Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

THE UNIVERSITY OF CHICAGO

Office of Science
U.S. DEPARTMENT OF ENERGY

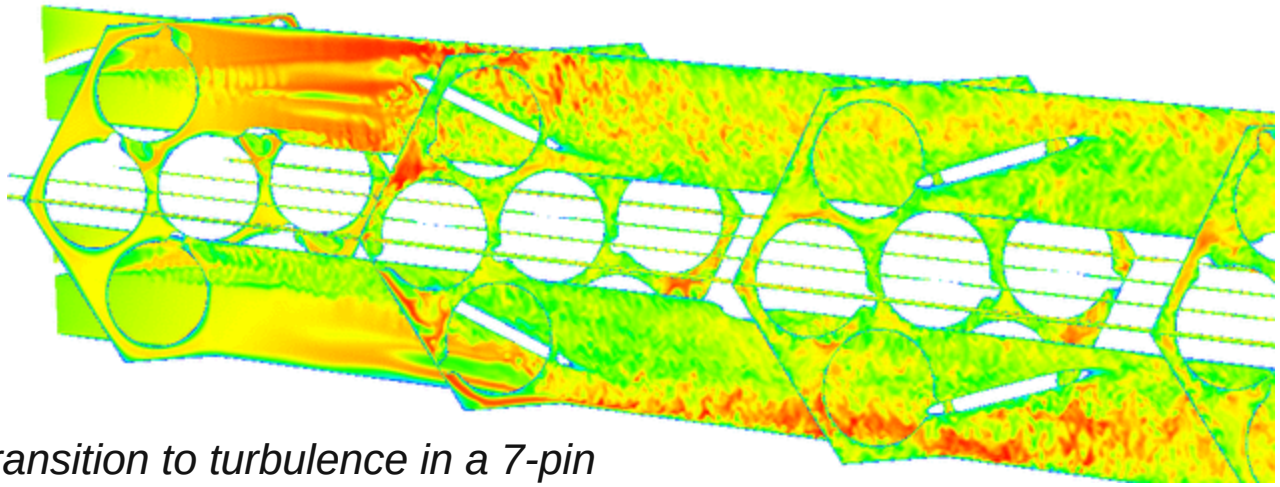
A U.S. Department of Energy laboratory managed by The University of Chicago

Advanced Reactor Simulation

Paul Fischer
James Lottes
David Pointer
Andrew Siegel

Argonne National Laboratory

Carlos Pantano
UIUC



Transition to turbulence in a 7-pin reactor subassembly with wire-wrapped fuel pins

Outline

- *Advanced Reactor Modeling Science*
- *Petascale Computational Issues*
- *Summary and Some Remarks on the Path Forward to a Million Processors*

Advanced Simulation & Modeling Effort for Fast Reactor Design

- By burning minor actinides, fast reactors offer the potential of 100x reduction in geological repository requirements and an increase in available fissionable materials.

- DOE's NE program has recently embarked on an ambitious simulation program for reactor modeling, reprocessing, seismic analysis, etc.

- Reactor development based at ANL. Two of the principal areas are:

  - Neutronics
    - *New scalable neutronics code, UNIC, designed specifically for fast reactor analysis (thousands of energy groups)*

  - Thermal hydraulics – ***focus of this talk***

Overview of TH Modeling Approach

Multiscale simulation hierarchy involving:

1. *experiments*
2. *DNS (direct numerical simulation of turbulence)*
3. *LES (large eddy simulation)*
4. *RANS (Reynolds-averaged Navier-Stokes)*
5. *Subchannel or lumped-parameter models*

Multiscale approach provides an important validation path:

- *In the past, only Options 1 and 5 were available.*

Thermal Hydraulics Simulation Effort

Two problem areas identified by the reactor design group:

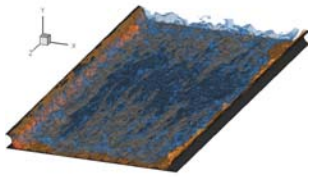
1. Mixing and pressure drop in **fuel rod bundles**,
   - *Controls peak temperature → power output*
   - *Influence of*
     - wire-wrap vs. grid spacers
     - wall effects are important → low pin count results do not extend to higher pin counts

2. **Thermal mixing** in the upper plenum
   - *Influences longevity of mechanical structures and places design constraints on reactor (outlet temperature differences)*

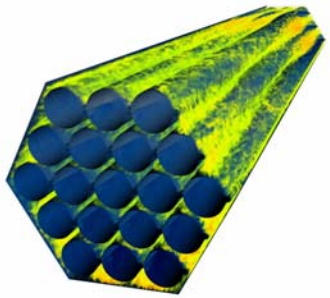
Fuel Bundle Subassembly Analysis

2007-8 INCITE Awards

- DNS of simple pin model: C. Pantano-UIUC

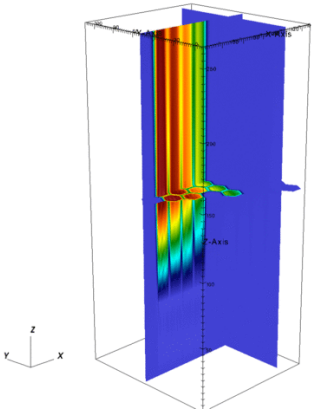


- LES of multipin assemblies:
  - *2007: 7-pin, 2008: 19- & 37-pin, 2009: 217 pin*



- RANS – up to 217 pins: D. Pointer, ANL
  - *16-64 proc. Linux cluster – k-ε model, Star CD*

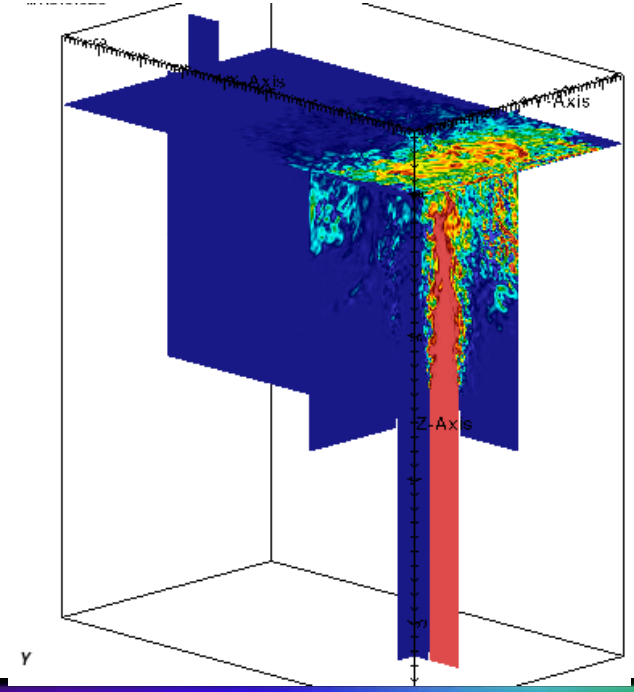
- Subchannel analysis – coupled neutronics/TH: entire SHARP team
  - *217 pins, 1/6 core– no wire detail*

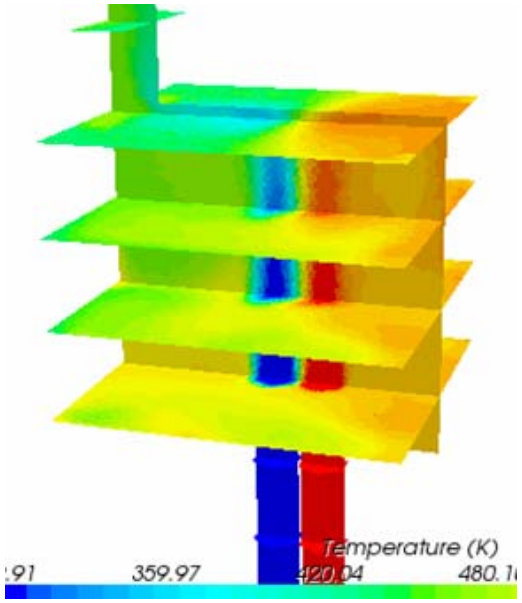


Thermal Mixing in the Upper Plenum

- Influences longevity of mechanical structures
- Places reactor design constraints on outlet temperature differences
- Not well-understood
- ANL investing $1 M in detailed experiment
- BG/P simulations supported through INCITE

