Performances and Tuning for Designing a Fast Parallel Hemodynamic Simulator

Bilel Hadri

University of Tennessee
Innovative Computing Laboratory

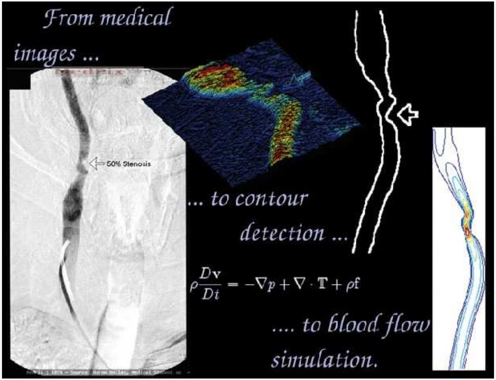
Collaboration: Dr Marc Garbey, University of Houston,
Department of Computer Science

July 10, 2008

Contents

- Cardiovascular Disease is the number one cause of death and disability in the US and Europe. (37.3% of all death in the US)
- In the United States, one person dies every 35 seconds from heart disease. Aneurysm, stenos are the main cardiovascular problems
- Need to make early and fast diagnostic

From an angiogram get the image segmentation and the flow simulation .



$\Rightarrow$ Need to design a close to real time simulation

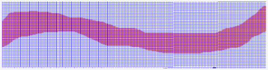
- **Fact** : The most consuming part of a code is the resolution of some linear system.
- **Focus**: Fast elliptic solver for incompressible Navier-Stokes(NS) flow code.
- **Context**:
  - Finite Volume,
  - Mesh topologically equivalent to Cartesian mesh,
  - Distributed computing with high latency network,
- **Goal :** Build a portable hemodynamic simulator that can be tuned for better performance.

Navier Stokes Formulation

$\Rightarrow$ We need a Fast Prototyping of NS flow.

$$\partial_t U + (U.\nabla)U + \nabla p - \nu \nabla.(\nabla U) = -\frac{1}{\eta}\Lambda_{\Omega_w}\,\{U - U_w(t)\},$$

$$div(U) \;=\; 0,$$



- Ω_w solid wall, and Ω_f flow domain.
- $U_w(t)$ speed of the wall.
- L^2 Penalty method: $\eta << 1$. - reference Caltagirone 84, Angot-Bruneau-Fabrie 99., Schneider et al 2005-
- Λ is a mask function provided by a level set method used in the image segmentation of the blood vessel.
- First Order algorithm , fast and robust.

Time Step: Projection Scheme (Chorin)
Momentum Equation

$$-dt\,\nu\,\Delta U \;+\; c\,U \;=\; RHS_1,\quad \nu << 1,\; dt << 1.$$

Pressure Equation

$$\Delta P = RHS_2$$

⇒ **Focus: : design of the optimum solver with the appropriate boundary conditions**

Design of the Elliptic Solver

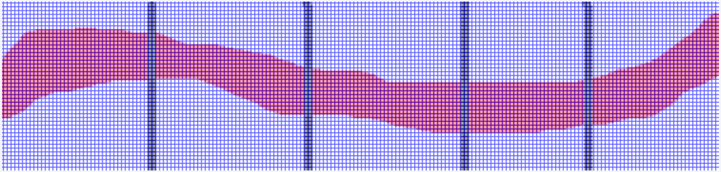
M. Garbey and D. Tromeur Dervout: "On some Aitken like acceleration of the Schwarz Method," International Journal for Numerical Methods in Fluids. Vol. 40(12),pp 1493-1513, 2002.

Aitken Schwarz is a domain decomposition method using the framework of Additive Schwarz and based on an approximate reconstruction of the dominant eigenvectors of the trace transfer operator.

