Parallel Flood Modeling Systems*

L. Hluchy, V. D. Tran, J. Astalos, M. Dobrucky, G. T. Nguyen

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 842 37 Bratislava, Slovakia
viet.ui@savba.sk

D. Froehlich

Parsons Brinckerhoff Quade and Douglas, Inc
909 Aviation Parkway, Suite 1500
Morrisville, North Carolina 27560 USA
Froehlich@pbworld.com

Abstract. Flood modeling is a complex problem that requires cooperation of many scientists in different areas. In this paper, the architecture and results of the ANFAS (Data Fusion for Flood Analysis and Decision Support) project is presented. This paper also focuses on the parallel numerical solutions of flood modeling module that are the most computational-intensive part of the whole ANFAS architecture.

1 Introduction

Over the past few years, floods have caused widespread damages throughout the world. Most continents were heavily threatened. Therefore, modeling and simulation of floods in order to forecast and to make necessary prevention is very important. There are several flood modeling software systems like MIKE [1], FLOWAV [2], however, they are suitable for small problems only. It is the motivation of the ANFAS (Data Fusion for Flood Analysis and Decision Support) project [16] that is supported by European Commission within IST Fifth Framework Programme.

The ANFAS project will:

- use data from the most advanced acquisition technology; in particular remote sensing imagery (optical, radar, interferometry radar) will be incorporated into a conventional Geographical Information System (GIS) database;

* This work is supported by EU 5FP ANFAS IST-1999-11676 RTD and the Slovak Scientific Grant Agency within Research Project No. 2/7186/20

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 543–551, 2002.

- develop advanced data processing tools for scene modeling and flood simulation; computer vision and scientific computing will be used together in order to take into account information extracted from real images for the calculation of parameters for the simulation model;
- have strong end-users involvement through the definition of the objectives, the conception of the system, the evaluation of the simulation results; it ensures that the system answers to "terrain needs" and are well adapted in practical use;
- involve industrial partners for the realization and design of the final system;
- implement a software modular, i.e. one composed of easily interchangeable blocks.

The functions of ANFAS system will be:

- to perform flood simulation: given a scenario, water flow propagation is simulated; the user can interact with the scene that models the real site in order to add/remove dikes or other man-made;
- to assess flood damage: flood assessment can be done using either the simulation results or the remote sensed images of a real flood event;
- at a prospective level, to analyze floodplain morphology including riverbed and subsurface properties changes due to repetitive floods for a given site.

The output of the system will be:

- visualization: consisting of the bi-dimensional maps of the extent of the flood; if simulation is performed, time sequence maps retracing the propagation of the simulated flood will be provided;
- direct impact assessment, evaluation of the whole affected area; regional and statistical analyses over the affected area; simple socio-economic assessment of damage directly caused by flood.

2 ANFAS Architecture

The ANFAS system is based on a 3-tier architecture (**Fig. 1**):

- the ANFAS client,
- the ANFAS core server,
- the servers – or wrappers – allowing to “connect” (possibly in remote access) external features such as the database, the models and possibly additional processes (e.g. computer vision ones).

Short description of each component follows:

- Database Server
  - GIS ArcView: here all data that represent the sites are stored;
  - GIS access component: responsible for GIS management.
- Modeling server
  - Numerical models such as FESWMS Flo2DH
  - Model access component: it pilots the numerical models
- ANFAS Core Server
  - Application server: it is the bridge between the client and the ANFAS core server

- Data manager: it handles the user sessions (i.e. set of data), especially for what concern scenarios (recording, loading) and collections
- File exchange component: it is responsible for file transfers between servers
- User management component: it manages all end users allowed to perform flood simulations
- Map server component: responsible for the manipulation of geographic data
- Model manager: it drives the numerical models
- Co-operative, explanatory component: it brings functions for the impact assessment.

