2004 NSF Division of Materials Research ITR Workshop

The Materials Computation Center, Duane D. Johnson and Richard M. Martin (PIs)

Funded by NSF DMR 03-25939

Computer Science and Scalable Parallel Methods for Materials Modeling

Hijacking the PS2 for Quantum Chemistry Supercomputing

B. Levine, T. J. Martinez, E. de Sturler, S. Patel,
Departments of Computer Science, Electrical and Computer Engineering, and Chemistry,
University of Illinois at Urbana-Champaign

Hijacking Game Consoles

Scientific computing began in mainframe-based centralized environments. These were replaced by workstations and finally over the last decade by "Beowulf" clusters of commodity computers. However, commodity computers, such as Intel and AMD-based machines, are general purpose machines and not necessarily ideal for materials simulation.

Recently, game consoles have begun to appear as a possible alternative where the economies of scale are increasingly favorable while the target applications are very similar to materials simulations.

Commodity-Off-The-Shelf (COTS) Computing

- *Utilizing clusters of standard PCs for high performance computing*
- *Economy of Scale – PC parts are built in large number so they cost less than proprietary high performance computers*

- *Ease of Upgrade – Backward compatibility is a must in the PC market, so the COTS strategy ensures easy upgrades without lengthy and difficult porting of codes.*

Commodity-Off-The-Toy-Shelf (COTTS) Computing

- *Utilizing clusters of video game consoles for high performance computing*
- *Economy of Scale – Game consoles are often sold near cost with profits coming from subsequent game software sales.*

- *Surprisingly, there is also a disparity in sales volume. Sony sold 60M PS2 consoles in 2002. Compaq, Dell and IBM together sold only 50M PCs in the same time.*
- *Performance – Modern video games involve the solution of differential equations (game physics) and linear algebra (graphic manipulations).*

Challenges

- *Complexity of Hardware – Game hardware is complex and proprietary just like a commercial workstation.*
- *Absence of software tools and libraries (e.g. BLAS)*
- *Limited Lifespan of Product – Companies only release new game consoles every several years, unlike PC hardware which is updated continuously.*

Sony Playstation 2 (PS2) Architecture

- *CPU is analogous to a typical PC processor*
- *VPUs are capable of 4 floating point multiply/accumulates per clock cycle*
- *VPU0 can act as a coprocessor to the CPU ('macromode') or work independently ('micromode')*
- *VPU1 only runs in micromode*
- *When in micromode VPUs run 'microprograms' of two simultaneous instruction streams*

Potential Performance

	PS2	PIII-600	P4-3200
Processor Clock (MHz)	300	600	3200
Floating Point Ops/Clock	20	2	2
MFLOPS	6000	1200	6400
System Bus Clock (MHz)	150	100	800
Bus Width (bits)	128	64	64
Data Transfer Rate (MB/s)	2400	800	6400
Memory (MB)	32	256	1024
Price (when first released)	$500	$2,000	$2,000

Performance of Game Consoles

Theoretical peak performance and actual performance can be very different things. Our preliminary work has focused on demonstrating that complex codes can be ported to the PS2 game console and determining what performance levels can be achieved in practice. Recently, we have begun collaborating with IBM to carry this work over to the new Cell processor which will be the foundation for Sony's PlayStation 3 game console. We plan to have codes ready to run at release, maximizing the price/performance advantages.

Linear Algebra Results

- *Data transfer is a key bottleneck. Fast transfer is possible using a specialized DMA, called the VIF (vector interface).*
- *An ATLAS-like strategy should be used to find transfer "sweet spots."*
- *Both VPUs should be used in parallel to optimize performance*
- *No hardware support for double precision*

Matrix-Vector Multiplication – VPU1 only, Micromode

Data Xfer to VPU	Calc?	Data Xfer from VPU	Time/s	MFLOPS	Data Xfer Rate (MB/s)
CPU	Yes	CPU	6.88	100.5	201.0
DMA/VIF	Yes	CPU	4.48	154.3	308.6
DMA/VIF	Yes	None	2.27	304.5	609.0
DMA/VIF	No	None	1.31	NA	1077.3
			Maximum Possible:	6000	2400

Matrix-Matrix Multiplication – VPU1 only, Micromode

Data Xfer to VPU	Calc?	Data Xfer from VPU	Time/s	MFLOPS	Data Xfer Rate (MB/s)
DMA/VIF	Yes	CPU	22.81	121.2	60.6
DMA/VIF	Yes	None	5.55	498.2	83.0
DMA/VIF	No	None	.53	NA	869.4
None	Yes	None	4.95	558.5	NA
			Maximum Possible:	6000	2400

