



Numerical modelling of free-surface flows.

Devising strategies for parallelism.

J.-M. Hervouet^a, J.-M. Janin^b & F. Marcos^b

^a Fluvial Hydraulics Group ^b Maritime Hydraulics Group
*Laboratoire National d'Hydraulique. Direction des Etudes et
Recherches. Electricité de France. 6 Quai Watier 78401
CHATOU. FRANCE*

Email: Jean-Michel.Hervouet@edf.der.fr

Jean-Marc.Janin@edf.der.fr Frederic.Marcos@edf.der.fr

Abstract

Three programs of the free surface flows solvers family are analysed in view of parallelism. A short description of their main features is done, with emphasis on the numerical methods used: finite elements with an E.B.E. (Element By Element) implementation. The analysis of the computational effort on scalar or vector machines is then carried out. Eventually, several strategies of parallelism are tested on different machines and compared. The results show that fine grain parallelism and matrix-vector decomposition can be efficient but only with a small number of processors, and with a maximum theoretical speed-up in the range 2-4. The rate of communications between processors, especially between workstations, is clearly a bottleneck.

Domain decomposition appears to be the most promising technique and a speed-up of more than 64 can be obtained on CRAY T3D with 128 processors.

1 Introduction

The PCECOWATER project (Parallel Computing of Environment COastal and lake shallow WATER dynamics) is part of the european ESPRIT project: Parallel Computing Initiative - II (PCI-II). This PCI-II project is a cluster of ten subprojects aiming at promoting within the European industries the Highly Parallel Computing technology and

culture. It is globally managed by CEPBA (Universitat Politecnica de Catalunya). Partners in our project are ENEL, CRS4 and Politecnico de Milano in Italy, Reiter Ltd and Kemijoki Oy in Finland. The goal is to provide the end-users with effective parallel simulation tools for large scale problems. The end-users may be large companies with super-computers as well as consultants with limited computing capabilities.

In this framework, it is planned to parallelize programs of the TELEMAC system, namely TELEMAC-2D for Shallow Water equations, TELEMAC-3D for Navier-Stokes equations with a free surface, and TOMAWAC for the wave action equation.

A number of test-cases were defined to assess the efficiency of different algorithms. One of them is the simulation with TELEMAC-2D of the Malpasset dam-break that occurred in France in 1959 (Figure 1). Other test-cases include flows in estuaries and storm surges in the Channel.

Tests were carried out on HP9000 workstations, on the parallel vector CRAY C98 (shared memory), on the massively parallel CRAY T3D (distributed memory), on a cluster of workstations CRAY CS6400.

Strong requirements for parallelism of our tools are that vectorization (with a speed-up ranging from ten to twenty) must be kept, and that the parallel versions of the software must be portable and as close as possible to the scalar versions.

2 Short description of the software

The TELEMAC system is based on the Finite Element technique with Element By Element (EBE) algorithms. It is fully vectorized for vector machines (see Hervouet [4]). The method of characteristics may also be used in some options. The numerical methods require all kinds of functionalities: matrix computations, vector assemblies, vector gradient computations, solvers for linear algebraic equations, etc... A key point is the solution of linear algebraic equations. Iterative methods are favoured within an EBE data structure since they require only matrix-vector products (see Hervouet & al. [5]).

EBE algorithms, mainly consisting of loops over the elements, *a priori* offer straightforward solutions for parallelism, but also raise specific problems, because of the 'grain size' of this parallelism.

All the modules in the TELEMAC system use the same library based on the same EBE techniques, but they solve different equations that may lead to different strategies of parallelism.

TELEMAC-2D solves the primitive shallow water equations written in the non-conservative depth-velocity form. Mass-conservative

schemes such as S.U.P.G. may be used to solve the continuity equation. The main features of TELEMAC 2D are the use of cartesian or spherical coordinates, the possibility of subcritical or supercritical regimes, a number of source terms such as wind stress, atmospheric pressure and Coriolis force, an equation of a tracer concentration, a treatment of tidal flats and many types of boundary conditions including free slip condition and incident waves.

TELEMAC-3D solves the Navier-Stokes equations with a free surface, together with the advection-diffusion equations of temperature, salinity and any other variables. Density effects, wind stress on the free surface, heat exchange with the atmosphere and the Coriolis force are included. The pressure is assumed hydrostatic and the free surface is computed by solving equations similar to the shallow-water equations. The solution procedure is made of three fractional steps: advection step, diffusion step and free surface-continuity-pressure step. The space discretization is done with prisms. The quadrangular sides of the prisms are vertical. The horizontal 2D projection of the mesh is made of triangles, and we only need to mesh the 2D horizontal domain and then duplicate it along the vertical. Domain decomposition is thus only done on 2D meshes.

