A STUDY OF EFFICIENCY AND SPEEDUP OF RESOLUTION OF LARGE LINEAR SYSTEMS ON A CLUSTER USING PETSC LIBRARIES

Torres López, Pedro Juan, torres.pedrozpk@gmail.com
Shin, Hyun Ho, hhs82528@yahoo.com
Mangiavacchi, Norberto, norberto.mangiavacchi@gmail.com
Rio de Janeiro State University - Rua Fonseca Teles, 121, São Cristovão, Rio de Janeiro, RJ

Schaerer Serra, Christian Emilio, cschaer@pol.una.py
Polytechnic-National University of Asuncion-Universitary Campus - P.O.Box: 211 SL San Lorenzo - Paraguay

Abstract. *The simulation of hydroelectric power plant reservoir flooding flows is becoming a very important issue to forecast the environmental impact of the drowned vegetation decomposition. To deal with this, we resolve the Navier-Stokes equation with scalar transport in a complex domain, using a finite element method approach as a discretization technique. As a part of that simulation process, several linear systems of equations have to be solved during the nonlinear iteration process at each time step, and this task is the most time consuming part of whole work. In order to improve that process, we study in this paper the performance characteristics of the resolution of those large linear systems on a Beowulf cluster, a set of parallel computers with a distributed and shared memory architecture. We employ a set of solvers and preconditioners available on PETSc, a library for a distributed and shared memory architecture, based on MPICH, BLAS and LAPACK. Finally, we present the performance obtained with several solver, precondition and reordering techniques.*

Keywords: *Hydroelectric reservoir, large linear system,parallel computer, PETSc*

1. INTRODUCTION

The enviromental impact of hydroelectric power plant reservoir flooding flows is becoming a very important issue to forecast the environmental impact of the drowned vegetation decomposition. In order to evaluate this impact we need to resolve complicate system of partial differential equations in quite complex domain, that is, RANS (Reynolds Average Navier Stoke) equations with scalar transport, using finite element aproach. At the core of this work, as a result of the space-time discretization, we need to solve several large linear system of equation. On the other hand parallel computation had been quite succesfull in cientific computing applications see [Gupta2003]. Thus, to deal with the current problem, it is natural that we take advantage of the extensive development of both hardward and software for parallel computation.

The main goal is to reduce the global simulation time. As a first step we focus in the resolution of the linear system, which is the most time consuming part. Basically there are two approachs, direct and iterative methods. The first approach has the advantage of being robust and accurate, however with high resolution of three-dimensional models as well as models that incorporate more complex phenomena, the computational costs increases, and the second approach become relevant, due to its lower computational costs, even though the covergence isn't guaranteed, see [Saad2000]. The critical succes of the iterative methods is the preconditioner.

As a starting point, we carry out experiments to evaluate the behavior of direct and iterative preconditioned solver, in a cluster platform. Also, we evaluate the influence of the reordering technique on both types of solvers.

2. GOVERNING EQUATIONS AND DISCRETE FORMULATION

The system of equations cosidered is the Reynolds Average Navier Stoke (RANS) and mean scalar transport, as show below,

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\overline{D}U_i}{Dt} = -\frac{1}{\rho_r}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu_{eff}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] - \beta\Theta g_i \tag{2}$$

$$\frac{\overline{D}\Theta}{Dt} = \frac{\partial}{\partial x_j}\left(\alpha_{eff}\frac{\partial \Theta}{\partial x_j}\right) + S_\theta \tag{3}$$

where U_i is the velocity vector, P the pressure, Θ the scalar quantity (e.g. temperature), β the thermal expansion

coefficient, S_θ a source or a sink of Θ, α_{eff} the effective molecular (heat or mass) diffusivity coeficient, ν_{eff} effective kinematic viscosity and g_i is the gravity component for details see [Pope2000] and [Batchelor2000].

