



ViscoSolve: Porting, Optimization and Performance Analysis of Viscoelastic Fluid Flow Code

O. Akinci¹, M. Sahin², B.O. Kanat³

¹National High Performance Computing Center of Turkey (UHeM), Istanbul Technical University (ITU), Ayazaga Campus, UHeM Office, Istanbul 34469, Turkey

²ITU, Ayazaga Campus, Faculty of Aeronautics and Astronautics, Istanbul 34469, Turkey

³ITU, Ayazaga Campus, Computer Engineering Department, Istanbul 34469, Turkey

Abstract

ViscoSolve is a stable unstructured finite volume method for parallel large-scale viscoelastic fluid flow calculations. The code incorporates the open-source libraries PETSc and MPI for parallel computation. In this whitepaper we report work that was done to investigate scaling the performance of the ViscoSolve code.

1. Introduction

ViscoSolve is a new stable unstructured finite volume method that is presented for parallel large-scale simulation of viscoelastic fluid flows [1]. The numerical method is based on the side-centered finite volume method where the velocity vector components are defined at the mid-point of each cell face, while the pressure term and the extra stress tensor are defined at element centroids. The main advantage of ViscoSolve is that it leads to a stable numerical algorithm and does not require any *ad-hoc* stabilization techniques. Therefore, the continuity equation is satisfied at the machine precision [1]. The log-conformation representation [2] has been implemented in order to improve the limiting Weissenberg number in the proposed finite volume method [1]. The time stepping algorithm used decouples the calculation of the polymeric stress by solution of a hyperbolic constitutive equation from the evolution of the velocity and pressure fields by solution of a generalized Stokes problem [1]. The resulting algebraic linear systems are solved using the FGMRES(m) Krylov iterative method with the restricted additive Schwarz preconditioner for the extra stress tensor and the geometric non-nested multilevel preconditioner for the Stokes system.

ViscoSolve has a potential to accurately predict the onset of viscoelastic fluid flow instabilities which is very important for many industrial polymer processing where output quality constraints require that operating conditions should be maintained in the stable flow regime. Therefore, the current algorithm has a potential for increasing the polymer processing speed while reducing the cost. In addition, it will help to reduce a huge industrial polymer processing wastage due to product defects which are caused by viscoelastic fluid flow instabilities. Furthermore, the present calculations will give very detailed information on the nature of viscoelastic fluid flow instabilities as well as the validations of current viscoelastic models. Although industrial polymer processing applications require more complex multi-mode models, this is possible by adding several source terms to the Oldroyd-B model.

ViscoSolve has been written based on MPI. Preconditioned iterative solvers are based on the PETSc library for improving the efficiency of the parallel code. The code has four main steps. The first one is that grid subroutine that reads mesh files and specifies the required boundary conditions. The second step consists of the setup subroutine that constructs the algebraic matrices for the Stokes system and sets up the geometric non-nested multigrid solver for the Stokes system. These first and second steps are called only once at the startup. The third step is the numerical time integration part which dominates the solver calculation time. A typical time dependent calculation requires approximately 1,000 to 10,000 iterations to obtain the final steady-state or the time-periodic

state.

In previous work, the code also has been ported to SGI Altix 3000 (1.3 GHz, Itanium 2) machine available at the Faculty of Aeronautics and Astronautics of ITU with 32 nodes and run with different data from the with in the present work [1].

The aim of this work was analyze the performance of the code on the PRACE Tier-1 system Karadeniz (Intel Xeon 5550) at National High Performance Computing Center of Turkey (UHeM) in Turkey and implement possible improvements to the application source code. During our work we focused on following tasks:

- Demonstrating capability of the code on the Karadeniz system,
- Examining the scalability of the code.

The Karadeniz system utilizes a Nehalem cluster architecture. It has 64 compute nodes, each offering two 4 core Intel Xeon 5550 (Nehalem) processors with a nominal clock speed of 2.67 GHz. Each node of Karadeniz has 24 GB RAM, they connect to each other with InfiniBand (20Gbps). A DDN SFA-10K disk system with Lustre filesystem is used as scratch filesystem.

The parallel performance of the ViscoSolve algorithm is presented in Table 1 for the initial setup phase and the total solve time for the first 4 iterations using the unstructured computational mesh with 1,279,200 hexahedral elements given in [3, 4] for the viscoelastic fluid flow past a confined circular cylinder. The first column represents the number of processors, the second column shows the initial setup time for the construction of the matrices for the Stokes system and the time required for the construction of the two-level non-nested multigrid method, the third column shows the total solve time for the first four iterations (one number for each iteration), the fourth column shows the number of multigrid (MG) iterations for the Stokes system and the fifth column represents the number of iterations for the restricted additive Schwarz preconditioner to solve the extra stress tensor. The final sixth column “N” represents the number of subdomain ratio between fine and course mesh levels. Total times of the first four iterations are shown in Figure 1.

