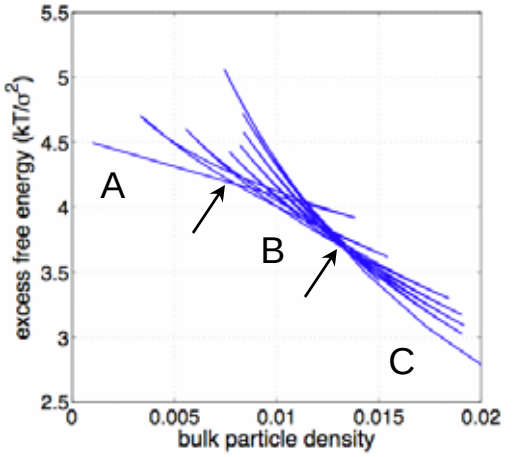


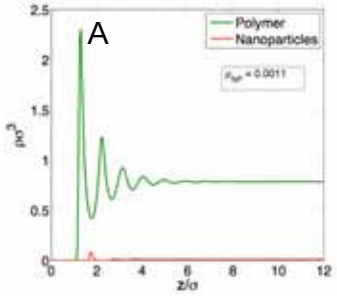


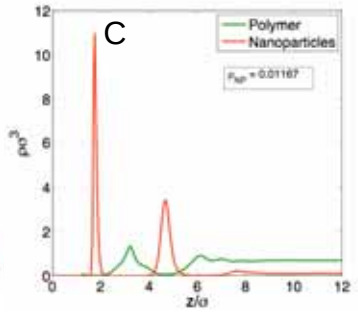
Sandia National Laboratories

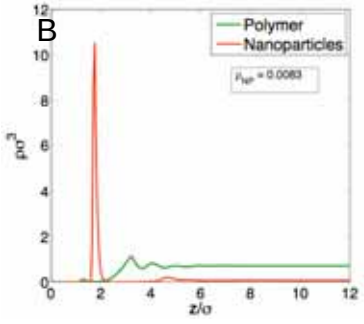
Competing phases / multiple solutions

1D: polymer/nanoparticles

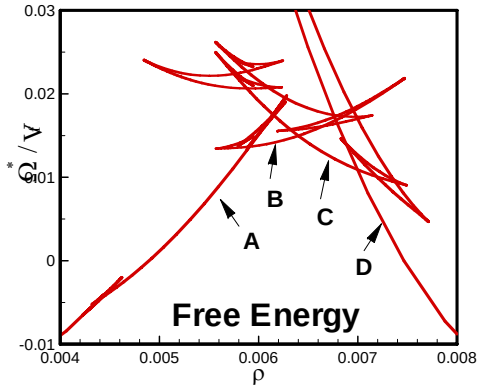


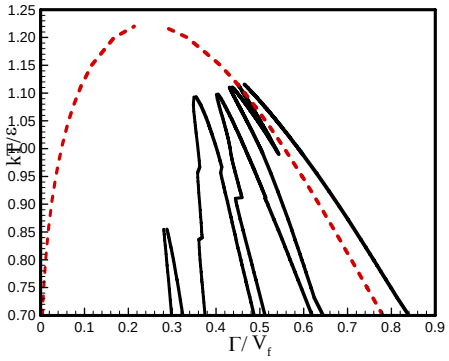


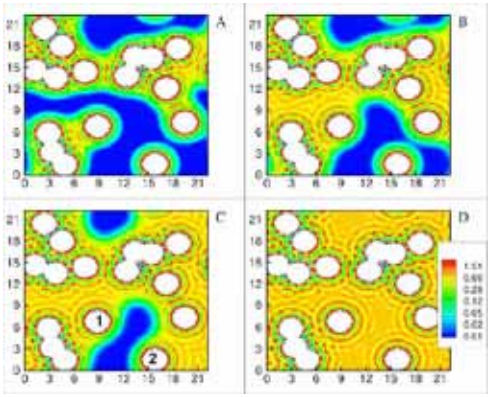




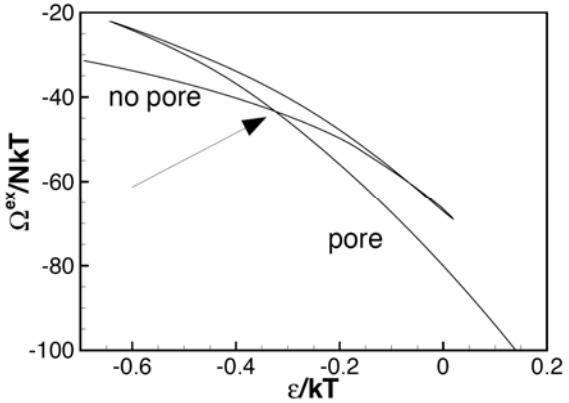
2D: porous media

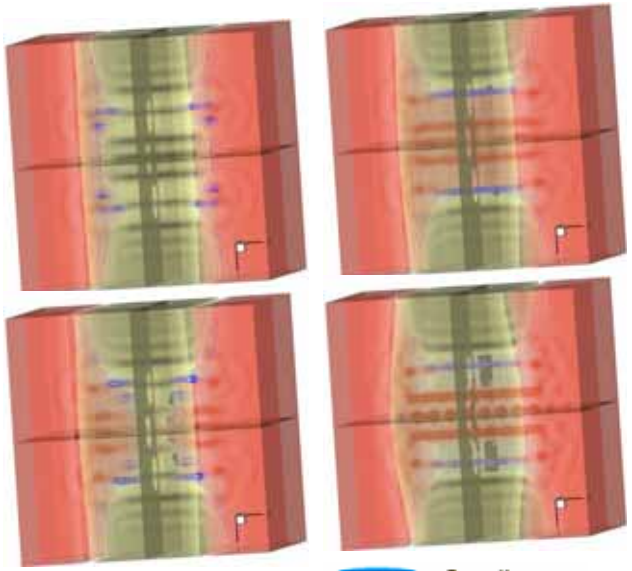






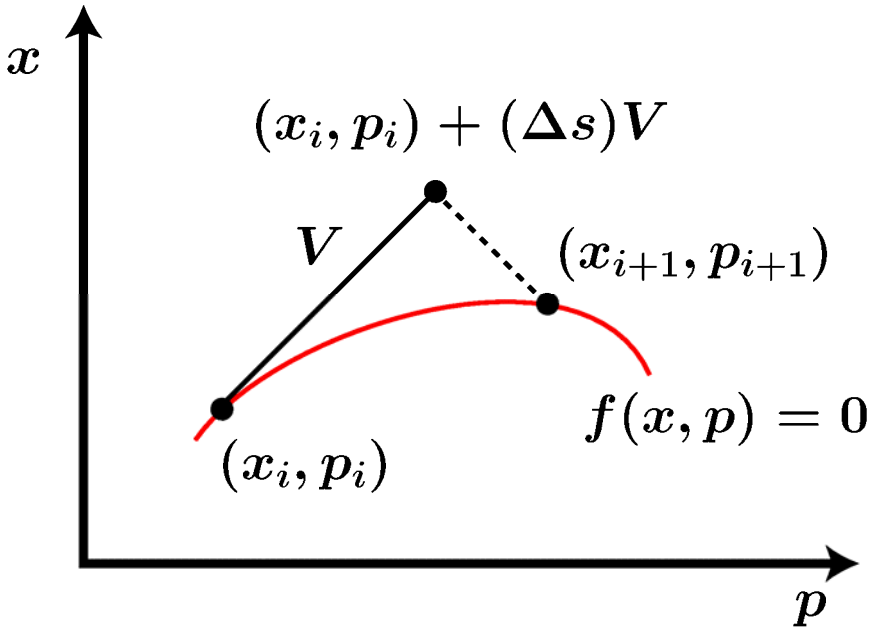
3D: AMP assemblies in CG Lipid bilayer membranes

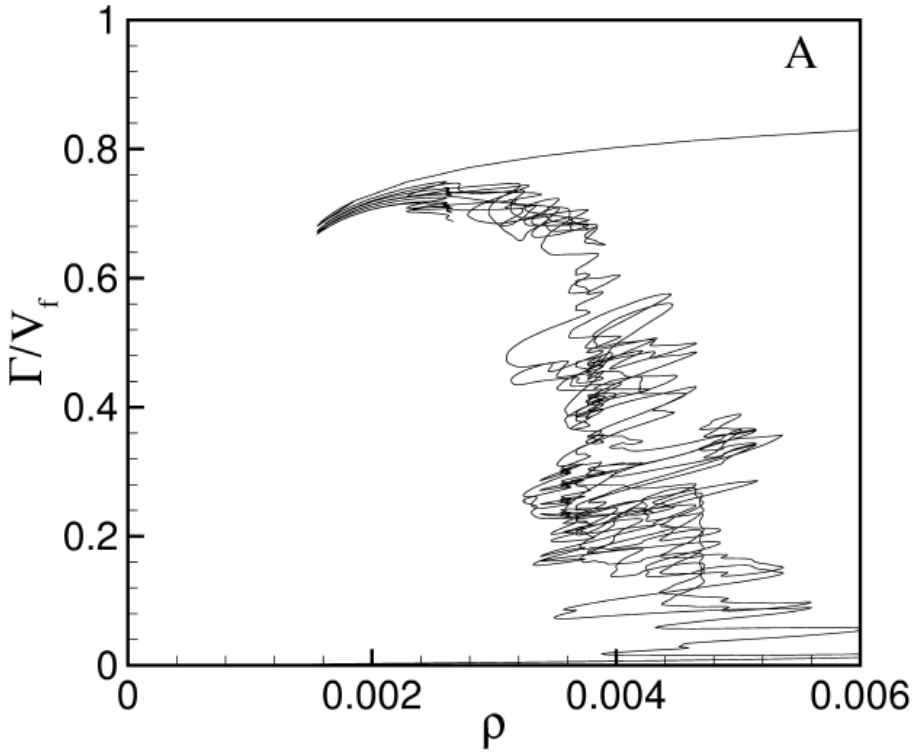




Sandia National Laboratories

Pseudo Arc-length Continuation Solves for Solution and Parameter Simultaneously





$$f(x,p) = 0$$

$$V_x^T(x - x_i) + V_p(p - p_i) = \Delta s$$

Sandia National Laboratories

Extra slides

Sandia National Laboratories

The trivial part - the ideal gas

The free energy of an ideal gas fluid can be written exactly as:

$$F_{id} = \int \rho(r)[\ln \rho(r) - 1]\, dr$$

$$\frac{\delta\Omega}{\delta\rho(z)}_{\mu,T} = \ln \rho(z) + V(z)/kT - \mu = 0$$

$$\mu = -\ln \rho_0$$

$$\ln \rho(z)/\rho_0 = -V(z)/kT$$

For an ideal gas in a gravitational field we find:

$$\boxed{\ln p(z)/p_0 = -gh/RT} \longrightarrow \text{Barometric pressure}$$

Sandia National Laboratories

The simplest non-trivial system - the hard sphere fluid.