- ANFAS Client
  - Modelling preparation: this component is responsible for the phase of data sets preparation
  - Modelling activation and follow-up: it permits to follow-up the execution of the numerical models
  - Results exploitation: it brings functions for manipulating the results of a model

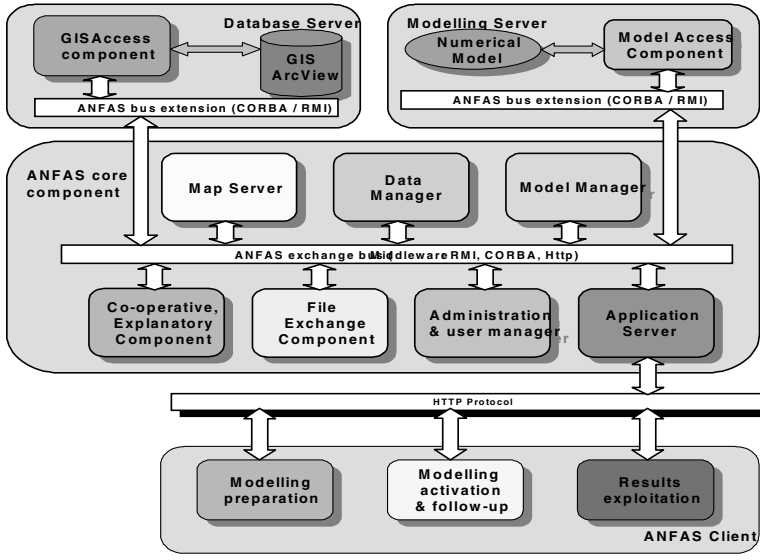

Fig. 1 ANFAS architecture

3 Modeling and Simulation at Vah River Pilot Site

Vah river is the biggest river in Slovakia, therefore it was chosen as a pilot site for the ANFAS project. The following input data are required for modeling and simulation:

- Topographical data: There are several sources of input topographical data: orthophotomaps, river cross-sections, LIDAR (Light Detection and Ranging) data. GIS (Geographical Information System) is used to manage and combine these data, and to product a final map for simulation (e.g. orthophotomaps are better for identifying objects, LIDAR cannot measure river depths).

- Roughness conditions: These data cannot be directly obtained, but are derived from orthophotomaps.
- Hydrological data: These data are obtained from monitoring of past flood events. These data can be used as boundary conditions as well as for calibration and validation of models.

Creating and calibrating models is a challenging process that can be done only by experts in hydrology. SMS (Surface-water Modeling System) [3] provides a convenient graphical interface for pre- and post-processing that can help the experts to speedup the modeling process. **Fig. 2** shows LIDAR data and orthophotomaps in SMS graphical interface.

