

Flux-limiting and non-linear solution techniques for simulation of transport in porous media

M. Awad* I. Turner*

(Received 7 August 2000)

Abstract

The conservation laws governing the flow of liquids in porous media are often non-linear and have steep fronts that require resolution in time. It is one of the aims of this work to analyse the use of higher order spatial weighting schemes and temporal methods for reducing

* School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, AUSTRALIA.

⁰See <http://anziamj.austms.org.au/V42/CTAC99/Awad> for this article and ancillary services, © Austral. Mathematical Soc. 2000. Published 27 Nov 2000.

numerical dispersion. Another important ingredient in the development of an efficient simulator is the treatment of the non-linear system that results from the discrete analogue of the conservation law. In this work a vertex-centered finite volume method is used for discretising a representative conservation law in one-dimension and two non-linear iterative methods, an inexact full Newton method and the modified Shamanskii method, are scrutinised. Two case studies are chosen to highlight the performance of the chosen numerical techniques. At first, the focus is on the accuracy and efficiency of the spatial weighting methods for a linear advection-dispersion equation and then, a two-phase flow problem is analysed to gauge the performance of the non-linear solvers. In both cases, comparisons with exact solutions are presented.

Contents

1	Introduction	C159
2	Mathematical Models	C161
2.1	The Linear case	C161
2.1.1	Computational Model	C162
2.2	The Non-Linear case	C165
2.2.1	Computational Model	C167
3	Results and Discussions	C169

1 Introduction

The equations that govern the flow of liquids in porous media are ubiquitous in science and engineering. For example, the governing equations find application in fields as diverse as drying, ground water flow, contamination and petroleum reservoir engineering. In this work, the transport equations that govern flow in porous media will be analysed numerically. Two specific case studies are chosen to benchmark the proposed numerical solution strategies. In the first case study, saltwater intrusion into aquifers is studied. Over exploitation of coastal aquifers has resulted in the undesirable situation of saltwater intrusion, whereby continual loss of groundwater resources reduces the related aquifer crop production per hectare. As a result, accurate knowledge of the salt water concentration distribution evolution within the aquifer is a key component of current research work in this field [11, 12]. Such knowledge can be obtained using computational methods to simulate the transport of salt using a simple linear advection dispersion transport equation. The configuration under investigation here is chosen to ensure an exact solution can be derived so that the accuracy of the computational model can be gauged. In the second case study, the simultaneous unsteady flow of two viscous fluids through a porous medium is investigated. Consider

as an example the petroleum industry, where one of the fluids would be water and the other would be one of a variety of hydrocarbon liquids. The difficulty with this problem is the highly non-linear nature of the term arising from the so-called capillary drive. Fokas and Yortsos[1] obtained a closed form, analytic solution for one-dimensional flow with capillary drive under the assumption that the capillary hydraulic functions have a prescribed form. This analytic solution is used to test the accuracy of the numerical techniques proposed to solve the same configuration.

In both cases, the vertex centred finite volume method [9] is used to discretise the one-dimensional transport equation. Various spatial and temporal weighting schemes are used to approximate the flow and dispersion terms. These methods range in both accuracy and in levels of sophistication. A full monotonicity analysis has been carried out on the final discrete analogue of the transport equations in order to identify the constraints that must be placed on the distance and time steps for each proposed spatial and temporal weighting method. In particular, flux limiting will be shown to be an extremely useful technique for reducing numerical dispersion of the concentration and saturation fronts. Such techniques have been utilised with great success in the past for drying [5] and for reducing numerical dispersion of the contaminant mole fraction fronts in multi-phase compositional computational models [2] and [3].

The primary objective for both case studies is to identify the most accurate and efficient temporal and spatial weighting strategy. The final decision will be based on the method that produces the least absolute error and the

fastest CPU time. Numerous plots highlighting the performance of all the proposed numerical methods are exhibited. The final section will summarise the general conclusions of the research.

2 Mathematical Models

2.1 The Linear case

It can be shown, after numerous simplifying assumptions, that salt water intrusion into aquifers can be modelled by a single one-dimensional advection dispersion equation [11] [12]. The relevant partial differential equation in this case is given as follows:

$$\frac{\partial C}{\partial t} + V_x \frac{\partial C}{\partial x} = D_{xx} \frac{\partial^2 C}{\partial x^2}. \quad (1)$$

Equation (1) is to be solved using a finite volume method that incorporates flux limiting. $C(x, t)$ represents the concentration of salt in both space and time. V_x is the velocity field that advects the salt concentration and D_{xx} is the dispersivity, which in its simplest form depends on the velocity field. The boundary conditions are: $C(0, t) = C_0$ and $\lim_{x \rightarrow \infty} C(x, t) = 0$ and the initial condition is: $C(x, 0) = 0$. It is assumed in this case that V_x and D_{xx} are constants. Furthermore, throughout this section we assume that $V_x > 0$

so that the flow will be from the left ($x = 0$) boundary in the direction of increasing x .