Initial transient for LES and steady-state RANS

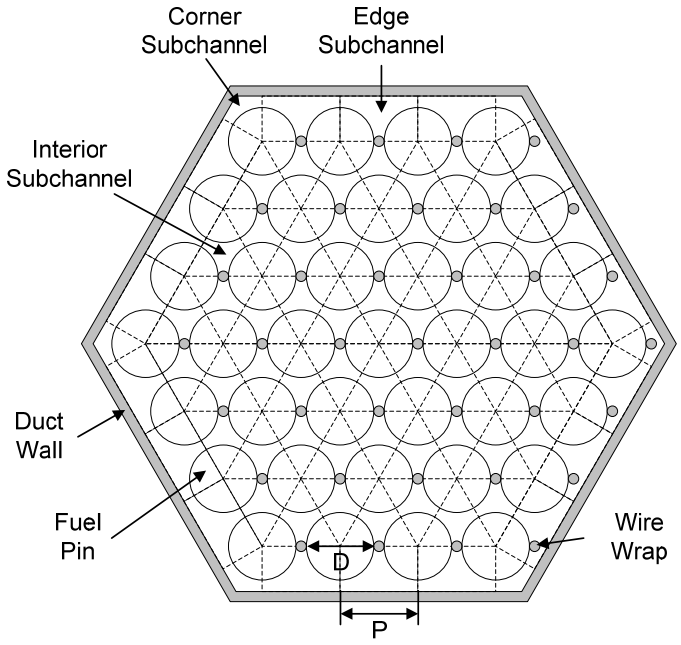


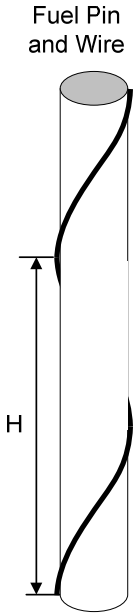


Results for Rod Bundle Flows

Coolant Flow in SFR Subassemblies

- *Interchannel cross-flow is principal cross-assembly energy transport mechanism*
  - *Uniformity of temperature controls peak power output*
  - *A better understanding of flow distributions is required to improved designs*
- Not accessible to DNS or subchannel codes
  - *Only through LES, RANS, or experiments*





Bogoslovskaya et al, IAEA 1157, 2000.

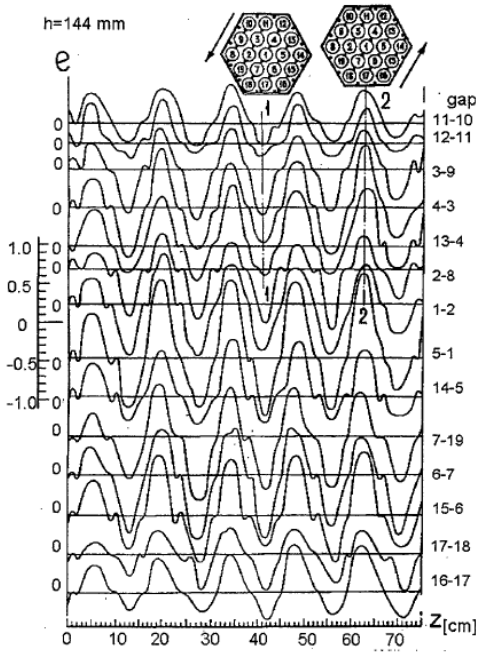


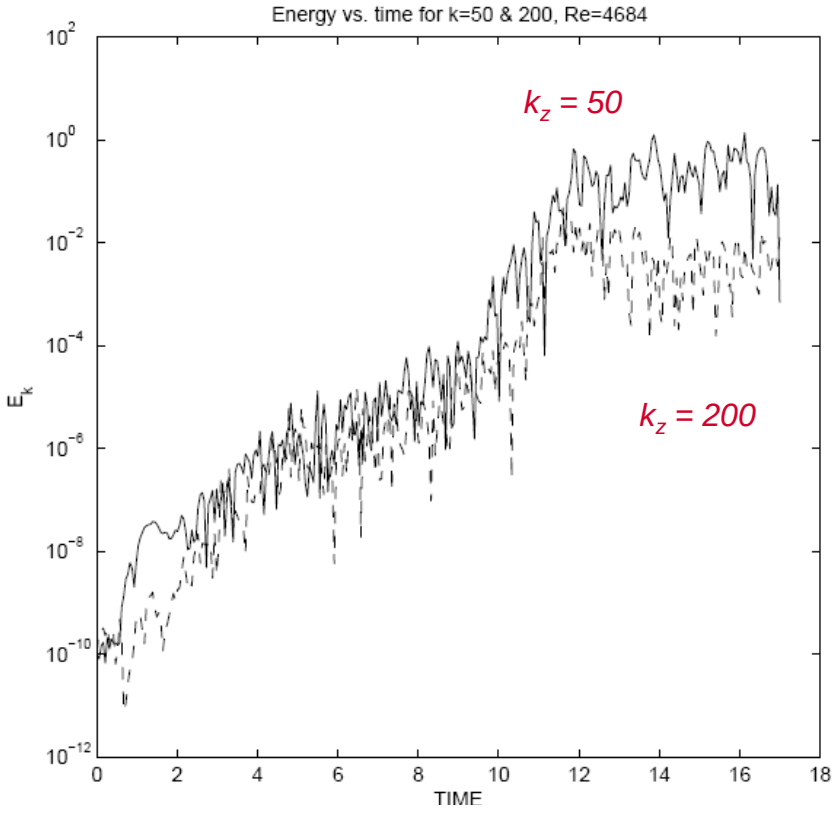
Fig. 1. Variation in transverse flow-over the height of gaps

Prediction of Transition from Earlier Single-Pin Simulations

- Flow establishes a fully turbulent state within ~ 1 flow-through time

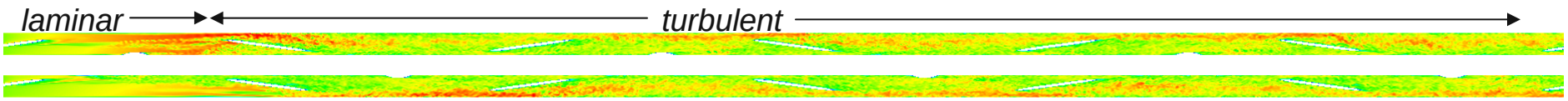
  → *spatial development length ~ H/D*

- In fact, H/D appears to be less relevant than z/D ~ 15

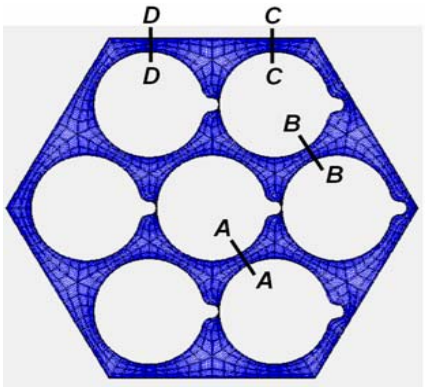


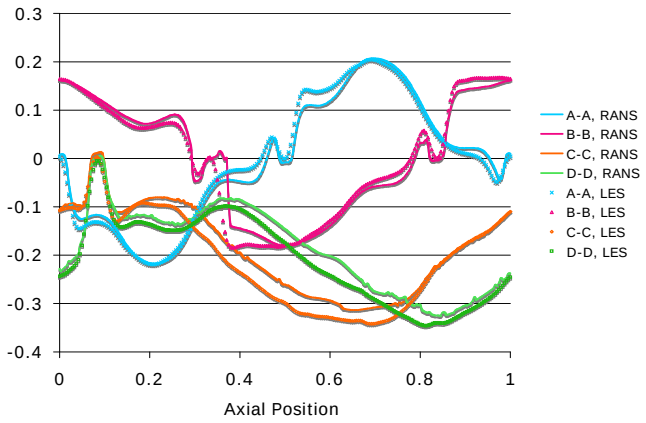
Key Findings: LES of Reactor Subassemblies

- *Transition to turbulence with inflow/outflow boundary conditions in 7-pin x 3H configuration occurs at z ~ H/2:*
  - *use of periodic BCs is warranted,*
  - *significant savings (10 x)*



