Appendix A

Facilities

A.1 JFF PC cluster at CTTC

The JFF PC cluster is sited at CTTC (Centre Tecnològic de Transferència de Calor) in Terrassa (Spain). This acronym has been adopted in memorial of Joan Francesc Fernandez, computer science professor of the Polytechnic University of Catalonia. The JFF cluster is a network of 35 commodity processors connected with fast ethernet NICs (Network Interface Cards) and two full duplex switches. Each processor runs the operating system Linux with a minimal installation with a source Message Passing library.

Details of the cluster configuration are given in table A.1.

element	type	model
main board	35×ASUS a7V	AMD-751 chipset
processor (32bits)	5×600MHz + 28×900MHz	AMD-K7
cache (L1+L2)	128KB of L1 + 512KB of off-chip L2	
RAM	2×1GB + 33×512MB	SDRAM 133MHz
HD	25GB on server + 34 × 18GB	IDE
SWAP	double RAM per processor	
NICs (100Mb/s)	2 on server + 34 × 1	3C905B-TX-NM
switch (100Mb/s)	24 full duplex ports, scalable	3C16980
matrix cable	1 matrix cable connecting 2 switches	3C16965
OS	Linux	Debian v2.2.17
batch system	none	
MP library	MPI	LAM 6.1
compilers	C,C++,F77	gcc v2.7, g77 v0.5.24

Table A.1: Cluster configuration

Few pictures A.1,A.2 and A.3 of the JFF cluster have been attached to show the distribution and connection of the commodity PCs and switches on shells.

Figure A.1: Front view of the JFF cluster

Figure A.2: Front of the switches and connections

Figure A.3: Back of the switches and connections

In order to manage the cluster, there is a server node with x-server and equipped with two NICs (Network Interface Cards), one connected to the 34 nodes of the private net and the other connected to the public network. Users access to the cluster on this second NIC. Tasks of compilation and debugging may be done in this node. The compiler and loader for sequential jobs is the `gcc`, with the following flags and mathematic library:

```
> gcc -O3 -w -Wall -pedantic source-files.c -o object-files.o
> gcc object-files.o -lm executable-file
```

For parallel jobs, the compiler and loader is the `hcc`,with the MPI library:

```
> hcc -O3 -w -Wall -pedantic source-files.c -o object-files.o
> hcc object-files.o -lm -lmpi executable-file
```

After that, directories and executable files needed for execution are copied to nodes with the remote commands `rsh` and `rcp`.

Since we are in the beginnings of the JFF cluster, there is not yet a batch system. Therefore the execution of sequential and parallel jobs is done either in interactive mode or in background with the command `nohup`.

For sequential jobs the following command line is used:

```
> nohup /work-diretory/executable-file &
```

For parallel jobs with LAM MPI library three steps must be followed:

```
> lamboot -v nodes
> nohup mpirun -v -c np -O -w /work-directory/executable-file &
> wipe -v nodes
```

The first step initialize the lam daemon process and 'awakes' the np node-processors listed in the nodes file. The second step runs with the `mpirun` command np copies of the executable file in np processors placed in the work directory. The `-O` flag tells to the lam daemond that the network of processors is homogeneous (it does not data conversion among nodes), and the `-w` flag waits for all process to complete before exiting `mpirun` and reports abnormal exit codes. Once the execution has ended, the daemon process is killed with the `wipe` command.

Although Linux is a multiprocess operating system it is desirable to minimize the number of processes for performance measures (i.e. timings of subroutines). During these measures, each process whether it is originated by the operating system (i.e. the daemons) or by the executable file (i.e. the output of results) produce perturbations in these measures. For this reason every experiment must be repeated many times to give an averaged measure.

