Computers & Fluids 68 (2012) 38–46

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Matrix-oriented implementation for the numerical solution of the partial differential equations governing flows and transport in porous media

Shuyu Sun [a,b], Amgad Salama [a,c,*], M.F. El Amin [a,d]

[a] Computational Transport Phenomena Laboratory (CTPL), Division of Physical Sciences and Engineering (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
[b] Center for Computational Geosciences, Xi'an Jiaotong University, Xi'an 710049, PR China
[c] Nuclear Research Center, AEA, 13759 Cairo, Egypt
[d] Mathematics Department, Faculty of Science, Aswan University, Aswan, Egypt

A R T I C L E I N F O

Article history:
Received 1 July 2011
Received in revised form 10 June 2012
Accepted 30 July 2012
Available online 11 August 2012

Keywords:
Porous media
Flow and transport
Numerical methods
Matrix-oriented techniques

A B S T R A C T

In this paper we introduce a new technique for the numerical solution of the various partial differential equations governing flow and transport phenomena in porous media. This method is proposed to be used in high level programming languages like MATLAB, Python, etc., which show to be more efficient for certain mathematical operations than for others. The proposed technique utilizes those operations in which these programming languages are efficient the most and keeps away as much as possible from those inefficient, time-consuming operations. In particular, this technique is based on the minimization of using multiple indices looping operations by reshaping the unknown variables into one-dimensional column vectors and performing the numerical operations using shifting matrices. The cell-centered information as well as the face-centered information are shifted to the adjacent face-center and cell-center, respectively. This enables the difference equations to be done for all the cells at once using matrix operations rather than within loops. Furthermore, for results post-processing, the face-center information can further be mapped to the physical grid nodes for contour plotting and stream lines constructions. In this work we apply this technique to flow and transport phenomena in porous media.

1. Introduction

Transport phenomena in porous media is described in the framework of the continuum hypothesis in which field variables are continuous functions of space and time defined at each point of the porous medium domain. This point of view enabled the description of basic conservation laws in the form of differential equations for which field variables are differentiable to at least as much as the governing differential equations require (Salama and Van-Geel [1,2]). Solutions to these equations are in most cases obtained numerically for which several methods have been applied ([3–10] to list but a few). The issue of efficiently solving partial differential equations numerically is not only related to the methods of solution but also to the possibility of designing efficient and robust algorithms. An efficient algorithm may be defined as the one which achieves the solution in the least required time. Many factors may be invoked which controls the time required to complete the solution. These may be divided into hardware requirements and programming methodologies. In cases when there may be no control on the available computing facilities, it remains important to design the solution algorithm in the most efficient and elegant way. Examples include the use of efficient solvers, the best utilization of language features that enable doing mathematical operations efficiently, etc. In this regard, we emphasize that some higher level programming languages are efficient in utilizing special kind of mathematical and logical operations, however, they show to be inefficient in other operations. It is therefore important, if one uses these programming languages, to put forward their coding into the form that utilizes those techniques which are the most efficient and to keep away as much as possible from those inefficient procedures. As an example, it is known that MATLAB, Python and the like are most powerful in using matrices. However, they are inefficient in loops because of the required interpreting procedures they require each time a loop is called. Therefore, it is important that one formulates his problem such that matrix operations dominate the looping processes. In this work we introduce the use of shifting matrices to transfer information between cell-centers and face-center and vice versa. Doing so eliminates the use of loops within the body of the code and therefore significantly reduces the CPU time. We apply this technique to the problem of transport in porous media. We use finite difference technique in

* Corresponding author at: Computational Transport Phenomena Laboratory (CTPL), Division of Physical Sciences and Engineering (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia.
E-mail address: amgad.salama@kaust.edu.sa (A. Salama).

our implementation because of its apparent simplicity; however, this technique may equally be used in finite volume and finite element settings.

2. Governing equations

In order to highlight the features of our scheme, we consider a simple 2D problem in a rectangular porous medium domain, Ω: $(a_1,b_1)\times(a_2,b_2)$, saturated with a fluid that moves as a result of pressure gradient. The equation of motion describing this system is given by:

$$\boldsymbol{u} = -\boldsymbol{K}\nabla p \quad \text{in } \Omega \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = q \quad \text{in } \Omega \tag{2}$$

$$p = p_B \quad \text{on } \Gamma_D \tag{3}$$

$$\mathbf{u}\cdot\mathbf{n} = u_B \quad \text{on } \Gamma_N \tag{4}$$

Furthermore, the transport of solute in this system is described by the following second order partial differential equation:

$$\frac{\partial\varphi c}{\partial t} + \nabla\cdot\mathbf{u}c = \nabla\cdot D\nabla c + q_c \quad \text{in } \Omega \tag{5}$$

$$c = c_B \quad \text{on } \Gamma_D \tag{6}$$

$$\mathbf{j}\cdot\mathbf{n} = j_B \quad \text{on } \Gamma_N \tag{7}$$

where q is a source/sink term, p is the pressure, $\mathbf{u}$ is the velocity vector, $\boldsymbol{K}$ is the permeability tensor, φ is the porosity, $\mathbf{j}$ is the diffusive flux vector, and c is solute concentration, $\boldsymbol{D}$ is the dispersion tensor, q_c is a source/sink term of solute, p_B, u_B, c_B, and j_B are pressure, velocity, concentration, and solute flux at the boundary. To solving this problem over a 2D rectangular domain, we define the rectangular mesh $\{x_0,x_1,x_2,\ldots x_i,\ldots,x_m\}\times\{y_0,y_1,y_2,\ldots,y_j,\ldots,y_n\}$ over which we approximate the solution. We use the cell-center finite difference scheme (CCFD) to approximate the governing equations because it satisfies mass conservation law locally as will be explained in the next section.