Porting Complex Codes

- *We have ported the GAMESS code for quantum chemistry*
- *MPICH has been ported and a cluster of PS2s run in parallel*

GAMESS – 11 Steps of a Hartree-Fock Geometry Optimization of Butadiene – CPU only

machine	PIII-600		PS2	
precision	single	double	single	double
time (s)	6.0	7.1	31.1	534.3

Conclusions and Work in Progress

- *When first released the Sony Playstation 2 had the potential to outperform the most powerful personal computers on the market in terms of FLOPS.*
- *Because of the nature of the game console market video game machines cost only a fraction of what personal computers cost.*
- *We have utilized the PS2's advanced architecture to do linear algebra, successfully ported the GAMESS electronic structure software package to the PS2, and employed MPICH to distribute a computational task over a cluster of PS2s.*
- *We are applying these lessons to the new Cell processor in collaboration with IBM. Scientific are being ported and developed and will be ready when the PS3 is released.*

Solving Sequences of Linear Systems

E. de Sturler, Duane D. Johnson, Philippe Geubelle, Robert H. Dodds, Misha Kilmer, Mike Parks, and Ben Kaap
Departments of Computer Science, Materials Science, Aerospace Engineering, and Civil and Environmental Engineering
University of Illinois at Urbana-Champaign, and Department of Mathematics, Tufts University

Solving Sequences of Linear Systems

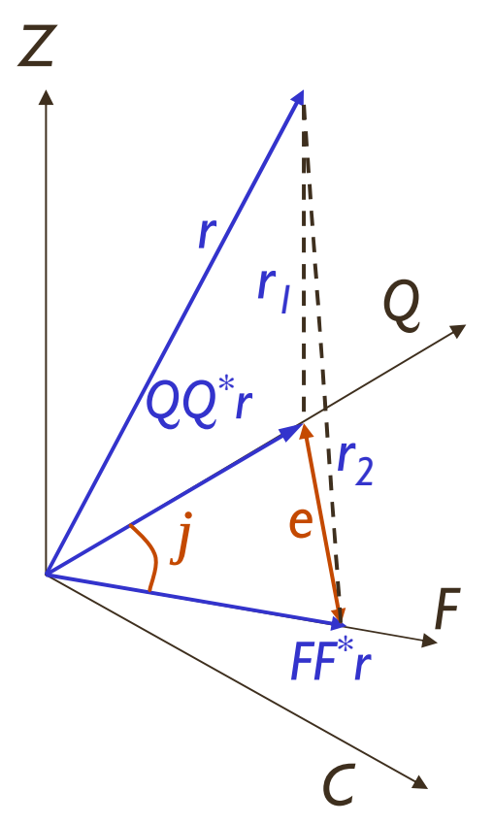
In many simulations in science and engineering we must solve a large sequence of closely related linear systems. In some cases only the right hand side changes, in others both the matrix and right hand side change, and in yet others, only the matrix changes. Applications range from the approximation of Green's functions in electronic structure, simulation of crack propagation in solids and structures, to fatigue, and nonlinear optimization problems in image restoration.

Here we adapt techniques we developed for fast and robust linear solvers for a single linear system to sequences of linear systems.

1. Linear Solvers

- *GMRES is the most robust method for linear systems, but becomes very expensive if convergence is not rapid.*
- *Restarting often leads to significantly slower convergence*
- *So, restart while recycling selected subspace from previous steps*
  - *Select recycle space to minimize loss of orthogonality*
  - *Select recycle space to approximate invariant subspace*
- *Often leads to convergence close to optimal but with reduced cost.*
- *If linear system changes slowly we can recycle selected subspace for subsequent linear system and improve convergence from the start*
- *Allows the method to learn while going from one system to the next, dealing effectively with slowly changing systems*

Minimizing Loss of Orthogonality *(Sturler'99)*



$range(Q) \perp range(C)$ in $range(C) \oplus range(F)$

$range(C)$, $range(Q)$, $range(Z)$ orthogonal

$r \perp range(C)$

$r_1 = (I - QQ^*)r$ optimal residual

$r_2 = (I - FF^*)r$ optimal residual ignoring C

The difference is determined by canonical angles j_k *between* $range(C)$ *and* $range(Q)$.

Force r_2 *closer to* r_1 *by maintaining orthogonality to selected subspace of* $range(C)$, *removing large* j_k.