A.2 SGI/Cray T3E at CIEMAT

The SGI/Cray T3E supercomputer of 32 air cooled nodes is sited at CIEMAT (Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas) in Madrid (Spain). Details of the configuration are given in table A.2.

element	type	model
processor (64bits)	32×300MHz, super scalar RISC	DEC 21164
cache (L1+L2)	8KB of L1 + 96KB 3-way of L2	
RAM	32×128MB	DRAM
HD	130GB distributed for every 8np	Giga Ring, SCSI
SWAP	none	
net (1Gb/s)	3-D torus of low latency	
OS	Unix	Unicos/mk v2.0.4.34
batch system	Network Queuing Environment	NQE v3.3
MP library	PVM,MPI	optimized for Cray
compilers	C,C++,F77,F90	optimized for Cray

Table A.2: SGI/Cray T3E configuration

The SGI/Cray T3E has two processors dedicated to the surveillance and maintenance of the global system (i.e. the Global Resource Manager) and file server for the rest of nodes. Four processors for interactive tasks such as user sessions, compiling and debugging. Only they support multiprocess (time sharing). The rest of processors (i.e. 26 processors) do parallel applications throughout the command `mpprun`. Each processor accepts only one process and the total amount of memory available without swap is 128MB.

Sequential and parallel jobs are submitted to a batch system (for instance, the Network Queuing Environment) using the following script, called `job.sh`:

```
# QSUB -l mpp_p=np limit
# QSUB -l mpp_t=hh:mm:ss limit
# QSUB -eo
# QSUB -o job.sh
cd /work-directory
mpprun -n np executable-file
```

Further details may be found at the Ciemat's web site www.ciemat.es.

Appendix B

Software layers in CFD code

B.1 Description of layers

The CFD code is structured in four layers: the user layer, the solver layer, the algebra layer and the communication layer. These layers are linked and integrated to build an structure which looks like an iceberg (see Fig. B.1).

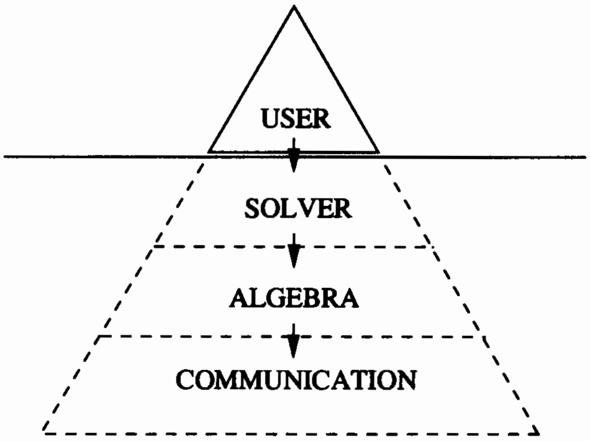

Figure B.1: Iceberg built by layers: user layer, solver layer, algebra layer and communication layer.

The concept of the iceberg comes from the idea that only the top of the iceberg or the last layer is visible while the rest of the iceberg or layers are hidden below the water. This top end is called the user layer, the only layer visible to the end user of the CFD code. The rest of layers are hidden from the user, i.e. the user does not access to the remaining layers. In order to build the CFD code in layers it is very important to specify clearly which are the subroutines of each layer and which layer is supported by another, i.e, to define the dependencies between layers in only one direction. This dependency goes from the top to the base as shown with the arrows in the in figure B.1.

The description of these layers is given below.

- The user layer can be structured into a set of modules, each of them with different functions. For example the graphic module, the data module and the model module. The graphic module represents the front-end of the CFD code. The data module contains all information needed for running a case. These, a specific information of the case (e.g. geometry, boundary conditions, initial conditions, properties of fluid and flow), and a specific information about how to run the case (e.g. algorithm or procedure, solver, iterations, convergence criteria) must be included. The model module performs the translation from the real case to the set of algebraic systems of equations (e.g. finite volume methods, schemes, set of physic hypotheses). This translation is the discretization of the partial differential equations involved in the case. Since this work has been focused on how to solve such algebraic systems by using solvers and parallel computing, the above modules are compacted in the concept of the user layer.