The boundary conditions

$$U_i = U_{\Gamma_i} \quad \text{on} \quad \Gamma_i; \qquad \frac{\partial U_i}{\partial x_n} = U_{\Gamma_i^c} \qquad \text{on} \quad \Gamma_i^c \quad (i = 1, 2, 3) \tag{4}$$

$$P = P_{\Gamma_p} \qquad \text{on} \quad \Gamma_p; \qquad \frac{\partial P}{\partial x_n} = P_{\Gamma_p^c} \qquad \text{on} \quad \Gamma_p^c \tag{5}$$

$$\Theta = \Theta_{\Gamma_\theta} \qquad \text{on} \quad \Gamma_\theta; \qquad \frac{\partial \Theta}{\partial x_n} = \Theta_{\Gamma_\theta^c} \qquad \text{on} \quad \Gamma_\theta^c \tag{6}$$

and initial conditions

$$U_i = U_{it_0} \quad P = P_{t_0} \quad \Theta = \Theta_{t_0} \quad \text{in } \Omega \quad \text{at } t = t_0 \tag{7}$$

where Ω is the domain of the differential equations system, and $\Gamma = \Gamma_\xi \oplus \Gamma_\xi^c$ ($\Gamma = \Gamma_\xi \cup \Gamma_\xi^c$ and $\Gamma_\xi \cap \Gamma_\xi^c = \varnothing$) are the boundary of Ω, *i.e.*, $\Gamma = \partial\Omega$, are applied.

Considering the discretized domain Ω_h, we obtain the semidiscrete variational aproach of (1), (2) and (3), as follows

$$\int_\Omega \frac{\partial w_p}{\partial x_i} U_i d\Omega = \int_{\Gamma_p^c} w_p U_i n_i d\Gamma \tag{8}$$

$$\begin{aligned} \int_\Omega w_i \frac{\overline{D}U_i}{Dt} d\Omega - \frac{1}{\rho_r} \int_\Omega \frac{\partial w_i}{\partial x_i} P d\Omega + \int_\Omega \nu_{eff} \frac{\partial w_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) d\Omega + \int_\Omega w_i \beta \Theta g_i d\Omega \\ = -\frac{1}{\rho_r} \int_{\Gamma_i^c} w_i P n_i d\Gamma + \int_{\Gamma_i^c} \nu_{eff} w_i \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) n_j d\Gamma \end{aligned} \tag{9}$$

$$\int_\Omega w_\theta \frac{\overline{D}\Theta}{Dt} d\Omega + \int_\Omega \alpha_{eff} \frac{\partial w_\theta}{\partial x_j} \frac{\partial \Theta}{\partial x_j} d\Omega = \int_{\Gamma_\theta^c} \alpha_{eff} w_\theta \frac{\partial \Theta}{\partial x_j} n_j d\Gamma + \int_\Omega w_\theta S_\theta d\Omega \tag{10}$$

where the w_p, $w_i (i = 1, 2, 3)$ and w_θ are weighting functions that belong to the subspace $\mathcal{H}_0^1$, more explanation of this formulation can be viewed in [Zienkiewiczb2000a], [Zienkiewiczb2000b], [Hughes2000] and [Lewis2004]. Thus, using the Galerkin approach and the finite element discretization technique, we can write (8), (9) and (10) in matrix form as follows,

$$\mathbf{D}\tilde{\mathbf{u}} = \boldsymbol{bcn}_1 \tag{11}$$

$$\mathbf{M}_\rho \dot{\tilde{\mathbf{u}}} - \mathbf{G}\tilde{\mathbf{p}} + \mathbf{K}_\rho \tilde{\mathbf{u}} + \beta \mathbf{g}\mathbf{F}\tilde{\theta} = \boldsymbol{bcn}_2 \tag{12}$$

$$\mathbf{M}_\theta \dot{\tilde{\theta}} + \mathbf{K}_\theta \tilde{\theta} = \boldsymbol{bcn}_3 \tag{13}$$

In this manner we obtain a system of ordinary differential equations, where $\mathbf{D}$ is the divergence matrix, $\mathbf{G}$ is the gradient matrix, $\mathbf{M}_\rho$ is the mass matrix, $\mathbf{K}_\rho$ is the momentum diffusion matrix, $\mathbf{M}_\theta$ is the scalar mass matrix, $\mathbf{K}_\theta$ is the scalar diffusion matrix.