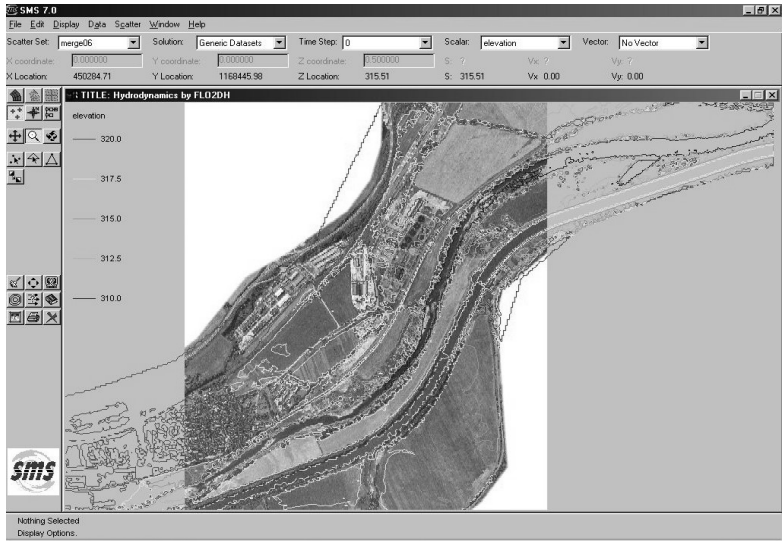

Fig. 2 LIDAR and orthophotomaps in SMS

The main modeling module in ANFAS is FESWMS (Finite Element Surface-Water Modeling System) Flo2DH [15] which is a 2D hydrodynamic, depth averaged, free surface, finite element model supported by SMS. Flo2DH computes water surface elevations and flow velocities for both super- and sub-critical flow at nodal points in a finite element mesh representing a body of water (such as a river, harbor, or estuary). Effects of bed friction, wind, turbulence, and the Earth's rotation can be taken into account. In addition, dynamic flow conditions caused by inflow hydrographs, tidal cycles, and storm surges can be accurately modeled. Since Flo2DH was initially developed to analyze water flow at highway crossings, it can model flow through bridges, culverts, gated openings, and drop inlet spillways, as well as over dams, weirs, and highway embankments. Flow through bridges and culverts, and over highway embankments can be modeled as either 1D or 2D flow. Flo2DH is ideal for modeling complex flow conditions at single- and multiple-bridge and culvert roadway crossings, such as the combination of pressurized flow, weir overflow, and submerged weir overflow, which are difficult to evaluate using conventional models. It is especially well-suited at analyzing bridge embankments and support structures to reduce and eliminate scour during flooding.

4 Numerical Solution of Flo2DH

In Flo2DH, the governing system of differential equations that describe the flow of water in rivers, floodplains, estuaries, and other surface-water flow applications are based on the classical concepts of conservation of mass and momentum. Flo2DH uses the Galerkin finite element method (FEM) [18] to solve the governing system of differential equations. Solutions begin by dividing the physical region of interest into sub-regions called elements. Two-dimensional elements can be either triangular or quadrangular in shape, and are defined by a finite number of node points situated along its boundary or in its interior. Approximations of the dependent variables are submitted into the governing equations, which generally will not be satisfied exactly, thus forming residuals, which are weighted over the entire solution region. The weighted residuals, which are defined by equations, are set to zero, and the resulting algebraic equations are solved for the dependent variables. In Galerkin's method, the weighting functions are chosen to be the same as those used to interpolate values of the dependent variables within each element.

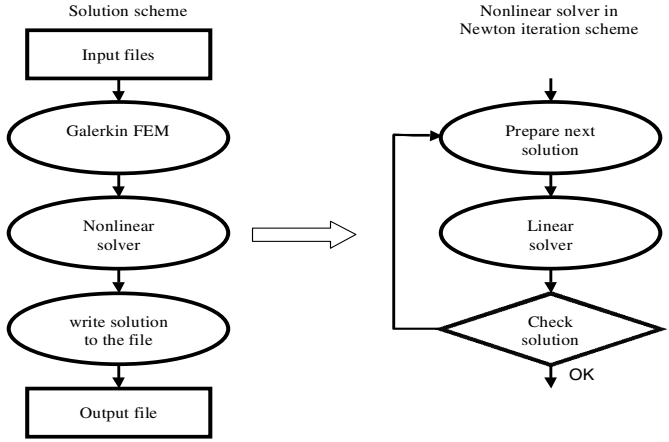

Fig. 3 Numerical scheme of Flo2DH

The weighting process uses integration, which is carried out numerically using Gaussian quadrature on a single element. Repetition of the integration for all elements that comprise a solution region produces a system of nonlinear algebraic equations when the time derivatives are discretized. Because the system of equations is nonlinear, Newton iteration, or its variation, is applied and the resulting system of linear equations is solved using a linear solver. Full-Newton iteration is written as follows:

$$M(a) = b$$
$$J(a_0)(a - a_0) = -(b - M(a_0))$$
$$J(a_0)\,\Delta a_0 = -R(a_0)$$
$$a_1 = a_0 + \omega \Delta a_0 \quad , \quad \omega \in (0,2);$$

generally,

$$J(a_j)\,\Delta a_j = R(a_j)$$
$$a_{j+1} = a_j + \omega \Delta a_j \qquad j = 0,1,2,\ldots n$$

where a_j is a newly computed solution vector, $J(a_j)$ is a Jacobian matrix computed for a_j, $R(a_j)$ is a residual load vector, Δa_i is the incremental solution value. An updated solution is computed by a relaxation factor ω_j. Solving the system of linear equations $J(a_j)\,\Delta a_j = R(a_j)$ is the most time- and memory-consuming part of Flo2DH. The complete numerical scheme of Flo2DH is shown in **Fig. 3**.