Algorithm :

- Step1: apply additive Schwarz with a subdomain solver
- Step 2:
  - compute the sine (or cosine) expansion of the traces on the artificial interface for the initial boundary condition u_Γ^0 and the solution given by the first Schwarz iterative u_Γ^1
  - apply generalized Aitken acceleration to get u_Γ^∞
  - recompose the trace in physical space.
- Step 3 : Compute in parallel the solution in each subdomain, with the new inner BCs u_Γ^∞.

$\Rightarrow$ Goal: solve quickly a linear system of a given problem



$\Rightarrow$ How do we choose the fastest method depending on the sub-domain size, the architecture ?
$\Rightarrow$ We are looking for a performance portability and tuning.

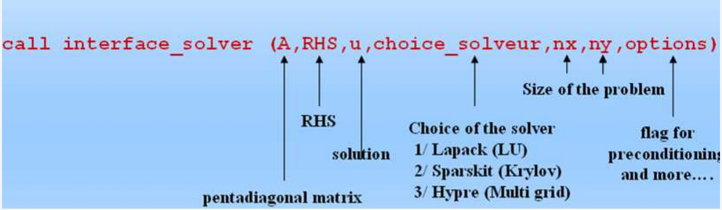
Many approaches to solve a linear system: $Ax = b$

- Direct solver
- Krylov methods
- Multigrid

	LAPACK	SPARSKIT	HYPRE
Resolution method	Direct Solver (LU decomposition)	Iterative Solver (Krylov Solver)	Algebraic Multigrid
Preconditioning method	Diagonal	ILU	Boomer AMG, PILUT, Jacobi....
Storage	Band	Sparse	Sparse
Language	Fortran 77 and C	Fortran 77	C + MPI

$\Rightarrow$ Need of an interface to help the user.

⇒ Interface that calls different libraries.



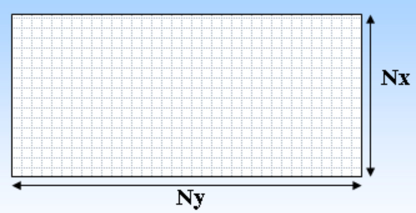
⇒ Performance evaluation thanks to Surface Response.

Performance Analysis Sequential

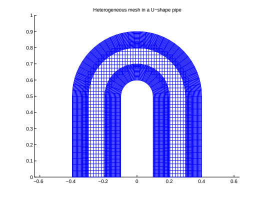
- Build a model prediction from least square quadratic polynomial approximation based on few runs.
- Predict the behavior for various subdomains sizes.
- Provide an indicator on the reliability of the model .

⇒ Model for the elapsed time T, depending on the size

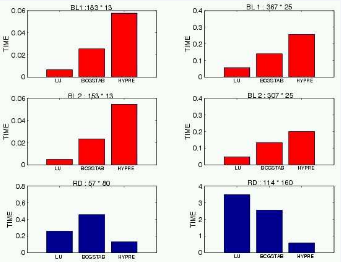
$$T(n_x, n_y) = \beta_0 + \beta_1.n_x + \beta_2.n_y + \beta_3.n_x^2 + \beta_4.n_y^2 + \beta_5.n_x n_y$$

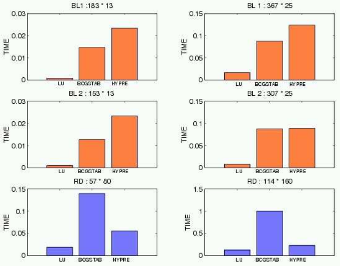


Performance of subdomain solvers with an incompressible flow in a curved pipe



- BL1 and BL2 fit the wall and have orthogonal meshes to approximate the boundary layer.
- The domain denoted RD for the central part of the pipe is polygonal and it is overlapping the boundary subdomains by few mesh cells.
- This is basically a Chimera approach that is convenient to compute fluid structure interaction.





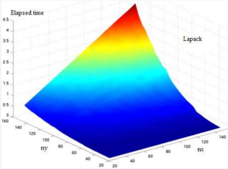
Comparison of the elapsed time for each subdomain with preconditioning(left graphic), precomputed(right) preconditioner.