Next, to time discretization we employed the Semi-Lagrangian method. This technique contributes to a significant enhancement of the efficiency of the semi-implicit integration scheme, see [Robert1984]. Equations 11, 12, and 13 with time discretization read as

$$\mathbf{D}\tilde{\mathbf{u}}^{n+1} = \boldsymbol{bcn}_1 \tag{14a}$$

$$\mathbf{M}_\rho \left(\frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_d^n}{\Delta t} \right) - \mathbf{G}\tilde{\mathbf{p}}^{n+1} + \mathbf{K}_\rho \left(\lambda \tilde{\mathbf{u}}^{n+1} + (1 - \lambda)\tilde{\mathbf{u}}_d^n \right) + \beta \mathbf{g}\mathbf{F}\tilde{\theta}_d^n = \boldsymbol{bcn}_2 \tag{14b}$$

$$\mathbf{M}_\theta \left(\frac{\tilde{\theta}^{n+1} - \tilde{\theta}_d^n}{\Delta t} \right) + \mathbf{K}_\theta \left(\lambda \tilde{\theta}^{n+1} + (1 - \lambda)\tilde{\theta}_d^n \right) = \boldsymbol{bcn}_3 \tag{14c}$$

where, λ is a parameter to obtain different methods of discretization in time. For, $\lambda = 0$ results a explicit discretization, $\lambda = 1$ results a semi-implicit discretization and $\lambda = \frac{1}{2}$ results the Crank-Nicolson method.

The linear equations system 14 is solved employing the block LU approximate factorization, also called projection method, resulting in the follows linear system,

$$\mathbf{M}'_\rho \hat{\mathbf{u}}^{n+1} = \boldsymbol{r}_u^n + \boldsymbol{bc}_1 \tag{15}$$

$$\Delta t \mathbf{D}\mathbf{M}'^{-1}_\rho \mathbf{G}\tilde{\mathbf{p}}^{n+1} = -\mathbf{D}\hat{\mathbf{u}}^{n+1} + \boldsymbol{bc}_2 \tag{16}$$

$$\mathbf{M}'_\theta \tilde{\theta}^{n+1} = \boldsymbol{r}_\theta^n + \boldsymbol{bc}_3 \tag{17}$$

$$\tilde{\mathbf{u}}^{n+1} = \hat{\mathbf{u}}^{n+1} + \Delta t \mathbf{M}'^{-1}_\rho \mathbf{G}\tilde{\mathbf{p}}^{n+1} \tag{18}$$

This method relies on the Helmholtz-Hodge decomposition, which say that any vector can be decomposed into a component of a zero divergence and another with zero curl.

We need to solve this large system of equations in each times step, consequently this task is the most consuming part. Clearly this job should be optimized reducing the execution time. In order to do that, we have been performing several test on both direct and precondiotioned interative solver combined with reordering technique. We know the propertis of matrix $\mathbf{M}'_\rho$ is symmetric and positive definite (SPD), therefore we use conjugate gradient method as a primary iterative method, that has well known behavior with this kind of matrix see The resulting linear system of equation are solved using precontioned GMRES on the case of $\mathbf{D}\mathbf{M}'^{-1}_\rho \mathbf{G}$.To improve the converge we employ reordering technique on the precondiotioner. We also employ direct solvers in both case.

3. DIRECT AND ITERATIVE METHODS ON PETSc

The Portable, Extensible Toolkit for Scientific Computation (PETSc) see [Balay et al.2009] and [Balay et al.2008], is a suite of data structures and routines that provide the building blocks for the implementation of large-scale application codes on parallel (and serial) computers. This library provide an important set of both parallel and serial solver collect in one generic context called KSP, allowing the user to choose the type of solver and preconditioning to be used for this context by changing a run-time option. This design makes it easy to compare the performance of many solver for a problem and determine the optimal method of solution. The design of PETSc was oriented to a problem rather than to an especific algorithm, see [Gropp1997a] and [Gropp1997b].