# of Cores	Setup time	Total solve time (secs)	MG iteration #	ASM ILU(0) iteration #	N
64	27	98/131/106/99	100/72/86/77	0/8/8/8	2
128	14	53/48/48/49	100/74/75/77	0/9/9/9	4
256	8	30/27/31/28	100/76/78/77	0/9/9/9	8

Table 1 The performance of the ViscoSolve on the Karadeniz system (Intel Xeon 5550) available at the National Center for High Performance Computing in Turkey.

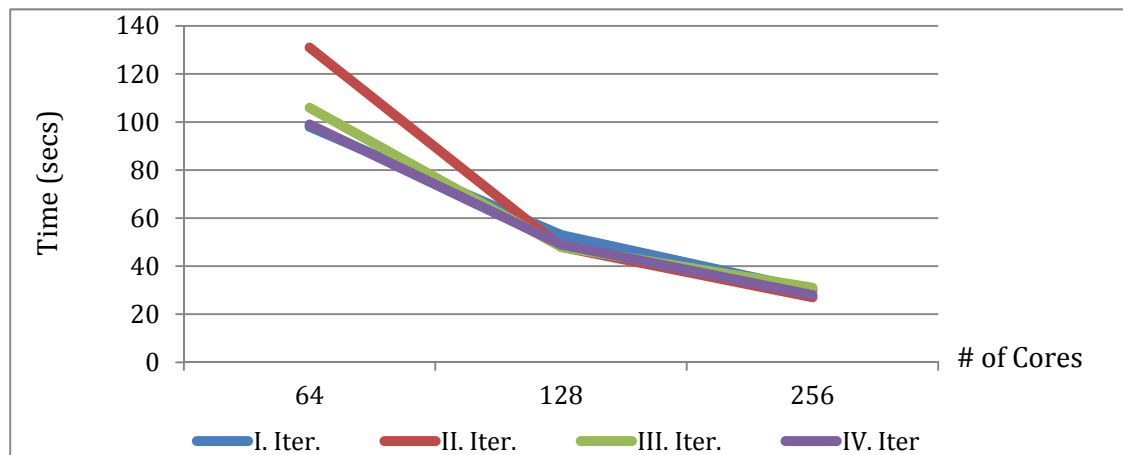


Figure 1 First four iteration total solve time for 127920 hexahedral elements.

Computation time results are shown in Table 1 and Figure 1. In Table 1, column 2, initial setup time for the construction of the matrices for the Stokes system and the time required for the construction of the two-level non-nested multigrid method decreases while number of cores increases. Total solve time for the first four iterations in column 3 is scaling well.

The implementation of the preconditioned iterative solvers is based on the PETSc library for improving the efficiency of the parallel code. Karadeniz system has 8 cores per node. The thread based version of the code might reduce computing time, because of communication between cores, and improve scaling. However, ViscoSolve supports PETSc 3.0.0 version which is not support thread safe processing [5]. Therefore thread based parallelization has not been implemented on PETSc routines. Functions of code must be modified for newer version of PETSc to become thread safe.

The code has a bug for more than 256 cores. There are some invalid values while decomposition of boundary conditions. It is possible to get over the problem via hybrid version of the code, which is not concluded yet. For writing hybrid version, thread safe version must be presented by using newer version of PETSc.

In conclusion, ViscoSolve is a powerful tool for predicting the onset of viscoelastic fluid flow instabilities, which is very important for many industrial polymer processing. Scaling test results run on Karadeniz Linux cluster at UHeM 64, 128 and 256 cores are presented. The code supports PETSc 3.0.0 version, which is not supported thread safe operations. There are some decomposition problems for more than 256 cores. Using hybrid version can solve the problem. For increasing performance of the code on multi-core systems, it must be modified for newer version of PETs that includes thread safe properties, and then hybrid version of the code can be implemented.

Acknowledgements

This work was supported through the PRACE-2IP project funded in part by the European Commission Framework Programme (FP7) under grant agreement RI-283493.

References

1. M. Sahin, Journal of Non-Newtonian Fluid Mechanics, No. 166 (2011) 779–791.
2. R. Fattal, R. Kupferman, Journal of Non-Newtonian Fluid Mechanics, No. 123 (2004) 281–285.
3. M. Sahin, Journal of Non-Newtonian Fluid Mechanics, No. 195 (2013) 46–56
4. M. Sahin, Parallel Large-Scale Numerical Simulations of Purely-Elastic Instabilities Behind a Confined Circular Cylinder in a Rectangular Channel (20 605 860 DOF)
5. <http://www.mcs.anl.gov/petsc/miscellaneous/petscthreads.html>