- The solver layer performs the evaluation of the solution of the algebraic systems of equations. This layer specifies a wide range of solvers based on classic methods (e.g. Jacobi, TDMA, Gauss), decompositions (e.g. complete and incomplete LU decompositions), Krylov space based methods helped by preconditioners (e.g. CG, BiCGSTAB, GMRESR) and acceleration techniques (e.g. Algebraic MultiGrid and MultiResolution Analysis with wavelets).

- Each of these solvers contains some common basic operations which have been integrated in a more basic layer so called algebra layer. By this, we refer mainly to the algebraic operations between vectors and matrices (e.g. matrix-vector product, inner product between vectors, addition and subtraction between vectors, norm of vectors), and some transformations over vectors, matrix and maps of scalars (e.g. discrete Fourier transform, discrete wavelet transform).

- The above layers are supported by the communication layer. It is also called the communication layer because its main task is the communication inside a layer between processors: mainly the domain decomposition at the user, solver and algebra layers . If the CFD code is thought in sequential, this layer can be eliminate at all, but in parallel, part of the task are done at the communication layer. For example, the parallel algebraic operations are performed in the algebra layer except the exchange of data among the processors that is performed in the communication layer (e.g. the matrix-vector product, the norm of a vector and the maximum or the minimum value of a map distributed among the processors). The subroutines embedded in this layer contain calls to the parallel library MPI. It is worth noting that the from the programming point of view of the CFD code it is based on the SPMD (Single Program and Multiple Data) paradigm.

Most of these subroutines have been summarized for each layer in Fig. B.2.

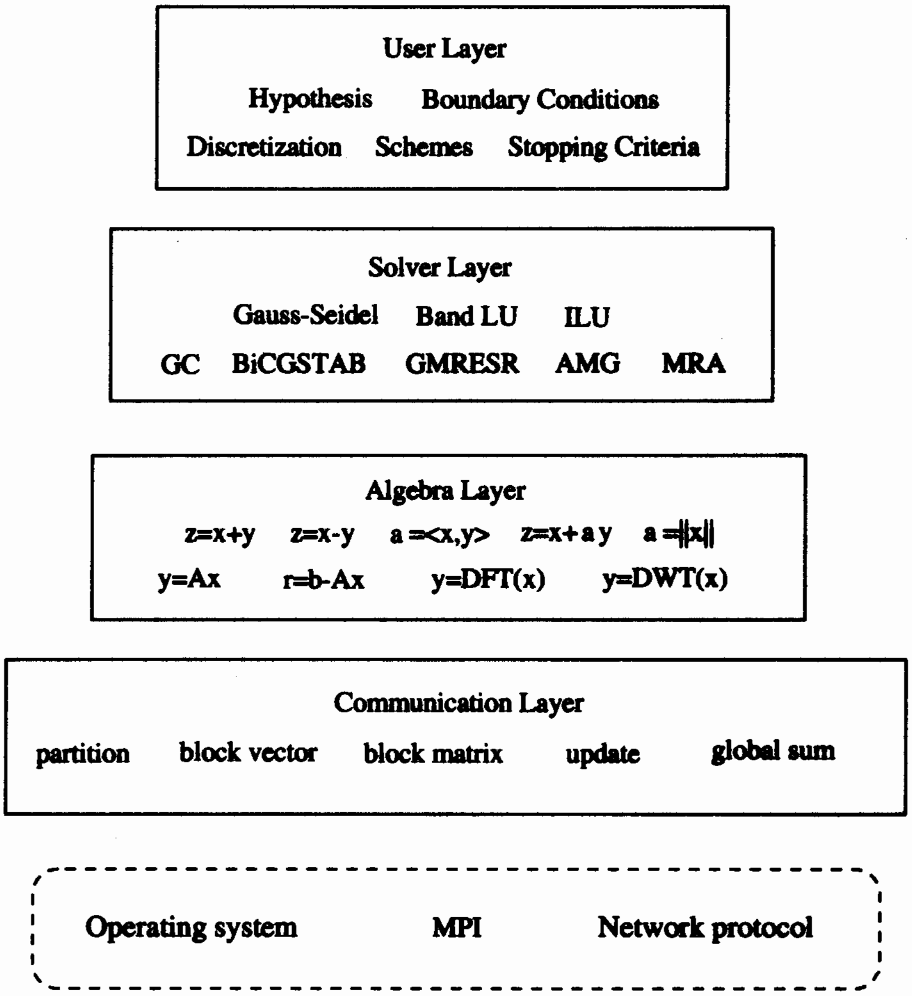

Figure B.2: Main subroutines of the CFD code grouped by layers.