In the bulk, a very good equation of state is known for the hard sphere Fluid - it is the Carnahan-Starling equation, and is exact.

$$p/kT = \frac{1+\eta+\eta^2+\eta^3}{(1-\eta)^3} \qquad \eta = \frac{\pi\sigma^3}{6}\rho$$

Local density approximations based on the Carnahan-Starling equation result in slowly varying and incorrect density profiles for hard spheres near hard surface. They overestimate the energy penalty associated with packing at the solid interface

The hard sphere fluid…

In practice, accurate DFTs take a nonlocal approach to defining the volume exclusion contributions to the free energy functional.

$$F_{hs} = \int \Phi(n^{(\gamma)})dr$$

$$n^{(\gamma)}(r) = \int \omega^{(\gamma)}(r - r')\rho(r')dr'$$

- These nonlocal density approaches can be very accurate in predicting the structure of interfacial fluids.

- Free energy density: a function of weighted average of all densities in nearby region of fluid.

Sandia National Laboratories

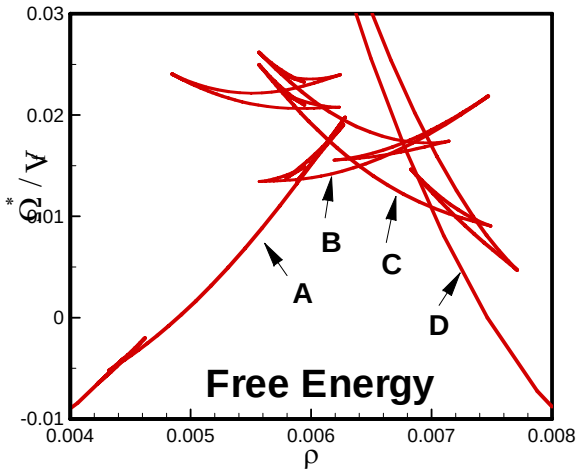
The Euler-Lagrange equations…

We seek the the stationary states of the free energy functional with the understanding that the thermodynamically relevant state should be found At the global free energy minimum.

$$\left(\frac{\delta\Omega}{\delta\rho(r)}\right)_{\mu,T} = 0$$

$$\Omega[\rho(r)] = F_{id} + F_{hs} + \int \rho(r)[V(r) - \mu]$$

$$0 = \ln\rho(r) + \int \sum_{\gamma} \frac{\partial\Phi}{\partial n^{(\gamma)}}(r') \frac{\delta n^{(\gamma)}(r')}{\delta\rho(r)} dr' + [V(r) - \mu]$$



Given that: $n^{(\gamma)}(r) = \int \omega^{(\gamma)}(r - r')\rho(r')dr'$

Our residual is: $0 = \ln\rho(r) + \int \sum_{\gamma} \frac{\partial\Phi}{\partial n^{(\gamma)}}(r')\omega^{(\gamma)}(r - r')dr' + [V(r) - \mu]$

Sandia National Laboratories

A closer look at the blocks…

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} \Delta n^{(\gamma)}(r) \\ \Delta\rho(r) \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

We are solving two residual equations simultaneously:

$$R_1 = n^{(\gamma)}(r) - \int \omega^{(\gamma)}(r-r')\rho(r')dr'$$

$$R_2 = 0 = ln\,\rho(r) + \int \sum_{\gamma} \frac{\partial\Phi}{\partial n^{(\gamma)}}(r')\omega^{(\gamma)}(r-r')dr' + [V(r)-\mu]$$

This linearized system leads to the following block Jacobian entries:

unity $A_{11}^{(\gamma\varepsilon)}(r,r') = \delta^{(\gamma\varepsilon)}(r,r')$

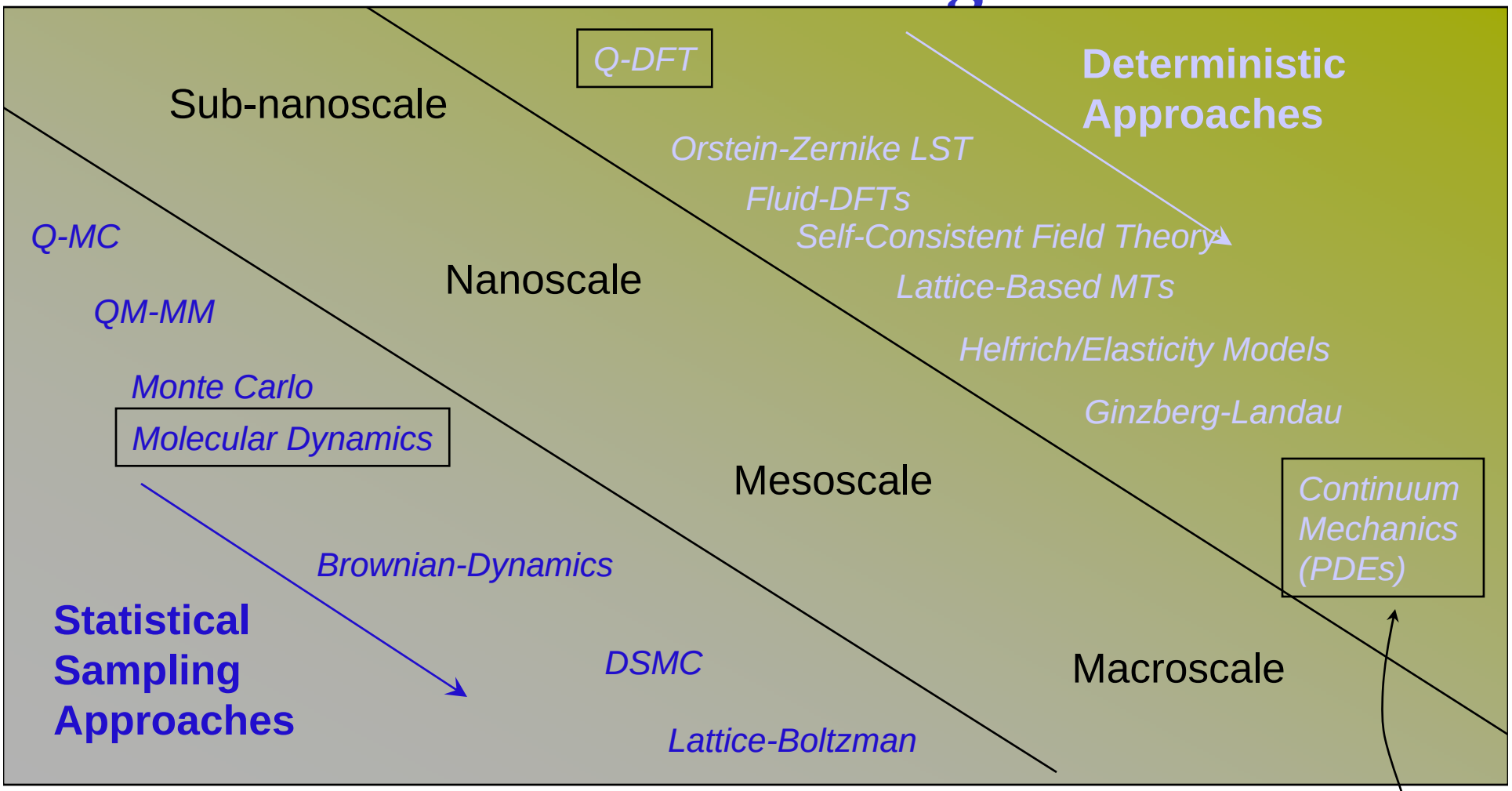
$A_{12}^{(\gamma)}(r,r') = \omega^{(\gamma)}(r,r')$ constant

$A_{21}^{\varepsilon}(r,r') = \frac{\partial}{\partial n^{(\varepsilon)}} \sum_{\gamma} \frac{\partial\Phi}{\partial n^{(\gamma)}}(r')\omega^{(\gamma)}(r-r')$