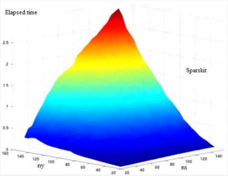
The optimum choice of the solver for each subdomain depends on

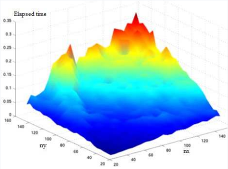
- the type of subdomain,
- the fact that one reuse or not the same preconditioner or decomposition
- the architecture of the processor,
- the size of the problem.

$\Rightarrow$The choice of the wrong solver for a specific domain can slow down the computation.

Systematic performance computation with Lapack, Sparskit and Hypre.

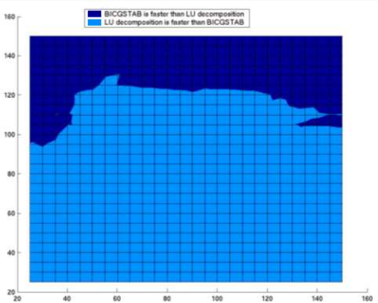


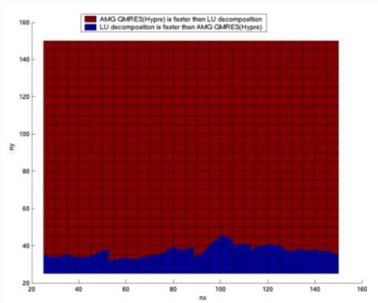




The surface is very smooth for Lapack and Sparkit while for Hypre, there are a lot of variation due to the high sensitivity of algebraic multigrid to grid sizes.

Comparison between different solvers

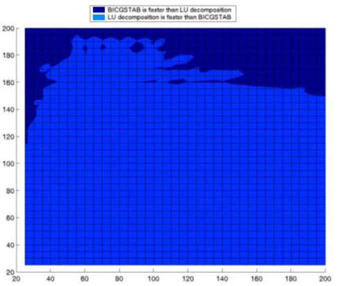


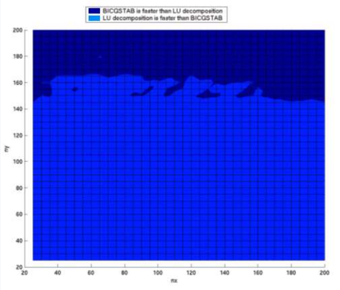


Surface Response on a AMD Athlom 1800 with 2GB of RAM depending on different libraries..

⇒ For small size problem, it is better to solve the linear system with the LU decomposition because it is faster than BICGTAB and AMG-GMRES for the Laplacian problem.

Comparison between different architectures





Surface Response on a AMD Athlom 1800 with 2GB of RAM (left) and on a Itanium2 with 3GB of RAM.

⇒ For the same problem, the elapsed time is not the same on two different architecture. The region where BICGSTAB is faster , is not the same. It depends on the architecture of the computer.

From the performance evaluation , the elapsed time depends on :

- the size
- the boundary condition
- the architecture of the machine

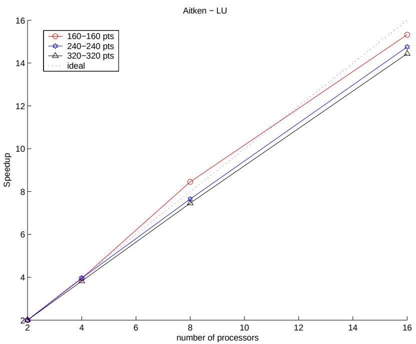
How can we choose the best solver ?
Regression along 9 points to get a model use that a least square quadratic polynomial approximation :