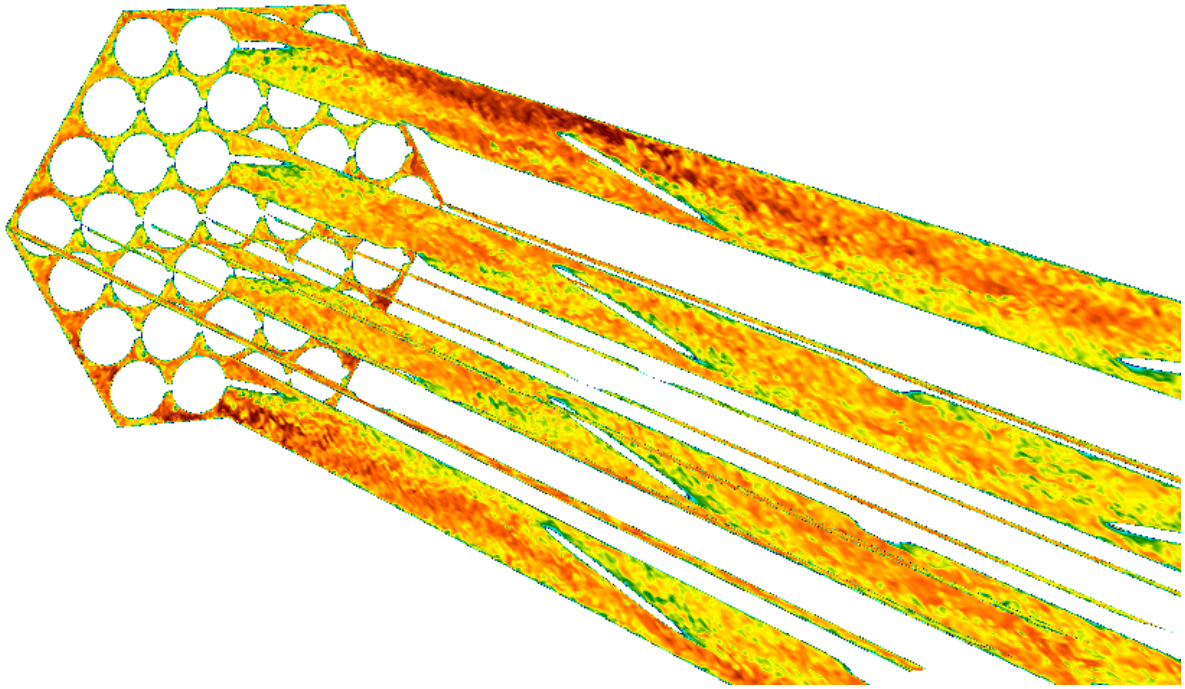
- *LES and RANS simulations give comparable results for cross-flow distributions in 7-pin case:*
  - *We have a mechanism for validating RANS, which gives considerable savings.*
  - ***Data being input to core-scale simulations***





Most Recent Results

- Turbulent flow in a reactor sub-assembly with 37 wire-wrapped fuel pins:
  - E=580000, N=7, n=200 million
  - 2-3 weekends on P=16384 of BG/P
  - Enabled through recent code developments (next topic)
  - Full data waiting to be analyzed w/ Eureka in production mode



Computational Science Issues

Computational Science Objectives

supported by DOE AMR Program

- Enable advanced scientific simulation at petascale and beyond
  - State of the art algorithms and discretizations
    - *High-order, to efficiently capture large/small scale interactions*
    - *Stable, able to accommodate challenging physics and general boundary conditions*
    - *Scalable O(n) solvers*
  - Implemented *at scale* (P > 1 million)
  - Physics focus is on fluid mechanics, heat transfer MHD, and electromagnetics
- This talk:
  - *Understand which computational strategies will / will not scale*
    - Example: all_to_all based schemes ??
  - *Discuss recent infrastructure developments enabling simulation at P > 100K*

Overarching Question: *(Petascale Workshop, March 05)*

- Can we scale to $P = 10^5$??

  The answer is strongly tied to the number of gridpoints per processor…. Fox et al., 1988, Gustafson et al. 1988 (1st Gordon Bell Pr.)

- For the problem class under consideration,

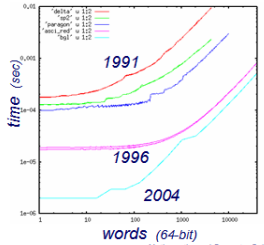
  (N / P) ~ 1000—10000 points per processor

  is sufficient, given current day parameters.

Analysis (Petascale Workshop, March 2005)

Assume a model, measure some parameters, do some analysis, and..

Communication Performance



Linear communication model :
$t_c(m) = \alpha^* + \beta^* m$, m: 64-bit words

Nondimensionalize by t_a [$c = a^*b$] :

$t_c(m) = (\alpha + \beta m)\, t_a$

$\alpha = \alpha^* / t_a$, $\beta = \beta^* / t_a$

Mathematics and Computer Science Division, Argonne National Laboratory

Two Decades of Nondimensional Machine Parameters

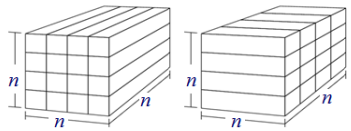
YEAR	t_a (us)	α^*	β^*	α	β	m_2	MACHINE
1986	50.00	5960.	64	119.2	1.3	93	Intel iPSC-1 (286)
1987	.333	5960.	64	18060	192	93	Intel iPSC-1/VX
1988	10.00	938.	2.8	93.8	.28	335	Intel iPSC-2 (386)
1988	.250	938.	2.8	3752	11	335	Intel iPSC-2/VX
1990	.100	80.	2.8	800	28	29	Intel iPSC-i860
1991	.100	60.	.80	600	8	75	Intel Delta
1992	.066	50.	.15	758	2.3	330	Intel Paragon
1995	.020	60.	.27	3000	15	200	IBM SP2 (BU96)
1996	.016	30.	.02	1800	1.25	1500	ASCI Red 333
1998	.006	14.	.06	2300	10	230	SGI Origin 2000
1999	.005	20.	.04	4000	8	375	Cray T3E/450
2005	.002	2.	.013	1000	6.5	154	BGL/ANL

- $m_2 := \alpha / \beta$ ~ message size → twice cost of single-word message
- t_a based on matrix-matrix products of order 10—13

Mathematics and Computer Science Division, Argonne National Laboratory

Global Spectral Methods

- $N = n^3$ points; N/P = number of points per processor
- Work: $\nabla^2 u = f$: 6 FFTs: $T_a \sim t_a\, 30\, n^3 \log_2 n / P = 10\,(N/P) \log_2 N\, t_a$
- Communication: 4 complete exchanges (all-to-all)
  - $4(P^{1/2} - 1)$ messages of length $N/P^{3/2}$:
  - without contention: $T_c \sim [4\, P^{1/2} + 4\,(N/P)/m_2\ ]\, \alpha\, t_a$
  - with contention on 3D torus: $T_c \sim [4\, P^{1/2} + (N/P)/(2m_2)\, P^{1/3}\ ]\, \alpha\, t_a$
  - with contention on 2D mesh: $T_c \sim [4\, P^{1/2} + (N/P)/m_2\ \ P^{1/2}\ ]\, \alpha\, t_a$
- Define $\eta = T_a / (T_a + T_c)$



Mathematics and Computer Science Division, Argonne national Laboratory

Example 2: Nearest Neighbor Algorithms

- Point Jacobi iteration (7-point stencil): $u_i = a_{ii} f_i + \Sigma_j\, a_{ij}\, u_j$
  - — Work: $T_{aJ} \sim 14\, N/P\, t_a$
  - — Communication: $T_{cJ} \sim (\,6 + (N/P)^{2/3}\,(1/m_2)\,)\, \alpha\, t_a$
- Conjugate gradient iteration (7-point stencil): (alt: *Chebyshev iteration*)
  - — Work: $T_{aCG} \sim 27\, N/P\, t_a$
  - — Communication: $T_{cCG} \sim T_{cJ} + 4 \log_2 P\, \alpha\, t_a$
- Multigrid-preconditioned conjugate gradient iteration:
  - — Work: $T_{aMG} \sim 42\, N/P\, t_a$
  - — Communication: $T_{cMG} \sim T_{cCG} + \log_2 (N/P)^{1/3}\, \alpha\, t_a$

Plus coarse-grid solve:

- — Std. "fast" coarse-grid solve: $T_{csd} \sim 2 \log_2 P\,(\,1 + P/m_2\,)\, \alpha\, t_a$
- — $A_c^{-1} = XX^T$ coarse-grid solve: $T_{cXX} \sim 2 \log_2 P\,(\,1 + 2.5/m_2\ P^{2/3}\,)\, \alpha\, t_a$