3. Traditional programming algorithms

For this system, we assume that neither the dependent variables nor the properties of the porous medium or the flowing fluid are dependent on solute concentrations and therefore the above system of equations is not two-way coupled. Substituting Eq. (1) into Eq. (2), one obtains an equation in the pressure only which may be solved numerically to obtain the pressure field. The velocity field is obtained by back substituting the pressure field into Darcy's law. Solute transport equation is then solved using velocity information obtained previously on solving flow equations. Traditionally, the pressure field at cells center as well as the velocity field at the edge centers surrounding each cell are usually saved in arrays characterized by number of indices that are as many as the dimensions of the space. For example, in our 2D problem the unknown pressure field, $p(i,j)$ and the solute concentration $c(i,j)$ are saved in $m\times n$ matrices where m is the number of segments in the x-direction and n is the number of segments in the y-direction. Likewise the velocity field, $u_x(i,j)$ and $u_y(i,j)$ are saved in arrays of dimensions $(m+1)\times n$ and $m\times(n+1)$ respectively. The difference scheme is usually done on a generic cell as shown in Fig. 1 and the discretization of the various terms is described in the next subsections.

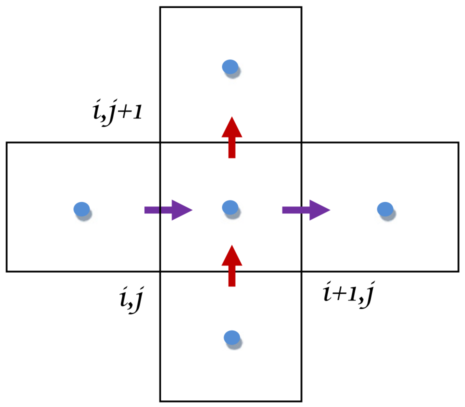

Fig. 1. Discretization on a generic cell.

3.1. Discretizing Darcy's law

Using the cell-center finite difference scheme, the various velocity components at the mid-edges of a generic cell are obtained as follows:

$$u_x\left(i,j+\frac{1}{2}\right) = -K_{xx}\left(i,j+\frac{1}{2}\right)\frac{p\left(i+\frac{1}{2},j+\frac{1}{2}\right)-p\left(i-\frac{1}{2},j+\frac{1}{2}\right)}{x\left(i+\frac{1}{2}\right)-x\left(i-\frac{1}{2}\right)} \tag{8}$$

$$u_x\left(i+1,j+\frac{1}{2}\right) = -K_{xx}\left(i+1,j+\frac{1}{2}\right)\frac{p\left(i+\frac{3}{2},j+\frac{1}{2}\right)-p\left(i+\frac{1}{2},j+\frac{1}{2}\right)}{x\left(i+\frac{3}{2}\right)-x\left(i+\frac{1}{2}\right)} \tag{9}$$

$$u_y\left(i+\frac{1}{2},j\right) = -K_{yy}\left(i+\frac{1}{2},j\right)\frac{p\left(i+\frac{1}{2},j+\frac{1}{2}\right)-p\left(i+\frac{1}{2},j-\frac{1}{2}\right)}{y\left(j+\frac{1}{2}\right)-y\left(j-\frac{1}{2}\right)} \tag{10}$$

$$u_y\left(i+\frac{1}{2},j+1\right) = -K_{yy}\left(i+\frac{1}{2},j+1\right)\frac{p\left(i+\frac{1}{2},j+\frac{3}{2}\right)-p\left(i+\frac{1}{2},j+\frac{1}{2}\right)}{y\left(j+\frac{3}{2}\right)-y\left(j+\frac{1}{2}\right)} \tag{11}$$

3.2. Discretizing mass conservation equation

Eq. (2) is discretized, likewise, in the following form

$$\frac{u_x(i+1,j+\frac{1}{2})-u_x(i,j+\frac{1}{2})}{x(i+1)-x(i)} + \frac{u_y(i+\frac{1}{2},j+1)-u_y(i+\frac{1}{2},j)}{y(j+1)-y(j)} = q\left(i+\frac{1}{2},j+\frac{1}{2}\right) \tag{12}$$

Now substituting Eqs. (8)–(11) into Eq. (12) one obtains an equation in the pressure only as explained earlier which may be solved to obtain the pressure field from which the velocity field may be determined.

3.3. Discretizing solute transport equation

Eq. (5) describes the transport of a passive solute moving with the fluid in the porous medium domain. Three terms are involved into this equation, namely an accumulation term, an advection term and a dispersive term. We highlight the fact that the treatment of the diffusion–dispersion term is similar to the pressure equation described earlier and will not, therefore, be repeated. Needless to mention that the velocity field at the mid-edge elements are known by solving the flow equations. The advection term is discretized based on the upwind scheme, such that:

$$\begin{aligned}\nabla\cdot c\mathbf{v} = \frac{1}{\Delta x(i)}\Bigg\{&\left[u_x^+\left(i+1,j+\frac{1}{2}\right)c\left(i+\frac{1}{2},j+\frac{1}{2}\right)\right.\\ &\left.+u_x^-\left(i+1,j+\frac{1}{2}\right)c\left(i+\frac{3}{2},j+\frac{1}{2}\right)\right]-\left[u_x^+\left(i,j+\frac{1}{2}\right)c\left(i-\frac{1}{2},j+\frac{1}{2}\right)\right.\\ &\left.\left.+u_x^-\left(i,j+\frac{1}{2}\right)c\left(i+\frac{1}{2},j+\frac{1}{2}\right)\right]\right\}\end{aligned}$$