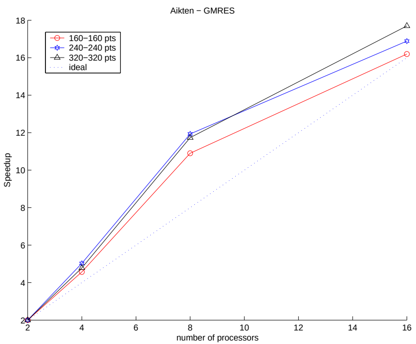
$$T(n_x, n_y) = \beta_0 + \beta_1 n_x + \beta_2 n_y + \beta_3 n_x^2 + \beta_4 n_y^2 + \beta_5 n_x n_y$$

	Lapack	Sparskit	Hypre
Least Square Error $\|\|\varepsilon\|\|^2$	6.088e-3	2.22e-3	0.3227
Mean of the percentage of the error μ	3.059	2.6059	50.69
Standard deviation σ	3.175	1.6142	23.803

Performance Analysis
Parallel Performance

Speedup of Aitken Schwarz with LU and BICGSTAB

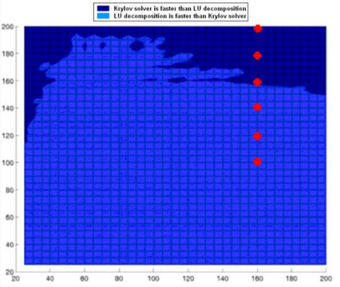




Aitken Schwarz performs very well on small problems. Further, the Krylov method seems to be more sensitive to the cache effect, since we have a superlinear speedup.

⇒ **Does the prediction model apply for the parallel runs ?**

Prediction of the best subdomain solver.

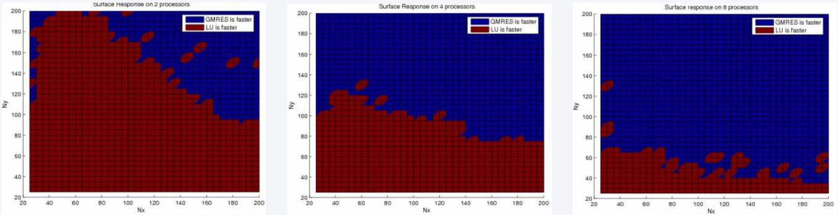


	2 processors		4 processors		8 processors		16 processors	
points per processors	LU	Saad	LU	Saad	LU	Saad	LU	Saad
100 x 160	1.44	2.26	1.58	1.87	1.43	1.43	1.79	1.82
120 x 160	2.08	3.19	2.45	2.54	2.29	1.98	2.75	2.50
140 x 160	3.28	4.17	3.61	3.36	3.54	2.75	3.89	3.45
160 x 160	4.67	5.21	4.81	4.36	4.95	3.49	5.37	4.37
180 x 160	6.46	6.39	6.43	5.63	6.52	4.31	7.72	5.45
200 x 160	8.24	8.18	8.89	6.69	8.27	5.25	9.69	6.54

This prediction is correct for the 2 processors computation. However as the number of processors grows, this prediction is slightly incorrect, and one should favor the Krylov solver.

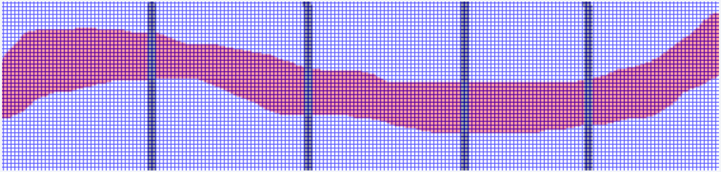
$\Rightarrow$ **Surface response modeling requires a 3rd dimension (= number of processors)**

Surface response depending on the numbers of processors



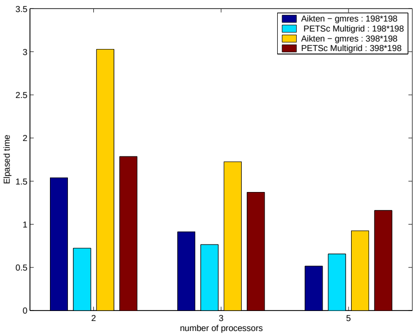
For the same number of unknowns, Krylov solver seems to be faster for small size when the number of processors is increased.