5 Parallelizing Flo2DH

Simulation of floods is very computation-expensive. Several days of CPU-time may be needed to simulate large areas. For critical situations, e.g. when a coming flood is simulated in order to predict threatened areas, and to make necessary prevention, long computation times are unacceptable. Therefore, using HPCN platforms to reduce the computational time of flood simulation is imperative [4] [12]. The HPCN version of FESWMS Flo2DH not only reduces computation times but also allows simulation of large-scale problems, and consequently provides more reliable results.

Solving the systems of linear equations is the most time- and memory-consuming part of Flo2DH. It is also the part of Flo2DH that is most difficult to parallelize because the matrices generated by Galerkin FEM method is large, sparse and unstructured. Galerkin FEM method has small computation time in comparison with linear solvers and it is trivially parallelized by domain decomposition. Therefore, the following part of this paper focuses on parallel linear solvers.

In the original sequential version, a direct solver based on a frontal scheme is used. The parallelism in the algorithms is relatively low; therefore it is not suitable for high performance platforms [10]. Furthermore, direct solvers need large additional memory for saving LU triangular matrices. For large simulation areas, there is not enough physical memory for saving the triangular matrices, so parts of the matrices have to be saved on hard disks and cause performance penalty.

In order to achieve better performance, iterative linear solvers based on Krylov subspace methods [19] are used in parallel version of Flo2DH instead of the direct solver. These iterative solvers consist of matrix and vector operations only; thus, they offer large potential parallelism. In comparison with the direct solvers, iterative solvers are often faster and require less additional memory. The only drawback of iterative solvers is that they do not guarantee convergence to a solution. Therefore, preconditioning algorithms [20] are used to improve the convergence of iterative algorithms. Recent studies [5], [8] show that the combination of inexact Newton iteration [7] with Krylov methods and additive Schwarz preconditioners [17] can offer high performance and stable convergence. PETSC [11] library, which is developed in Argonne National Laboratory, is used in implementation of parallel Flo2DH version. It is available for both supercomputers and clusters of workstations.

6 Case Study

Input data are generated by GIS; the data from LIDAR, cross-sections and orthophotomaps are combined. Hydrological data are provided by Vah River Authority which monitors and records all past events in Vah river. Roughness is defined based on image processing procedures of orthophotomaps, which divided the simulated pilot site into areas with the same coverage.

6.1 Hardware Platform

Experiments have been carried out on a cluster of workstations at the Institute of Informatics. The cluster consists of seven computational nodes, each has a Pentium III 550 MHz processor and 384 MB RAM, and of a dual processor master node and 512 MB RAM. All of the nodes are connected by an Ethernet 100Mb/s network. The Linux operating system is used on the cluster. Communication among processes is done via MPI [9].

6.2 Experimental Results

The steady-state simulation converged after 17 Newton iterations. The original Flo2DH with a direct solver takes 2801 seconds for the simulation. The parallel version of Flo2DH takes 404 seconds on a single processor and 109 seconds on 7 processors (Tab. 1). One of the reasons why the parallel version (with iterative solvers) is much faster than original version (with a direct solver) is that the direct solver in the original version needs a lot of additional memory for saving LU triangular matrices. As there is not enough physical memory the matrices, a part of them have to be saved on hard disk, which causes performance penalty. Speedup of the parallel version on our cluster is shown in **Fig. 4.**

Tab. 1 Simulation times (in seconds)

Original version	Parallel version						
	1 processor	2 p.	3 p.	4 p.	5 p.	6 p.	7 p.
2801	404	220	150	136	122	118	109

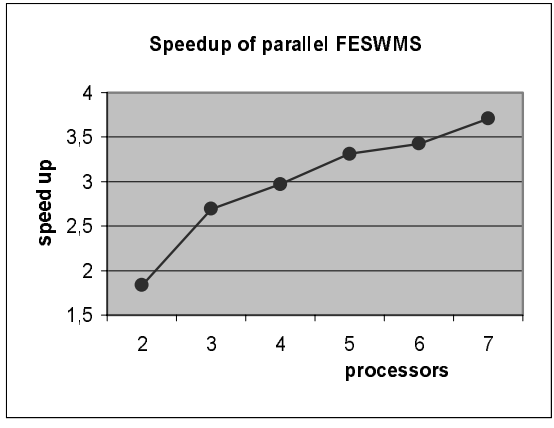

Fig. 4 Speedup of the parallel version of FESWMS

The experiments have revealed that from the implemented Krylov interactive algorithms in PETSTC, Bi-CGSTAB [13] is the fastest and relatively stable, especially when it is used with additive Schwarz/ILU preconditioner. GMRES [6] has the most stable convergence; however, it is much slower than Bi-CGSTAB, and CGS [14] has unstable convergence.