This library also allow the user to log program performance and obtain information of the code behavior such us floating point operation (flop) rates, message passing activity and memory usage. Another interesting feature allows users to utilize external packages and solvers including parallel direct and multigrid solvers, whithout changing the users code.

The complete list of solvers can be founded on PETSc Web Page see [Balay et al.2009].

4. NUMERICAL EXPERIMENTS

In the numerical experiments we tested the direct solver LU and Cholesky factorization, combined with following reordering tecniques *Nested Dissection, One-way Dissection, Reverse Cuthill-McKee, Quotient Minimum Degree*, all available in the library. The direct solver are available only on uniprocess mode, for multiprocessor mode we should use an external package.

Next, we test the solvers GMRES(k) ([Saad1986]) and Cojugate Gradient CG with block jacobi precondiotioners, in both serial and parallel form, using ILU(k) and ICC(k) on each block. In uniprocess the precondiotioner lead to an ILU or ICC factorization case.The initial guess was the zero vector.

For this experiment we used two different 3-dimensional meshes with homogeneus Dirichlet boundary conditions. Mesh 1 (4500 elements) and Mesh 2 (131422 elements) have the form showed in Fig. 1, these represent the confluence of a river on a brand of reservoir. A feactures summary of the used meshes are showe in Tab (1).

Table 1. Mesh Feactures Summary

Feactures	Mesh 1	Mesh 2
Number of Elements	4500	131422
Number of Node	1632	25322
NDoF/node	5	5
Type	Unstructured	Unstructured
Element Type	Tetraedral	Tetraedral

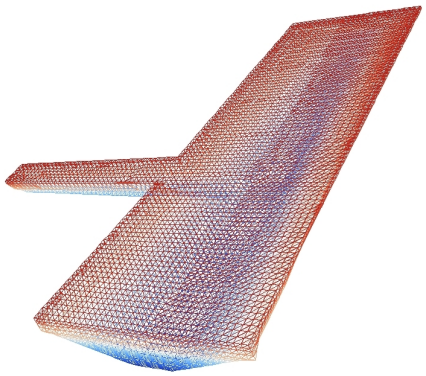


Figure 1. The used domain representing the confluence of a river on a brand of reservoir

4.1 Numerical Results

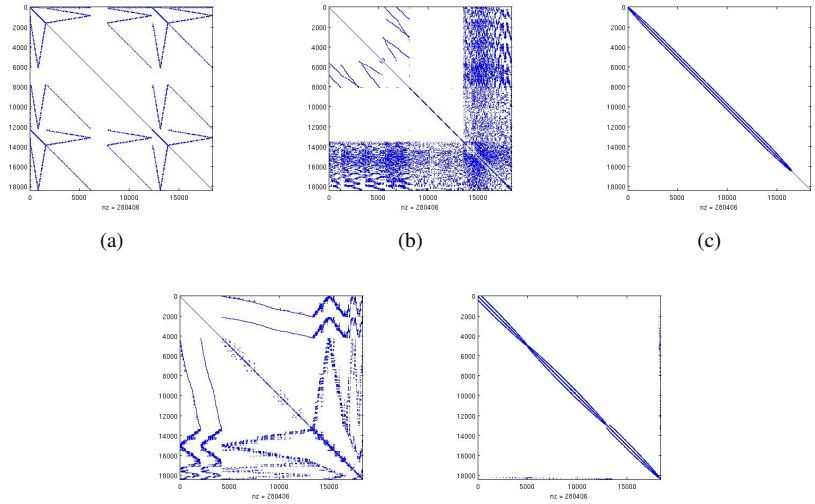
We presents the numerical results in two parts. Firts using direct solver and second using iterative solver, on each mesh. In the first part (4.1.1) we tested the behavior of combinations of direct solver with different reorder technique, showing the *fill-in* ratio between number of non zeros new entries in the factorized matrix and the number of non zeros on the original matrix, *Flops* numbers of floating points operation per seconds, and the CPU time in seconds.