Taking the Laplace transform of equation (1), the analytical solution can be shown to be [8]:

$$C(x, t) = \frac{C_0}{2} \left[\operatorname{Erfc} \left(\frac{x - V_x t}{2\sqrt{D_{xx}t}} \right) + e^{\frac{V_x x}{D_{xx}}} \operatorname{Erfc} \left(\frac{x + V_x t}{2\sqrt{D_{xx}t}} \right) \right]. \quad (2)$$

2.1.1 Computational Model

The vertex centred finite volume discretisation method now is applied to equation (1). In this work, a uniform grid spacing is used, although the model developed here can cater for nonuniform meshes. After the generation of the node locations, control volumes are built around the nodes x_i . See Figure 1 where the cell boundaries are located at positions $x_{i+\frac{1}{2}}$ and $x_{i-\frac{1}{2}}$ respectively.

The discretisation process proceeds by integrating equation (1) in space

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[\frac{\partial C}{\partial t} + V_x \frac{\partial C}{\partial x} - D_{xx} \frac{\partial^2 C}{\partial x^2} \right] dx = 0. \quad (3)$$

Using the Mean Value Theorem and defining the flux as: $J = V_x C - D_{xx} \frac{\partial C}{\partial x}$ allows equation (3) to be rewritten as

$$F_i \cong \Delta x_i \frac{\partial C_i}{\partial t} + \{ J_{i+1/2} - J_{i-1/2} \} = 0. \quad (4)$$

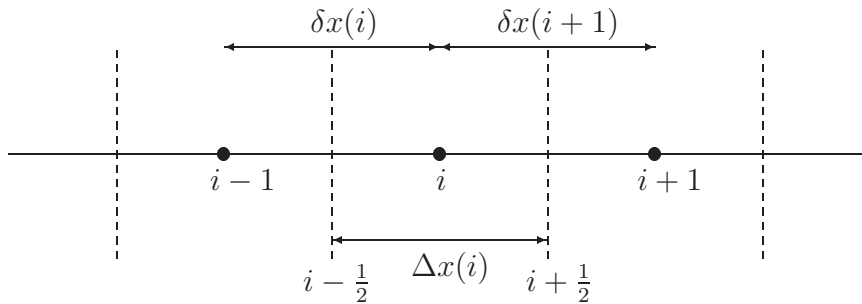


FIGURE 1: Schematic of a typical vertex centred finite volume scheme where the cell faces are located at the mid point of the vertices or nodes.

Integration of equation (4) in time from t_n to t_{n+1} yields:

$$\Delta x_i \left[C_i^{n+1} - C_i^n \right] + \left[\alpha \left(J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1} \right) + (1 - \alpha) \left(J_{i+\frac{1}{2}}^n - J_{i-\frac{1}{2}}^n \right) \right] \delta t = 0, \quad (5)$$

where a value of $\alpha = 1$ corresponds with the fully implicit method, which is first order in time, and the value $\alpha = \frac{1}{2}$ corresponds with the Crank Nicolson method, which is second order in time. Equation (5) is called the discrete analogue of the conservation law.

It can be seen from (5) that the fluxes must be approximated at the control volume (CV) faces $i + \frac{1}{2}$ and $i - \frac{1}{2}$, see Figure 1. $D_{xx} \frac{\partial C}{\partial x}$ is the dispersion term and $V_x C$ is the advection term. The required derivatives of concentration in the dispersion term are approximated at the CV faces

using a Taylor series expansion:

$$\left(\frac{\partial C}{\partial x}\right)_{i+\frac{1}{2}} = \frac{C_{i+1} - C_i}{\delta x_{i+1}} \quad \text{and} \quad \left(\frac{\partial C}{\partial x}\right)_{i-\frac{1}{2}} = \frac{C_i - C_{i-1}}{\delta x_i}. \quad (6)$$