TOMAWAC is a third generation spectral wave model for the computation of waves due to wind. It solves the wave action conservation equation in spherical or cartesian coordinates for infinite or finite depth. The result is a directional wave energy spectrum at any point of a 2D grid, as a consequence of wind forcing, bottom friction, white capping and non linear interactions (quadruplets). The combination of a 2D mesh and of the directional aspect (many wave directions and frequencies represented) results in a large demand of storage capacity. A consequence for parallelism is that not only computer time must be addressed, but also the memory space, because the number of unknowns, due to a discretisation of wave energy in frequency and direction, may be several hundreds per node.

3 Analysis of the computational effort:

Telemac-2D: The analysis has been performed with the help of Gunter Röth (CRAY Research France), on a CRAY C98. He worked on version 3.0 of TELEMAC-2D with a computation of a M2 tidal wave in the Channel and Atlantic (1210 time steps with a mesh of 9414 elements). The results show that 87 % of the time is spent in only 7 subroutines. These subroutines perform the element by element part of the matrix-vector product, the vector assembly after the element by

element computation, operations on vectors (sum, difference, etc) and on matrices (sum, difference, multiplication by a diagonal, etc), and computation of characteristics. The vector assembly vectorization has been obtained by a special numbering of the elements. As a matter of fact, this loop is of a scatter-gather type and is not purely vectorizable. The same problem is raised for parallelism, and requires special attention in shared memory. The computation of characteristics is made of complex loops with a number of clock-periods in the range 280 000 to 1 000 000. It is thus suitable for a simple parallel mode. The subroutines building the matrices are remarkably absent from the top seven subroutines. This is due to the EBE implementation and storage, without assembly, and with exact polynomial formulas replacing the classical Gauss quadratures. Other test-cases and other machines give similar results and clearly show that we must concentrate on the matrix-vector product, and, in a lesser extent, on the method of characteristics.

Taking into account the degree of parallelism of each subroutine and their percentage of time, the overall degree of parallelism of TELEMAT-2D, before any modification is estimated to be of 65 %. The theoretical speed-up given by the Amdahl law would be thus for fine grain parallelism:

CPU's	2	3	4	8
theoretical speed-up	1.48	1.76	1.95	2.32

We shall see that a coarse grain analysis applied to the matrix-vector product of the linear system in TELEMAT-2D gives in fact a degree of parallelism of 81%. The analysis of TELEMAT-3D will not be presented here for brevity but gives the same kind of conclusion. This is no wonder since the solution procedures are very close.

Tomawac: For TOMAWAC, the analysis has been performed on a Cray C98 super computer, on a CS6400 (cluster of SUN workstations) in scalar mode and on a HP-9000/700 scalar computer. The test-case is the generation and propagation of waves on a flat bottom (3200 elements, 144 time steps, frequency discretized by 25 values). The six top subroutines depend on the machine used, but always correspond to source terms computations and interpolations related to the propagation step. The method of characteristics does not appear because it is used only once at the beginning of the computation. The overall degree of parallelism on a parallel-vector super computer,

before any modification is estimated to be of 75%. The theoretical speed-up given by the Amdahl law would be thus:

CPUs	2	3	4	6
theoretical speed-up	1.6	2	2.29	2.66

On scalar computers, the overall degree of parallelism is higher than 90%. The theoretical speed up given by the Amdahl law would be:

CPUs	2	3	4	6
theoretical speed-up	1.82	2.5	3.1	4

4 Devising strategies for parallelism

Different techniques have been tested:

- Fine grain parallelism, i.e. parallelising every possible loop.
- Matrix-vector decomposition for TELEMAT-2D (see next paragraph).
- Domain decomposition techniques. We benefited here from the work performed at the University of Hannover by the team of Professor Zielke. The program is run on every sub-domain and the sub-domains communicate on their borders to ensure correct matrix-vector and dot products.

