

**IMPLEMENTATION AND PERFORMANCE ANALYSIS
OF A PARALLEL MULTICOMPONENT GROUNDWATER
TRANSPORT CODE**

*G. (Kumar) Mahinthakumar
Center for Computational Sciences
Oak Ridge National Laboratory*

and

*Faisal Saied
National Center for Supercomputing Applications
University of Illinois at Urbana—Champaign*

March 24, 1999

OUTLINE

● Background

- Main features of transport code
- Contaminant Transport Equation

● Implementation Details

- Numerical Implementation
- Parallelization

● Results

- Cross platform comparison
- Parallel scalability and efficiency
- Cache effects on single node performance

● Conclusions

MAIN FEATURES OF TRANSPORT CODE

- **Multicomponent 3D advection – dispersion equation with first order kinetic sorption, bioremediation and reversible kinetic (or ion exchange) reactions.**
- **Components can be either mobile or immobile but matrix solves are required only for the mobile species.**
- **Generalized biodegradation with multiple – electron acceptors and multiple substrates. Special option available to handle cometabolic biodegradation of chlorinated organics.**
- **Equilibrium speciation reactions, multiple bacterial populations, precipitation/dissolution, and colloid transport are not yet implemented.**
- **Mainly a research code but applications include field simulations, laboratory column simulations, computational steering, stochastic studies, and inverse problems.**

TRANSPORT EQUATIONS

$$\frac{\partial C_i}{\partial t} = \nabla \cdot (\mathbf{D} \cdot \nabla C_i) - \nabla \cdot (C_i \mathbf{v}) + \frac{q}{\theta} (C_T - C_{0i}) + R_i \quad i = 1, 2, 3, \dots, nc$$

**MULTICOMPONENT
TRANSPORT**

$$D_{ij} = \alpha_L |\mathbf{v}| \delta_{ij} + (\alpha_L - \alpha_T) \frac{v_j v_i}{|\mathbf{v}|} + D_m \quad i, j = x, y, z$$

**DISPERSION
TENSOR**

$$R_i = E_i + B_i - (\lambda_i + \lambda_{wi}) C_i - \frac{\rho}{\theta} \frac{\partial S_i}{\partial t} \quad i = 1, 2, 3, \dots, nc$$

REACTIONS
(reversible, biodegradation, decay, sorption)

$$E_i = (\eta_i - \eta'_i) \left[k_f \prod_{j=1}^{j=nc} C_j^{\eta_j} - k_r \prod_{k=1}^{k=nc} C_k^{\eta'_k} \right] \quad i = 1, 2, 3, \dots, nc$$

**REVERSIBLE
ION EXCHANGE**

$$B_i = -\mu_{\max} F_i X \prod_{j=1}^{j=nc} f_{ji} \left(\frac{C_j}{K_j + C_j} \right) \quad i = 1, 2, 3, \dots, nc$$

BIODEGRADATION

$$\frac{\partial X}{\partial t} = \mu_{\max} X Y \prod_{i=1}^{i=nc} g_i \left(\frac{C_i}{K_i + C_i} \right) - bX$$

**BIOMASS
GROWTH AND DECAY**

$$\frac{\partial S_i}{\partial t} = k_{ri} (K_{di} C_T - S_i) - (\lambda_i + \lambda_{si}) S_i \quad i = 1, 2, 3, \dots, nc$$

SORPTION

SOME PERFORMANCE FEATURES OF TRANSPORT CODE

- **Largest problem we have solved**
 - Single component transport: 40 million grid cells
 - Multicomponent transport: 120 million degrees of freedom (20 million grid cells with 6–component reactive transport)
 - Required about 5–10 seconds per timestep on 1024 processors of Intel Paragon XPS/150
- **Commonly we solve 1–2 million cells problems**
 - Takes 1–2 seconds per time step for most of these problems
- **When nonlinear reactions are involved timings are usually higher since each non–linear SLA iteration requires a full matrix solve for each component**
 - For example, a 6 component transport bioremediation problem using 1.0 million cells (6.8 million unknowns) takes about 4 seconds per time step to solve using 64 processors of an IBM SP.