GCRODR *(Parks, Sturler, et al. '04)*

Consider the following bound on the residual after m iterations

$$\min_{z \in K^m(A, r_0)} \|r_0 - Az\| = \min_{p_m(0)=1} \|p_m(A) r_0\| \le \|r_0\| \|U\| \|U^{-1}\| \min_{p_m(0)=1} \max_{\lambda \in \Lambda(A)} |p_m(\lambda)|$$

If condition number of U *not too large, reduce bound by removing eigenvalues that make* $\min_{p_m(0)=1} \max_{\lambda \in \Lambda(A)} |p_m(\lambda)|$ *large.*

These are often, but not always, small eigenvalues. Remove these by augmenting Krylov space with corresponding approximate eigenvectors.

For strongly nonhermitian problems this approach is dubious.

Implemented in GMRESDR (Morgan'03). This method cannot use recycling; hence, alternative proposed based on GCRO (Sturler'96), GCRODR.

2. KKR Electronic-Structure Method

KKR-CPA code in k-space and r-space *is an* $O(N^{1+\varepsilon})$ *DFT - Green's Function method with:*

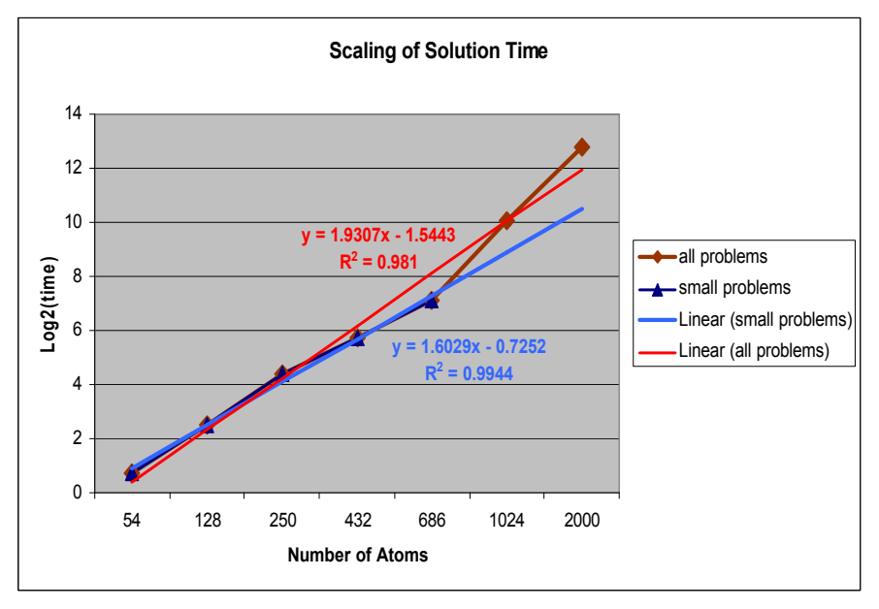
- *Screened structure constants (sparse matrices)*
- *Sparse matrix techniques (memory & size)*
- *Iterative inversion techniques (size)*
- *Parallel computation on E-points and/or N atoms*
- *Coherent-potential approximation for chemical disorder*

Current scaling behavior (in metals)

For SCF Total Energy:	$O(N^{1+\varepsilon})$	ε_1 ˜ 0.1-0.5
For DOS calculations:	$O(N^{1+\varepsilon})$	ε_2 ˜ 0.5-1.0
For Bands:	$O(N^{2+\varepsilon})$	$\varepsilon_3(k)$ ˜ 0.5-1.0

Memory scales as $NM(L+1)^4$, *with M atoms in screening cluster (2-6 n.n. shells) and maximum angular momentum, L (2-4)*