The advection term will be approximated using various methods, including first order upstream, second order averaging and the van Leer Flux limiter [7]. The different approximations of the advection term at the CV faces can be cast succinctly into two equations that are:

$$C_{i+\frac{1}{2}} = C_i + \frac{\sigma_R}{2} (C_{i+1} - C_i) \quad \text{and} \quad C_{i-\frac{1}{2}} = C_{i-1} + \frac{\sigma_L}{2} (C_i - C_{i-1}), \quad (7)$$

where the limiters σ_R and σ_L are evaluated at the right and left faces of the control volume respectively. If $\sigma_R = \sigma_L = 0.0$, then the one point upstream spatial scheme is applied, and if $\sigma_R = \sigma_L = 1.0$, the second order spatial averaging scheme is applied. The flux limiters, σ_R and σ_L are calculated using a sensor r , which for the linear case is taken as the ratio of the gradient of the concentration of the upstream CV face divided by the gradient of the concentration of the so called second upstream CV face:

$$(r_R)_i = \frac{\left(\frac{dC}{dx}\right)_{i-\frac{1}{2}}}{\left(\frac{dC}{dx}\right)_{i+\frac{1}{2}}} = \frac{\frac{C_i - C_{i-1}}{\delta x_i}}{\frac{C_{i+1} - C_i}{\delta x_{i+1}}} = \frac{\delta x_i}{\delta x_{i+1}} \left(\frac{C_i - C_{i-1}}{C_{i+1} - C_i} \right). \quad (8)$$

The sensor at the CV left face is defined in a similar fashion. For the case of the van Leer flux limiter, the limiter used for both CV faces is $\sigma(r) = \frac{2r}{1+r}$ [7].

Substituting equations (6) and (7) into equation (5) and rearranging such that the n^{th} time level terms are on the RHS of the equation and the $(n+1)^{\text{th}}$ time level terms are on the LHS, the general equation representing the linear system to be solved numerically is given by:

$$\begin{aligned} & \left(\frac{\alpha \sigma_L V_x}{2} - \alpha V_x - \frac{\alpha D_{xx}}{\delta x_i} \right) C_{i-1}^{m+1} + \left(\frac{\alpha \sigma_R V_x}{2} - \frac{\alpha D_{xx}}{\delta x_{i+1}} \right) C_{i+1}^{m+1} + \\ & \left(\frac{\Delta x_i}{\delta t} + \alpha V_x - \frac{\alpha V_x}{2} (\sigma_L + \sigma_R) + \alpha D_{xx} \left(\frac{1}{\delta x_i} + \frac{1}{\delta x_{i+1}} \right) \right) C_i^{m+1} = \\ (\alpha - 1) & \left(\frac{V_x \sigma_L}{2} - V_x - \frac{D_{xx}}{\delta x_i} \right) C_{i-1}^n + (\alpha - 1) \left(\frac{V_x \sigma_R}{2} - \frac{D_{xx}}{\delta x_{i+1}} \right) C_{i+1}^n + \quad (9) \\ & \frac{\Delta x_i}{\delta t} C_i^n + (\alpha - 1) \left(V_x - \frac{V_x}{2} (\sigma_L + \sigma_R) + D_{xx} \left(\frac{1}{\delta x_i} + \frac{1}{\delta x_{i+1}} \right) \right) C_i^n. \end{aligned}$$

Clearly, once all the nodes of the mesh are visited, a tri-diagonal matrix system arises that can be solved efficiently using the Thomas algorithm for the salt concentrations at each node.

2.2 The Non-Linear case

The partial differential equation for the one-dimensional, horizontal flow of two immiscible, incompressible fluids is well known [12]. The non-linear

partial differential equation under consideration is:

$$\frac{\partial S}{\partial t} - \frac{\partial}{\partial x} \left[g(S) \frac{\partial S}{\partial x} + f(S) \right] = 0. \quad (10)$$

Initially $S(x, 0) = 1 - S_{wr}$.

The boundary conditions are:

$$\begin{aligned} x = 0, \quad \frac{\partial S}{\partial x} &= \frac{\alpha}{\beta} (\beta S + \gamma) + \frac{\delta}{\beta} (\beta S + \gamma)^2 \\ x \rightarrow \infty, \quad S &= 1 - S_{wr}. \end{aligned}$$

The functions $f(S)$ and $g(S)$ are defined as follows:

$$f(S) = \frac{\alpha}{\beta^2} \left[\frac{1}{1 - S_{wr} + \frac{\gamma}{\beta}} - \frac{1}{S + \frac{\gamma}{\beta}} \right] \text{ and } g(S) = \frac{1}{(\beta S + \gamma)^2} \quad (11)$$

where,

$$\frac{\alpha}{\beta^2} = - \frac{\left(S_{or} + \frac{\gamma}{\beta} \right) \left(1 - S_{wr} + \frac{\gamma}{\beta} \right)}{(1 - S_{wr} - S_{or})}. \quad (12)$$

$S(x, t)$ is the wetting phase saturation, $S_{wr} = 0.0375$ and $S_{or} = 0.15$ are constants called the irreducible saturation of water and oil respectively. $f(S)$ and $g(S)$ are the capillary-hydraulic properties of the fluid-porous system, $\delta = \beta - \alpha (\beta - \beta S_{wr} + \gamma)^{-1}$

and

$\frac{\gamma}{\beta}$ is a constant ratio.

The analytical solution presented in [1] uses a transformation to linearize (10) to a standard diffusion equation for $\phi(\tilde{x}, t)$. Once the solution $\phi(\tilde{x}, t)$ is determined, then a value of \tilde{x} must be computed for given x and t values by solving the nonlinear equation $\phi(\tilde{x}, t) = e^{\alpha x}$ using Newton's method. This value of \tilde{x} is used thereafter to enable the saturation to be evaluated at this particular (x, t) using the expression:

$$S(x, t) = \frac{1}{\beta} \left[\frac{\alpha e^{\alpha x}}{\phi'(\tilde{x}, t)} - \gamma \right]. \quad (13)$$

Further details on the numerical scheme used to tabulate the saturation for various x and t values can be found in [6].

2.2.1 Computational Model

In a similar fashion to that described above for the linear advection dispersion equation, the discretization process transforms the non-linear partial differential equation (10) into a system of non-linear equations.

$$F_i(S_{i-1}^{n+1}, S_i^{n+1}, S_{i+1}^{n+1}) = \Delta x_i \left(\frac{S_i^{n+1} - S_i^n}{\delta t} \right) - \alpha \left[J_{i+\frac{1}{2}}^{n+1}(S_i^{n+1}, S_{i+1}^{n+1}) - J_{i-\frac{1}{2}}^{n+1}(S_{i-1}^{n+1}, S_i^{n+1}) \right] + \quad (14)$$

$$(\alpha - 1) \left[J_{i+\frac{1}{2}}^n (S_i^n, S_{i+1}^n) - J_{i-\frac{1}{2}}^n (S_{i-1}^n, S_i^n) \right] = 0$$

Where the flux at the CV right face is defined as:

$$J_{i+\frac{1}{2}}^{n+1} = g_{i+\frac{1}{2}}^{n+1} \left(\frac{S_i^{n+1} - S_i^n}{\delta x_i} \right) + f_{i+\frac{1}{2}}^{n+1}.$$

The flux at the CV left face is defined in a similar manner and

$$g_{i+\frac{1}{2}} = \begin{cases} g(S_i) + \frac{\sigma}{2} [g(S_{i+1}) - g(S_i)] & \text{if } S_i > S_{i+1} \\ g(S_{i+1}) + \frac{\sigma}{2} [g(S_i) - g(S_{i+1})] & \text{if } S_i < S_{i+1} \end{cases} \quad (15)$$

$$g_{i-\frac{1}{2}} = \begin{cases} g(S_{i-1}) + \frac{\sigma}{2} [g(S_i) - g(S_{i-1})] & \text{if } S_{i-1} > S_i \\ g(S_i) + \frac{\sigma}{2} [g(S_{i-1}) - g(S_i)] & \text{if } S_{i-1} < S_i \end{cases} \quad (16)$$

with $f_{i+\frac{1}{2}}$ and $f_{i-\frac{1}{2}}$ defined similarly.

This system is solved using an Inexact Newton method [10] whereby the saturations at all the nodes can be updated at every iteration using the Newton step. The Jacobian matrix, \mathbf{J} , in this case is tridiagonal and the Newton search direction can be resolved using the Thomas algorithm. Note that the derivatives required for the Jacobian are generated numerically as discussed in [4]. Obviously, there is a large computational cost associated with generating and subsequently solving the Jacobian system. One way to overcome this computational overhead is to use the Shamanskii method along with a strategy that enables a decision on when to re-generate the Jacobian matrix [10].

Note that two sensors have been examined for the flux limiter in the non-linear case. The first is identical to that described above, which is based on the ratio of gradients at the upstream and second upstream points. The second, which is found to be more accurate than the gradient ratio, concerns the ratio of the entire flux at the upstream and second upstream locations (see [5]).

Two FORTRAN 90 codes were written for each case studied. The results of the simulations and comparisons with exact solutions, with some plots to highlight the most important findings for determining the most efficient computational techniques, will be presented in the next section.

3 Results and Discussions

The linear case monotonicity analysis shows that the fully implicit temporal method with first order upstream spatial weighting (Fup) has no constraint associated with it, implying that when different combinations of the parameters such as mesh size, dispersivity, velocity and time step δt are used the approximated solution remains stable and physical. The meaning of a physical solution is that no non-physical oscillations appear in the computed profiles and that the $C(x, t)$ and $S(x, t)$ values always remain positive and less than C_0 and 1 respectively. It is clear from Figure 2 that as the number of mesh points increases, the accuracy of the approximated solution also increases, with the smearing of the concentration front diminishing with decreasing

mesh size. This phenomenon is justified by recalling that this method is a first order process in both time and space. As we investigate this method more closely, one can see that increasing the number of mesh points, decreases the absolute error, however the CPU time increases by a factor of two. We conclude that this method is very costly in terms of computation time for producing realistic and accurate concentration profiles in comparison with the exact solution.

Consider now the Crank Nicolson method together with second order spatial averaging (*CAve*). The monotonicity argument shows that this method has two constraints associated with it: $P_e < 2$ and $\frac{C}{P_e} < 1$, where P_e is the Peclet number and C is the Courant number. Note that provided the two monotonicity conditions are satisfied, the approximated solution will remain stable and physical however, if either condition is breached, then oscillatory behaviour appears in the solution. Observe Figure 3 for the non-physical behaviour resulting from breaching the monotonicity constraint. Examining Figure 4, one can see that none of the constraints are breached and the results are in good agreement with the analytical solution. It is clear that this later method is more accurate and more efficient than the first method, however it too is costly in terms of computation time, and a penalty must be paid in terms of mesh size and magnitude of the time step due to the monotonicity constraints.

Next, flux limiting techniques are employed to ensure monotone solutions for concentration. These methods compute a sensor value that is used in the van Leer flux limiter for each CV face at every time level. Here the use

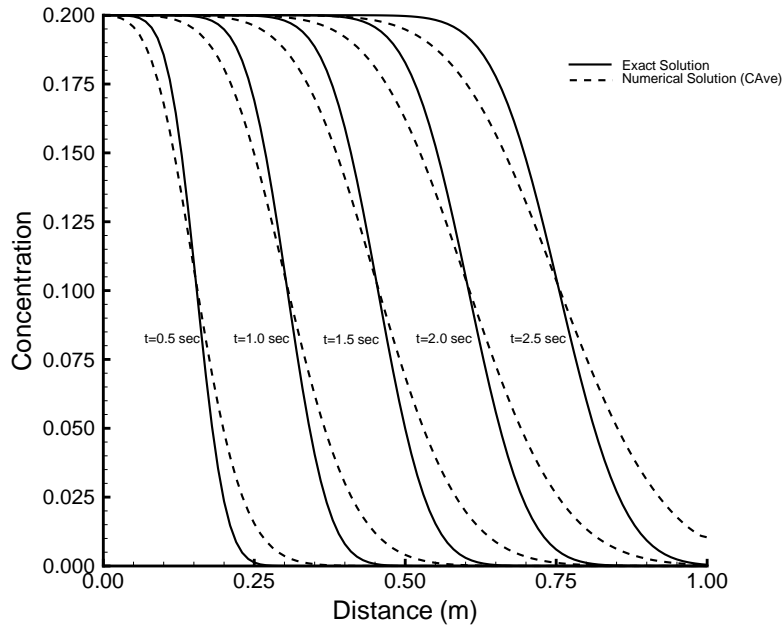


FIGURE 2: Evolution of concentration fronts using the fully implicit method and first order upstream weighting (*Fup*) for 100 mesh points.

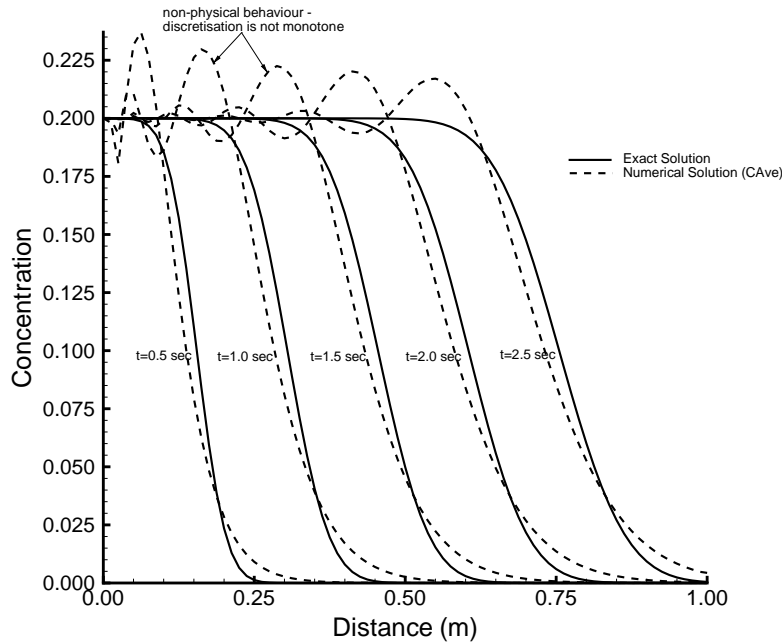


FIGURE 3: Evolution of concentration fronts using the Crank Nicolson method and averaging (*CAve*) for 80 mesh points - Non-physical behaviour.

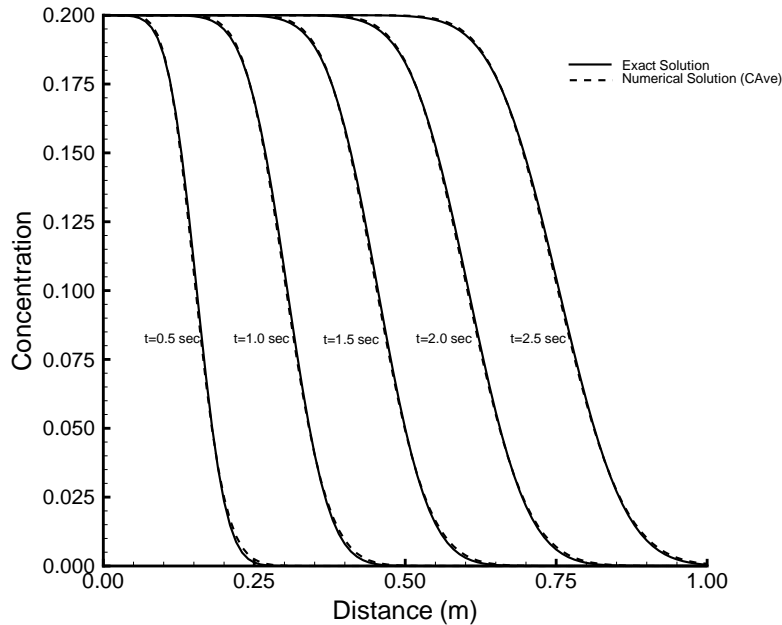


FIGURE 4: Evolution of concentration fronts using the Crank Nicolson method and averaging (*CAve*) for 100 mesh points - Physical behaviour.

TABLE 1: Results of the four study cases for the linear concentration transport model.

Case	Absolute Error	CPU time	P_e	$\frac{C}{P_e}$
1	0.02457	0.0837 sec	2.0	0.3
2	0.03127	0.0683 sec	2.5	1.92
3	0.001934	0.0772 sec	1.6	0.5
4	0.02752	0.0354 sec	6.66	0.027

of flux limiting is being investigated in terms of accuracy and efficiency in comparison with the other spatial weighting schemes described in this paper.

Four cases with total simulation time = 2.5 seconds are studied:

1. *Fup* with $V_x = 0.3$, $D_{xx} = 0.0015$, $\delta t = 0.02$, mesh points= 100.
2. *CAve* with $V_x = 0.3$, $D_{xx} = 0.0015$, $\delta t = 0.2$, mesh points= 80.
3. *CAve* with $V_x = 0.4$, $D_{xx} = 0.0025$, $\delta t = 0.02$, mesh points= 100.
4. *CVan* with $V_x = 0.3$, $D_{xx} = 0.0015$, $\delta t = 0.02$, mesh points= 30.

Analysing Figure 5 and Table 1 for the four cases listed it can be concluded that the use of flux limiting techniques in combination with Crank Nicolson second order temporal weighting (*CVan*) require approximately one third of

the total mesh points used by the fully implicit one point upstream method to maintain the same accuracy. Furthermore, the flux limiting method achieves this result in approximately half the CPU time. This finding is important when developing an efficient and accurate simulator for studying transport in porous media.

It is well known that Newton's method provides quadratic convergence if the initial approximation is sufficiently close to the root of the non-linear system, however, the modified Shamanskii method, called here the Definitive Newton method, is more efficient in terms of computation because the Jacobian matrix need not be computed at every iteration [10]. On the other hand, the Shamanskii method offers only super-linear convergence rates [10]. The parameters used for the two non-linear solvers are $\beta = 5$, $\gamma = 0.12$, 20 mesh points, print time interval = 0.1 seconds, $\delta t = 0.01$ seconds, and total time = 0.6 seconds. Figures 6 and 7 exhibit the simulation results for the different spatial and temporal weighting schemes used. Figure 6 depicts the results for the first and second order methods plotted against the analytical solution, while Figure 7 shows the results offered by the flux limiting technique for two different definitions of the sensor (ratio of the gradients and ratio of the fluxes). Clearly, the flux limiting again is successful in reducing numerical dispersion of the saturation fronts and when the sensor based on the ratio of fluxes is used, the results are in good agreement with the exact solutions. This finding was consistent when different combinations of the parameters β , or γ were used.

Finally, the overall performance of the Newton method (Table 2) and

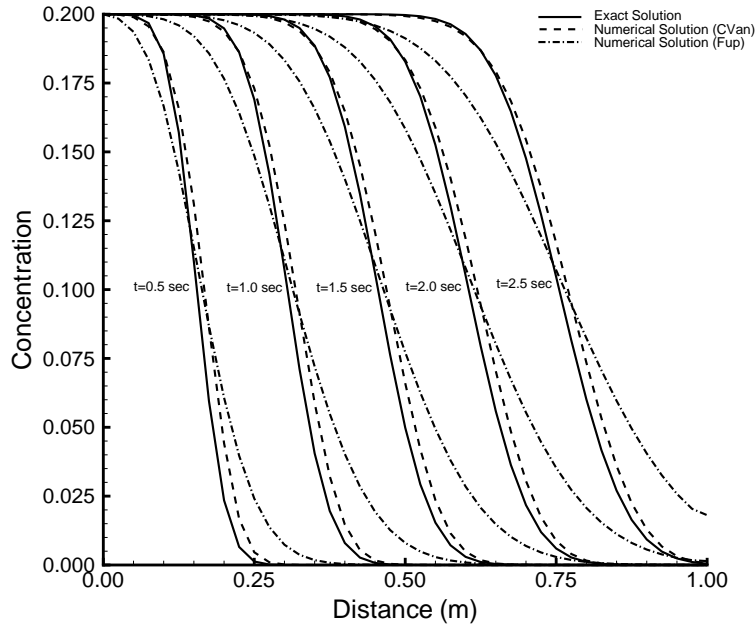


FIGURE 5: Comparison of concentration fronts using (Fup) and van Leer flux limiting and the Crank Nicolson method ($CVan$) for 30 mesh points.

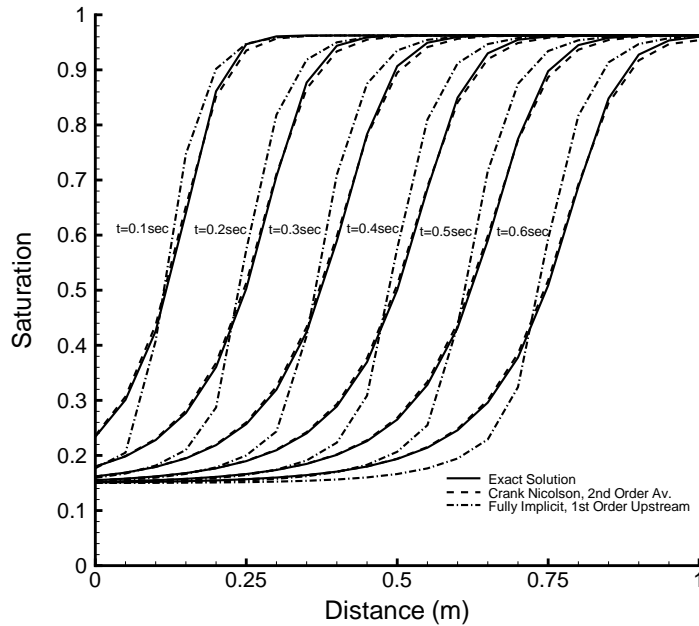


FIGURE 6: Exact solution for saturation plotted against the computed results for Fup and $CAve$ using 20 mesh points.

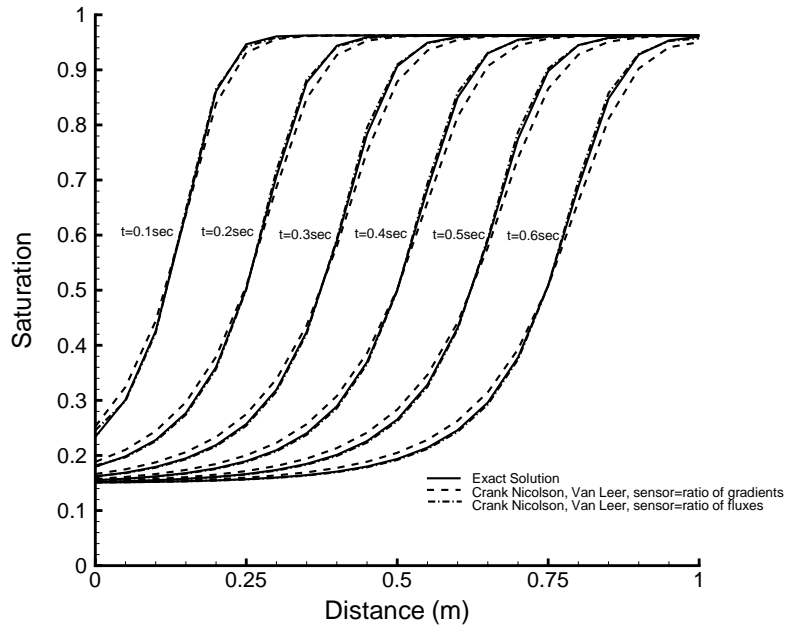


FIGURE 7: Exact solution for saturation plotted against the computed results for *CVan* using different sensors for 20 mesh points.