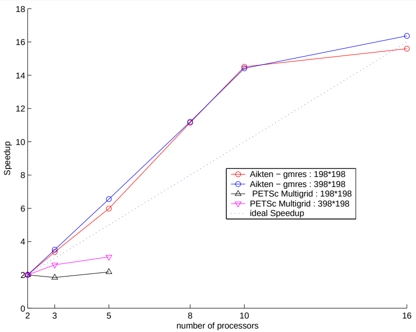
⇒ Goal: solve quickly a linear system of a given problem



⇒ **Are we competitive** ?

Comparison AS with PETSc





PETSc is faster than AS with 2 and 3 processors. As the number of processors increases:

- the PETSc multigrid solver does not speed up well, while AS is performing better.
- AS gives a better elapsed time than the multigrid solver .
- For simple problems, and with high latency network, the AS algorithm is very efficient.

⇒ **Best compromise with PETSc to solve each subdomain**

Navier-Stokes Applications

Performance on the Itanium2

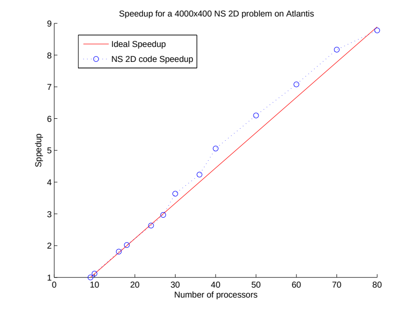


Figure: Speedup Performance

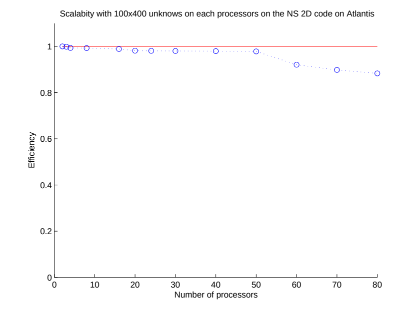
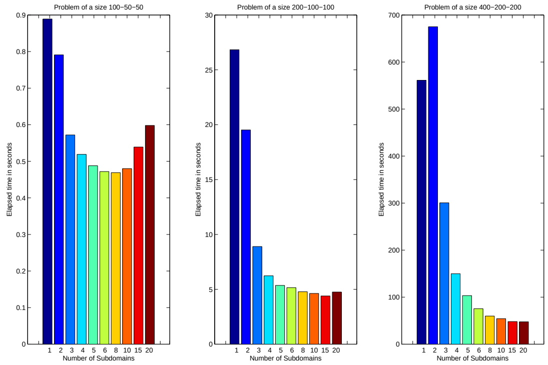


Figure: Scalability Performance

Elapsed time for one step time depending on the number of subdomains and the grid size.

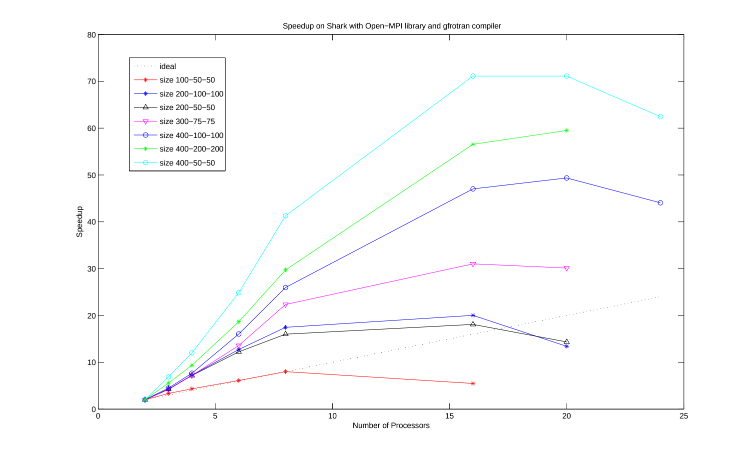


Size grows by factor **8** / elapsed time by **10** Speedup!