NUMERICAL IMPLEMENTATION

- Galerkin finite–elements with 8–node hexahedral elements
 - 27–point stencil → 27–diagonal non–symmetric matrix
- Logically rectangular grid structure (with “natural ordering” of nodes and elements)
- Irregular geometries supported by distorted elements.
- Krylov solvers (diagonal or block ILU preconditioning for all solvers) (**Mahinthakumar et al., 1997**)
 - BICGSTAB
 - GMRES(m)
 - ORTHOMIN(k)
 - CGS
- Non–linearity handled by a Sequential Iterative Algorithm (SIA) (**GS99**)

COUPLED MATRIX SYSTEM

$$\begin{array}{|c|} \hline A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline \end{array}
 \begin{array}{|c|} \hline A_{12} \\ \hline A_{22} \\ \hline A_{32} \\ \hline \end{array}
 \begin{array}{|c|} \hline A_{13} \\ \hline A_{23} \\ \hline A_{33} \\ \hline \end{array}
 \begin{array}{|c|} \hline x_{11} \\ \hline x_{1n} \\ \hline x_{21} \\ \hline x_{2n} \\ \hline x_{31} \\ \hline x_{3n} \\ \hline \end{array}
 =
 \begin{array}{|c|} \hline b_{11} \\ \hline b_{cn} \\ \hline b_{21} \\ \hline b_{2n} \\ \hline b_{31} \\ \hline b_{3n} \\ \hline \end{array}$$

Component 1
Component 2
Component 3

- A lumped formulation is used for all zeroth derivative and time derivative terms including non-linear reaction terms
 - this in turn greatly simplifies the implementation by making the off-diagonal coupling blocks to be unit diagonal
- We fully decouple the system to solve for each component independently. Iterations are performed to resolve the fully coupled system.

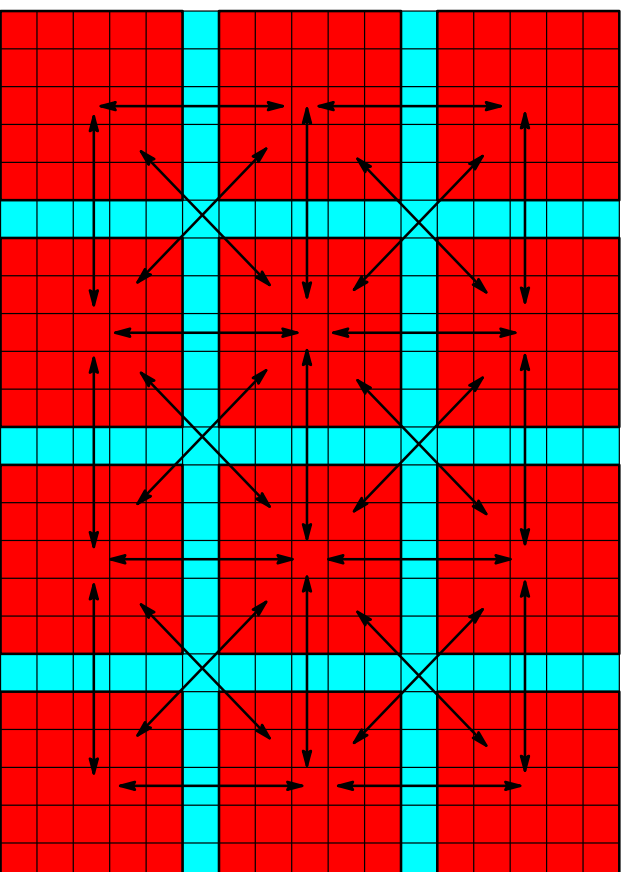
MEMORY AND COMPUTATIONAL SAVINGS IN MULTICOMPONENT TRANSPORT

- Only changing matrix entries are updated during assembly of global matrix during each time step
 - when the flow field is steady only the diagonal entries of the matrix changes (due to reactions) each time step
 - in a majority of groundwater problems the flow field is either steady or the flow time step is 10 – 100 times larger than the transport time step
- Matrix entries common to all components are not duplicated in storage
 - for single phase flow all components use the same velocity field
 - the dispersivity values are usually same for more than 1 component if not all components.
- These features have one minor drawback
 - we cannot take advantage of high Mflop dense block operations usually associated with vector p.d.e's (multicomponent transport) since we have to solve each component in a decoupled manner
- But benefits due to computational and memory savings far outweigh slight degradation in Mflop performance