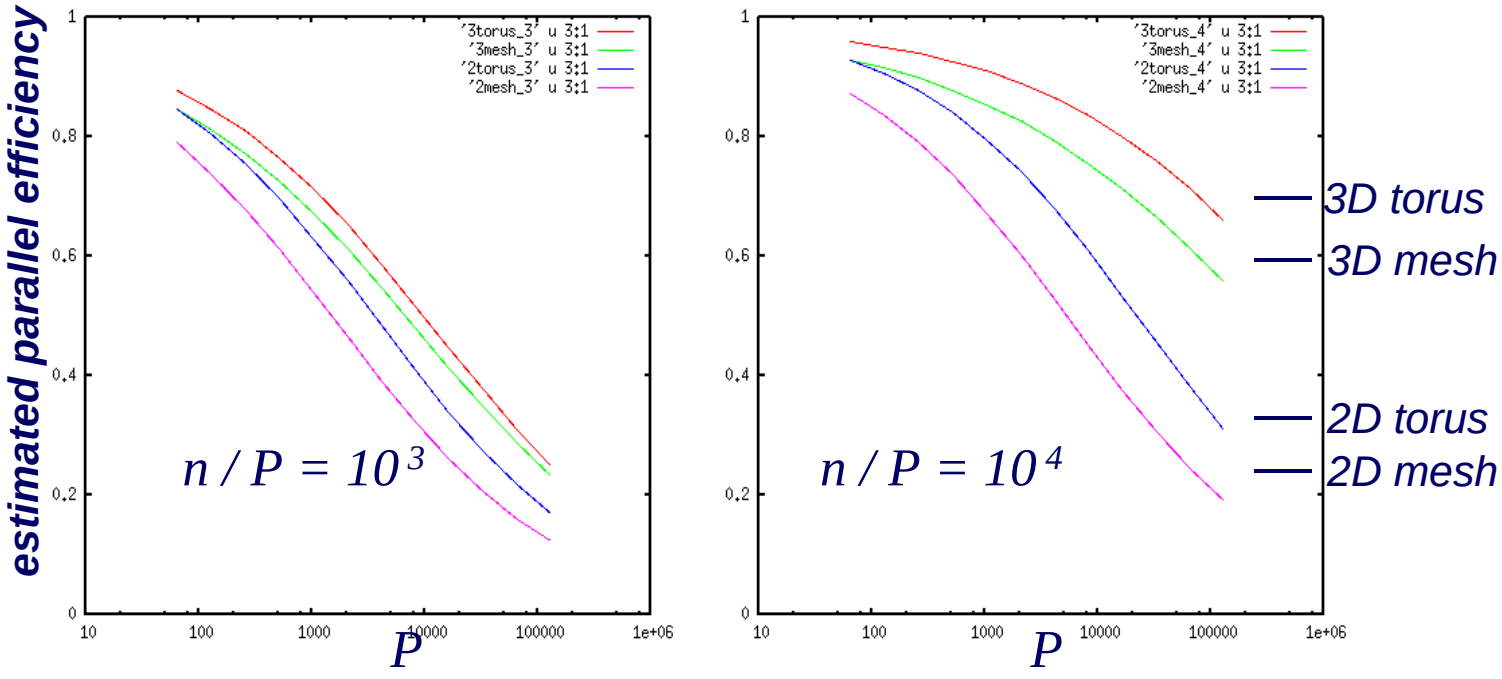
(Tufo & F, JDPC 01)

Mathematics and Computer Science Division, Argonne National Laboratory

Surprise!

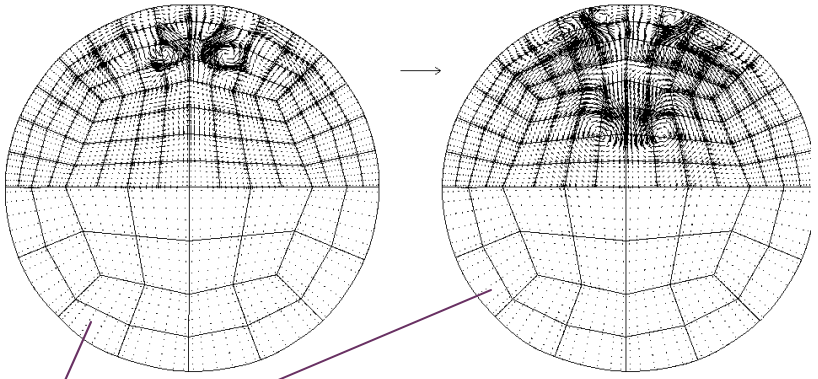
- All-to-all (e.g., ***global FFT***) based schemes not so bad, provided…

  rich enough interconnect network
- *3D is rich enough, 2D is not.*
- *Take home message –* ***No need for a lot of hand wringing over occasional all_to_all*** *(at least, not for now)*

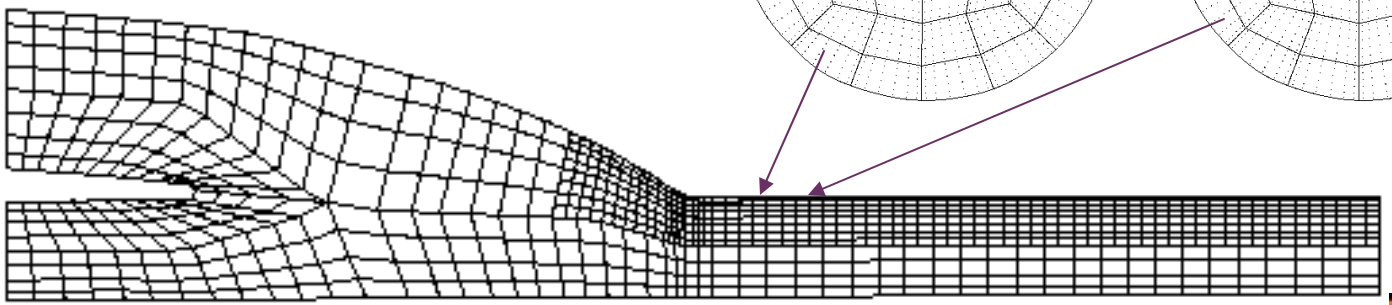


A Domain Decomposition Example: Spectral elements for incompressible flow simulation

- Variational method, similar to FEM, using *GL* quadrature.
- Domain partitioned into E high-order quadrilateral (or hexahedral) elements (decomposition may be nonconforming - *localized refinement*)
- Trial and test functions represented as N th-order tensor-product polynomials within each element. (N ~ 4 -- 15, typ.)
  - Fast local operator evaluations (***low memory, mat-mat product based***)
- Converges *exponentially* with N
- $n \sim EN^3$ gridpoints in 3D



3D nonconforming mesh for arterio-venous graft simulations: E = 6168 elements, N = 7



Incompressible Flow Simulations
Pressure Poisson Solve: $A\underline{p}^n = \underline{g}^n$

- Intrinsic to the incompressible (or low-Mach number) model
  - *elliptic solve at each step*
  - *multilevel solver required → parallel coarse grid solve*
- The matrix *A* is SPD and evaluated in matrix-free form:
  - never form the global stiffness matrix
  - never form the local stiffness matrix
    - *storage:* $O(N^3)$ *vs* $O(N^6)$
    - *work:* $O(N^4)$ *vs* $O(N^6)$

Scalable Gather-Scatter Communication Kernel

- Spectral element coefficients stored on element basis ($\underline{u}_L$ not $\underline{u}$)

$$\underline{w} = A\underline{x} = Q^T A_L Q\underline{x}, \qquad \underline{w}_L := Q\underline{w}, \qquad \underline{u}_L := Q\underline{u}$$

$$\underline{w}_L = QQ^T A_L \underline{u}_L$$

local work (matrix-matrix products)

