A Concept Paper On A Fast Numerical Solver For Multiphase Flow.

group 1

April 2017

1 Introduction

In fluid mechanics, multiphase flow is simultaneous flow of materials with different states or phases (i.e. gas, liquid or solid) and materials with different chemical properties but in the same state or phase (i.e. liquid-liquid systems such as oil droplets in water). In nature and environment, rain, snow, fog, avalanches, mud slides, sediment transport, de-bris flows are all examples of multiphase flow where the behavior of the phases are studied in different fields of natural science. In oil and gas industries, multiphase flow often implies to simultaneous flow of oil, water and gas. The term is also applicable to signify the properties Of a flow in some field where there is a chemical injection or various types of inhibitors. The resulting models are difficult to solve due to large number of strongly coupled nonlinear differential equations in the systems. In this paper we solve numerically global pressure model for isothermal two-phase immiscible flow problems. The model is described in Section 2. In Section 3 we present the cho-sen finite volume discretization, Section 4 describes the concept of our software tool for the solution of multiphase flow problems, called MfsolverC++ as well as the current state of the implementations.

1.1 Background to the problem.

In this paper we discuss algorithms of a fast numerical solver for multiphase flow as well as the proper object oriented implementation of this algorithm. Different time stepping discretization and linearization approaches are discussed. Numerical results for one realistic problem are presented and problems involving the multiphasefl ow, heat transfer, and multi component. Each of the phases is considered to have a separately defined volume fraction (the sum of which is unity) and velocity field. Conservation equations for the flow of each species (perhaps with terms for interchange between the phases) can then be written down straightforwardly

2 Scope.

2.1 Main Objective.

Our goal is to find a fast numerical algorithm for solving problems of multiphase flows thus we have decided to start with not the complicated but flexible and robust approximations. We try to avoid the time-consuming computation of approximate Riemann solvers and the related characteristic decompositions. Since dimensional splitting and splitting in physical processes also introduce numerical and non-physical errors, we avoid splitting also.

2.2 Main objective.

Our goal is to find a fast numerical algorithm for solving problems of multiphase flows thus we have decided to start with not the complicated but flexible and robust approximations. We try to avoid the time-consuming computation of approximate Riemann solvers and the related characteristic decompositions. Since dimensional splitting and splitting in physical processes also introduce numerical and non-physical errors, we avoid splitting also.

2.3 Specific Objectives.

• To avoid the time-consuming computation of approximate Riemann solvers and the related characteristic decompositions. • To find a fast numerical algorithm for solving problems of multiphase flows. • To describe the numerical method and test its performance. Rather than relying on the periodic Green's function as classical BIE (boundary integral equation) methods do.

3 Literature Review.

This chapter reviews a fast numerical solver for multiphase flows and other algorithms. In this algorithm, finite volume discretization and time stepping algorithms are used to construct a fast and efficient numerical solver.

3.1 Existing algorithims.

The following are the algorithms which are currently being used to solve multiphase flows. MfsolverC++. It allows developers to reduce the time spent on the programming, de-bugging and it makes all implementation aspects cleaner and simpler. Global Pressure Model for Isothermal Two-Phase Immiscible Flow. It has two equa-tions for mass conservation in each phase. Time- stepping algorithm.

3.2 Weaknesses of the existing algorithms

There is time consuming in computation of the results. There is also large number of strongly coupled nonlinear differential equations in the system which are difficult to read.

3.3 The fast numerical solver will consider the following.

Finite volume discretization: its a method for representing and evaluating partial differential equations in form of algebraic equations. It is easily formulated to allow for un structured meshes. Time-stepping algorithm: Here the convergence methods based on the convergence properties of the underlying iterative methods are discussed, and an accurate perfor-mance model from which the speedup and other quantities can be estimated is pre-sented. Dimensional splitting is an effective method for solving multidimensional problems in multiphase flows by constructing he integration algorithm from one-dimensional sub-problems to one dimension at a time. It has been proved that dimensional splitting encounters several limitations when applied for solving conservation laws. Local characteristic decomposition is used for accurate resolution of complicated structure of solutions. I t increases the computational costs and controls spurious os-cillations when order of accuracy is high.