In the second part (4.1.2) we show the result for parallel tests using precondiotioned iterative solve. Basically, we use block Jacobi precondiotioner combining with different reordering technique. Tests were made from 1 to 8 processors, in each trial we measure the CPU time in seconds, number of iterations and Flops.

Tables (2) to (8)show the numerical result, that we are obtained. From (1) to (3) correspond to the velocity matrix $\mathbf{M}'_\rho$, Tab. (1) and (2) correspond to the test with direct solvers and Tab. (3) correspond to the parallel case with iterative method. In the case of LU factorization, from Tab. (1) we can see that the best result obtained was with Quotient Minimun Degree in terms of fill-in and MFlops, but in terms of CPU time was Nested Disection. In the case of Cholesky factorization we can conclude the same.

4.1.1 Using Direct Solvers

As an ilustration in the Fig. 2, we draw the reordered matrix in each case. Tables (2), (3) and 4 correspond to the test with direct solvers on both meshes. In the case of LU factorization, from Tab. (2) and (4) we can see that the best result obtained was with Quotient Minimun Degree in terms of fill-in and MFlops, but in terms of CPU time was Nested Disection. In the case of Cholesky factorization we can conclude the same thing. For the pressure matrix $\mathbf{M}'_\rho$ case we see that the best result was obtained with Reverse Cutchill McKee, in terms of fill-in and MFlops. On the other hand we obtain the lower CPU time with Nested Dissection.



(a) (b) (c)

(d) (e)

Figure 2. Differents Reordering Techniques for Velocity Matrix - $\mathbf{M}'_\rho$: (a) Natural Ordering. (b) With QMD algorithm. (c) With RCM algorithm. (d) With ND algorithm. (e) With 1WD algorithm.

Table 2. Measurement of performance with LU factorization of the velocity matrix - $\mathbf{M}'_\rho$.

Velocity - $\mathbf{M}'_\rho$							
		Mesh 1			Mesh 2		
Solver	Reorder Technique	Fill-in	MFlops	CPU Time(s)	Fill-in	MFlops	CPU Time(s)
LU	Natural	628.13	2.27e+06	4430.11	N.E.M*	N.E.M	N.E.M
	QMD	2.376	41.97	0.615853	9.932	1.708e+05	588.452
	RCM	5.42494	132.0	0.29305	N.E.M	N.E.M	N.E.M
	ND	2.53	46.0	0.1664421	8.76567	1.327e+05	271.26058
	1WD	7.63281	300.0	0.705649	58.5736	1.057e+06	2710.68

Table 3. Measurement of performance with Cholesky factorization of the velocity matrix - $\mathbf{M}'_\rho$.

Velocity - $\mathbf{M}'_\rho$							
		Mesh 1			Mesh 2		
Solver	Reorder Technique	Fill-in	MFlops	CPU Time(s)	Fill-in	MFlops	CPU Time(s)
Cholesky	Natural	314.084	591.1	7349.0	N.E.M*	N.E.M	N.E.M
	QMD	1.20781	3.28	0.771177	4.983	2.95e+02	739.27634
	RCM	2.73	6.14	0.544599	5.424	132.0	0.29305
	ND	1.2866	3.43	0.2896601	4.40	2.646e+02	402.806
	1WD	3.83599	8.21	1.329501	N.E.M	N.E.M	N.E.M

N.E.M.: Not Enough Memory

4.1.2 Using Precondiotioned Iterative Solvers

The results in the parallel case are reported in Tab. (5) to (8). In the velocity case we obtain a reasonable scalability, see Tab. (5) and (6). Anyway, the best combinationg result with CG plus natural reordering, in terms of CPU time. Tables (7) and (8) shows the results for pressure matrix case, the best performance obtained was with Reverse Cutchill Mckee.

