caffa3d.MB: a fully implicit finite volume method for solving the 3D incompressible Navier-Stokes equations in complex geometry

Sergio Nesmachnow, Gonzalo Rodríguez
uy-Grid Team
Universidad de la República, Uruguay

Joint GISELA/EPIKH School for Application Porting

Valparaíso, Chile
November, 2010

Abstract
‘caffa3d.MB’ is a project under development since 2004, which includes a finite volume incompressible 3D fluid flow solver, with several extensions. caffa3d.MB is implemented in FORTRAN, mainly using standard FORTRAN 77 features, but also including some FORTRAN 95/90 extensions. Parallel programming techniques have been applied to caffa3d.MB through OpenMP directives to take advantage of shared memory architectures, while a distributed memory parallel implementation using MPI is currently in development phase.
1. Introduction

The software package ‘caffa3d.MB’ is a three-dimensional extension of the original ‘caffa.f’ two-dimensional code to solve the Navier-Stokes equations by Ferziger and Peric [1]. The solver in caffa3d.MB includes block-structured meshes for the spatial discretization and several other features in order to deal with complex geometries.

This article provides a brief description of the caffa3d.MB software, describing how the application works, providing a high-level pseudocode of its workflow, and also listing the hardware and software requirements to perform its execution (including libraries, compilers, etc.).

Next section provides the general description of caffa3d.MB and section 3 presents a brief performance analysis of the main time-consuming tasks in caffa3d.MB. Section 4 summarizes the requirements on hardware and software to execute the model, while specific comments about the parallelization strategies are summarized in section 5. Last, section 6 describes the exploration of parameters space proposed to be implemented in parallel in a grid environment.

2. General Program Description

The software ‘caffa3d.MB’ is a FORTRAN implementation of a fully implicit finite volume method for solving the 3D incompressible Navier-Stokes equations in complex geometry. Spatial discretization is based on block-structured, non-orthogonal, body fitted, collocated meshes with first order (UDS) and second order (CDS) schemes for the convective terms. For the time discretization fully implicit two-level first order (implicit backward Euler) and three-level second order schemes are available. The SIMPLE algorithm is implemented for the coupling between velocity and pressure.

A generic transport equation is provided for solving transport of different scalars. In particular, heat transport is implemented together with buoyancy effects on the momentum equations.

caffa3d.MB is mainly implemented using standard FORTRAN 77 features, but also including some FORTRAN 95/90 extensions.

The main tasks performed in the basic core of caffa3d.MB are the resolution of the momentum and mass balance equations. Algorithm 1 presents the pseudocode of the basic core of caffa3d.MB model.

---

Algorithm 1: Basic core of caffa3d.MB (pseudocode)

0. Read problem name from command line
1. CALL INIT: Read problem data from <name>.ini file
   1.1 CALL INITGEO: Read block data <name>.bck file
   1.2 CALL READMESHS: Read mesh data <name>.grd file(s)
2. Start Main time loop
   2.1 Start Relaxations loop
      2.1.1 CALL CALCUVW: Build and iterate UVW equations.
         2.1.1.1 CALL GRADFI: Compute UVW gradients
         2.1.1.2 CALL FLUXUVW: Compute momentum fluxes
         2.1.1.3 CALL SIPSOLO: Call Solver for U,V,W
      2.1.2 Build P equation. Call Solver for P
         2.1.2.1 CALL GRADFI: Compute P gradients
         2.1.2.2 CALL FLUXM: Compute mass fluxes
         2.1.2.3 CALL SIPSOLO: Call Solver for P
         2.1.2.4 Correct velocities and mass fluxes
   2.1.3 Check Convergence
   2.1.4 Print out iteration information
2.2 Print out time step information
2.3 Print out intermediate results
3. Print out final results
3. **Performance analysis: principal time-consuming tasks in caffa3d.MB**

Within caffa3d.MB there are three main groups of time consuming tasks:

1. Routines for updating the heptadiagonal equation coefficient matrix for each equation (momentum, mass balance, etc).
2. Routines for computing the gradients of each field through gauss theorem.
3. Iterations in the heptadiagonal equation system for each equation, solving the system using the Stone ILU SIP Solver [2].

Routines in the groups 1 and 2 mainly comprise visiting each cell interface in an ordered fashion (East faces Loop, North faces Loop, Top faces Loop) and computing the flux contributions to either the coefficient matrices or the gradients. A neat example of this type of tasks is the GRADFI routine which organizes the computation of gradients.

The operations in group 3 mainly involve a series of backward and forward substitutions (solving upper triangular and lower triangular subsystems of equations). In fact the routine SIPSOL is the single most time consuming routine in the code, accounting for up to 30% of the total computing time.

4. **Requirements**

The caffa3d.MB source code currently supports being compiled with GNU gFORTRAN, Intel ifort, and Portland Group gf90 compilers. No additional libraries are required for either the compilation or the solver execution.

RAM requirements vary according to the mesh size, with some cases reaching up to 6 GB RAM. For the selected test cases, RAM requirements are below 1 GB RAM.

5. **Parallelization strategies**

In the current version of caffa3d.MB, OpenMP directives are included in the code for implementing shared memory parallelism at mesh-block level. Whenever a computation must be applied to all inner cells in the domain, each mesh block is attributed to one thread and the computation is done in parallel. The treatment of boundary conditions and mesh block interfaces is done serially due to potential dependency issues. The linear solver SIPSOL has been modified to accommodate this mesh block strategy without sensible deterioration of convergence properties.

An MPI version is currently under development, with parallelization at block-regions level. Another interesting option for improving the practical applicability of caffa3d.MB using high performance computing techniques is to perform the exploration of the parameters space in a parallel/distributed environment, as it is described in the next section.

6. **Exploration of caffa3d.MB parameters space in a grid environment**

In engineering practice is often not enough to run one single simulation with a specific set of values for the parameters involved in the problem. The user usually needs to scan a large phase space –defined by a range of values– for a given number of parameters. For example, a given flow problem might exhibit critical behaviors at different values of the Reynolds number and the user might be interested in running the simulation for Reynolds numbers in, for instance, the range [50,3000], incrementing the value of the Reynolds number in steps of 50 units. To perform this experimental analysis, about 60 independent runs need to be executed. Afterwards, the user might want to refine some interesting region of the parameter space at, say, intervals of 10 units in the value of the Reynolds number. Thus, an efficient way of organizing and distributing the execution of a series of simulations to explore varying values of a set of parameters would be a useful tool in practice. This is the basis of the specific proposal to implement in parallel in a grid environment.
The proposed application is conceived as a master-slave parallel program in which the master process controls the parameters exploration, which is specifically performed by a set of distributed slave processes. The master process controls the search by performing the domain decomposition and assigning each slave the parameters values to execute caffa3d.MB, while it also takes the decision of when to refine a promising region of the parameter space. A dynamic model will be used for the application: the slave processes will wait for specific data to be sent by the master to perform the caffa3d.MB execution. Load balancing strategies have to be applied by the master in order to deal with different simulation times for each of the parameter values explored. The proposed model is well suited for a grid environment, since the slave processes will run independent tasks with no communications among them.

7. Conclusions

This document has presented the main details of the caffa3d.MB model for solving the 3D incompressible Navier-Stokes equations in complex geometry using a fully implicit finite volume method. The main features of the application have been described, along with the requirements to compile and execute the caffa3d.MB code. In addition, the opportunities for applying parallel computing techniques have been presented, and a specific master-slave parallel version has been conceived to perform the exploration of the parameters space in a parallel/distributed environment.

References