The presentation of all combinations between techniques, machines, and programs would be far too long and we report hereafter only on a selection of experiments:

Experiments with TELEMAT-2D on CRAY C98. This machine was used in parallel vector mode for testing shared memory strategies:

- parallelism at a "small grain" level. This is the most straightforward way for parallelising, as most of the computer time is spent in a small number of loops. The algorithm of the vector assembly must be rewritten, with a first assembly on each processor, and then a final sum.
- parallelism at the level of the matrix-vector product when solving the linear system at the end of every iteration in time. As a matter of fact, the three unknowns H, U and V (depth and the two components of the velocity) are coupled in the following linear system:

$$\begin{pmatrix} \text{AM1} & \text{BM1} & \text{BM2} \\ \text{CM1} & \text{AM2} & 0 \\ \text{CM2} & 0 & \text{AM3} \end{pmatrix} \begin{pmatrix} \text{H} \\ \text{U} \\ \text{V} \end{pmatrix} = \begin{pmatrix} \text{SM1} \\ \text{SM2} \\ \text{SM3} \end{pmatrix}$$

AM1, BM1, BM2, etc. are the classical matrices obtained in finite elements, e.g. mass-matrix, etc. The matrix is thus a block of nine matrices (two of them being nil) and the result of the matrix-vector product is a block of three vectors. The idea is to compute the three vectors independently on 3 different processors. Taking into account this coarse grain parallelism, the degree of parallelism of TELEMAC-2D becomes 81%. The theoretical speed-up now is:

Number of CPUs	2	3	4	8
theoretical speed-up	1.68	2.17	2.55	3.08

The "fine grain" and "coarse grain" approaches have first been tested separately, with poor results. Even if they are applied simultaneously, the computer times obtained on CRAY C98 are:

	elapsed time	speed-up
1 CPU	128.04 s	1
2 CPU	86.78 s	1.48
3 CPU	67.42 s	1.91
4 CPU	64.37 s	1.99
8 CPU	61.77 s	2.07

The theoretical speed-up is not obtained for two reasons: the "fine grain" parallelism requires too much communication, and the "coarse grain" is suitable only for 3 processors. The difference between 3 and 8 processors is thus not very significant. An additional problem is that the tests have been done on a machine dedicated to parallelism, with special access rights eliminating the competition with other programs. The results are even worse when the processors may happen to be busy because of another program.

Experiments with TOMAWAC on a cluster of workstations.

The CRAY SPARC 6400 is a computer developed by CRAY in collaboration with SUN. This machine is a shared memory machine.

The processors are SUN SPARC 60MHz. The key of the efficiency of this computer is the capability of the XDBus to transport information between memories and CPUs.

The compiler of the CS6400 detects and parallelizes the loops which are obviously parallel, there is thus no specific implementation. The results for TOMAWAC are listed in the following table:

Number of CPUs	2	3	4	6
theoretical speed-up (90%)	1.82	2.5	3.1	4
Observed speed-up	1.9	2.4	2.9	2.5

The efficiency of the parallelism seems correct up to 4 processors. The cost of communications then overcomes the benefit of parallelism. Small grain parallelism of TOMAWAC is however efficient on shared memory machines and is obtained without effort.

For distributed memory machines (and with a lot of processors) like Cray T3D, the preceding solution is not relevant because a high amount of memory should be available for each processor. Domain decomposition seems to be more interesting. Nevertheless, in TOMAWAC, we will not have a classical domain decomposition: an algorithm for distributed memory machines integrating could be the following:

- Propagation step: memory distribution by 2D nodes.
- Source term step: memory distribution by frequencies.

Experiments with TELEMAT-3D on a CRAY T3D. A domain decomposition technique could be tested on TELEMAT-3D thanks to a Ph. D. thesis currently being done at the University of Hannover (Hinkelmann & Zielke [3]). The Simon's Recursive Spectral Bisection technique is used (Simon [1]). To ensure a good load balance between processors, and in view of EBE techniques, it is necessary to ensure the same number of elements in each processor. For optimization of the communications it is necessary to reduce the number of points in the boundaries. This is achieved in two steps. First the mesh partitionning is done on the dual graph, every element being replaced by its center point, and those center points connected according to the element neighbourhood. Then the Kernighan-Lin heuristic technique is used to minimize the number of boundary points (elements are switched from

one sub-graph to the other when the number of common nodes is decreased).

Each processor calculates with its data its part of the element by element matrix. No communication is necessary for this. For the matrix-vector product, a problem arises with boundary points because every processor has only a part of the information, so a communication is necessary. In the scalar product, a node belonging to several sub-domains will be taken into account several times and its contribution must thus be multiplied by a correction factor. For the method of characteristics, option 3 explained in next paragraph has been chosen.