PARALLEL IMPLEMENTATION

- **Two-dimensional domain decomposition**
 - well suited for groundwater applications where the vertical dimension is much smaller than the lateral dimensions
 - communication with at most 8 neighboring processors
 - natural node ordering for individual processor regions
- **Explicit message passing required to exchange information at processor boundaries especially during assembly and matrix-vector product stages**
 - NX or MPI communication library for Intel architectures
 - MPI for other architectures
- **Recently Open MP directives incorporated into the solver portion of the code to use the code in a pure MPI, pure Open MP, or hybrid mode**
 - But preliminary experiments on an Origin 2000 suggest more tuning required to use Open MP.
- **Portable parallel binary I/O for all grid-scale reads and writes including restart.**
 - Options to use either MPI-IO or single processor I/O

PARALLEL DOMAIN DECOMPOSITION



■ overlapping processor regions

■ individual processor regions

(arrows show communication pattern)

Plan View of Two-Dimensional Domain Decomposition
(showing a 4x3 processor decomposition)

PARALLEL ARCHITECTURES USED IN THIS STUDY

- **Intel Paragon XPS/150 (CCS – ORNL) – 1024 nodes**
 - 1024 nodes with 64 Mb/node
 - 150 or 75 Mflop/node peak (dual or single threaded mode)
- **Origin 2000 (NCSA)**
 - 128 nodes (max allowed) with 256 Mb/node
 - 500 Mflop/node peak
- **IBM SP (Argonne)**
 - 64 nodes (max allowed) with 128 Mb/node
 - 120 Mflop/node peak
- **Cray T3E (NERSC)**
 - 512 nodes total with 256 Mb/node
 - 900 Mflop/node peak
- **Convex Exemplar (NCSA)**
 - 64 nodes total (max 32 allowed at one time) with 128 Mb/node
 - 360 Mflop/node peak