TABLE 2: Results of the four study cases of the non-linear transport equation using the Inexact Newton method.

Case	Absolute Error	CPU time	Total iterations	Jacobians
1	0.12865	4.98 sec	246	246
2	0.01046	4.05 sec	210	210
3	0.03712	6.23 sec	332	332
4	0.0105	9.21 sec	536	536

the Definitive Newton method (Table 3) have been compared via the total number of non-linear iterations and the total number of Jacobian evaluation. From those two tables, one notices that the number of iterations required by the Definitive method is either the same or, slightly higher than, that required by the Newton method. This finding can be justified because of the loss of quadratic convergence and probably due to the fact that the non-linearity of the problem presented here is not strong. On the other hand, the number of Jacobian evaluations dropped significantly for the Definitive method and this is where the computational overheads can be reduced.

TABLE 3: Results of the four study cases of the non-linear transport equation using the Definitive Newton method.

Case	Absolute Error	CPU time	Total iterations	Jacobians
1	0.12865	3.62 sec	253	194
2	0.01046	3.74 sec	246	187
3	0.03712	5.44 sec	333	274
4	0.01050	7.65 sec	535	420

4 Conclusions

Through the discussion of results and comparison of methods, it was concluded that the *Fup* technique produces an approximate solution that is always stable and physical. However, this method requires more than 100 mesh points to capture the physics apparent in the exact solution. In order to remedy this large computational overhead, we implement the method *CAve*. This method increases the accuracy as well as the efficiency, however a penalty is paid due to the monotonicity constraints, which, if breached, give rise to non-physical behaviour in the approximated solution. We conclude that using *CVan* requires one third of the mesh points, when compared to the fully implicit method with one point upstream, to maintain the same accuracy and needs only half the CPU time to ensure efficiency. This conclusion can be drawn for both the linear and non-linear cases. We conclude also that the flux limiting technique based on using the ratio of fluxes rather than the

ratio of the gradients for the sensor gives better results in terms of absolute error and CPU time. Finally, implementing the Definitive Newton method decreases the CPU time due to the reduction of Jacobian evaluations.

References

- [1] A. S. Fokas and Y. C. Yortsos. On the exactly solvable equation $S_t = [(\beta S + \gamma)^{-2} S_x]_x + \alpha (\beta S + \gamma)^{-2} S_x$ occurring in two-phase flow in porous media. *SIAM J. Appl. Math.*, 42(2):318–332, 1982. [C160](#), [C167](#)
- [2] P. A. Forsyth. Three-dimensional modelling of steam flush for DNAPL site remediation. *Int. J. Num. Methods in Fluids*, 19:1055–1081, 1994. [C160](#)
- [3] P. A. Forsyth, A. J. A. Unger and E. A. Sudicky. Nonequilibrium multiphase subsurface flow: An object oriented approach. Technical report CS-96-19, Dept. Comp. Sci., University of Waterloo, 1996. [C160](#)
- [4] I. W. Turner and P. Perré'. A synopsis of the strategies and efficient resolution techniques used for modelling and numerically simulating the drying process. In *Mathematical Modeling and Numerical Techniques in Drying Technology*, pages 1–82. Marcel Dekker Inc., 1996. [C168](#)

- [5] I. W. Turner and P. Perre'. The use of implicit flux limiting schemes in the simulation of the drying process: A new maximum flow sensor applied to phase mobilities. *J. of Applied Mathematical Modelling*, to appear 2000. C160, C169
- [6] M. Awad. *An investigation of flux-limiting and non-linear solution techniques for efficient simulation of transport in porous media* Master of Applied Science thesis, Queensland University of Technology, 2000.
- [7] B. van Leer. Towards the ultimate conservative difference scheme, II: Monotonicity and conservation combined in a second order scheme. *J. Comp. Phys.*, 14:361–370, 1974. C164, C164
- [8] A. Ogata and R. B. Banks. A solution of the differential equation of longitudinal dispersion in porous media. Professional Paper N0. 411-A, United States Geological Survey, 1961. C162
- [9] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemishpere Publishing Corporation, McGraw Hill, 1980. C160
- [10] C. T. Kelly. *Iterative Methods for Linear and Non-Linear Equations*. SIAM, Philadelphia, 1995. C168, C168, C175, C175
- [11] J. Bear and A. Verruijt. *Modeling Groundwater Flow and Pollution*. D. Reidel Publishing Company, 1998. C159, C161
- [12] J. Bear. *Dynamics of Fluids in Porous Media*. Elsevier, 1972. C159, C161, C165