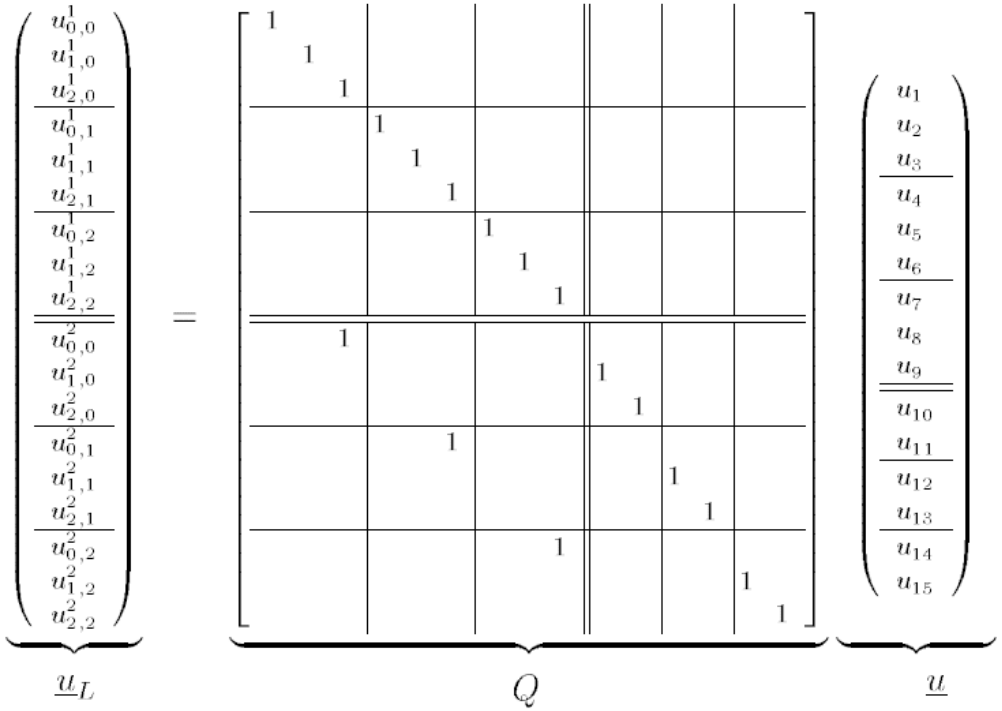
nearest-neighbor (gather-scatter) exchange

$$A_L := \begin{bmatrix} A^1 & & & \\ & A^2 & & \\ & & \ddots & \\ & & & A^E \end{bmatrix}$$

- Decouples complex physics (A_L) from communication (QQ^T)

Example of Q for E=2

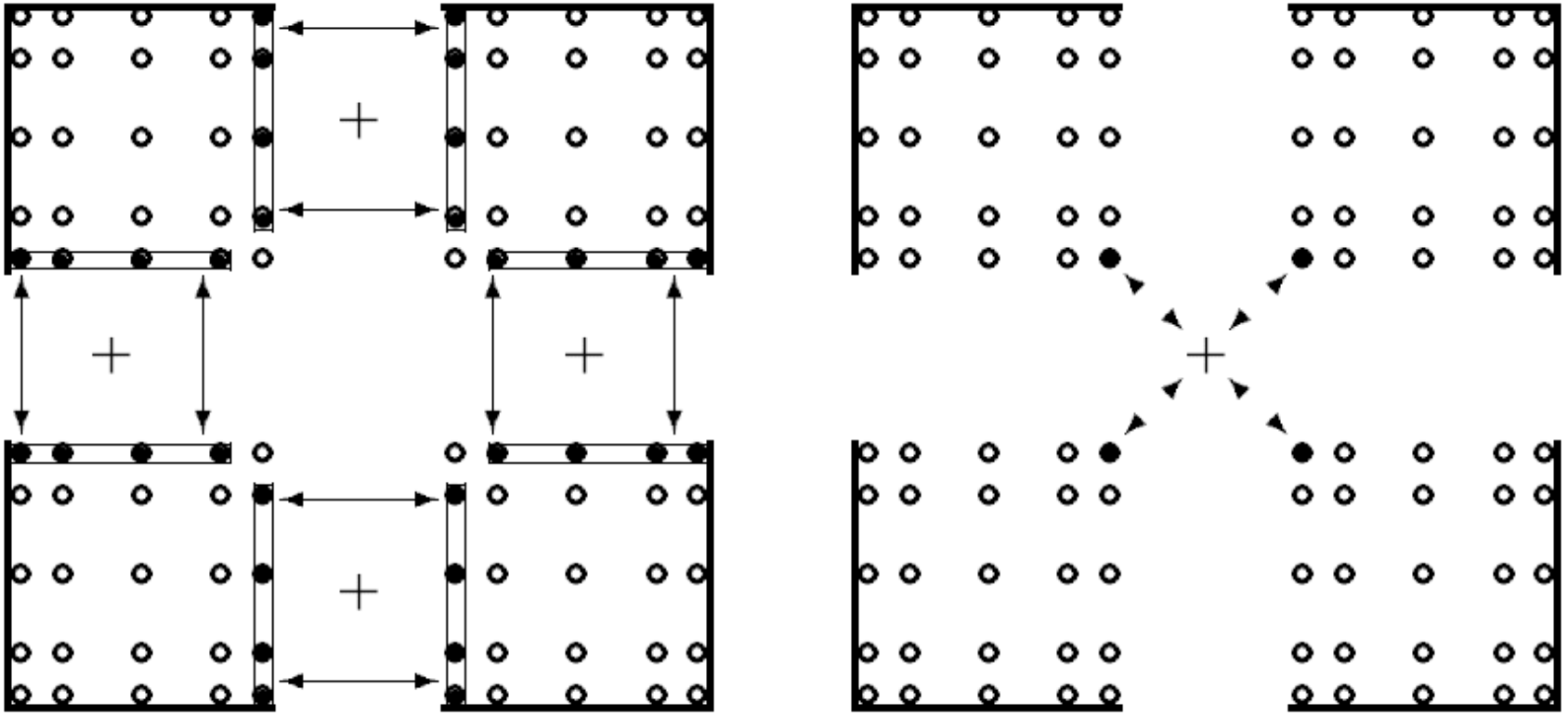
$$\underline{u}_L = Q\underline{u}$$



Communication is required, and the communication pattern must be established a priori (for performance)

→ set-up phase, ***gs_setup()***, and excecute phase, ***gs()***

QQ^T Pictorially
(gather-scatter or direct-stiffness summation)



Central Kernel: General Purpose Gather-Scatter

- Handled in an abstract way. Given index sets:

  proc 0: *global_num* = { 1, 9, 7, 2, 5, 1, 8 }

  proc 1: *global_num* = { 2, 1, 3, 4, 6, 10, 11, 12, 15 }

  On each processor: ***gs_handle = gs_setup(global_num,n,comm)***

- In an *execute()* phase, exchange and sum:

  proc 0: $u = \{ u_1, u_9, u_7, u_2, u_5, u_1, u_8 \}$

  proc 1: $u = \{ u_2, u_1, u_3, u_4, u_6, u_{10}, u_{11}, u_{12}, u_{15} \}$

  On each processor: ***call gs(u,gs_handle)***

Central Kernel: General Purpose Gather-Scatter

- Simple, lightweight, fast, general, not error prone.
  - Handles arbitrary Boolean QQ^T, Q, or Q^T
  - ***Supports 64-bit index sets*** (!)
  - QQ^T supports arbitrary associative/commutative operatrors (+,*,min,max)
  - Being using in a variety of codes (Nek5000, NekCEM, MOAB, others,…)
  - ***gs_setup*** requires a ***disovery phase***:
    - *For every global index* $\mathbf{i}$ *on proc.* $\mathbf{p}$*, find all procs* $\mathbf{q}$ *that also have* $\mathbf{I}$

  This was restrictive in the past… (90 minutes setup time on P=8192)

Discovery Phase: scalable gs_setup()

- ***all_to_all*** required, send index ***i*** to proc. ***p := mod (i,P)***

- **crystal_router()** exchange of Fox et al. (1988):
  - *For all $p < P/2$, if p has data needed by any processor $q > P/2-1$, send to processor $p + P/2$.*
  - *All processors $p > P/2-1$ reciprocate.*
  - *Divide processor set in half and recur on subsets.*

- **properties**:
  - *$\log_2 P$ messages – not 100,000 messages*
  - *potentially taxes bisection bandwidth of the network*
    - but not likely, based on our earlier analysis for 3D interconnect networks

      3D or richer interconnect is necessary and sufficient

Performance: *gs_setup()* and *gs()*

- Problem size: E=360K, N=11, n=471 million, n_{suface} = 120 million

P	n_unique shared	*gs_setup* time (s)	*gs()* pairwise max time	*gs()* crystal max time
16384	53687932	1.5159	0.00160	0.00821
32768	66734284	0.9700	0.00164	0.00592
65536	80216148	0.6208	0.00116	0.00414
131072	93440680	0.4615	0.00124	0.00392

