An Analytical Approach of Computational Complexity for the Method of Multifluid Modeling

A. K. Borah[1]   P. K. Singh[2]

1. Department of Mathematics, R G Baruah College, Fatasil Ambari, Gauhati University, Guwahati- 781025 , India
2. Department of Mathematics, Allahabad University, Allahabad UP- 211002, India

ABSTRACT
In this paper we deal the building blocks of the computer simulation of the multiphase flows. Whole simulation procedure can be viewed as two super procedures, the implementation of VOF method and the solution of Navier Stoke's Equation, Moreover, a sequential code for a Navier Stoke's solver has been studied.
Keywords- SIMPLE algorithm, Krylov Subspace, Bi-Conjugate Gradient Stabilized (Bi-CGSTAB), ILUT function, Preconditioner, multifluid flows.

1   Introduction

The solutions of multifluid model have been developed in this studies. Implementation of both steps of VOF (PLIC) method has been derived. The SIMPLE algorithm is used for solving the momentum equation generates large sparse linear systems of equations. These methods are solved by iterative method such as K-S methods. On the other hand convergence rate of this method can be accelerated by preconditioning techniques. Development and improvement of numerical schemes have encouraged researchers to investigate almost every branch of fluid dynamics and its application to real life problem.

Multiphase flows occur in mant industrial and natural phenomena, petroleum refining (Mayer and Lenhard, 1998), biological flows (Christopher, 2005) and interaction of air with the sea surface (Melville, 1996). The simulation of multiphase fluid flows in one of the most challenging problem in CFD as it involves in modeling of sharp interfaces separating multiple fluids. The numerical simulation—into fluid flow modeling and multiphase modeling in Fig. (1)
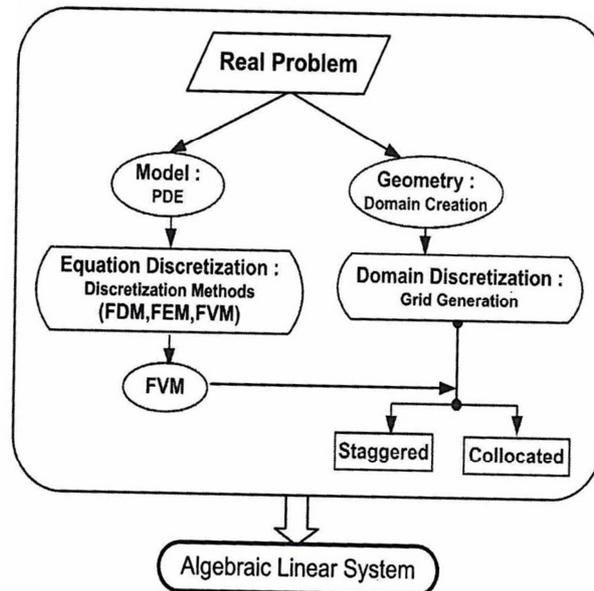

Figure 1. PDE to Linear System

The fluid flow properties (velocity, pressure, etc.,) can be represented by partial differential equations such as N-S equation. Furthermore, numerical solutions of these equations constitute fluid flow modeling. Multiphase modeling involves defining the interface between various phases and then calculating the flux in all directions by using the solution from fluid flow model.

2   Multiphase Modeling

Maintaining sharp interface during fluid transportation is a difficult task in the modeling of interfacial flows (Rider & Kothe, 1998). The interface between two phases can be modeled by scalar transport equation (Greaves, 2004). The modeling involves the construction and movement of the interface. An effective approach for

interface modeling is interface capturing, we study a VOF method based on interface capturing approach has been applied. This method has two steps *reconstruction* and *advection* of the interface between two fluids (Rudman, 1997; Denis *et al*. 1999). Moreover, the interface is established by calculating the volume fractions of each fluid. There is a transportation algorithm is employed for the movement of the interface. The main challenging of the modeling of the interfacial fluid flow is the implementation of a method which can efficiently move the sharp interface without stretching and wrinkling (Alibadi & Shujaee, 2001; Rudman, 1997). In the latter step the interface is approximated by a straight line (a plan 3 dimension) (Scardovelli & Zaleski, 2000). On the other hand, the area of the geometrical shape represented by triangles or rectangle below the lines is calculated to evaluate the volume of the fluid based on the position of the interface. The interface is approximated in the subsequent time steps by using the volume of fluid of the previous time step and this is where time difficulty maintaining the sharpness of the 8interface in the interface approximation arises (Ruben & Zaleski, 2003).