Results

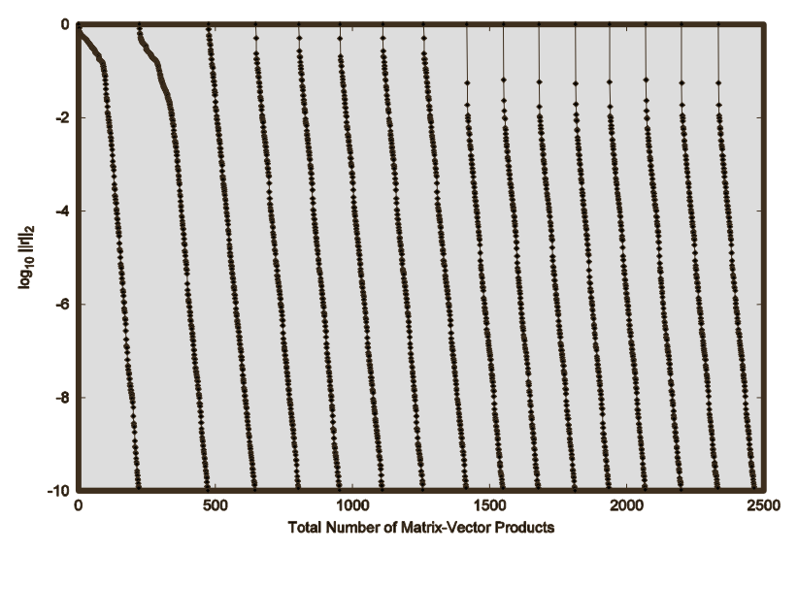


Scaling for 54 to 2000 atoms (Cu) for complex energy near the real axis.

Time for largest two ensembles can be improved using larger subspace (to do).

Gives much better scaling than direct methods.

Subspace recycling will improve solution time further.



Solution of 16 right hand sides corresponding to a single atom (54 atoms) using subspace recycling.

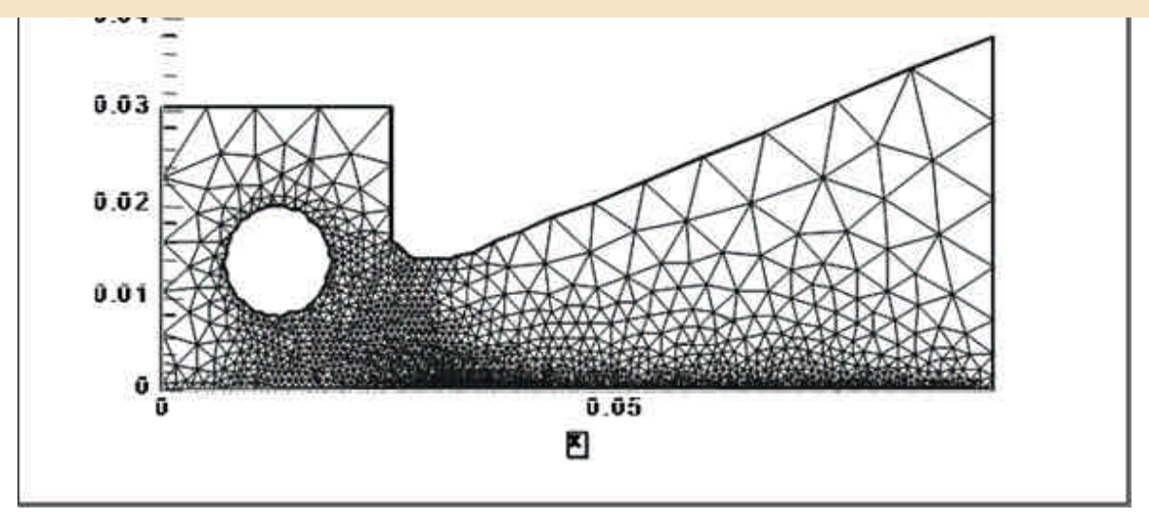
Each separate curve indicates convergence for 1 rhs.

Note the improved convergence rate for later right hand sides (roughly factor two).