$A_{22}(r,r') = \frac{\delta(r,r')}{\rho(r)}$ diagonal

Sandia National Laboratories

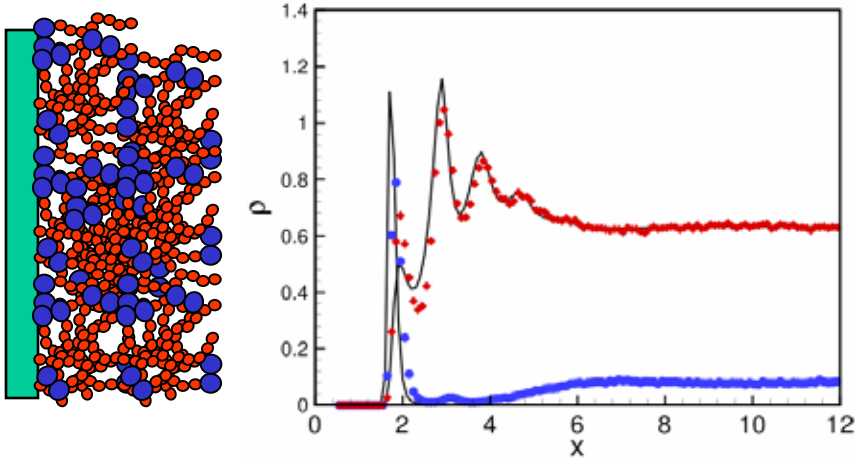
Materials modeling tools



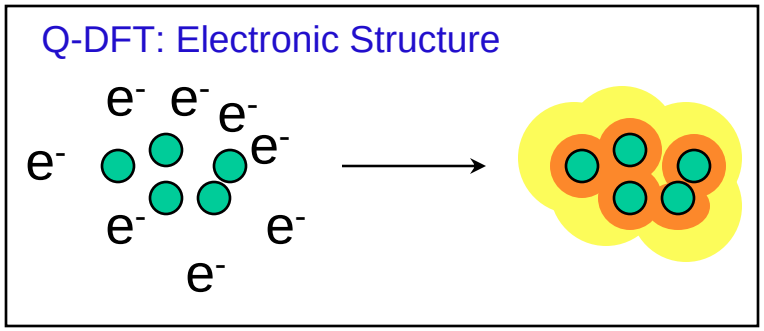
DFT for fluids…

$$V(r) \rightarrow \rho(r)$$

External field — Density profile



- General and flexible approach with broad application space
- Reasonably accurate in many cases - well developed for reference fluids.
- May be applied at many length scales
  - o Nanoscale: atomic / molecular /polymer systems
  - o Mesoscale: coarse-grained models (Colloids, Proteins, Cells)
  - o Mesoscale: lattice models (porous media)



Sandia National Laboratories

From a computing perspective…

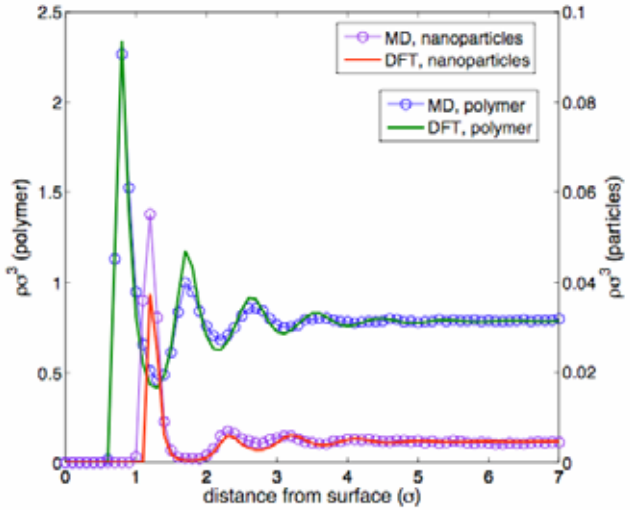
- **Free energy functionals are approximate … many flavors.**
- **Difficult to find a canonical problem for methods development.**
- **Numerical methods for F-DFTs lags behind more widely used computational methods (PDEs / MD / Electronic Structure).**

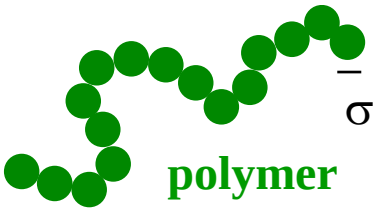
Systems with 2 dimensions of symmetry have been most widely studied.

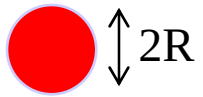
Nanocomposite thin films

Experiment:
PS nanoparticles go to the surface
(Krishnan et al., Langmuir, 2005)

DFT/MD:
E. S. McGarrity and M. E. Mackay (MSU),
A.L. Frischknecht (Sandia)







nanoparticle

Sandia National Laboratories

Algorithms work

Goal : Develop general and robust algorithms for F-DFTs in complex geometries (including 3D) for broad classes of fluids (atomistic to polymers).

- Develop solver strategies specific to F-DFTs for parallel (and serial) computational platforms.
  - Some general strategies
  - Some problem specific implementations

- Couple solver methods with engineering analysis tools
  - Arc-length continuation
  - Multi-state tracking
  - Optimization

Sandia National Laboratories

One class of DFTs: perturbations to a hard sphere reference fluid…

Coulomb interactions | Associations (H-bonding) | [Applied field]

$$\Omega[\rho(r)] = F_{id} + F_{hs} + F_{vdW} + F_c + F_{assoc} + \int \rho(r)[V(r) - \mu]$$

Ideal gas | Hard sphere | Dispersion attractions

Legendre Transform from Canonical to Grand canonical ensemble

$$\frac{\delta\Omega}{\delta\rho_i(r)}_{\mu,T} = 0$$

$$\frac{\delta}{\delta\rho}\left(\int f[\rho] * g[n[\rho]]dr\right) = \cancel{\frac{\partial f}{\partial\rho} g[n[\rho]]} + \cancel{\int f[\rho]} \frac{\partial g}{\partial n}\frac{\delta n}{\delta\rho} dr$$

1

$$f[\rho] = 1$$

$$g[n] = \Phi$$

Sandia National Laboratories

Residuals and Jacobians

$$\frac{\delta\Omega}{\delta\rho_i(r)} = \int \sum_\gamma \frac{\partial\Phi}{\partial n_\gamma}(r') \frac{\delta n_\gamma(r')}{\delta\rho_i(r)} dr'$$

$$n_\gamma[\{\rho_i(r)\}] = \sum_i \int dr' \rho_i(r') w_i^{(\gamma)}(r - r'; R_i)$$

$$w_i^{(\gamma)}(|r-r'|) = C_\gamma \delta(|r-r'| - R_i)$$

$$w_i^{(\gamma)}(|r-r'|) = C_\gamma \theta(|r-r'| - R_i)$$

Integral Eqns
Of Finite Range!