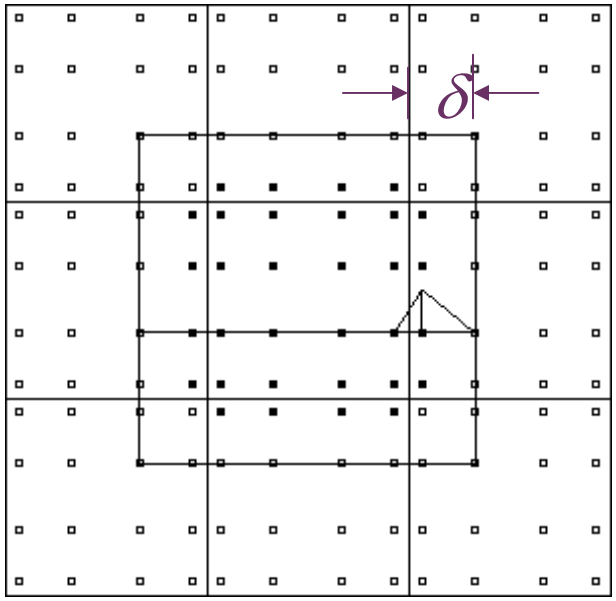
- ***gs_setup()*** requires three calls to cr(), plus 10 timing executions of each exchange strategy to identify the fastest. (more on this later…)

- Setup times of ~0.5 second, for all to all on 131000 processors.
  - Very tolerable overhead. Suitable for adaptive meshing.

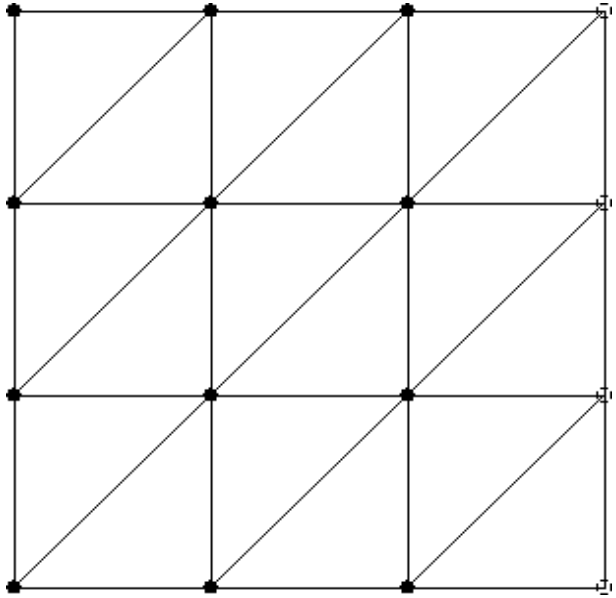
Coarse-Grid Solver Developments

Pressure Solve: $A\underline{x}^n = \underline{b}^n$

- P-type MG preconditioning GMRES,
  - using additive overlapping Schwarz for smoother
  - plus AMG for scalable coarse grid solve
  - many right hand sides



***Local Overlapping Solves**: FEM-based Poisson problems with homogeneous Dirichlet boundary conditions, A_e .*

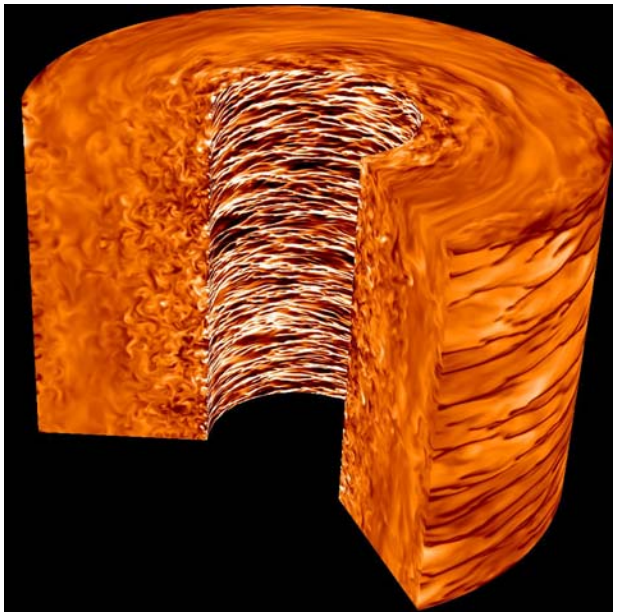


***Coarse Grid Solve**: Poisson problem using linear finite elements on entire spectral element mesh, A_0 (GLOBAL).*

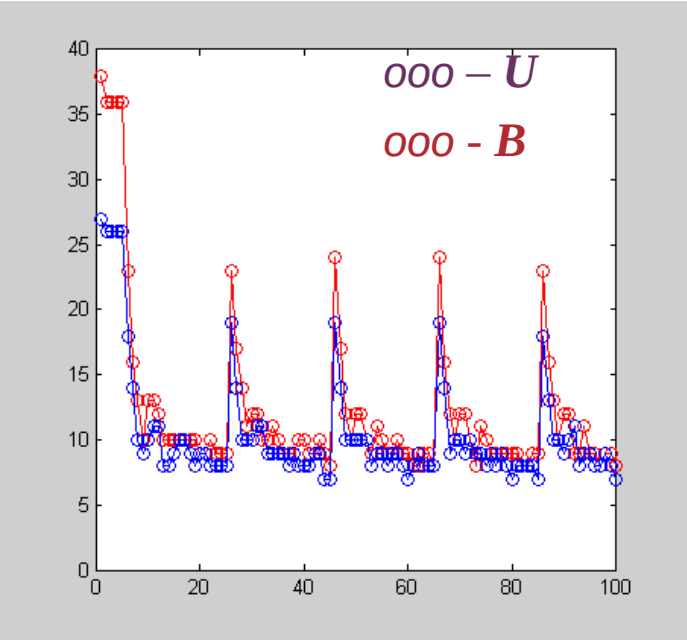
Solver Performance: hybrid-Schwarz/MG

(Lottes & F 05)

- *Magneto-rotational instability* *(Obabko, Cattaneo & F.)*
  - *E=140000, N=9 (n = 112 M), P=32768 (BG/L)*
  - *~ 1.2 sec/step*
  - *~ 8 iterations / step for U & B*
  - *Key is to have a fast coarse-grid solver*



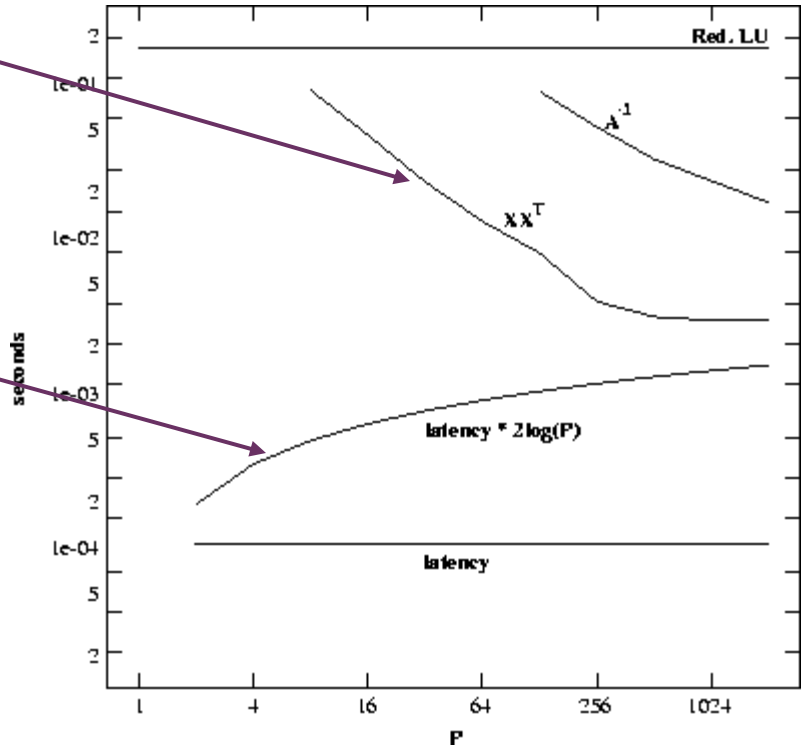
Iterations / Step



XX^T Coarse Grid Solver Timings: 127^2 Poisson Problem on ASCI Red

- *XX^T- approach projects solution onto sparse basis.* *(Tufo & F 01)*
  - *$O(n^{5/3} / P)$ work*
  - *$O(n^{2/3})$ $\log_2 P$ comm.*
  - *Only 2 $\log_2 P$ messages*
- *latency * 2 log P* curve is best possible lower bound
- Fine for P ~ 10,000