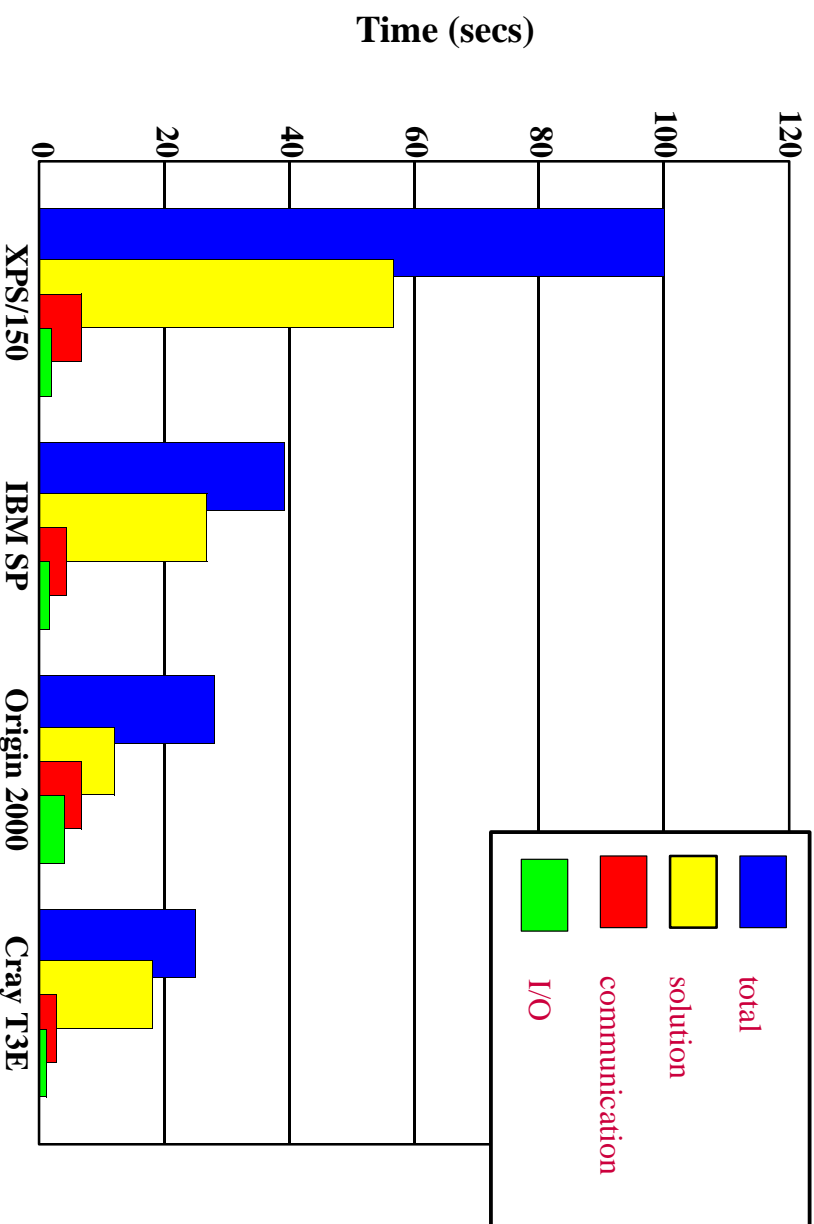
PERFORMANCE RESULTS

- Overall scalability
- Cache effects on single – node performance
- Solver comparisons and performance [Mahinthakumar et al., 1997]
 - Compared relative performances of BICGSTAB, GMRES, CGS, and ORTHOMIN for different problems on a variety of parallel architectures
- Parallel I/O implementation and performance [Mackay et al., 1998].

TEST PROBLEM FOR MULTICOMPONENT TRANSPORT

- Six component bioremediation field test outlined in **Sempini and McCary (1992)**.
- Homogeneous K–field with essentially 1–D flow.
- The components are Methane, Dissolved Oxygen, TCE, VC, t–DCE, and c–DCE.
- Four of the 6 components involved kinetic sorption reactions.
- Periodic injection using third–type boundary conditions.
- Results were first verified with 1–D finite–difference solutions reported by **Sempini and McCary (1992)** **simple** for a coarse–grid model.
- For the scalability tests problem size is changed by increasing domain size (not representative of field conditions).
- Parallel I/O limited to final concentration field output of each component.