Residual $$\frac{\delta\Omega}{\delta\rho_i(r)} = \ldots + \int \sum_\gamma \frac{\partial\Phi}{\partial n_\gamma}(r') w_i^{(\gamma)}(r - r') dr' + \ldots$$

Jacobian
$$A_{ij}(r,r') = \frac{\delta^2\Omega}{\delta\rho_i(r)\delta\rho_j(r')} = \ldots + \int \sum_\varepsilon \sum_\gamma \frac{\partial^2\Phi}{\partial n_\gamma \partial n_\varepsilon}(r'') w_i^{(\gamma)}(r - r'') w_j^{(\varepsilon)}(r' - r'') dr'' + \ldots$$

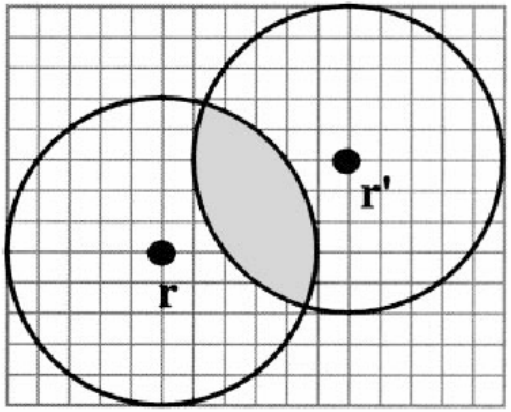
Sandia National Laboratories

Two ways to form the matrix problem…

(1) “second order” in complexity in forming the system of equations.

$$[A][\Delta\rho] = [b]$$

$$A_{ij}(r,r') = \frac{\delta^2\Omega}{\delta\rho_i(r)\delta\rho_j(r')} = \ldots + \int \sum_{\varepsilon}\sum_{\gamma} \frac{\partial^2\Phi}{\partial n_\gamma \partial n_\varepsilon}(r'')w_i^{(\gamma)}(r-r'')w_j^{(\varepsilon)}(r'-r'')dr'' + \ldots$$



---

(2) Reduced fill complexity.

$$R_1 = 0 = n(r) - \int w(r-r')\rho(r')dr'$$

$$R_2 = 0 = \ldots + \int \frac{\partial\Phi}{\partial n}(r')w(r-r')dr' + \ldots$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} \Delta n(r) \\ \Delta\rho(r) \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$A_{11} = I$$

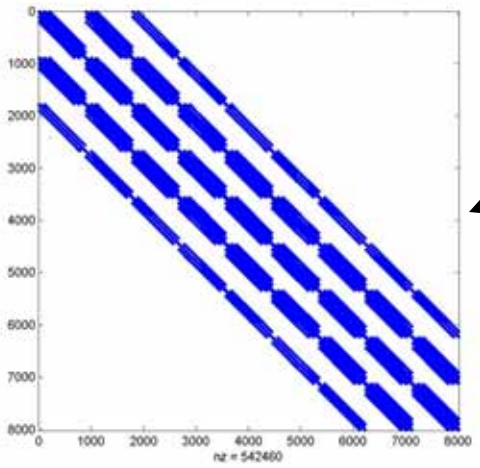
$$A_{12} = -\omega(r-r')$$

$$A_{21} = \frac{\partial^2\Phi}{\partial n^2}\omega(r-r')$$

$$A_{21} = \frac{\delta(r,r')}{\rho(r)}$$

Sandia National Laboratories

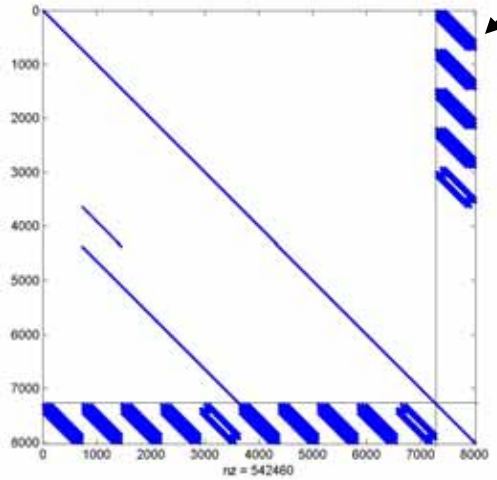
Two ways to order the matrix…



Nodal Ordering

Example:
Hard-sphere problem

Physics Block Ordering



Block Matrix Nonzeros

Subblock	Nonzeros
A11	10,935
A12	176,932
A21	353,864
A22	729

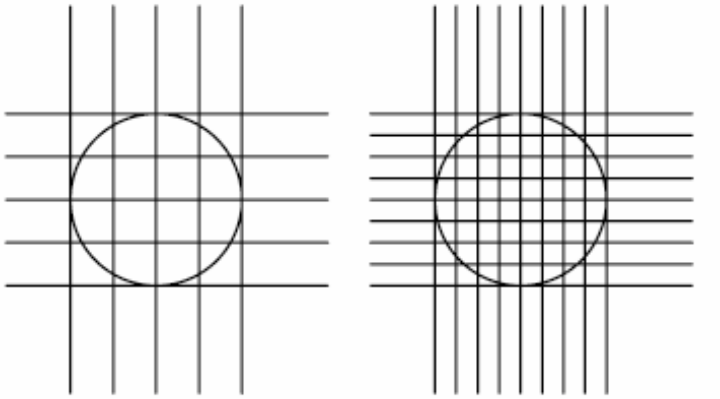
Sandia National Laboratories

Properties of *F-DFT systems*

DFT - Integral equations of finite range
(matrix density is system size dependent)
PDE - matrix density independent of system size.