3   Interface Reconstruction

In the volume of fluid (PLIC) method, the interface between two fluids in a grid cell is approximated by a line segment which intersects the cell's faces. But the line segment divides the cell into two parts- each of them containing one of the two fluids as shown in polygon ABFCGD in Figure 2, the notations are as follows
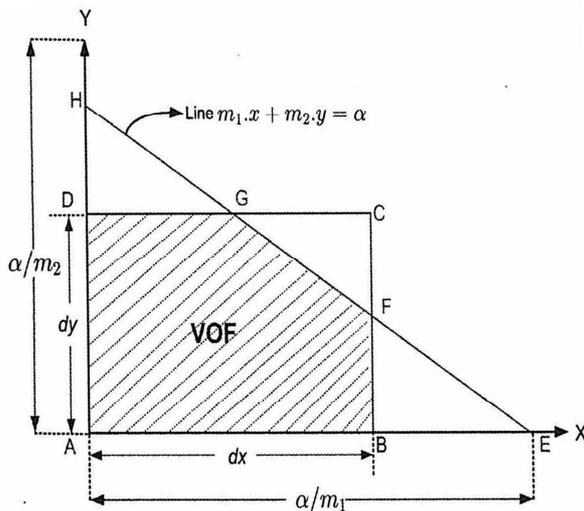

\Figure 2. Cell ABCD is cut by the straight line EH and contains fluid 2 in region ABFGD and fluid 1 in region FCG

  rectangle ABCD, represents a grid cell
  dx the length of the cell in X direction
  dy the length of the cell in Y direction
  line EH approximation of the interface
  polygon ABFGD: volume of one fluid in the cell and
  $\frac{\alpha}{m_1}$, projection of the segment EH on the X axis.

The general equation of straight line L1 with normal $\tilde{m}$ may be represented as $m_1 x + m_2 y = 0$ (1)

Where m_1 and m_2 are the components of the normal vector in the x and y directions respectively and α being the constant which is related to the distance of the line from the origin.   The coordinate at the points at which the line intersects the axes X and Y are respectively $(\frac{\alpha}{m_1}, 0)(0, \frac{\alpha}{m_2})$ to the points E and H in Figure 2.

However, in the simulation the values of volume fractions are provided initially for all the cells. But at the next time step the fluid mixture moves and the interface changes its position and hence new values of the volume fractions have to be calculated. In order to evaluate the volume fraction of one fluid in a cell, one has to calculate the area below the line L1, polygon ABFGD in Figure 2.

3.1 *The Estimation of the normal vector*   In the first part of reconstruction, a normal vector is estimated by a nine-point finite difference formula (Rudman, 1997),
$$m = \nabla C$$ (2)
Eq.(2) represents the gradient of the colour function *C* in the direction of coordinates axes. The discrete approximation to Eq.(2) is given by

$$\nabla C = \begin{pmatrix} \nabla^x & C \\ \nabla^y & C \end{pmatrix} \equiv \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \tag{3}$$

Where ∇^x and ∇^y denotes the gradient in the x and y direction respectively. For approximating the values of these gradient terms, we choose eight nearest neighbours in 2 dimension all the neighbours sharing the vertex, (Scardovelli and Zaleski, 2003). For a uniform mesh, in a grid cell at location (i, j), the gradient terms of Eq.(3) in the coordinate form can be expressed (Rudman, 1997)

$$(m_1)_{i,j} = \frac{1}{\partial x}(C_{i+1,j+1} + 2\,C_{i+1,j} + C_{i+1,j-1} - C_{i-1,j+1} - 2C_{i-1,j} - C_{i-1,j-1}) \tag{4}$$

$$(m_2)_{i,j} = \frac{1}{\partial y}(C_{i+1,j+1} + 2\,C_{i,j+1} + C_{i-1,j+1} - C_{i+1,j-1} - 2C_{i,j-1} - C_{i-1,j-1}) \tag{5}$$

Eq.(3)-(4) represents the normal vector estimation formula for x, y directions (Denis et al., 1999), this scheme produces a good estimation of the normal vector. An investigation of accuracy test of different normal estimations schemes have been demonstrated (Scardovelli and Zaleski, 2003) that the linear fit (using the nine-point) stencil produce the same order error as other methods such as quadratic fit which require more computations.