COMPARISON AMONG PARALLEL PLATFORMS

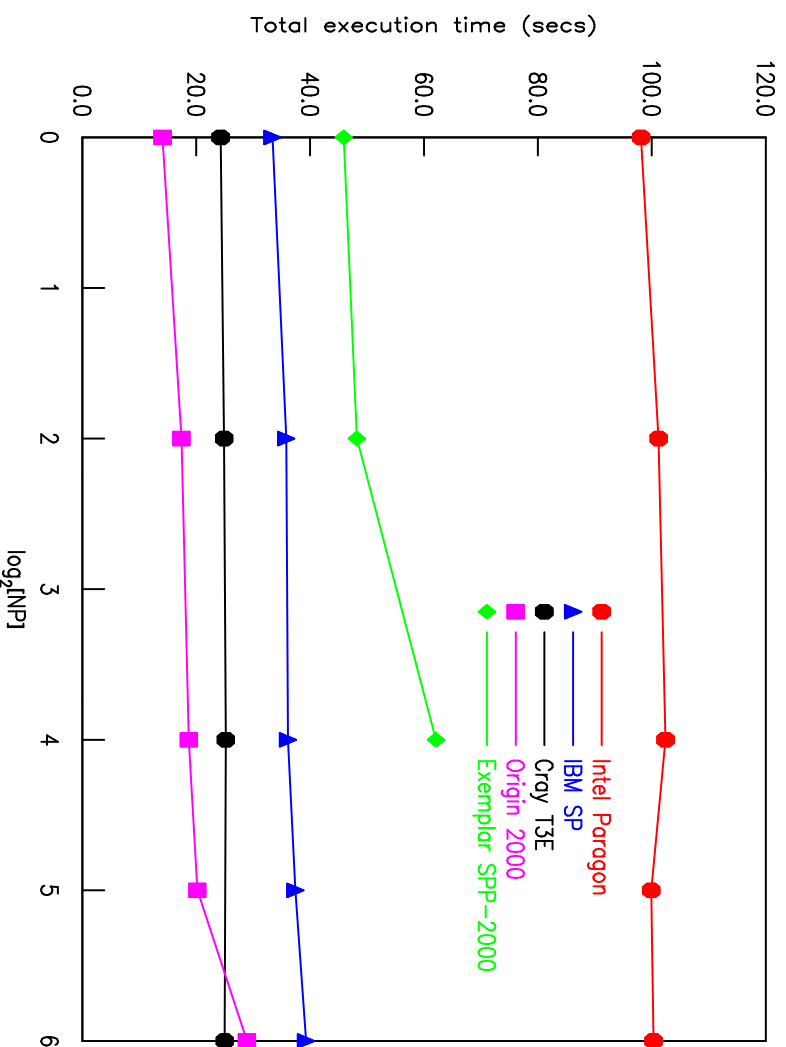


Performance of transport code on three parallel platforms

number of processors = 64, number of components = 6, number of time steps = 10,

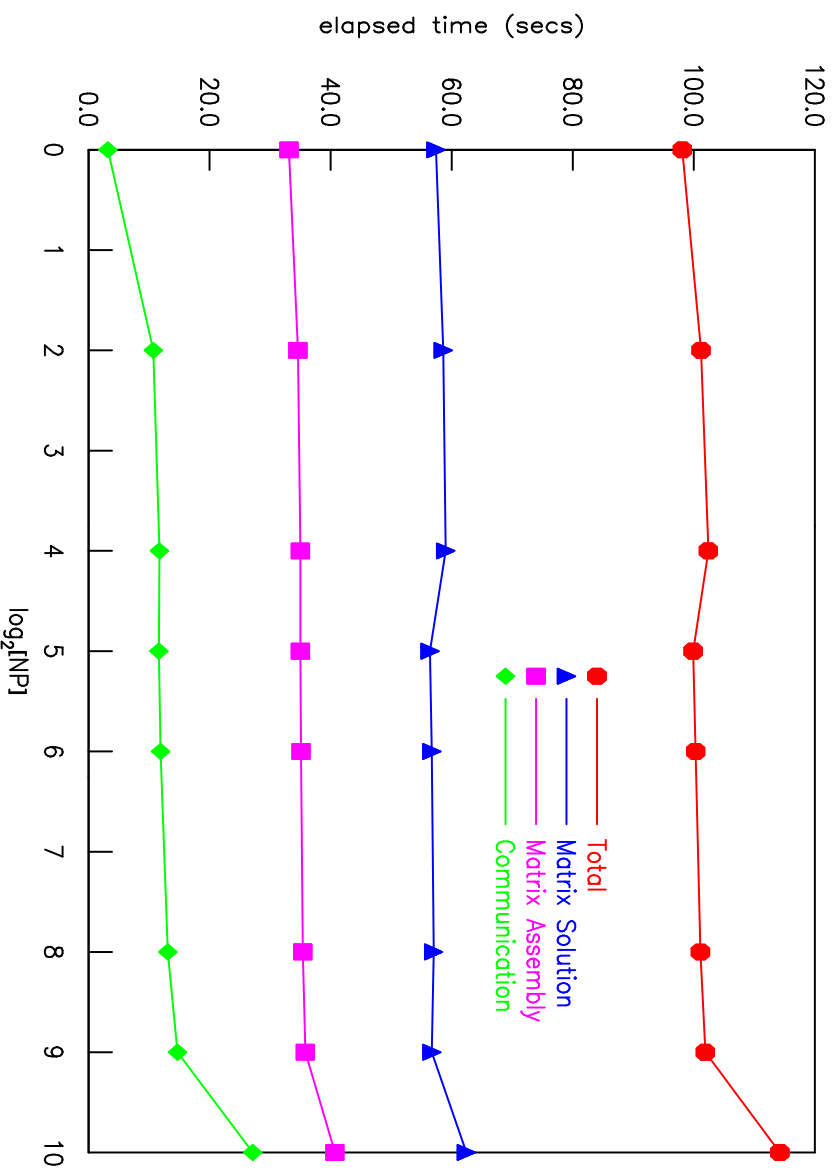
number of finite elements = 1 M, number of degrees of freedom 6.8 M