Table 4. Measurement of performance with LU factorization of the pressure matrix - $\mathbf{DM}_{\rho}^{\prime -1}\mathbf{G}$

Pressure - $\mathbf{DM}_{\rho}^{\prime -1}\mathbf{G}$							
		Mesh 1			Mesh 2		
Solver	Reorder Technique	Fill-in	MFlops	CPU Time(s)	Fill-in	MFlops	CPU Time(s)
LU	Natural	9.78	92.15	0.30	628.13	2.27e+06	4430.11
	QMD	4.385	19.36	0.17	24.71	3.768e+04	76.84751
	RCM	3.83964	13.09	0.18	27.99	3.477e+04	66.89325
	ND	4.88	24.32	0.04052393	22.96	3.328e+04	66.927
	1WD	6.23	35.54	0.05319892	28.48	3.598e+04	69.48

Table 5. Measurement of parallel performance with preconditioned iterative method of the velocity matrix - $\mathbf{M}_{\rho}^{\prime}$.

Velocity - Mesh 1						
Preconditioned Solver	Reorder Tecnique	Number of Process				
		1	2	4	6	8
CG+Block Jacobi	Natural	0.230/40/94.45	0.0721/37/39.03	0.0867/69/29.56	0.132/70/23.8	0.13368344/70/20.60
	QMD	0.584/8/21.95	0.148/8/9.28	0.108/62/26.26	0.088/61/20.65	0.100/62/18.24
	RCM	0.1404/13/32.47	0.0503/15/16.1	0.0827/66/27.93	0.1054/64/21.63	0.145/68/19.950
	ND	0.128/8/22.02	0.048/10/11.25	0.0804/64/27.09	0.136/62/20.98	0.093/63/18.53

Table 6. Measurement of parallel performance with preconditioned iterative method of the velocity matrix - $\mathbf{M}_{\rho}^{\prime}$.

Velocity - Mesh 2						
Preconditioned Solver	Reorder Tecnique	Number of Process				
		1	2	4	6	8
CG+Block Jacobi	Natural	2.27819/4/963.9	1.04783/6/333.5	0.831251/8/129.3	0.855934/8/98.7	0.88327/8/83.31
	QMD	2.2794/4/963	28.5604/6/234.5	5.9203/8/108.4	2.67/8/87.1	2.302/8/76.44
	RCM	2.2837/4/963.9	1.45026/6/242	0.83278/8/109.5	0.7897/8/87.48	0.7966/8/76.42
	ND	4.37846/3/408.8	1.74282/6/233.9	1.03316/8/108.3	1.00/8/87.03	1.009145/8/76.37

Table 7. Measurement of parallel performance with preconditioned iterative method of the pressure matrix - $\mathbf{DM}_{\rho}^{\prime -1}\mathbf{G}$

Pressure - Mesh 1						
Preconditioned Solver	Reorder Tecnique	Number of Process(*)				
		1	2	4	6	8
GMRES+Block Jacobi	Natural	0.011/28/9.63	0.01786/63/10.41	0.022/70/5.63	0.0384/107/5.65	0.071798/127/5.01
	QMD	0.03029/43/14.20	0.025/65/10.72	0.0234/70/5.65	0.0024/110/5.83	0.0576/123/4.89
	RCM	0.01324/19/6.30	0.011/32/5.50	0.0173/60/4.96	0.0302/70/3.71	0.0627/117/4.631
	ND	0.0283/57/19.08	0.022/63/10.51	0.033/73/5.89	0.0567/111/5.89	0.0698/131/5.15

(*) The data should be read as the following CPU time (s)/Iterations/MFlops

Table 8. Measurement of parallel performance with preconditioned iterative method of the pressure matrix - $\mathbf{DM}_{\rho}^{\prime -1}\mathbf{G}$

Pressure - Mesh 2						
Preconditioned Solver	Reorder Tecnique	Number of Process(*)				
		1	2	4	6	8
GMRES+Block Jacobi	Natural	39.07/2144/1.586e+04	118.33/9000/3.76e+04	41.236/5609/1.06e+04	33.93/4908/6.15e+03	38.543/5543/5.16e+03
	QMD	38.6379/1868/1.38e+04	48.42005/4493/1.69e+04	40.238/5996/1.13e+04	49.48/7826/9.8e+03	24.668/4169/3.88e+03
	RCM	14.43348/747/5.55e+03	107.05/5000/3.76e+04	43.046/5665/1.076e+04	24.51/3526/4.42e+03	42.693/5920/5.51e+03
	ND	38.029/1900/1.407e+04	107.36/7000/3.763e+04	37.86/5688/1.081e+04	51.08/8248/1.03e+03	30.41/5098/4.75e+03