3.2 *The calculation of VOF from the normal vector and the line constant.*
The area $\square$ polygon ABFGD can be calculated geometrically (Greaves, 2004)

$$a?? \text{ ABFGD} = \underbrace{\Delta AEH}_{\Delta big} - \underbrace{\Delta BEH}_{\Delta_1} - \underbrace{\Delta BEH}_{\Delta_2} \tag{6}$$

$$\text{Area} = \underbrace{\frac{\alpha^2}{2m_1 m_2}}_{\Delta big}\left\{ 1 - H(\alpha - m_1\,dx)\left(\frac{\alpha - m_1 dx}{\alpha}\right)^2 - \underbrace{H(\alpha - m_2\,dy)\left(\frac{\alpha - m_2\,dy}{\alpha}\right)^2}_{\Delta_2} \right\} \tag{7}$$

where H(x) be the Heaviside step function

$$H(x) = \{0 \quad x < 0 \; ; \; 1 \quad x > 0 \tag{8}$$

Eq.(6)-(7) calculates the area below the line in Fig.(2), the procedure for calculating area below the line in two cases for the x direction using Eq.(7) the case of a line with positive slope intersecting the grid cell at the axes as shown in Fig.(3) represents the three triangles
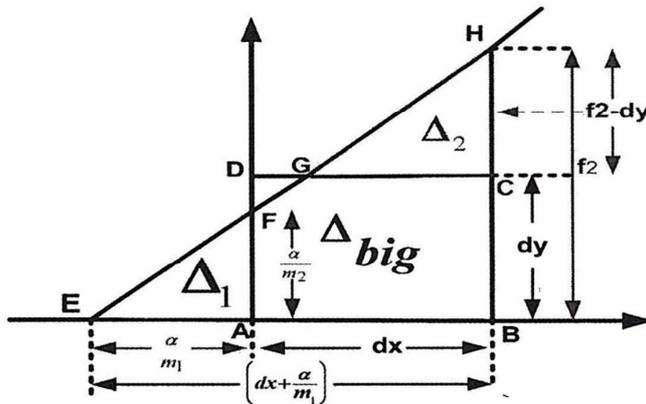

Figure 3. Line of positive slope intersects the cell faces
Δ_{big-} triangle BEH
$\Delta_{1\,-}$ triangle AEF
Δ_{2-} triangle CGH

Now the area of the polygon $\square$ is required to calculate the actual volume of the fluid contained in the cell.

$$\Delta_{big=}\left|0.5 \times \left(dx + \frac{\alpha}{m_1}\right)f_2\right|$$

$$\Delta_{1=}\left|0.5 \times \frac{\alpha}{m_1}\frac{\alpha}{m_2}\right|$$

$$\Delta_2 = \Delta_{big\frac{(CH)^2}{(BH)^2}}$$

$$\Delta_2 = \Delta_{big\frac{\{(f_2 - dy)^2\}}{f_2^2}} = \Delta_{big}\;(1 - \frac{dy}{f_2})^2$$

$$\left(\frac{\alpha}{m_1} < 0\right) \Rightarrow h1 = 1\ and\ (f_2 - dy) > 0 \Rightarrow h2 = 1$$

Area $= \Delta_{big}\{1 - h1.\Delta_1 - h2.\,\Delta_2\}$ \hfill (9)

In Eq. (9) h1 and h2 represents the Heaviside step-function $H(\alpha - m_1 dx)$, $H(\alpha - m_2 dx)$ respectively. A clearer picture of this case is depicted in Fig.(4).
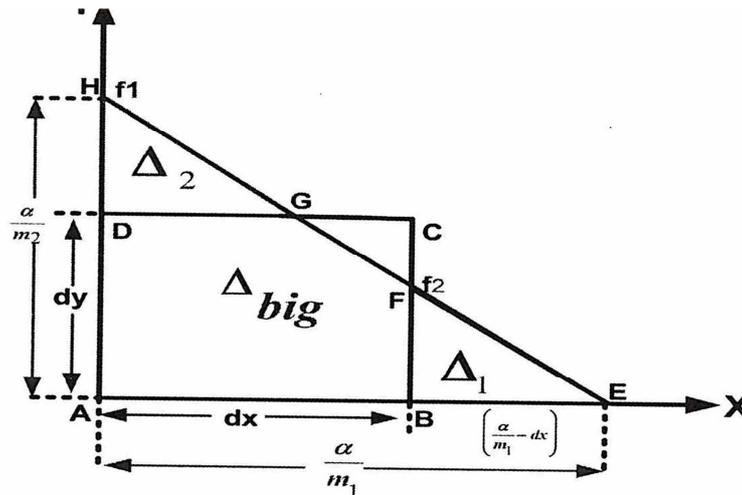
Figure 4. Line of negative slope intersects the cell faces.