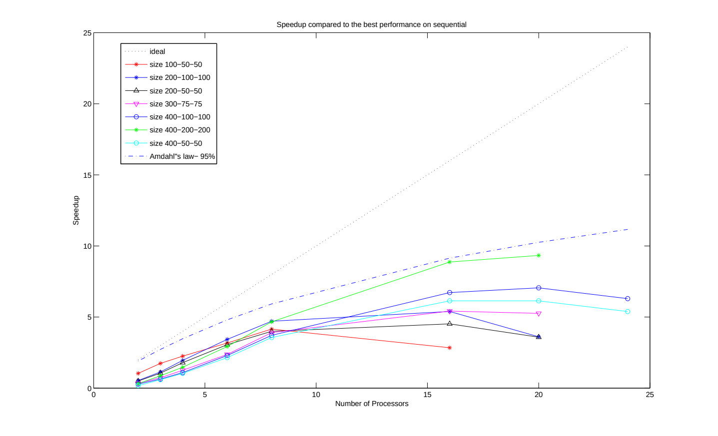
Tests performed on a SUN cluster. The systems gathers $24\ X2100$ nodes, 2.2 GHz dual core AMD Opteron processor, 2 GB main memory each, with an Infiniband Interconnect.

Table: Elapsed time for the resolution of one step time

	Number of Processors						
Grid Size	2	3	4	6	8	16	20
100-50-50	0.52	0.1	0.24	0.17	0.13	0.2	
200-50-50	2.08	0.98	0.58	0.34	0.26	0.23	0.29
400-50-50	12.80	3.72	2.13	1.03	0.62	0.36	0.36
300-75-75	10.39	4.96	2.89	1.53	0.93	0.67	0.69
200-100-100	8.91	4.21	2.46	1.4	1.02	0.89	1.15
400-100-100	34.8	15.5	9.08	4.34	2.68	1.48	1.41
400-200-200	188	67.7	40.2	20.16	12.65	6.65	6.32



Speedup: the number of subdomains equal the number of processors



"True" Speedup: the number of subdomains equal the number of processors, Amdahl's law since 95% of the code is parallelized.