Fig. 5 Simulation results

An additive Schwarz/ILU preconditioner was used in the experiments. However, the preconditioner has several parameters that have to be tuned for better performance. Finding optimal values for these parameters required a large number of experiments and it is a part of future work on parallel version of FESWMS Flo2DH.

7 Conclusion

In this paper, we have presented ANFAS architecture for 3-tier flood modeling system. The paper was focused on parallelization of numerical module Flo2DH of the ANFAS architecture. Experiments with real data from Vah river pilot site show good speedups of our parallel version of Flo2DH. Similar results are also achieved on real data from Loire river pilot site, France. Experiments on larger computer systems (Origin 2800 with 128 processors and Linux clusters with 200 nodes) are planned.

References

1. FLDWAV: 1D flood wave simulation program
   http://hsp.nws.noaa.gov/oh/hrl/rvrmech/rvrmain.htm
2. Mike21: 2D engineering modeling tool for rivers, estuaries and coastal waters
   http://www.dhisoftware.com/mike21/
3. SMS: 2D surface water modeling package
   http://www.bossintl.com/html/sms_overview.html
4. V. D. Tran, L. Hluchy, G. T. Nguyen: Parallel Program Model and Environment, ParCo'99, Imperial College Press, pp. 697-704, August 1999, TU Delft, The Netherlands.
5. I.S. Duff, H.A. van der Vorst: Developments and Trends in the Parallel Solution of Linear Systems, Parallel Computing, Vol 25 (13-14), pp.1931-1970, 1999.
6. Y. Saad, M. H. Schultz: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Scientific and Statistical Computing, Vol. 7, pp. 856-869, 1986.
7. R.S. Dembo, S.C. Eisenstat, T. Steihaug: Inexact Newton methods. SIAM J. Numerical Analysis, Vol. 19, pp. 400-408, 1982.
8. W.D. Gropp, D.E. Keyes, L.C. McInnes, M.D. Tidriri: Globalized Newton-Krylov-Schwarz Algorithms and Software for Parallel Implicit CFD. Int. J. High Performance Computing Applications, Vol. 14, pp. 102-136, 2000.
9. MPICH - A Portable Implementation of MPI
   http://www-unix.mcs.anl.gov/mpi/mpich/
10. Selim G. Akl: Parallel Computation Models and Methods, Prentice Hall 1997.
11. PETSc The Portable, Extensible Toolkit for Scientific Computation. http://www-fp.mcs.anl.gov/petsc/
12. Pfister G.F.: In Search of Clusters, Second Edition. Prentice Hall PTR, ISBN 0-13-899709-8, 1998.
13. H. Vorst: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Scientific and Statistical Computing, Vol. 13, pp. 631-644, 1992.
14. P. Sonneveld: CGS, a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J. Scientific and Statistical Computing, no. 10, pp. 36-52, 1989.
15. FESWMS - Finite Element Surface Water Modeling.
   http://www.bossintl.com/html/feswms.html
16. ANFAS Data Fusion for Flood Analysis and Decision Support.
   http://www.ercim.org/anfas/
17. X.C. Cai, M. Sarkis: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Scientific Computing Vol. 21, pp. 792-797, 1999.
18. D. Froehlich: Finite element surface-water modeling system: Two-dimensional flow in a horizontal plane. User manual.
19. Y. Saad, H. Vorst: Iterative Solution of Linear Systems in the 20-th Century. J. Comp. Appl. Math, Vol. 123, pp. 1-33, 2000.
20. K. Ajmani, W.F. Ng, M.S. Liou: Preconditioned conjugate gradient methods for the Navier-Stokes equations. J. Computaional Physics, Vol. 110, pp. 68-81, 1994.