Bibliography

[1] M. Costa, A. Oliva, and C. Pérez Segarra, "A three-dimensional numerical study of melting inside a heated horizontal cylinder," *Numerical Heat Transfer*, vol. Part A, no. 32, pp. 531–553, 1997.

[2] H. Schweiger, A. Oliva, M. Costa, and C. Pérez Segarra, "Numerical experiments on laminar natural convection in rectangular cavities with and without honeycomb-structures," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 5, no. 5, pp. 423–445, 1995.

[3] M. Soria, H. Costa, M. Schweiger, and A. Oliva, "Design of multifunctional ventilated facades for mediterranean climates using a specific numerical simulation code," in *Eurosun98 Conference, Slovenia*, 1998.

[4] C. Pérez Segarra, J. Cadafalch, J. Rigola, and A. Oliva, "Numerical study of turbulent fluid-flow through valves," in *International Conference on Compressors and Their Systems, London*, 1999.

[5] A. Ivancic, A. Oliva, C. Pérez Segarra, and M. Costa, "Heat transfer simulation in vertical cylindrical enclosures for supercritic rayleigh number and arbitrary side-wall conductivity," *International Journal of Heat and Mass Transfer*, vol. 42, no. 2, pp. 323–343, 1999.

[6] Myricom, "Creators of myrinet." http://www.myri.com.

[7] M. P. I. Forum, "Mpi forum, mpi: A message-passing interface standard." http://www.mpi-forum.org.

[8] M. Zedan and G. Schneider, "A coupled strongly implicit procedure for velocity and pressure computation in fluid flow problems," *Numerical Heat Transfer, part B*, vol. 8, pp. 537–557, 1985.

[9] X. Chuan and D. Keyes, "Nonlinear preconditioned inexact newton algorithms," 2000. Technical Report, Departement of Computer Science, University of Colorado, Boulder.

[10] H. Stone, "Iterative solution of implicit approximation of multidimensional partial differential equations," *SIAM Journal of numerical analysis*, vol. 5, pp. 530–558, 1968.

[11] M. Zedan and G. Schneider, "A three-dimensional modified strongly implicit procedure for heat conduction," *AIAA*, vol. 21, pp. 295–303, Feb 1983.

[12] M. Peric, "An efficient semi-implicit solving algorithm for nine-diagonal coefficient matrix," *Numerical Heat Transfer*, vol. 11, no. Part B, Fundamentals, pp. 251–279, 1987.

[13] S. Lee, "A strongly implicit solver for two dimensional elliptic differential equations," *Numerical Heat Transfer*, vol. 16, no. Part B, Fundamentals, pp. 161–178, 1989.

[14] I. Duff, A. Erisman, and J. Reid, *Direct Methods for Sparse Matrices*. New York: Oxford University Press, 1986.

[15] C. Craig, "Mgnet." Web site www.mgnet.org.

[16] P. Wesseling, *An Introduction to Multigrid Methods*. Pure and Applied Mathematics, John Wiley and Sons, 1992.

[17] P. Zeeuw, *Acceleration of Iterative Methods by Coarse Grid Corrections*. PhD thesis, Amsterdam University, 1997.

[18] C. Wagner, "Introduction to algebraic multigrid," 1999. Course Notes of an Algebraic Multigrid Course, Heidelberg University.

[19] J. Mora, M. Soria, and A. Oliva, "Uso de multigrid algebráico para la resolución de los sistemas de ecuaciones discretos obtenidos en transferencia de calor y dinámica de fluidos," in *XIII Congreso Nacional de Ingeniería Mecánica*, 1998.

[20] J. Williams and K. Amaratunga, "Introduction to wavelets in engineering," *I. J. Numerical Methods in Engineering*, vol. 37, pp. 2365–2388, 1994.

[21] R. Wells and X. Zhou, "Wavelet interpolation and approximate solutions of elliptic partial differential equations," 1992. Technical Report 92-03, Computational Mathematics Laboratory, Rice University.

[22] A. Rieder, R. Wells, and X. Zhou, "A wavelet approach to robust multilevel solvers for anysotropic elliptic problems," 1992. Technical Report 93-07, Computational Mathematics Laboratory, Rice University.

[23] R. DeVore and B. Lucier, "Wavelets," 1992. Technical Report, Preprint 92-026, University of Minnesota.

[24] E. Simons, "Domain decomposition methods for separable elliptic equations suitable for les of complex flows," 1995. Technical Report, von Karman Institute for Fluid Dynamics, Belgium.

[25] Y. Saad, "Krylov subspace method for solving large unsymmetric linear systems," *Math. Comput.*, vol. 37, pp. 105–126, 1981.

[26] Y. Saad and J. Zhang, "Diagonal threshold techniques in robust multi-level ilu preconditioners for general sparse linear systems," *Numerical Linear Algebra*, vol. 6, no. 4, pp. 257–280, 1999.

[27] A. Basermann, B. Reichel, and C. Schelthoff, "Preconditioned cg methods for sparse matrices on massively parallel machines," *Parallel Computing*, vol. 26, pp. 381–398, 1997.

[28] M. Grote and T. Huckle, "Parallel preconditioning with sparse approximate inverses," *SIAM J. Scientific Computing*, vol. 18, no. 3, pp. 838–853, 1997.

[29] M. Benzi and M. Tuma, "A sparse approximate inverse preconditioner for nonsymmetric linear systems," *SIAM J. Sci. Comput.*, vol. 19, pp. 968–994, 1998.

[30] S. Zweben, S. Edwards, B. Weide, and J. Hollingsworth, "The effects of layering and encapsulation on software development and quality," *IEEE Transactions on software engineering*, vol. 21, no. 3, pp. 200–208, 1995.

[31] W. Rohsenow, J. Hartnett, and E. Ganic, *Handbook of Heat Transfer Fundamentals*. Mc. Graww-Hill Book Company, 1985.

[32] S. Patankar, *Numerical Heat Transfer and Flow*. New York: Hemisphere, 1980.

[33] J.H.Ferziger and M.Peric, *Computational Methods for Fluid Dynamics*. Springer-Berlag, 1996.

[34] M. Darwish and F. Moukalled, "The normalized variable and space formulation methodology for high resolution schemes," *Numerical Heat Transfer*, vol. Part B, fundamentals, no. 26, pp. 79–96, 1994.

[35] J. Van Doormal and G. Raithby, "Enhancements of the simple method for predicting incompressible fluid flows," *Numerical Heat Transfer, part B*, vol. 7, pp. 147–163, 1984.

[36] K. Aksevol and P. Moin, "Large eddy simulation of turbulent confined coanular jets and turbulent flow over a backward facing step," 1992. Technical Report TR-63, Research Center of Turbulence,U.S.

[37] J. Mora, "A nine point formulation of poisson equation with a fourth order scheme," 2000. Technical Report of CTTC, Polytechnical University of Catalonia.

[38] M. Quispe, J. Cadafalch, M. Costa, and M. Soria, "Comparative study of flow and heat transfer periodic boundary conditions," in *ECCOMAS*, 2000.

[39] M. Soria, *Parallel Multigrid Algorithms for Computational Fluid Dynamics and Heat Transfer*. PhD thesis, Polytechnical University of Catalonia, 2000.

[40] R. Barrett *et al.*, "Templates for the solution of linear systems: Building blocks for iterative methods." Downloadable document at ftp.netlib.org/templates/templates.ps.

[41] S. Rump, "Ill conditioned matrices are component wise near to singularity," *SIAM Review*, vol. 41, no. 1, pp. 102–112, 1999.

[42] W. Spotz and G. Carey, "A high-order compact formulation for the 3d poisson equation," *Numerical Methods for Partial Differential Equations*, vol. 12, pp. 235–243, 1996.

[43] W. Press *et al.*, *Numerical Recipes in C. The art of Scientific Computing.* Cambridge University Press, 1994.

[44] C. Vuik, "Solution of the discretized icompressible navier-stokes equations with the gmres method," *I. J. for Numerical Methods in Fluids*, vol. 16, pp. 507–523, 1993.

[45] H. Vorst, "Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, pp. 631–644, 1992.

[46] M. Saad, Y. ahd Schultz, "Gmres: A generalized minimum residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856–869, 1986.

[47] H. Vorst and C. Vuik, "Gmresr: A family of nested gmres methods," 1991. Technical Report, Delft University of Technology.

[48] M. Trummer, "Iterative methods in linear algebra," 1997. Technical Report, Dept. Mathematics and Statistics. Simon Fraser University.

[49] H. Vorst and G. Sleijpen, "The effect of incomplete decomposition preconditioning on the convergence of conjugate gradients.," 1992. Incomplete Decompositions, Proceedings of the Eight GAMM Seminar, Delft University of Technology.

[50] S. Dupont and J. Marchal, "Preconditioned conjugate gradients for solving the transient boussinesq equations in three-dimensional geometries," *I. J. for Numerical Methods in Fluids*, vol. 8, pp. 283–303, 1988.

[51] G. Larrazábal and J. Cela, "Study of spai preconditioners for convective problems," 1999. Technical Report, Dept. Computer Architecture. Polytechnical University of Catalonia.

[52] B. Hutchinson and G. Raithby, "A multigrid method based on the additive correction strategy," *Numerical Heat Transfer*, vol. Part B, no. 9, pp. 511–537, 1986.

[53] I. Daubechies, *Ten Lectures on Wavelets.* Pure and Applied Mathematics, Philadelphia: SIAM, 1992.

[54] S. Mallat, "A theory for multiresolution signal decomposition the wavelet representation," *Communications in Pure and Applied Mathematics*, vol. 41, pp. 674–693, 1988.

[55] J. Xu and W. Shann, "Galerkin-wavelet methods for two-point boundary value problems," *Numerical Mathematics*, vol. 63, pp. 123–144, 1992.

[56] J. Dongarra and T. Dunigan, "Message passing performance of various computers," 1996. Technical Report, University of Tennessee. Oak Ridge National Laboratory.

[57] G. Editorial, "Parallel computing on clusters of workstations," *Parallel Computing*, vol. 26, pp. 295–303, 2000.

[58] J. Cadafalch *et al.*, "Domain decomposition as a method for the parallel computing of laminar incompressible flows," 1996. Proceedings of the Third ECCOMAS Computational Fluid Dynamics Conference.

[59] H. Gilbert *et al.*, "Sparse matrices in matlab: design and implementation," *SIAM J. Matrix Analysis Applications*, vol. 13, no. 1, pp. 333–356, 1992.

[60] J. Koseff, "On end wall effects in a lid driven cavity flow," *Journal of Fluid Engineering*, vol. 106, pp. 385–389, 1984.

[61] H. Gortler, "On the three-dimensional instability of laminar boundary layers on concave walls," *NACA Techinical mem.*, no. 1375, 1954.

[62] M. Soria, C. Perez-Segarra, and A. Oliva, "A direct algorithm for the efficient solution of the pressure-correction equation of incompressible flow problems using loosely coupled parallel computers," *(Submitted) Numerical Heat Transfer*, vol. Part B, Fundamentals, 2001.

[63] S. Paolucci, "Direct numerical simulation of two-dimensional turbulent natural convection in an enclosed cavity," *Journal of Fluid Mechanics*, vol. 215, pp. 229–262, 1990.