Coarse-Grid Solve Times



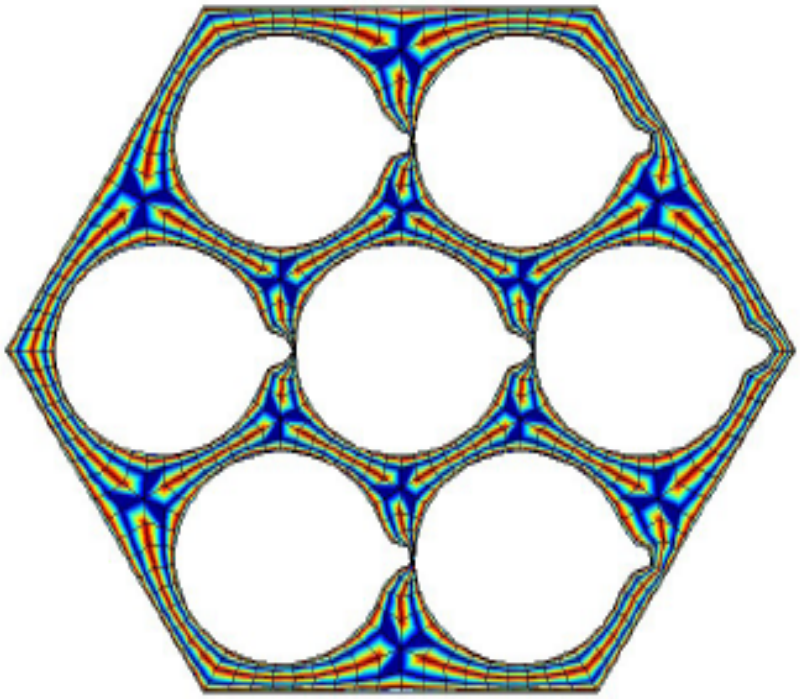
n=16129, 2D Poisson problem

AMG Coarse-Grid Solver

James Lottes (ANL / Oxford)

- Uses coarse/fine (C-F) AMG
  - *C-points selected to eliminate max. Gerschgorin disks of* $D^{-1/2}AD^{-1/2} - I$
- Energy minimal prolongation weights *(Chan, Wan, Smith)*

  $W \sim - A_{ff}^{-1} A_{cf}$
- Diagonal smoothing on F points only, with Chebyshev acceleration
- AMG automatically identifies proper semi-coarsening
- Communication exploits ***gs()*** library



coarse (red) and fine (blue) points

AMG vs. XX^T Performance

Solution time break down for n=120 M.

Case/P	Total	QQ^T	Coarse	all_reduce()
x4096	1994	125	1180	1.2
a4096	1112	125	192	1.4
b4096	846	126	25	1.
8192	460	88	22	1.
16384	266	64	20	1.

- Cannot consider XX^T on larger problems.
- "a4096" case is relies on pairwise + all_reduce
  - First version, pairwise-only, was not much faster than XX^T. Why?

Number of rows and nonzeros in AMG *(E=580,000)*

- Key observations:
  - n_{dofs} < P → idle some processors. OK.
  - Number of nonzeros does not drop as rapidly as number of rows
  - Stencil width grows at lower levels
    - → 100s of nonzeros per row
    - → More messages per processor
    - → Alternative message exchange strategy at lower levels.
    - → Rewrite ***gs()***

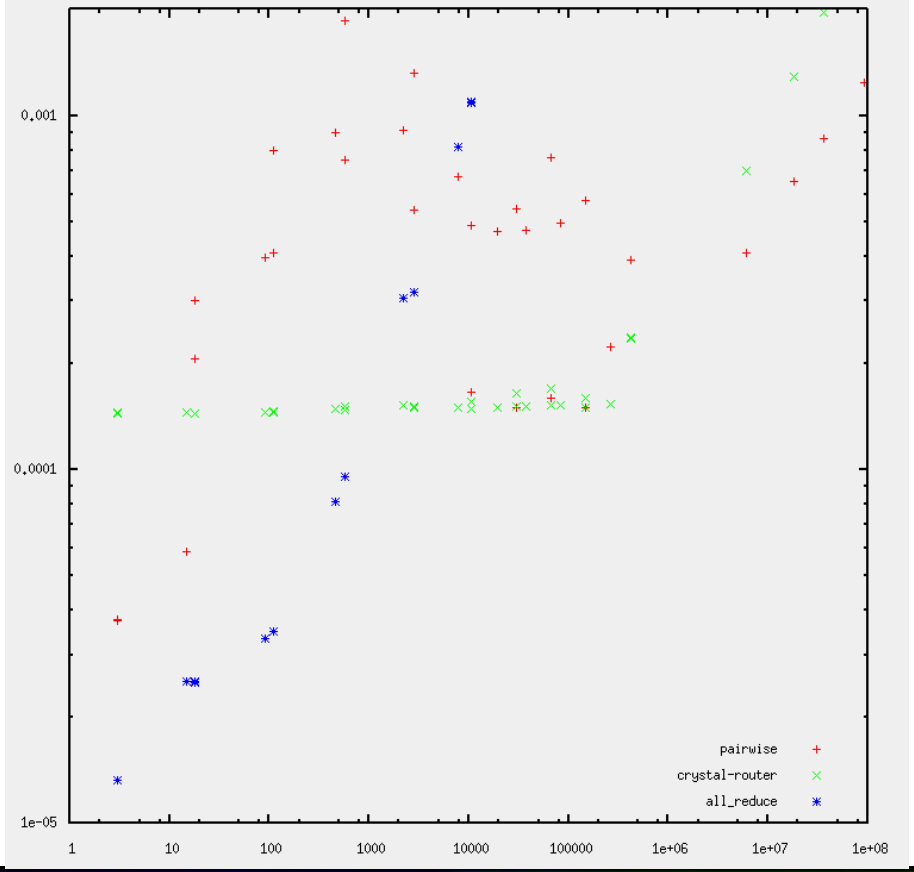
      3 exchange strategies:

      pairwise, all_reduce, cr()

Level	*n_{dofs}*	*nnz*
0	*665820*	
1	*304403*	*15668640.*
2	*204979*	*20863046.*
3	*96379*	*11293784.*
4	*38094*	*5095546.*
5	*16123*	*2051300.*
6	*4754*	*459490.*
7	*927*	*25760.*
8	*138*	*506.*
9	*18*	*20.*

gs() times – P=131K

- Red – pairwise, green – cr(), blue – all_reduce
- Horizontal axis – number of nontrivial (shared) columns in matrix
- cr() and all_reduce *> 5-10 X* faster in many cases



AMG vs. XX^T Performance

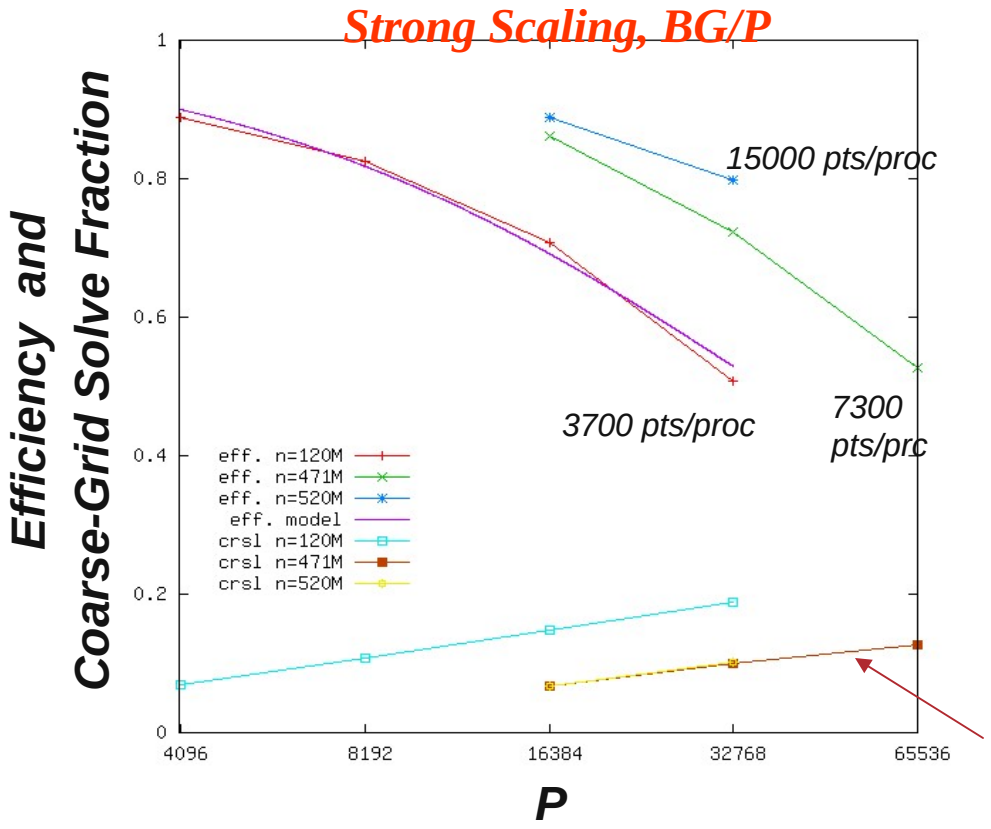
Solution time break down for n=120 M.