The different steps involved in the procedure of area calculations are performed below

$$\Delta_{big} = 0.5 \times \frac{\alpha \times \alpha}{m_1 m_2}$$

$$\Delta_1 = \Delta_{big} \left(\frac{BE}{AE}\right)^2 = \Delta_{big} \times \left(\frac{\frac{\alpha}{m_1} - dx}{\frac{\alpha_1}{m_1}}\right)^2$$

$$\Delta_2 = \Delta_{big} \left(\frac{DH}{AH}\right)^2 = \Delta_{big} \times \left(\frac{\frac{\alpha}{m_2} - dy}{\frac{\alpha_2}{m_2}}\right)^2 \qquad (10)$$

$$\Delta_2 = \Delta_{big} \times \left(\frac{\alpha - m_2 dy}{\alpha}\right)^2$$

$$(\alpha - m_1 dx) > 0 \Rightarrow h1 = 1 \ and \ (\alpha - m_2 dy) > 0 \Rightarrow h2 = 1$$

Area $= \Delta_{big} \{1 - h1. \Delta_1 - h2. \Delta_2\}$

From the Eqs (9)- (10) it has been demonstrated that when the line intersects the cell such that it generates two small triangles. But other cases may arise when there is only one triangle h1= 0 or h2= 0 or there is no triangle present at all as such h1=h2=0.

3.3 *Performance of preconditioners*   The preconditioners accelerates the convergence rate of the Krylov-Subspace method. During simulation the matrix is being generated at each time step. Thus the matrix entries changes at each time step which changes the matrix properties. In this study, the Bi-CGSTAB method has been employed because it has been found in literature to provide small convergence for non-symmetric matrices. Now the effects of preconditioners applied to the Bi-CGSTAB method implemented on different problem are demonstrated.

To simulate multifluid flow, the VOF method has been implemented. This method treats the mixture of two fluids as one fluid which is determined by the interface. During the advection the mesh is kept fixed and the interface is reconstructed from the values of the colour function and its gradient in a grid cell. These values contribute to the calculation of the coefficients matrix entries.

Since the fluid moves at each time step, it has been observed that the magnitude of the matrix entries change because of the changes in the interface position. Due to this change the condition number- the ratio of the highest to the lowest eigenvalues of the matrix may vary.

A matrix with high condition number makes Krylov-Subspace solvers converge slowly and so, in order to increase the convergence rate preconditioning techniques are applied (Saad, 1992; Sun et al., 2009). Incomplete to L-U Symmetric Successive Over

Relaxation (SSOR), Diagonal Scaling (DS) are the are the most appropriate preconditioners for the solvers.

4. Computational Complexity In ILU factorization, the original matrix A is decomposed into two matrices L and U. Its Algorithm in dense format is shown in 4.1. In this Algorithm, there three nested *for* loops required (Steps -1, 2 and 9), these nested loops

1   for ($i = 0$ to n)
2   for ($j = i+1$ to n) do
3     $U_{j,i} = A_{j,i}$;
4   if ($U_{j,i} == 0.0$) then
5     $L_{j,i} = 0.0$;
6   else
7     $L_{j,i=} \dfrac{U_{j,i}}{U_{i,i}}$;
8   end if
9   for ($k = i+1$ to n) do
10   if ($U_{j,k} \neq 0.0$) then
11    $U_{j,k} \text{—} = L_{j,i} \times U_{i,k}$
12   end if
13   end for
14   end for
15    $L_{i,,i} = 1.0$;
16   end for

Algorithm 4.1 Dense ILUT algorithm (Saad, 1996) generates the data depending of the matrix elements of L and U. This depending implies that for calculating the elements of the $(i+1)^{th}$, $(i+2)^{th}$ rows, the elements from the ith or previous row are required . Further, this data dependency is a hindrance to parallelising the Algorithm (Basermann, 2000). In the parallel version the matrix is divided into different parts which are available on different processors of the parallel computer (Li, 2005). Therefore, the elements of previous rows may not available on the same processor and those elements have to be brought from other processors.

In Algorithm 4.1 there are three *for* loops, its computational complexity can be calculated by observing the number of counts in each loop. The outer loop runs from 0 to (n—1) and other loop run from (i+1) to (n—1). The total number of counts can be expressed

$$\sum_{i=0}^{n}(n - i)^2 \qquad\qquad (11)$$

Which has complexity has order $\mathcal{O}(n^3 - n^2)$ but in algorithm ILUT algorithm in diagonal format, there is one *for* loop so its complexity is given as $\mathcal{O}(n)$. Moreover, the matrix vector product in Algorithm requires only one *for* loop. Therefore, its computational complexity can also be given as *n*.

The computational complexity of the matrix-vector products in other sparse formats (Shahnaz *et al*. 2006; Straubhar, 2008) are 5*n*.