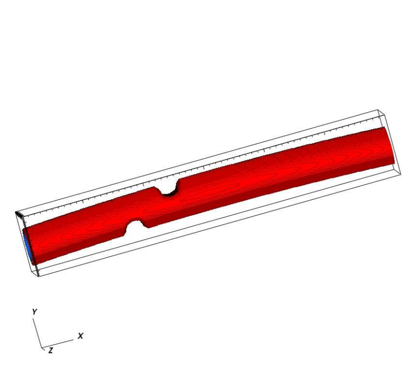


Figure: Geometry of the artery

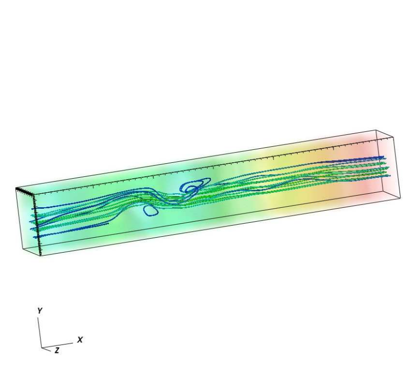


Figure: Velocity and Pressure inside the artery

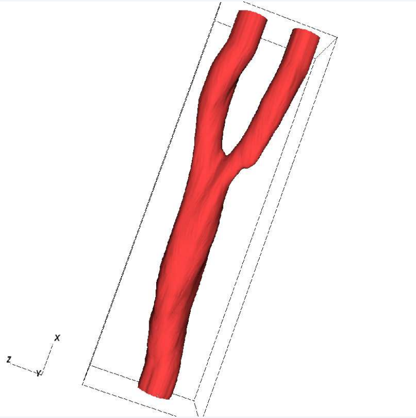


Figure: Carotid bifurcation

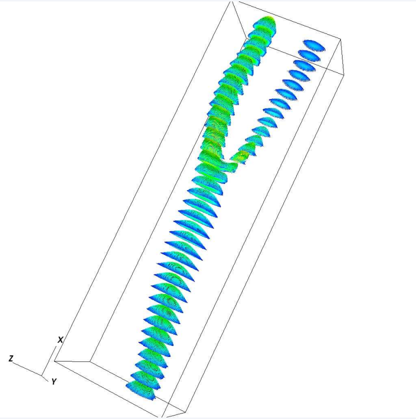


Figure: Velocity

Conclusions

- Optimum tuning of the solver provides us the fastest subdomain solver.
- Aikten Schwarz a domain decomposition framework for elliptic solver, is efficient and robust for distributed computing.
- This attractive approach provides scalability and improves performances for small problems with large number of processors.
- Brings blood flow simulation close to the level of efficiency of image processing .
- Parallel processing is one way to deal with the complexity of computational medicine.

How can we achieve a better performance ?

- Enhance the parallelism: 2D topology (Parallelize the subdomain)
- Choose adapted linear algebra both for distributed and shared memory systems (PLASMA ☺)
- Optimize the matrix-matrix multiplication or the FFT decomposition
- Tune collective communications(ADCL Adaptive Data and Communication Library from E. Gabriel, UH)

- **Interdisciplinary research collaboration is indeed needed to achieve the best performance for HPC applications!**
  Scientific computing groups + Algorithm groups + Compiler groups.

- B. Hadri, H. Ltaief, M. Garbey, An Hemodynamic Application on a Distributed Computing Environment, AIAA 46th Sciences Meeting and Exhibit Conference, Reno January 2008.
- B. Hadri, M. Garbey, Optimization of a Fast 2D Parallel Navier Stokes Code, Journal of Parallel Computing, in submission, 2007
- M. Garbey, B. Hadri, V. Hilford, C. Karmonik, Parallel Image-Based Hemodynamic Simulator, IEEE, ICSNC'07 , workshop HPC-Bio07, Cap Esterel, France, August 2007
- B. Hadri, M. Garbey, A Fast Parallel Blood Flow Simulator, 19th International Conference on Parallel Computational Fluid Dynamics, Antalya, Turkey, May 2007
- M. Garbey and B. Hadri, Image Based CFD for Blood Flow Simulation. 45th Aerospace Sciences Meeting and Exhibit Conference, Reno January 2007, paper number: AIAA-2007-0718.
- B. Hadri, M. Garbey, and W. Shyy, Improving the Resolution of an Elliptic Solver for CFD Problems on the Grid. Parallel Computational Fluid Dynamics, Theory and Applications, A.Deane et Al edit. Elsevier, pp 317-324, 2006.
- M. Garbey and B. Hadri, Toward Real Time Image Based CFD. 17th International Conference on Domain Decomposition Methods, July 2006, Austria.
- M. Garbey, B. Hadri and W. Shyy. Fast Elliptic Solver for Incompressible Navier Stokes Flow and Heat Transfer Problems on the Grid. 43rd Aerospace Sciences Meeting and Exhibit Conference, Reno January 2005, paper number: AIAA-2005-1386.
- M. Garbey, W. Shyy, B. Hadri, and E. Rougetet. Efficient solution techniques for CFD and heat transfer. In ASME Heat Transfer/Fluids Engineering Summer Conference, Charlotte, July 2004.

Thank you !