Case/P	Total	QQ^T	Coarse	all_reduce()
x4096	1994	125	1180	1.2
a4096	1112	125	192	1.4
b4096	846	126	25	1.
8192	460	88	22	1.
16384	266	64	20	1.

- ***50x speed up for AMG vs XX^T (2 x for total solution time)***
- ***Almost no time in vector reductions because of fast tree network***
- ***gather-scatter() is leading-order overhead***

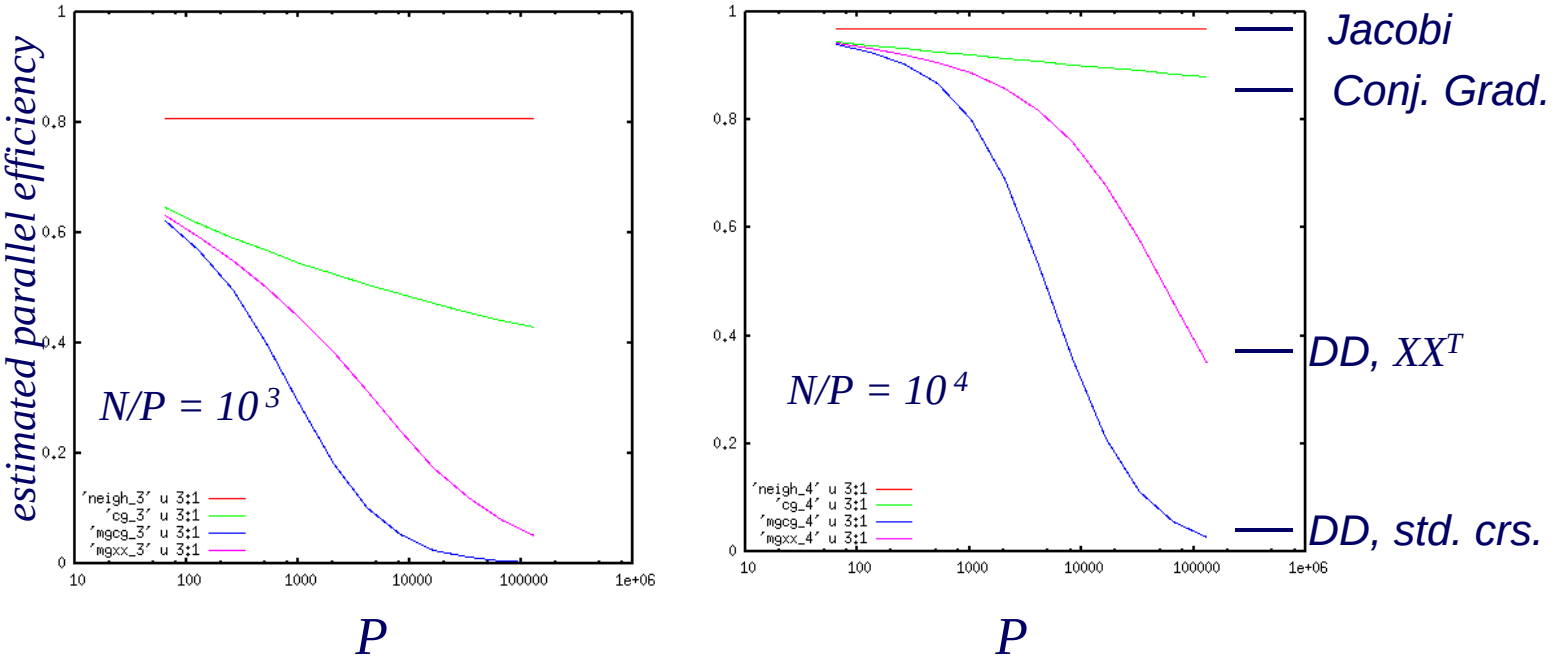
Putting It All Together

- Efficiency on P=65K ~ 50 % for n/P ~ 7000. Reasonable ?
- Back of the envelope computation of 2005 says Yes.



Coarse-grid solve time/total

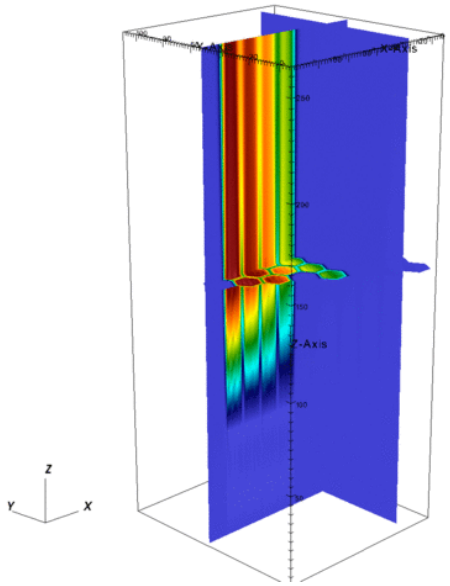
Nearest Neighbor Scaled-Speedup Models (05 workshop)



Summary and Path Forward

Summary: TH Modeling

- Turbulent entrance length established
- LES / RANS equivalence established for cross-flow velocity distributions
- Recent success of 37-pin analysis (2-3 weekends on P=16384) indicates that design configuration of 217 pins is within reach.
- Now using LES and validated RANS to provide base velocity inputs to high-fidelity sub-core coupled neutronics/TH simulations



Next Steps: TH Modeling

- Detailed analysis of 19- and 37-pin data – submit in Jan 09.
  - I/O and user intensive… Eureka now online.
- Simulation / analysis of 217-pin case and detailed comparison to reference experiment
- Coupled TH/neutronics with detailed flow distributions in whole-core model
- Core-scale upper plenum analysis of thermal striping phenomena
  - Boundary conditions have a profound influence → core scale required.

Summary: Computational Science

- Flexible and lightweight ***gs()*** communication utility is enabling petascale deployment of many codes: Nek5000, NekCEM, MOAB, AMG,…

- New AMG coarse-grid solver has overcome a major impediment to scaling beyond P=10,000.

  – Coarse-grid solves account for ~15% of CPU time at n/P ~ 5,000.

  – This behavior appears to scale, though more analysis is needed.

Next Steps: Computational Science

- Viz: a major problem
  - metadata and in situ running of VisIt are promising avenues to resolving this serious bottleneck.
  - Otherwise, we're going to need a ton of hardware.
    - *Our group has a dedicated 128-core cluster for reactor analysis.*
    - *A typical LES simulation will produce ~2 TB of data.*
    - *It takes a long time to analyze…*
    - *New territory for us because of the size of these problems –*

Next Steps: Computational Science

Battle Plan for a million cores: (2008—2017):

Straight MPI, no hybrid programming models

- *The clearest path to parallel memory access is through the distributed memory model.*
- *To date, straight MPI is often the most efficient path to multicore usage.*
  - Tufo & Fischer ’99, Mavriplis 06, Lin et al. (Sandia) 08,…
  - Even if a hybrid programming approach offers a 1.5x speedup, the lack of portability and stability would not warrant a major code rewrite
- *A radical change to programming model is only warranted through transformational paradigm shifts, e.g.,*
  - emergence of distributed memory parallelism in 80s
  - emergence of GPU-based clusters (now)