4 Discussion.

4.1 Global Pressure Model for Isothermal Two-Phase Immiscible Flow in Porous Media.

Traditionally multiphase flow in porous media is described by the macroscopic models, in which various phases are considered as distinct fluids with individual thermodynamic and transport properties [6]. Macroscopic equations are obtained by averaging microscopic equations over the representative elementary volume. The microscopic equations are the equations of mass, momentum and energy conservation for all considered phases. The change of scale enables to convert the real discontinuous medium to a fictitious continuous one. Each macroscopic term is obtained by averaging microscopic ones . The averaging (integration) is done by using various closing assumptions (homogeneity, pe-riodic cell structure etc.) .So, for isothermal two-phase immiscible flow in porous media we have two equations for mass conservation in each phase (k= 1,2):

((k s k)/t)+ .(kuk) = 0 . . . .(1)

where is the porosity of the porous medium, k is the phase density, sk is the phase saturation denoting the volumetric fraction of the void space occupied by phase k.Obviously we hav s1+s2=1.

Uk is the macroscopic velocity vector (volume rate of flow through a unit cross-sectional area of multiple fluids and porous medium). It is also called bulk velocity. It is assumed that there are no external mass sources or sinks, no mass transfer between phases, and the porous medium is not deformable. It is also assumed that multiphase extension of Darcy's law is valid for conservation of phase momentum:

uk = - K . . . .(2)

where K is the absolute permeability tensor of the porous medium, krk is the relative permeability of phase k, k is the phase dynamic viscosity, pk is the phase pressure, and g is the gravitational vector. The pressures of two phases are related through the capillary pressure,

pc=p2p1. Relative permeability functions and capillary pressure are assumed to be functions of the phase saturations only

krk= krk(Sk), pc=pc(s1)...(3)

One of the assumptions in generalized Darcy's law is that flow is slow, i.e. inertial effects can be neglected. The governing equations (1)–(2) can be mathematically manipulated into several alternate forms with different choices of primary variables: saturation–pressure, pressure–pressure, saturation–saturati on formulations. We have chosen global pressure model, sometimes also called fractional flow formulation or mixture model. Comparing to other formulations, equations in this model are less coupled and entering quantities are smoother , because most of them describe mixture properties: velocity, density, etc. First we define the properties of multiphase mixture through the proper- ties of its components (phases).

= 1s1 + 2 s2 , ...(4)

u=1u1 + 2u2, ...(5) he mixture kinematic viscosity is given by =((kr1/ 1) + (kr2/ 2)) power -1 . . . .(6)

where k = k/k is the kinematic viscosity of phase k. Then, the mobility of each phase in the multiphase mixture is defined as

k = (krk / k) . . . .(7)

In the literature, k is often called the fractional flow function. We have the relations for these function.

1 + 2 = 1.

The main idea of the model is an introduction of the global (mixture) pressure. The mixture pressure is defined so that the following differential equation holds:

p =(1 p1 + 2p2) = p1+2pc

It is not always possible to find such p. A necessary condition for this is he interchangeability of partial derivertives:

/xi (2 . (pc / xi))= /xj (2 .(pc/ xj)) , I not equall to j

For our assumptions this condition is satisfied, and mixture pressure can be defined explicitly as:-

. . . .(8) where Sc is obtained from the equation pc(Sc) = 0. Adding the phase mass conservation equations .1) we obtain the continuity equation for the mixture: . . . .(9) Multiplying equations (2.2) by corresponding densities and summing them up, we obtain Darcy's law for the mixture: . . . .(10) Substituting (10) into (9) we get

. . . .(11)

Obtained equation is sometimes called pressure equation and is used to find global (mixture) pressure p. The mixture kinematic viscosity is positive. The absolute permeability tensor K of porous medium is positive-definite, so is K/. Consequently, it follows from (11) that the pressure equation is elliptic (for known). The diffusive mass flux of phase k within the multiphase mixture is defined as

....(12)

As a consequence we get that

Substituting the momentum equations for the first phase (2) and for the mixture (10) into equation (12), the diffusive mass flux j1 can be expressed as follows:

The system is completed by the constitutive relations (2.3). Equations are defined in , which is assumed to be a polyhedral domain. Both, Dirichlet and full flux (or Neumann) type boundary conditions can be specified on the boundary of the region. This system can be used to model isothermal two-phase immiscible fluid flow in porous media, when the phase compositions are of no importance and molecular diffusion and hydraulic dispersion effects can be neglected. .... (14) Substituting equations (13) with k = 1 and (14) into mass conservation equation of the first phase (1), we obtain

..........................(15) Equation (15) is called a saturation equation. It is used to find the saturation s1. Note that dpc/ds1 ¡ 0 by the property of capillary pressure. Hence if K is positive-definite, then .15) is a degenerate parabolic equation. The degeneracy is caused by the fact that fractional mobilities 1 and 2 can become zero. When the capillary forces are small, saturation equation (15) is advection dominated. It is purely hyperbolic in the absence of the diffusive forces. The saturation equation .15) in conjunction with the pressure equation (11) and the Darcy law (10) makes up a full system of equations for unknowns p, u, and s1:

.... (16) The system is completed by the constitutive relations (2.3). Equations are defined in , which is assumed to be a polyhedral domain. Both, Dirichlet and full flux (or Neumann) type boundary conditions can be specified on the boundary of the region. This system can be used to model isothermal two-phase immiscible fluid flow in porous media, when the phase compositions are of no importance and molecular diffusion and hydraulic dispersion effects can be neglected.

4.2 Numerical Algorithms.

The major numerical difficulties for problems arising in the simulation of processes in porous media are identified by Ewing [14] and Helmig [17] (see also the dissertation of Bastian [4]). They deal with • solution of convection dominated parabolic problems, • time–stepping algorithms. Many problems describing multi-phase flows in porous media are convection dominated parabolic problems. When capillary forces are equal to zero the saturation equation in (2.16) becomes hyperbolic. Sharp fronts of solutions appear often in various applications. They should be resolved accurately by discrete schemes. It is also very important to preserve the monotonicity of the solutions. Here we should balance between two possibilities. Upwind approximations are monotone, but the approximation is only first order accurate. Centered differences are second order accurate but can yield oscillatory approximations if the solution is not smooth enough.

In many papers splitting and decomposition methods are used to construct computationally efficient numerical algorithms A) Dimensional splitting is an effective method for solving multidimensional problems by constructing the integration algorithm from one-dimensional subproblems – one dimension at a time

B) Splitting in physical processes consists in solving separately a system of conservation laws and stiff diffusion and reaction problems

C) Local characteristic decomposition is used for the accurate resolution of complicated structure of of solutions. The development of such methods started from the Godunov method, and now a large number of special classes of methods, e.g. ENO, WENO algorithms, are proposed .While this decomposition increases the computational cost, it controls spurious oscillations when the order of accuracy is high

Thus we have decided to start with not the most complicated but flexible and robust approximations. We try to avoid the time–consuming computation of approximate Riemann solvers and the related characteristic decompositions. Since dimensional splitting and splitting in physical processes also introduce numerical and non-physical errors, we avoid splitting also.

Finite volume discretization: We discretize equations (16) on structured nonuniform grids. A 2D space is used as an example in this section. We have a set of vertices:

Vh = uij = (x1i , x2j), 0 i Ni , 0 j Nj , where: x1,i = x1,i1 + h1,i, x2,j = x2,j1 + h2,j. The grid Vh defines a structured mesh Eh, which covers the region : Eh = eij , 1 i Ni , 1 j Nj.

Elements eij of the mesh are called cells or control volumes (thus, in fact we define a dual mesh in the terminology of [4]). In 2D we get quadrilateral elements. These elements can be cell-centered or vertex-centered. Our scheme is based on cell-centered mesh, and these vertices make a primary grid h = Xij , 1 i Ni , 1 j Nj. For boundary elements, one or more faces of which belong to the boundary , the vertices are taken on the boundary. Let denote: h1,i+1/2 = Xi+1,j Xij , h2,j+1/2 = Xi,j+1 Xij . Based on the given grid we can define a finite dimensional function space of piecewise linear functions: ....(3.1) where ij (x) are the usual nodal basis functions. We describe the finite-volume scheme for the saturation equation, since the pressure equation is approximated in a similar way. Integrating the saturation equation (2.16) over the control element e Eh and applying the divergence theorem, we obtain