DFT- Inter-physics coupling dominates
PDE - Inter-nodal coupling dominates

DFT - Stencils based on physical constants
PDE - Stencils based on nearest neighbors



DFT - May have large numbers of DOF per node
HS (3D) 10+
Polymer (20 beads) 42+
PDE - Usually a few DOFs per node

Sandia National Laboratories

Solving F-DFTs on parallel computers

1. A_{11}^{-1} easy to compute (or apply) --> can form (or apply implicitly) S easily in parallel

2. Dimension of S is much smaller than A: iterative methods (e.g. GMRES) will typically converge faster in parallel.

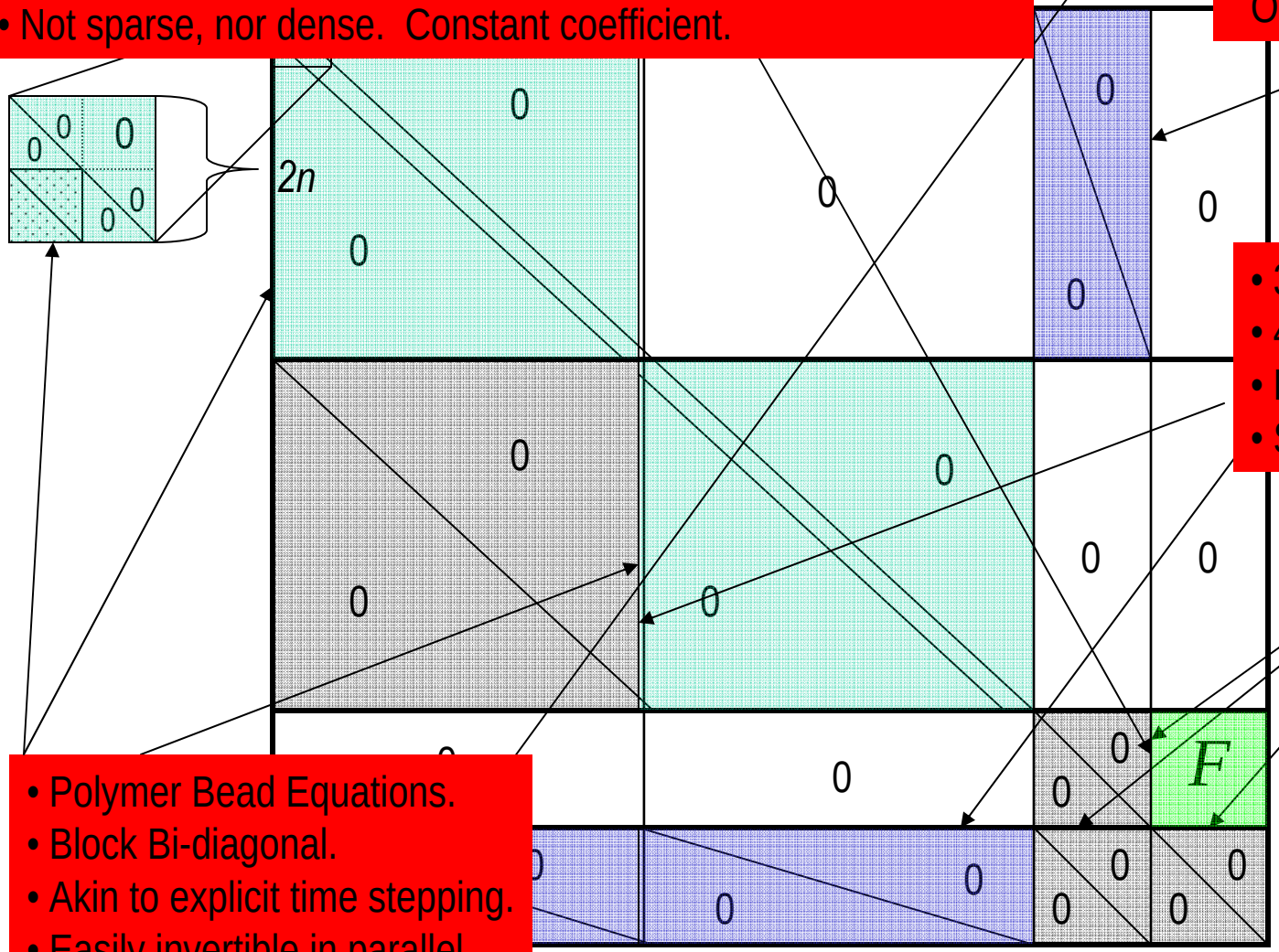
3. Given an equal partitioning, parallel execution will be well-balanced and produce identical results independent of the number of processors.

Sandia National Laboratories

Proble

- There is only ONE interesting block in this whole matrix!!
- F describes CMS field dependence on primitive densities.
- 2.5 ◆ radius integral at each grid node (mesh independent).
- Well-conditioned matrix with strong main diagonal.
- Not sparse, nor dense. Constant coefficient.

- Diagonal-like.
- One non-zero per row/col in long dimension.
- Like Prolongation/restriction Operators?