CROSS PLATFORM SCALABILITY



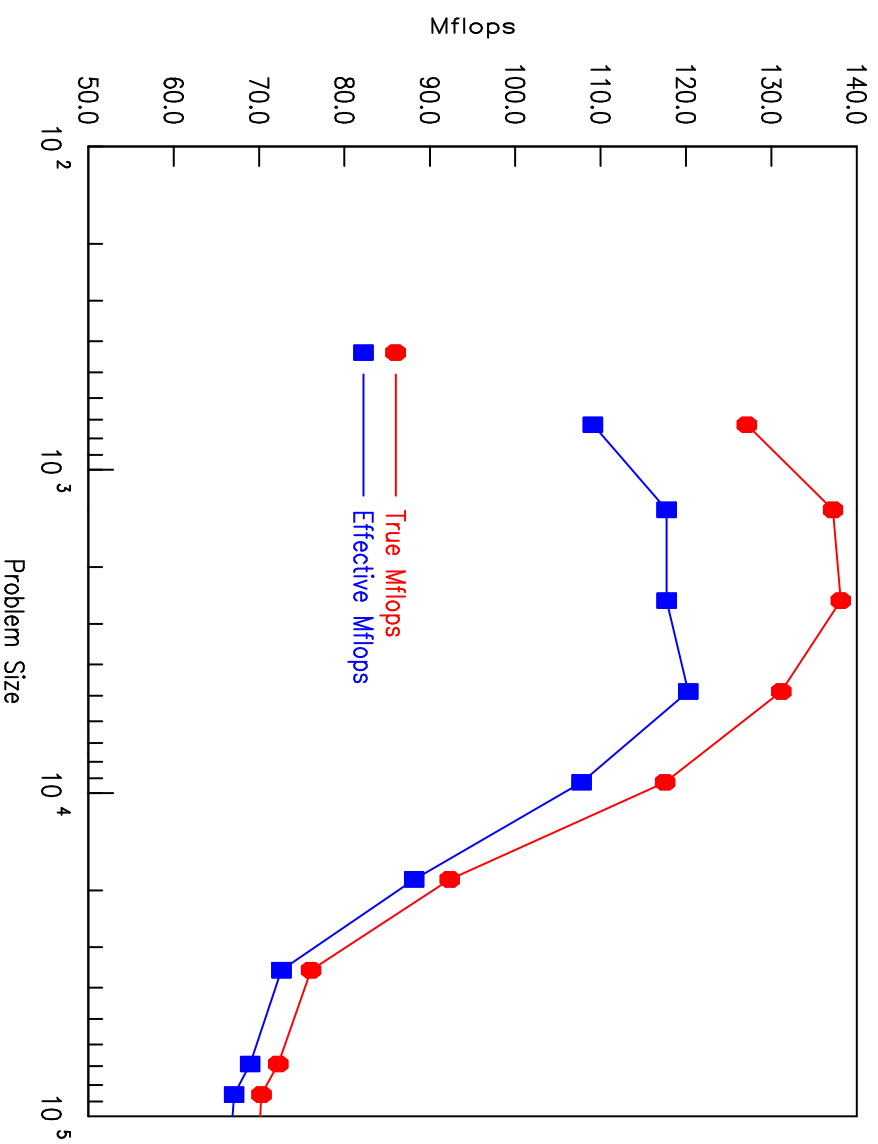
Cross-platform scalability up to 64 processors. NP = number of processors. Local problem size is fixed at $41 \times 41 \times 11$. Six transporting species are simulated to 10 time steps with approximately 110,000 degrees of freedom per processor. Total execution time includes all operations of the code.

SCALABILITY ON THE PARAGON XPS/150



Paragon scalability up to 1024 processors. NP = number of processors. Local problem size is fixed at 41x41x11. NP = number of processors. Local problem size is fixed at 41x41x11. Six transporting species are simulated to 10 time steps with approximately 110,000 degrees of freedom per processor. Total execution time includes all operations of the code.

CACHE EFFECTS ON ORIGIN2000



Effect of cache on the single node performance of the BICGSTAB matrix solver on the Origin 2000.
Problem size is the number of grid points. Six component transport.

CONCLUSIONS

- Our implementation is scalable on architectures with good communication bandwidth to peak performance ratio (e.g. Intel Paragon and Cray T3E).
- On architectures with large cache sizes (> 4 MB) better percentage of peak can be achieved for moderate problem sizes (e.g. Origin 2000).
 - Single node solver performance of > 120 Mflops for 3000–5000 node problems on the Origin 2000.
- The special memory and computational saving features enable rapid solution for most commonly encountered class of problems.
 - e.g. 5–10 secs per time step for problems with greater than 100 million degrees of freedom on the 1024 node paragon.

ACKNOWLEDGEMENTS

This work was supported in part by the Partnership in Computational Sciences Program administered by the Mathematics, Information and Computer Science section of the U.S. Department of Energy

The use of High Performance Computing Facilities at the Oak Ridge National Laboratory, National Center for Supercomputing Applications, Argonne National Laboratory, and the National Energy Research Supercomputer Center are gratefully acknowledged .