The numerical diffusion flux over control element face is computed by using harmonic or arithmetic average of the diffusion coefficient. For example, in 2D case taking only one part of the element interface we get the following numerical diffusion flux:

The evaluation of the advective part of the flux is done by using the fully upwinding method

Fully upwinding discretization of the advective flux leads to positive diagonal and negative off-diagonal entries in the matrix obtained after linearization of nonlinear discrete equations by the Newton method and the discrete solution obeys a discrete maximum principle. The drawback of such approximation is that artificial (numerical) diffusion is introduced. It can be split into two components: one along the streamline and a second component perpendicular to the streamline [17]. The second component exists only for multi – dimensional problems. A multidimensionalupwind method of first order is considered in [17], which try to reduce the grid orientation effects by taking into account the direction of the velocity field. Our strategy is to reduce the approximation error by decreasing the mesh step, since the obtained large systems of equations can be solved efficiently with parallel computers. For very fine mesh central differencing can be used also. In our tool for 1D problems we have also implemented an explicit higher order TVD approximation of the advection flux, when the central difference approximation is combined with the upwind approximation by using the minmod limiter. Let consider the case, when u1,i+1/2,j 1 (Sij) 0, then the advection flux is approximated as follows

where the limiter is defined by

After discretization the mass is conserved locally in each element and therefore the discrete problem satisfies the mass conservation property in the whole region .

4.3 Time–stepping algorithms.

After approximation of space derivatives in system (2.16) a large system of nonstationary nonlinear DEs is obtained . . . .(2) Here the first equation approximates the global pressure equation, Ch(S, P) denotes the convection part and Dh(S, P) the diffusion part of the semi–discrete saturation equation. We consider incompressible case. Let denote saturation and pressure values at time t n by S n and P n, respectively. The aim of time–stepping algorithms is to balance the stability and robustness of the integration algorithm with computational efficiency. The one way is to use a some kind of segregated solution procedure. With this approach the new pressure field P n+1 is computed first from . . . . (3) which is a linear elliptic equation in P for a given saturation at previous time level t n. Next, saturation at time level t n+1 is computed with a velocity field obtained from the new pressure field. The most simple way is to use the forward Euler scheme:

This time–stepping procedure is named IMPES (for implicit pressure and explicit saturation). It was originally developed by Sheldon et al. [33] and Stone and Garder [34] for solving partial differential coupled system for two– phase flow in porous medium (see also [4, 17]). Due to the explicitness of the scheme for saturation equation IMPES is stable only when time steps are sufficiently small. An improved IMPES method which utilizes an adaptive control strategy for selection of time step is proposed in [9]. The stability of IMPES scheme for solving a three–phase flow problem is investigated in [11]. For many applications the strongest restriction on time step is due the diffusion part of saturation equation. The following implicit–explicit modifi- cation of (3.4) is used in some papers (see, e.g. [2]):

which consists of applying an explicit discretization for the advection term and an implicit discretization for the diffusion and source terms. We note that the approximation of different terms on different time levels leads to truncation errors similar to additive scheme case. In fact scheme (3.5) is equivalent to the following splitting scheme

but the additive scheme is more flexible, since different time steps can be used for integration of each subproblem. The next step to increase the stability of the discrete scheme is to use backward Euler (or more generally an implicit) integration method for the saturation equation: . . . .(6) At each time level we get a nonlinear elliptic equation with advection– dominated space operator. It can be solved by some iterative procedure, e.g. the Newton method. The whole time–stepping procedure (3.3), (3.6) is named IMPIMS. Another way is to use the coupled schemes. In our case, the fully implicit/fully coupled (FIFC) scheme gives the system . . . . (7) when both equations are solved simultaneously. The FIFC approach is usually selected for a maximum of robustness and stability. However, it can be very expensive computationally. Our goal is to have a tool working with both: segregated and coupled solvers, to be able to choose the best for the specific problem. Currently, the IMPIMS type iterative algorithm is implemented in our tool with additional "k"-iterations between separate equation solvers at each time level: . . . .(8) An quasi-Newton method is used to solve the nonlinear saturation equation. Let us write the diffusion term as Dh(S) = h • K(S)hPc(S) . Let denote the m-th iteration of quasi-Newton method. Then the next iterative approximation is defined as

where S m satisfies the following equation:

4.4 MfsolverC++:

A Tool for Computations Recently, it becomes very popular in the scientific computing to exploit the object-oriented programming (OOP) techniques. This allows developers to reduce the time spent on the programming and debugging and makes all implementation aspects cleaner and simpler. We also use OOP in the design and implementation of our software tool MfsolverC++ for computations of multiphase flows in porous media. We want to have a flexible set of efficient C++ modules to reduce the overhead for creation of different solvers for different schemes and problems. Implementation of the solver for the solution of isothermal two-phase immiscible flow problems using global pressure model is shown in Figure 1. For this and other segregated solvers we use strategy proposed in [24] and used in Diffpack software library. According to it, solvers for the systems of PDEs are built by merging together the independent solvers for the alone-standing equations that enter the system.

Design of the PDE system solver with relations between solver classes, manager and pool of common relations. Solid arrows indicate inheritance (”is-a” relationship, with arrows pointing from subclass to base class). Dashed arrows indicate pointer (”has-a” relationship).

Class Pressure and his children are independent solvers of pressure equation (2.11). Class Saturation and his children are independent solvers of saturation equation (2.15). Equations become coupled into the system through the coefficients. In our PDE solvers, these coefficients, including the source, initial and boundary conditions functions, are represented by virtual functions. Subclasses of PDE solvers override these functions and implement the physically relevant versions, when the coefficients are coupled to other unknown fields in the PDE system. All these functions are often built of a common set of relations (constitutive relationships, model definitions, etc.). Therefore, they are collected in class hierarchies and accessed from PDE solvers through a base class CommonRel interface (pointer). Note that constitutive relationships can be easily changed without affecting the code in PDE solvers. A class Manager acts as the solver class for the whole PDEs system and manages the whole solution of the given problem. This class contains two waypointers to the subclasses for solving the pressure and saturation equations, PressureM and SaturationM, respectively, which enable the coupling by overriding the virtual functions of base classes with the functions from common relation hierarchy. The manager is also responsible for creating a space grid and time discretization. It allocates a common linear system and solver object, and distributes all these data to the PDE solver classes. It is clear that segregated solvers have ”implementational” advantage over coupled solvers. They are easier to implement, flexible and naturally extensible if the differential model is extended by additional PDEs. Next we review very briefly some similar projects. First we mention a general PDE software tool Diffpack, which is an object oriented development framework for the solution PDE. It is a commercial product and many application examples are described in [24]. The toolbox UG is a framework for unstructured grid computations. Development of UG started in 1990 at the University of Heidelberg and proceeded at the University of Stuttgart from 1994. It is a flexible software tool for the numerical solution of partial differential equations on unstructured meshes in two and three space dimensions using multigrid methods. A number of applications of this tool for computations of complex fluid flows in porous media are described in [5, 17]. For the time discretization either backward Euler, Crank-Nicholson or BDF(2) are used. The fully implicit/fully coupled time stepping strategy is selected for a maximum of robustness and stability. In UG the Newton method is used to solve large systems of nonlinear algebraic equations at each time step. The linear systems arising within the Newton method are solved with a multigrid method. TOUGH2 is a general-purpose numerical simulation program for multiphase fluid and heat flow in porous and fractured media [29]. It is developed at Lawrence Berkeley National Laboratory and is widely used for computations on unstructured meshes (see e.g [28]). In TOUGH2 code various strategies are implemented. In order to take into account the hyperbolic nature of the saturation equation very accurate explicit time–stepping algorithm TOUGH2-EOS3f with front tracking is used.

5 Methodology.

5.1 Mathematical Model.

In this method we study adjectives flow in a porous medium, that is, we neglect gravity and capillary forces and consider viscous flow driven only by pressure forces. The fact that vis-cous flow is unidirectional along streamlines will be the key property used to derive our highly efficient upwind solvers.

5.2 Spatial Discretisation.

The discretisation in a discontinuous Galerkin method starts with a variation formulation, as in a standard Galerkin method. The difference is that discontinuous Galerkin methods allow discontinuities at element interfaces.

5.3 Treatment of Cycles.

The number and size of cycles in the discrete fluxes depend on the heterogeneity and on the numerical method used to discretize the pressure equation.