- 3rd block: CMS Field
- 4th block: Prim Densities
- Diagonal matrices.
- See? No spatial coupling!

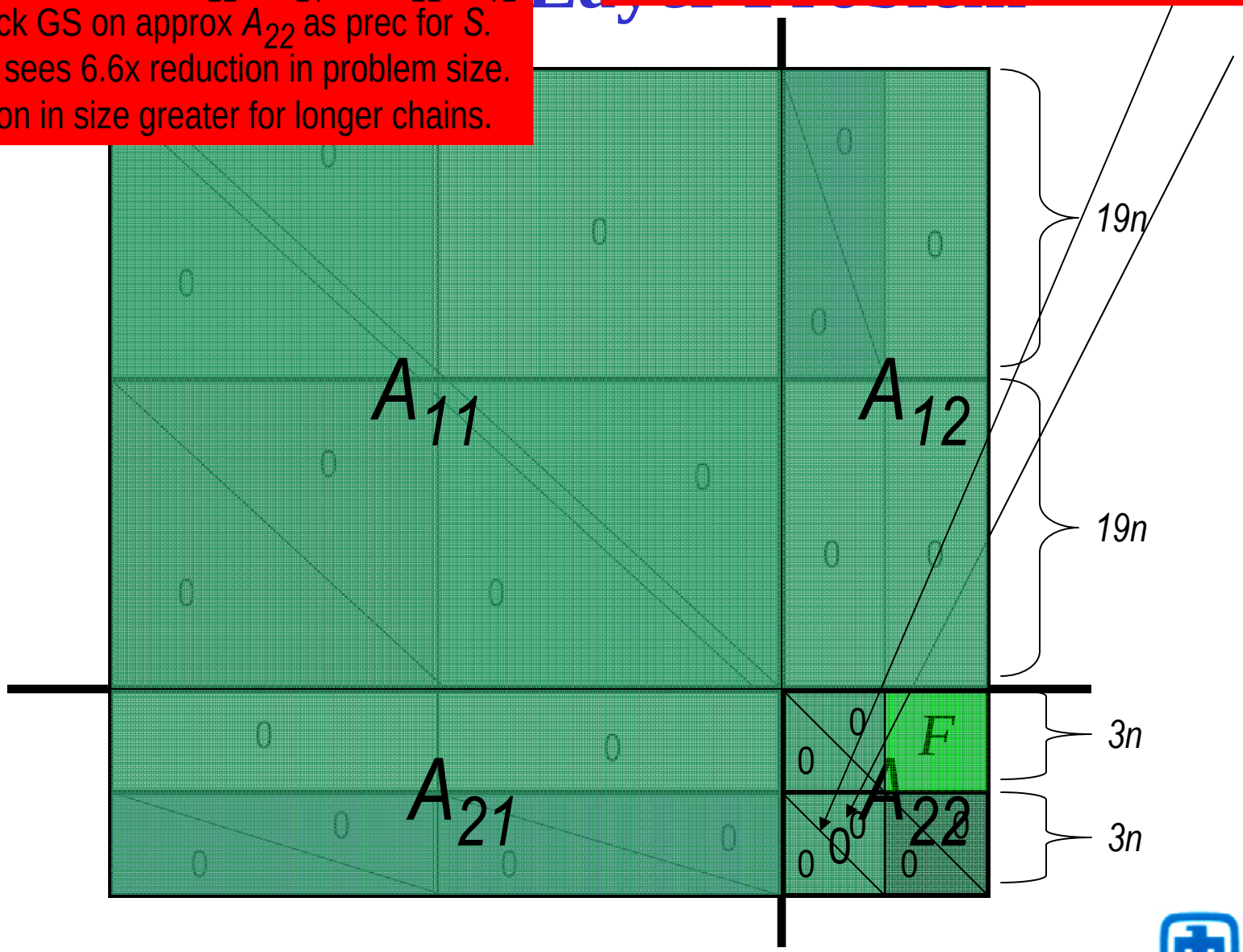
- Polymer Bead Equations.
- Block Bi-diagonal.
- Akin to explicit time stepping.
- Easily invertible in parallel.

Sandia National Laboratories

Layer Reduction

- Last layer of structure: 2-by-2 partitioning.
- A_{11} solve easily applied in parallel.
- Apply GMRES to $S = A_{22} - A_{21}*inv(A_{22})*A_{12}$
- Use block GS on approx A_{22} as prec for S.
- GMRES sees 6.6x reduction in problem size.
- Reduction in size greater for longer chains.

- Ignore these small terms for preconditioning.
- Introduces spatial parallelism.
- Approximate *inv(A22)* using block Gauss-Seidel.



Sandia National Laboratories

Properties of New Solver

- This general approach has many favorable properties:
  - If mesh nodes are uniformly distributed, work will also be.
  - Each substep of preconditioner is naturally parallel:
    - Results invariant to processor count up to round-off.
  - Preconditioner requires almost no extra memory over storage of matrix: Memory reduction of 4-10 X over previous approach.
  - GMRES subspace and storage reduced 6X-10X or more.
  - Speedup 20-2X (difference goes down as PE count grows).
  - Solver has:
    - No tuning parameters.
    - Near linear scaling.
  - Increased problem sizes (in domain size and mesh refinement).