The language used for transmitting the data between processors is PVM. On CRAY T3D, a speed-up of 64 with 128 processors, for meshes of about 10000 elements, is obtained. However, the T3D processors are not vector processors and the speed-up compared with one processor of C98 is only 4.

The specific problem of characteristics. The method of characteristics consist of computing a number of trajectories (one for every point in the domain). For one characteristic pathline, the knowledge of all the domain may be necessary because the trajectory may cross all the domains (depending of the C.F.L. number). For shared memory parallelism, it is thus parallelised straightforwardly, with a degree of parallelism of about 98%. This was done in our tests on CRAY machines. The problem is much more difficult if domain decomposition is to be used on distributed memory machines. Several options may be chosen (see for example Creach [2]):

- 1) The basic approach consists of a computation of characteristics on every sub-domain. If one trajectory gets out of the sub-domain to enter another one, informations are transmitted to this latter sub-domain, to complete the computation.

- 2) An overlapping of sub-domains limits the need to exchange information. Coupled to a limitation of the C.F.L. number (this number is in fact the number of cells crossed per time-step), exchanges may even be suppressed.

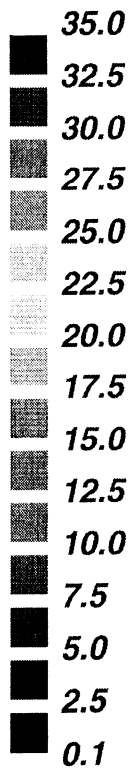
- 3) The time-step is limited so that no boundary of sub-domain is crossed. This is equivalent to a limitation of the C.F.L. number.

We discard other techniques that would not give the same solution than a scalar implementation.

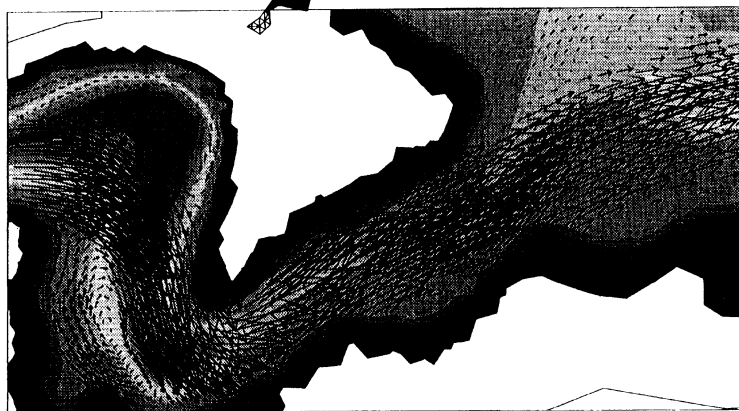
TELEMAC-2D: a test-case for parallelism

Malpasset dam-break flood wave
Water depth and velocity after 1200 s

DEPTH (m)



Position of
the dam



VELOCITY
→ 10

FIGURE 1



5 Conclusion

The tests on shared memory machines with a fine grain parallelism are not fully satisfactory, with the exception of TOMAWAC. For TELEMAC-2D and 3D, we can expect theoretically only a factor in the range 2 - 3, and the efficiency rapidly decreases with the number of CPUs. The optimum is obtained with 3 processors, when the matrix-vector product of the TELEMAC-2D system is split into 3 parts. However this trick is not valid for other programs in the system.

The domain decomposition technique seems to be the most promising and a parallel efficiency of more than 50% was reached with 128 processors. However the results highly depend on the network efficiency. In domain decomposition a full implementation of the method of characteristics is very tedious and one generally resorts to a limitation of the Courant number.

References

- [1] Simon, H.D.: Partitionning of unstructured problems for parallel precessing, NASA Ames research Center, Moffett Field, Colorado, USA 1994.
- [2] Creach V., Décomposition de domaines pour problèmes hyperboliques sur ordinateurs parallèles. Rapport de stage de fin d'études de FSIMACS 1994.
- [3] Hinkelmann R., Zielke W., A parallel 3D Lagrangian-Eulerian Model for the Shallow Water and Transport Equations. To appear in Notes on Numerical Fluid Mechanics, Verlag Vieweg.
- [4] Hervouet, J.-M., Telemac, a fully vectorised finite element software for shallow water equations. Computer Methods and Water Resources. Rabat, Morocco. 7-11 Oct. 1991.
- [5] Hervouet J.-M., Janin J.-M., Moulin C, Preconditioning and solving linear systems for the computation of free surface flows. X International Conference on Computational Methods in Water Resources. Heidelberg. Allemagne. 19-22 Juillet 1994.