5. Discussion and Conclusion

The main conclusions can be summarized from the above studies:

(i)   the diagonal format occupies less memory storing penta diagonal matrices.
(ii)   solving the linear system consumes most of the computational time of the simulation.
(iii)   a short description of the iterative methods has been investigated.
(iv)   the Bi-CGSTAB method requires four inner products and two matrix vectors products.
(v)   these products are developed a diagonal format reducing the computational complexity of the solver.
(vi)   the need for preconditioners has been highlighted.
(vii) the main computational steps of Krylov-Subspace methods have been demonstrated.
(viii) the parallel Bi-CGSTAB method has been integrated into Navier Stokes solver. Furthermore, all the parallel computational tools (C_{++} code with MPI) for simulating the multiphase flow phenomena have been investigated.

References
Alibadi, S., and Shujaee, K.(2001), "Free surface flow simulations using parallel finite element method", *SIMULATION*, **76**(5), 257-262.
Basermann, A. (2000), Parallel Block ILUT/ILDT Preconditioning for Sparse Eigen Problems and Sparse Linear system. Numerical Linear Algebra with Applications **7**, 635-648.

Christopher, E. (2005), "Fundamentals of Multiphase Flows", Cambridge University, Press U.K.

Denis, G ., Ali, N., Scardovelli, R., and Zaleski, S. (1999), "Volume-of-fluid interface tracking with Smoothed Surface Stress Methods for Three Dimensional Flows", *Journal of Computational Physics*, **152,** 423-456.

Greaves, D. (2004), "A Quadrative Adaptive method for Simulating Fluid flow with moving Interfaces", *Journal of Computational Physics*, **194,** 35-56.

Li, K. (2005), First and Scalable Parallel Matrix Computations on Distributed memory Systems. In *19th IEEE International Parallel and Distributed Processing Symposium*.

Mayer, A. and Lenhard, R. (1998), "Special Issue on Multiphase Flows and Chemical Transport", *Advances in Water Resource*, **21,** 75-76.

Melvile, W. (1996), "The Role of Surface Wave Breaking in Air-Sea Interaction", *Annual Review of Fluid Mechanics*, **28,** 279-321.

Rider, W. and Kothe, D.(1998), "Reconstructing Volume Tracking", *Journal of Computational Physics*, **141,** 112-152.

Rudman, M. (1997), "Volume Tracking Methods for Interfacial Flow Calculations", *International Journal of Numerical methods in Fluids*, **24,** 671-691.

Ruben, S and Zaleski, S (2003), "Interface Reconstruction with Least-Square Fit and Split Eulerian Lagragian Advection", *International Journal for Numerical Methods in Fluids*, **41,** 251-274.

Scardovelli, R. and Zaleski, S. (2000), "Analytical relations connecting Linear Interfaces and Volume Fractions in Rectangular Grids", *Journal of Computational Physics,* **164**(6), 228-237.

Scardovelli, R and Zaleski, S. (2003), "Interface Reconstruction with Least Square fit and split Eulerian – Lagragian advection", *International Journal of Numerical Methods in Fluids*, **41,** 251-274.

Saad, Y. (1992), "Preconditioning Techniques for Nonsymmetric Indefinite Linear systems", *Technical Report, Centre for Supercomputing Research and Development,* University of Illinois at Urbana Champaign.

Sun, J, Cao, J and Yang, C.(2009), "Paralled Preconconditioners for large scale partial differential equations systems", *Journal of Computational and Applied mathematics*, **226,** 125-135.

Shahnaz, R., Usman, A. and Chugati, I. (2006), "Implementation and Evaluation of parallel sparse Matrix-Vector products on Distributed Memory Parallel Computers", Barcelona, Spain. *In IEEE International Conference on Cluster Computing* (*CLUSTER*), Barcelona, Spain pages 1-6.

Straubhaar, J. (2008), "Parallel Preconditioners for the conjugate Gradient Algorithm using Gram Schmidt and least square methods", *Parallel Computing*, **34**(10), 551-569.

(a) **A K Borah**, is Head of the Department of Mathematics and Associate Professor, R. G. Baruah College, Fatasil Ambari, Gauhati University, Assam. He has completed his Ph. D degree from Gauhati University in the faculty of Science Mathematics in 2007. He has attended six International conferences held in USA, UK. France, Germany (Berlin), Australia (Perth), Thailand during last five years and presented paper as a **S**peaker and role as **S**ession **C**hair.

(b) **P. K. Singh**, Professor in the Department of Mathematics, University of Allahabad Uttar Pradesh India. He has published research papers in National and International Journals near about **30** publications.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar