

Multi-resolution adaptive modeling of groundwater flow and transport problems

H. Gotovac^{a,b}, R. Andricevic^{a,*}, B. Gotovac^a

^a Department of Civil and Architectural Engineering, University of Split, Matice Hrvatske 15, 21000 Split, Croatia

^b Division of Water Resources Engineering, KTH, Brinellvagen 32, 10044 Stockholm, Sweden

Received 9 January 2006; received in revised form 13 October 2006; accepted 21 October 2006

Available online 20 December 2006

Abstract

Many groundwater flow and transport problems, especially those with sharp fronts, narrow transition zones, layers and fingers, require extensive computational resources. In this paper, we present a novel multi-resolution adaptive *Fup* approach to solve the above mentioned problems. Our numerical procedure is the Adaptive Fup Collocation Method (AFCM), based on *Fup* basis functions and designed through a method of lines (MOL). *Fup* basis functions are localized and infinitely differentiable functions with compact support and are related to more standard choices such as splines or wavelets. This method enables the adaptive multi-resolution approach to solve problems with different spatial and temporal scales with a desired level of accuracy using the entire family of *Fup* basis functions. In addition, the utilized collocation algorithm enables the mesh free approach with consistent velocity approximation and flux continuity due to properties of the *Fup* basis functions. The introduced numerical procedure was tested and verified by a few characteristic groundwater flow and transport problems, the Buckley–Leverett multiphase flow problem, the 1-D vertical density driven problem and the standard 2-D seawater intrusion benchmark–Henry problem. The results demonstrate that the method is robust and efficient particularly when describing sharp fronts and narrow transition zones changing in space and time.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: *Fup* basis functions; Compact support; Method of lines; Adaptive *Fup* collocation method; Multi-resolution approach; Numerical dispersion; Groundwater flow and transport problems

1. Introduction

Many groundwater flow and transport problems exhibit a wide range of space and/or temporal scales characterized by sharp gradients resulting in fingering and layering with the existence of sharp interface and narrow transition zones. These characteristics are commonly present in problems of unsaturated and multiphase flow [5,13,23,34], density driven flow and transport [6,17,43] and reactive transport [3,39].

The numerical modeling of such processes usually presents significant difficulties in resolving numerical oscillations and dispersion. In order to overcome these difficulties demanding computational resources with a very fine grid and small time steps are needed. In recent numerical approaches adaptive methods with low computational costs are being developed. The first attempt to apply them has been focused upon using classical finite difference and finite element methods [2,17]. The main difficulty in applying these methods is finding a stable solution at the transition between zones having different discretization. Significant improvements have been obtained by the adaptive discontinuous finite element method, e.g. [5,34]. Recently, there have been many attempts to develop new adaptive procedures which, among others, are focused upon using the adaptive wavelet Galerkin methods [9,12] and collocation methods [7,8,15,25,26,46,47]. The adaptive wavelet Galerkin methods have three potential difficulties: treatment of general boundary conditions, treatment of

* Corresponding author. Tel.: +385 21 303 325; fax: +385 21 465 117.
E-mail addresses: Hrvoje.Gotovac@gradst.hr (H. Gotovac), Rokoand@gradst.hr (R. Andricevic), Blaz.Gotovac@gradst.hr (B. Gotovac).

nonlinearities and solving problems with complex domains. The first two difficulties can be successfully solved using the collocation procedure, while the third is still an open research topic.

The wavelet’s main feature is to facilitate the grid space adaptation and multi-resolution data compression. It enables solving problems with a sharp interface and narrow transition zone by changing their location and steepness in time and space. Wavelets utilize non-uniform grids dynamically adaptive according to the solution development. Any function, signal or data can be represented by a linear combination of basis functions (discrete wavelet transform) in multi-resolution fashion with different scales or frequencies and locations. This means that specific frequencies are associated with a particular spatial location that is not possible in classical Fourier transform (coefficients of linear combinations are wavelet coefficients which are associated with a specific resolution level [scale or frequency] and [collocation] point in space/time domain). This procedure is also known as multi-resolution analysis (MRA). Besides the solution variables (pressure, concentration, velocity) any other variables can be represented in the same multi-resolution fashion such as electrical or hydraulic conductivity, porosity, natural recharge or pumping. Furthermore, it is possible to use other basis functions with compact support (non-zero values only in one part of the domain) within the collocation method [46]. The spline adaptive collocation methods are described in [10,36,49].

Apart from wavelets and splines, there is a relatively lesser known class of atomic or R_{bf} basis functions (Rvachev’s basis functions) [37,38]. Atomic functions are classified between classic polynomials and spline functions. However, in practice, their application as basis functions is closer to splines or wavelets. In this paper we use *Fup* basis functions which are one type of atomic basis functions. Gotovac and Kozulić [20] systemized the existing knowledge on atomic functions and presented its usage and calculation into a numerically applicable form. The application of *Fup* basis functions has been demonstrated in signal processing [32,51], in initial value problems [21], and in the non-adaptive collocation method for boundary value problems [22,31].

The main objective of this paper is to develop the Adaptive Fup Collocation Method (AFCM) and demonstrate its application to groundwater flow and transport problems. Presented is a novel adaptive *Fup* collocation method which is well suited to dealing with strong nonlinear groundwater problems with sharp fronts and narrow transition zones. A numerical procedure is implemented through a method of lines. Spatial discretization and grid adaptation are obtained by *Fup* collocation transform, while time integration is obtained by solving the system of Differential-Algebraic Equations (DAE). Furthermore, this method enables the adaptive multi-resolution evolution of a solution with resolved spatial and temporal scales and a desired level of accuracy. The numerical method has

been tested and verified with a few characteristic groundwater flow and transport examples.

The following section presents a brief review of *Fup* basis functions. Section 3 describes the proposed collocation method; numerical examples are in Section 4, followed by conclusions in Section 5. Finally, two appendixes are added in order to show construction of the finite difference operator and convergence properties of the AFCM.

2. *Fup* basis functions preliminaries

Atomic or Rvachev’s basis functions – R_{bf} have a compact support and they are infinitely differentiable functions [20,37]. They are classified in between classical polynomials and spline functions, but in practice their application as basis functions is closer to splines and wavelets.

Atomic functions, $y(\cdot)$, are defined as solutions of differential-functional equations of the following type:

$$Ly(x) = \lambda \sum_{k=1}^M C_k y(ax - b_k) \tag{1}$$

where L is a linear differential operator with constant coefficients, λ is a scalar different than zero, C_k are coefficients of the linear combination, $a > 1$ is a parameter defining the length of the compact support and b_k are coefficients which determine displacements of basis functions.

The simplest function, which is the most studied among atomic basis functions, is the $up(x)$ function (Fig. 1a). Function $up(x)$ is a smooth function with compact support $[-1, 1]$, which is obtained as a solution of a differential-functional equation

$$up'(x) = 2up(2x + 1) - 2up(2x - 1) \tag{2}$$

with the normalized condition

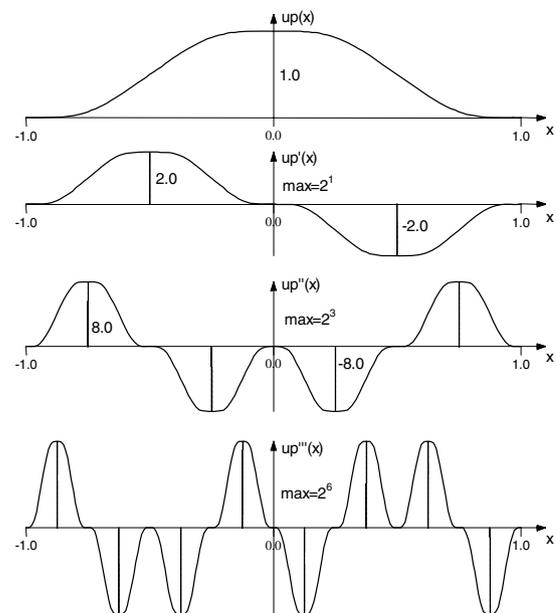


Fig. 1a. Function $up(x)$ and its derivatives.

$$\int_{-\infty}^{\infty} up(x) dx = \int_{-1}^1 up(x) dx = 1 \tag{3}$$

Function $up(x)$ can be expressed as an inverse Fourier transform

$$up(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itx} \prod_{j=1}^{\infty} \left(\frac{\sin(t2^{-j})}{t2^{-j}} \right) dt \tag{4}$$

Since Eq. (4) represents the exact, but not mathematically tractable expression, Rvačev [38] and Gotovac and Kozulić [20] provided tractable means for calculating function $up(x)$

$$up(x) = 1 - \sum_{k=1}^{\infty} (-1)^{1+p_1+\dots+p_k} p_k \sum_{j=0}^k C_{jk} (x - 0, p_1 \dots p_k)^j \tag{5}$$

where coefficients C_{jk} are rational numbers determined according to the following expression:

$$C_{jk} = \frac{1}{j!} 2^{j(j+1)/2} up(-1 + 2^{-(k-j)}); \quad j = 0, 1, \dots, k, \tag{6}$$

$$k = 1, 2, \dots, \infty$$

Calculation of the few first $up(x)$ values in Eq. (6), as well as all details regarding the calculation of the function $up(x)$ values, is provided in [20,21]. The argument $(x - 0, p_1 \dots p_k)$ in Eq. (5) is the difference between the real value of coordinate x and its binary form with k bits, where $p_1 \dots p_k$ are digits, 0 or 1, of the binary development of the coordinate x value. Therefore, the accuracy of the coordinate x computation and, thus, the accuracy of the $up(x)$ function in an arbitrary point, depend upon machine accuracy.

From Eq. (2) it can be seen that the derivatives of the $up(x)$ function can be calculated simply from the values of the function itself. The general expression for the derivative of the m th degree is

$$up^{(m)}(x) = 2^{C_{m+1}^2} \sum_{k=1}^{2^m} \delta_k up(2^m x + 2^m + 1 - 2k), \quad m \in N \tag{7}$$

where $C_{m+1}^2 = m(m+1)/2$ is the binomial coefficient and δ_k are the coefficients with ± 1 value which determine the sign of each term. They change according to the following recursive formulas:

$$\delta_{2k-1} = \delta_k, \delta_{2k} = -\delta_k, \quad k \in N, \delta_1 = 1 \tag{8}$$

Fig. 1a shows the $up(x)$ function and its derivatives. It can be observed that the derivatives consist of the $up(x)$ function compressed to the interval of 2^{-m+1} length with ordinates “extended” with the $2^{C_{m+1}^2}$ factor. The $Fup_n(x)$ function can be defined as a linear combination of the $up(x)$ function. The general form of the Fourier transform $F_n(t)$ for the function $Fup_n(x)$ follows:

$$F_n(t) = \left(\frac{\sin(t2^{-n-1})}{t2^{-n-1}} \right)^{n+1} \prod_{j=n+2}^{\infty} \frac{\sin(t2^{-j})}{t2^{-j}} \tag{9}$$

Function $Fup_n(x)$ can be written as an inverse Fourier transform

$$Fup_n(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itx} F_n(t) dt \tag{10}$$

Eq. (10) is not numerically tractable for the calculation of the $Fup_n(x)$ function. It is numerically more convenient to construct the $Fup_n(x)$ function in the form of a linear combination of displaced $up(x)$ functions. Index n denotes the highest degree of the polynomial which can be expressed accurately in the form of a linear combination of $Fup_n(x)$ basis functions displaced by a characteristic interval 2^{-n} . For $n = 0$, $Fup_0(x) = up(x)$, since the $Fup_n(x)$ function values are calculated using a linear combination of displaced $up(x)$ functions

$$Fup_n(x) = \sum_{k=0}^{\infty} C_k^*(n) up\left(x - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{11}$$

where coefficient $C_0^*(n)$ is

$$C_0^*(n) = 2^{C_{n+1}^2} = 2^{n(n+1)/2} \tag{12}$$

and other coefficients of a linear combination in Eq. (11) are determined as $C_k^*(n) = C_0^*(n) \cdot C'_k(n)$, where a recursive formula is used for calculating auxiliary coefficients $C'_k(n)$

$$C'_0(n) = 1, \quad \text{when } k = 0; \text{ i.e. when } k > 0$$

$$C'_k(n) = (-1)^k C_{n+1}^k - \sum_{j=1}^{\min\{k, 2^{n+1}-1\}} C'_{k-j}(n) \cdot \delta_{j+1} \tag{13}$$

The $Fup_n(x)$ function support is determined according to

$$\text{supp } Fup_n(x) = [-(n+2)2^{-n-1}; (n+2)2^{-n-1}] \tag{14}$$

where ‘supp’ denotes the length of the compact support. Derivatives of the $Fup_n(x)$ function are also obtained by a linear combination of derivatives of displaced $up(x)$ functions according to Eq. (11). Fig. 1b shows the $Fup_2(x)$ function and its first two derivatives. Thus, a quadratic

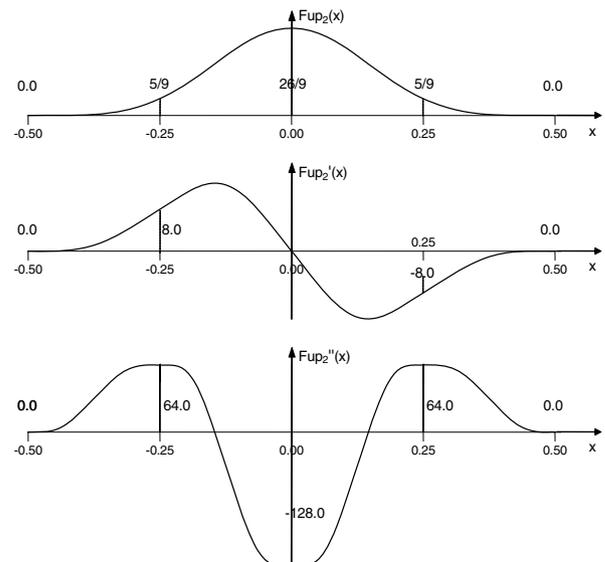


Fig. 1b. Function $Fup_2(x)$ and its first two derivatives.

polynomial on a characteristic interval 2^{-n} can be exactly expressed in the following way:

$$x^2 = 2^{-6} \sum_{k=-1}^2 (k^2 - 5/18) \text{Fup}_2(x - k/4) \quad (15)$$

Generally, $n + 2$ $\text{Fup}_n(x)$ or 2^{n+1} $up(x)$ basis functions are needed for the development of an n -order polynomial on a characteristic interval 2^{-n} . This clearly shows that $\text{Fup}_n(x)$ basis functions are more suitable and efficient than $up(x)$ basis functions for numerical purposes.

Fup basis functions, wavelets and splines are similar mainly due to the compact support and possible numerical implementation. Generally, approximation properties of all these basis functions are related to the developing of algebraic polynomials. The basic difference between Fup basis functions and wavelet and spline basis functions is that, generally, atomic basis functions are exact solutions of differential-functional equations or linear combination of these exact solutions, but splines and wavelets are obtained from some types of mathematical transforms. This is the reason why Fup basis functions have infinite number of derivatives and non-vanishing moments. Therefore, Fup basis functions belong to the universal vector space such that more basis functions give better solution or at least maintain the obtained accuracy. This requirement is not always satisfied in numerical procedures utilizing splines and wavelets [46] or in conventional finite element methods. A more detailed discussion about Fup and atomic basis functions is given in [20,21].

3. Adaptive Fup collocation method

This section deals with a detailed description of the **A**daptive **F**up **C**ollocation **M**ethod (AFCM) including its features in space and time. Presented are all necessary steps for AFCM implementation to be reproducible.

3.1. Fup collocation transform

The Fup collocation transform (FCT) is an efficient numerical tool for describing various types of signals and functions using a linear combination of the Fup basis functions. It is a discrete type of transform, similar to the classic discrete Fourier transform (DFT), where linear combination coefficients are called Fup coefficients. However, the main disadvantage of DFT lies in the fact that unresolved locations of important frequencies has not been defined due to non-localized properties of classic trigonometric basis functions. Thus, the essential problem with DFT becomes a natural advantage of a presented transform based on the chosen basis function with a compact support (Fig. 1). In other words, the specific frequencies are associated with a particular spatial location which is not possible in the classic Fourier transform. Fup coefficients are associated with a specific resolution level and location in the space/time domain. This resolution level defines the spatial

discretization level prescribed by a specific number of collocation points used to describe the given function. For example, a smooth function is presented only by a few frequencies in the DFT or a few coarse resolution levels in the FCT. On the other hand, for a function with sharp fronts and large gradients, the DFT shows a wide range of frequencies without any information on their spatial locations, while the FCT adds higher resolution levels and frequencies only in the front regions and resolves all spatial scales and their locations. This procedure is also known as a multi-resolution analysis. The transform is obtained through a collocation procedure and is therefore called the Fup collocation transform. The high efficiency of the FCT lies in the transform property which keeps only significant Fup coefficients which accurately describe the chosen function. Other Fup coefficients present a residual between a true function and their Fup presentation which must be less than the prescribed spatial threshold $-\varepsilon$. This threshold has a fundamental meaning for the FCT because it presents the Fup approximation accuracy or the FCT precision level. In this way, any functions in a multi-resolution fashion are decomposed using only a few significant Fup basis functions with appropriate scales (frequencies) and locations, a desired level of accuracy and minimum computational cost.

As in usual transformations (e.g., Fourier), if the Fup coefficients are known, the function can be calculated and *vice versa*. For example, the multi-resolution expansion of the $u(x)$ function can be expressed in the following way:

$$u(x) = \sum_{j=0}^{J \rightarrow \infty} \sum_{k=-n/2}^{2^{\min+j+n/2}} d_k^j \phi_k^j(x) \quad (16)$$

where j is the resolution level, from zero to a maximum level J , needed for the Fup presentation in Eq. (16), n is the Fup order, j_{\min} is the resolution at the zero level, d_k^j are Fup coefficients, ϕ_k^j are Fup basis functions and k denotes the location index at the current level. We consider a set of dyadic grids:

$$G^j = \{x_k^j \in R : x_k^j = 2^{-j}k, k \in Z\}, \quad j \in Z \quad (17)$$

where x_k^j are the grid collocation points. Note that even a numbered collocation point of G^{j+1} already exists in G^j ($x_{2k}^{j+1} = x_k^j$). It implies the relation $G^j \subset G^{j+1}$. The example of a dyadic grid is displayed in Fig. 2. We use regular grid terminology for a grid containing all possible points at all levels. The grid is irregular if at least one collocation point, at any resolution level, is omitted. If we define the domain $\Omega = [X_1, X_2]$, then the characteristic interval at each level is equal to the scale or distance between adjacent collocation points

$$\Delta x_j = \frac{X_2 - X_1}{2^{j_{\min}+j}} \quad (18)$$

For demonstrating FCT, consider the following test function:

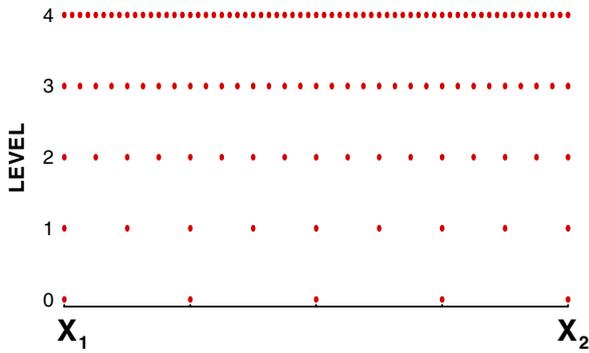


Fig. 2. Example of the regular dyadic grid.

$$f(x) = -\tanh\left(\frac{x - 2/3}{0.02}\right) \tag{19}$$

with a relatively high threshold of $\varepsilon = 0.07$ which implies that the residual between the *Fup* approximation and function in Eq. (19) must be less than the prescribed threshold. Other parameters are $j_{\min} = 2$, $X_1 = 0$, $X_2 = 2$ and $n = 4$. Fig. 3 shows the location of internal and external basis functions and the corresponding collocation points. Basis functions are characterized by vertices or peaks having maximum values. All basis functions whose vertices are located inside the domain are called internal basis functions. Other functions are external basis functions and only their influence within the domain is considered (bold parts of the external basis functions in Fig. 3).

The best choice for the location of the collocation points is vertices of the internal basis functions as proven numerically for splines in [35], wavelets in [46] and *Fup* basis functions in [20]. Moreover, the main difficulty in transformations with localized basis functions is the special treatment of the boundary. For a complete *Fup* approximation in every characteristic interval Δx_j , we need $n + 2$ *Fup*_{*n*}(*x*) basis functions (i.e. Eq. (15)) which exactly develops a *n*-order polynomial. This request is violated near the boundary if external basis functions are not used because accuracy is less than inside the domain (Fig. 3). If external basis functions are used then a problem arises when defining locations of additional collocation points and consistent conditions for their implementation in the collocation procedure. Bertoluzza and Naldi [8] reported three possible solutions for solving this problem:

(1) without external basis functions which leads to a stable, but inaccurate solution, as mentioned above; (2) by constructing internal basis functions near the boundary of higher order accuracy than other internal basis functions and (3) by replacing additional collocation points at the locations which belong to higher levels near the boundary [46].

Another approach has been employed in this paper which arises from the properties of the *Fup* basis function. For all $n/2$ external basis functions at the left and right boundaries, the collocation points are located at the (at X_1 and X_2) boundary as shown in Fig. 3. The approximation for internal and external basis functions should satisfy the function values at corresponding collocation points and first $n/2$ derivatives at the boundary collocation points (at X_1 and X_2), respectively.

The location of each basis function is actually determined by the location of the vertices and defined by $b_k^j = X_1 + k\Delta x_j$. The calculation of basis function values and their derivatives at a general characteristic interval Δx_j should be done in the following form with respect to a basic characteristic interval 2^{-n} :

$$\varphi_k^{j(m)}(x) = \frac{1}{(2^n \Delta x_j)^{(m)}} Fup_n^{(m)}\left(\frac{x - b_k^j}{2^n \Delta x_j}\right) \tag{20}$$

where m is the order of the derivative. The compact support of the basis function at every level has $(n + 2)\Delta x_j$ length.

Fig. 4 presents the adaptive multi-resolution *Fup* collocation transform for a chosen function (Eq. (19)). Fig. 4a shows an adaptive grid for all levels and internal basis functions for the zero and first level. Every next level includes two times more internal basis functions with two times less support and scale (Eq. (18)). Note that smaller scales at higher levels involve higher frequencies and detailed approximation properties which are particularly important for zones with large gradients. Zero level is the starting (coarsest) level which is always present in the grid. The FCT satisfies function values in all collocation points and for the first two derivatives in boundary points (Fig. 4b). The key step of the FCT is the transfer from the current level to the next level. The residual between the true function and the previous level approximation is checked and the points with a residual below the prescribed

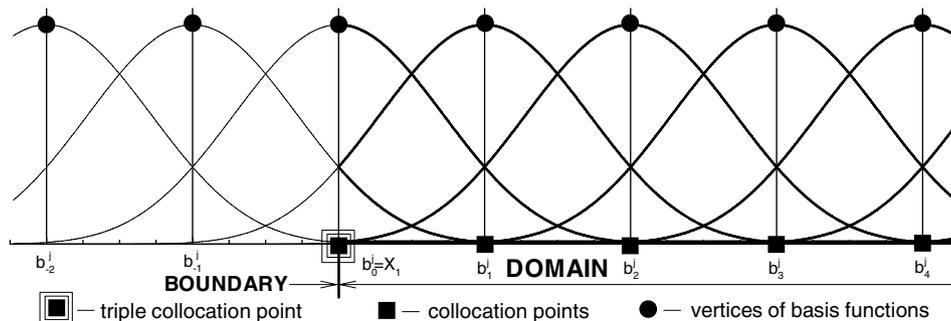


Fig. 3. Vertices location of the internal and external *Fup*₄(*x*) basis functions and the corresponding collocation points.

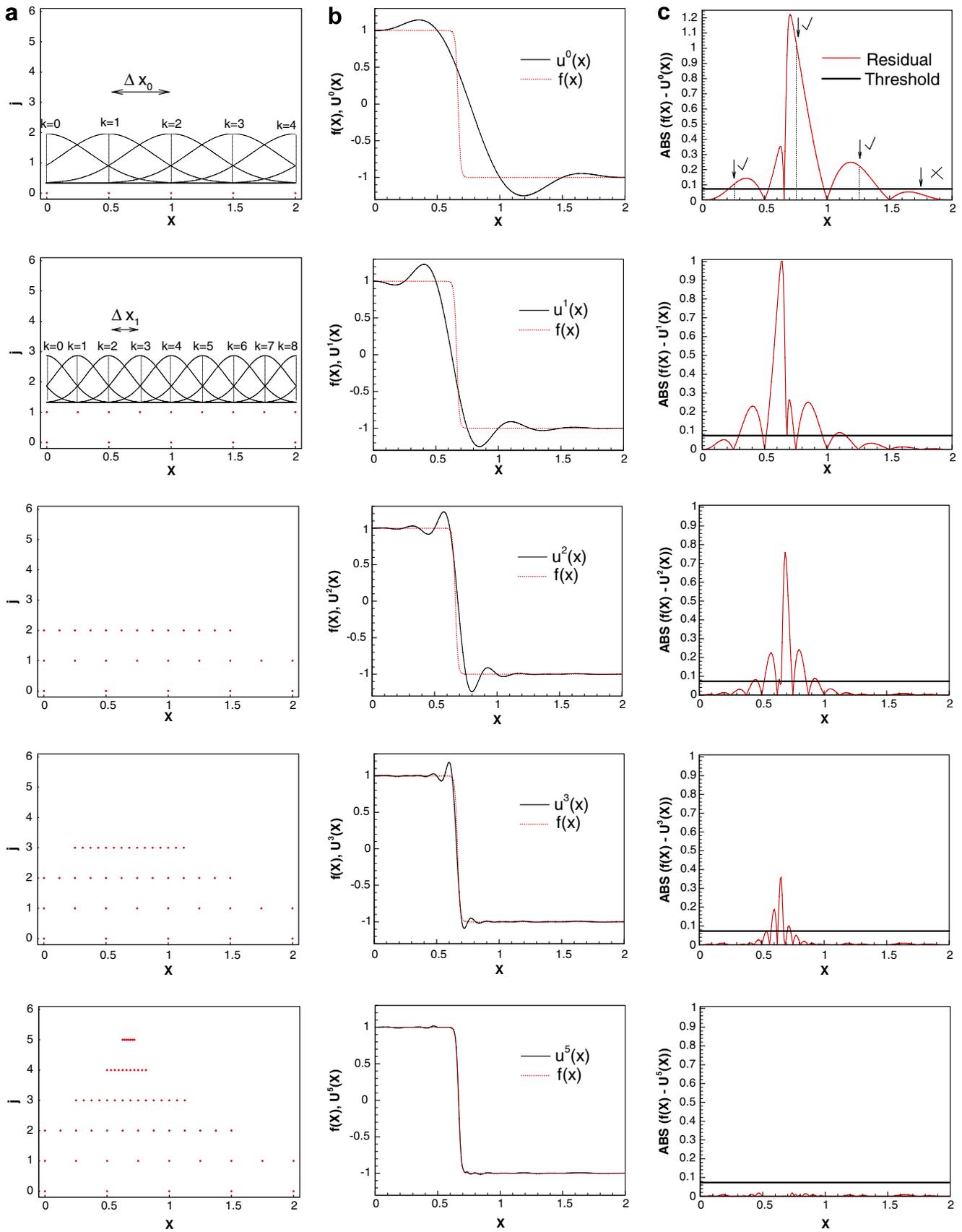


Fig. 4. Multi-resolution approximation of the function (19), (a) adaptive grid development and internal basis functions, (b) FCT approximation and function (19) and (c) a-priori adaptive criterion for new collocation points based on residual between function (19) and their FCT approximation.

threshold are dropped from the grid (Fig. 4c). This procedure presents a-priori adaptive criterion for defining the new collocation points at the next level (different from a classical a-posteriori criterion in the adaptive finite element method). Note that residuals are always zero for even collocation points. Each retained point must be surrounded by $n + 2$ basis functions which enable a consistent approximation for the transfer to the next level. In addition, external basis functions should be added if points near the boundary are present in the grid.