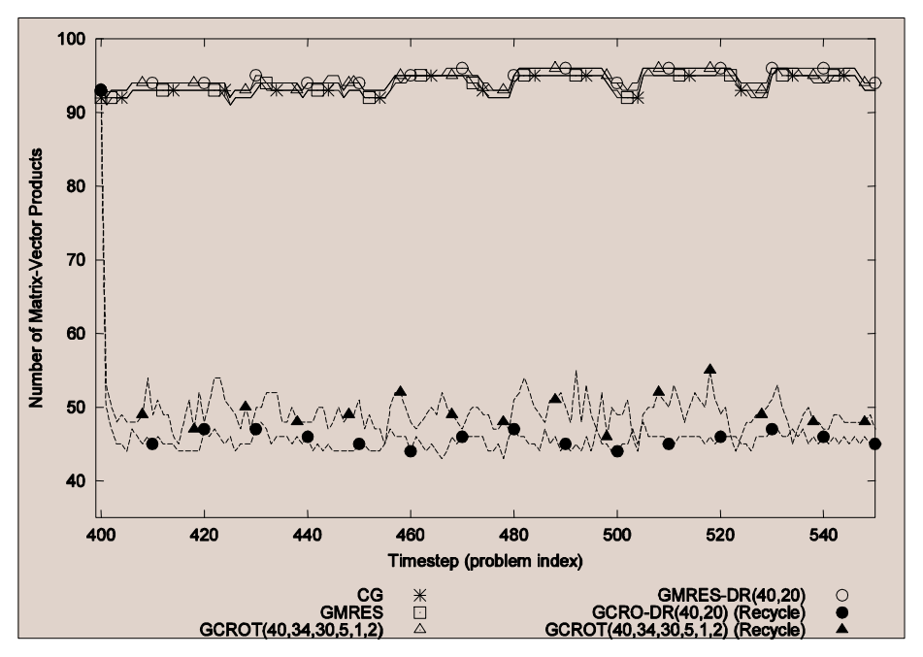
Crack Propagation

Metal plate pulled apart at top left corner, breaking along the symmetry axis, which coincides with the x-axis. Along fracture line cohesive finite elements are used to model nonlinear response and breaking.

Thousands of loading steps are needed to model crack propagation.



Results with and without Subspace Recycling



Number of matrix-vector products to solve linear system per loading step with and without subspace recycling for 150 consecutive loading steps.

Systems are symmetric positive definite, with IC(0) preconditioner

Diffuse Optical Tomography

Collaboration with Misha Kilmer (Tufts University)

Reconstruct 3D absorption and scattering information by matching solutions from a parameterized model to measured data.

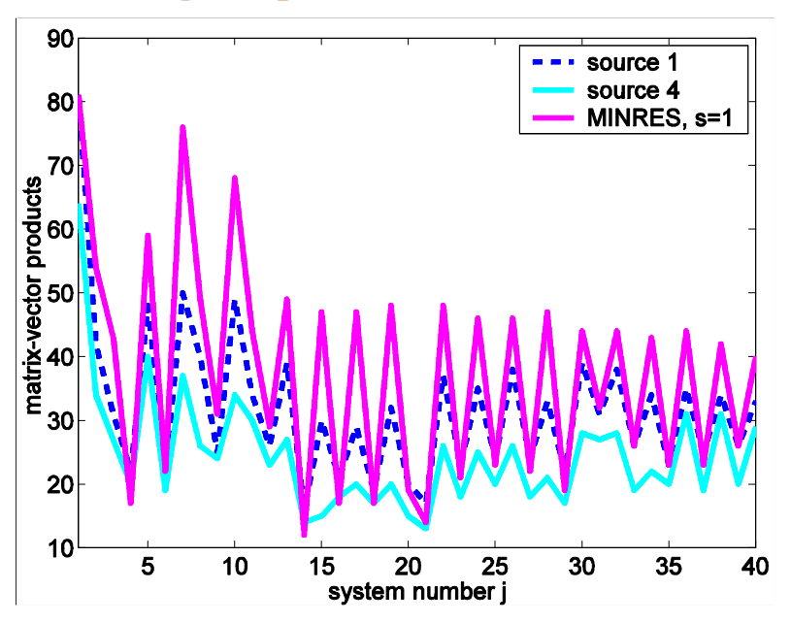
Gauss-Newton with line search for nonlinear least squares.

Each nonlinear step requires the solution of $\left(A(p_j) + i g I\right) x_{s,g}^{(j)} = b_s$ *for multiple shifts and right hand sides. Each nonlinear step the matrix* $A(p_j)$ *changes.*

Carefully tune the subspace to be recycled to the phases of the nonlinear optimization algorithm.

Recycle both invariant subspaces and selected subspaces of previous solutions.

Results for Tomography



Number of matrix-vector products to solve linear system for 40 nonlinear steps for sources 1and 4 using subspace recycling compared with MINRES using an appropriate previous solution as starting guess.

After source 4, other sources have number of matrix-vector products similar to source 4. All shifts are solved with a single Krylov subspace.

Additional Applications

Several other applications are planned:

- *Topology optimization for tailoring functionally graded materials and structures (with Glaucio Paulino, CEE),*
- *Computational Electromagnetics (with Eric Michielsen, ECE),*
- *Reduce occasional extreme ill-conditioning in multiple spawning method (Todd Martinez, Chemistry)*

Conclusions

Large sequences of linear systems ubiquitous in computational science and engineering.

Obtained significant improvement in convergence rate and solution time.

Developed for use in electronic-structures (KKR) code, but useful in many other applications. Effect on scalability in KKR must be evaluated.

Combined with block method to increase versatility. Testing for optimal combination is future work (application dependent).