(*) The data should be read as the following CPU time (s)/Iterations/MFlops

5. CONCLUSIONS

In this work we evaluate direct and iterative solver available in PETSc library. In the velocity case, with direct solver the best result obtained was with RCM algorithm, and with iterative was with CG with natural reordering. In the pressure case, good results are obtained with RCM algorithm on both direct and iterative solver. As a future work, we are planning to use parallel direct solver to perform the factorizations.

6. ACKNOWLEDGEMENTS

We would like to thank to FAPERJ and CNPQ for the financial support.

7. REFERENCES

Batchelor, G.K., 2000, "An Introduction to Fluid Dynamics", Cambridge University Press, UK.

Balay, Satish; Gropp, William D.; McInnes, Lois Curfman; Smith, Barry F. 1997. "Efficient management of parallelism in object oriented numerical software libraries", Modern Software Tools in Scientific Computing, Conference, pp. 163-202.

Balay S.,Buschelman K.,Gropp W.,Kaushik D.,Knepley M.,Curfman M. L.,Smith B. F., and Zhang H. 2009. "PETSc Web Page http://www.mcs.anl.gov/petsc".

Balay, S.,Buschelman, K.,Gropp, W.,Kaushik, D.,Knepley, M.,Curfman, M.,Smith, B.,Zhang, H., 2009, "PETSc Users Manual. ANL-95/11 - Revision 3.0.0", Argonne National Laboratory.

Hughes, T., 2000, "The Finite Element Method: Linear Static and Dynamic Finite Element Analysis", Dover Publications, New York.

Lewis, R.W., Nithiarasu, P. and Seetharamu, K.N., 2004, "Fundamentals of the Finite Element Method for Heat and Fluid Flow", John Wiley & Sons, Ltd, UK.

Pope, S.B., 2000, "Turbulent Flows", Cambridge University Press, Cambridge, UK.

Robert, A., Yee, T.L. and Ritchie, H, 1984, "A Semi-Lagrangian and Semi-Implicit Numerical Integration Scheme for Multilevel Atmospheric Models", Monthly Weather Review 113, pp. 388-394.

Rosenberg, D.M., McCully, P. and Pringle C.M., 2000, "Global-Scale Environmental Effects of Hydrological Alteration: Introduction", BioScience, Vol. 50 No. 9, pp. 746-751.

Gupta, A.,Gupta, A.,Karypis, G.,Kumar, V., 2003, "Introduction to Parallel Computing", Addison-Wesley, England.

Gropp, W., 1997. "Why we Couldn't Use Numerical Libraries for PETSc". Mathematics and Computer Science Division, Argonne National Laboratory.

Gropp, W., 1998, "Exploiting Existing Software in Libraries: Successes, Failures, and Reasons Why". Mathematics and Computer Science Division, Argonne National Laboratory.

Saad, Y.,van der Vorst, H., 2000, "Iterative Solution of Linear System in the 20-th Century", Journal of Computational and Applied Mathematics, Vol 123, pp. 1-33.

Saad, Y., 1995,"Precondiotioned Krylov Subspace Methods for CFD Applications", Proceedings of the International Workshop on Solution Techniques for Large-Scale CFD Problems, Vol. 26, pp. 179-195.

Saad, Y.,Schultz, M., 1986, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", Vol 7, Issue 3, pp 856-869.

Zienkiewicz, O.C. and Taylor, R.L., 2000, "The Finite Element Method. Volume 1: The Basis", Butterworth-Heinemann, Ma.

Zienkiewicz, O.C. and Taylor, R.L., 2000, "The Finite Element Method. Volume 3: Fluid Dynamics", Butterworth-Heinemann, Ma.

8. Responsibility notice

The author(s) is (are) the only responsible for the printed material included in this paper