For the first and for each subsequent level the collocation algorithm should only satisfy the residual between the true function and the previous level approximation. Boundary derivatives for the first and each subsequent level are homogeneous (zero value) since they are satisfactory at the zero level.

Higher levels include only higher frequencies and show a more detailed description of the chosen function. The collocation points are added only around the front where the residual from the previous level is greater than the prescribed threshold (Fig. 4a and c). Finally, the residual between true function and the *Fup* approximation up to five levels is less than the threshold within the entire domain. In this way, we can decompose any function in a multi-resolution fashion by employing only a few significant *Fup* basis functions with appropriate scales (frequencies) and locations with a desired level of accuracy and a near minimum computational cost. Finally, the meaning of the threshold is twofold: (1) it presents a-priori adaptive criterion in such a way that points can be dropped from the grid where the residual between the real function and the *Fup* approximation is less than the threshold and (2) it defines the accuracy of the approximation because the final absolute difference between the *Fup* approximation and the real function must be less than the threshold.

In general, the *Fup* collocation transform modifies Eq. (16) in order to use the adaptive procedure and can be presented by

$$u^j(x) = \sum_{j=0}^J \sum_{k \in Z^j} d_k^j \phi_k^j(x) \quad (21)$$

where Z^j is the irregular grid containing only the significant collocation points and the *Fup* basis functions obtained using the above presented adaptive procedure (Fig. 4). The function values are satisfied at collocation points

$$\sum_{k \in Z^j} d_k^j \phi_k^j(x_p^j) = \Delta_j(x_p^j), \quad p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j}; \quad j = 0, \dots, J \quad (22)$$

The boundary derivatives are satisfied at points X_1 and X_2

$$\sum_{k \in Z^j} d_k^j \phi_k^{j(i)}(X_b) = \Delta_j^{(i)}(X_b), \quad i = 1, \dots, n/2; \quad b = 1, 2; \quad j = 0, \dots, J \quad (23)$$

The residual vector has the following form:

$$\Delta_j(x_p^j) = \begin{cases} f(x_p^j), & p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j}; \quad j = 0 \\ f(x_p^j) - u^{j-1}(x_p^j), & p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j}; \quad j = 1, \dots, J \end{cases}$$

$$\Delta_j^{(i)}(X_b) = \begin{cases} f^{(i)}(X_b), & j = 0; \quad b = 1, 2 \\ 0, & j = 1, \dots, J \end{cases} \quad (24)$$

3.2. Adaptive numerical algorithm

This section presents the adaptive *Fup* collocation method with its main numerical properties. The main idea behind AFCM is to incorporate the FCT spatial description into an adaptive algorithm for solving PDEs for groundwater flow and transport problems. The main feature of the AFCM is the adaptive change of the grid in time. In this way, the grid follows the system dynamics, i.e. changes in space and time according to different spatial and temporal scales determined during the adaptation procedure.

The AFCM is designed by a method of lines using the separation between spatial and temporal evolution. After each time step, the space discretization on a dyadic grid is obtained by a *Fup* collocation transform and the corresponding spatial adaptive strategy. Time integration is obtained by solving the system of differential-algebraic equations written in a general form suitable for groundwater flow and transport problems

$$A(t, u) \frac{\partial u}{\partial t} = F(t, x, u, u^{(m)}) \quad (25)$$

$$0 = G(t, x, u, u^{(m)}) \quad (26)$$

where u is the solution, m is the order of derivatives and A , F and G are linear or nonlinear operators depending upon the considered problem. Eq. (25) represents time-dependent partial differential equation which describes time evolution of the solution while the algebraic equation (26) presents the boundary conditions (Dirichlet, Neumann or Cauchy mixed type). Fig. 5 presents a flow chart of the adaptive *Fup* collocation method. This paper is mainly focused on the spatial approximation, while temporal integration is solved using classic multi-step routines.

The general numerical algorithm consists of three commonly used basic steps [15,16,47,49]:

1. Spatial grid adaptation procedure.
2. Calculation of spatial derivatives.
3. Time integration procedure.

Subsequently, the above steps are described in more details:

3.2.1. Spatial grid adaptation

The spatial grid adaptation procedure means changing the grid in order to resolve different spatial scales. The spatial adaptive procedure is performed after each time step according to the prescribed FCT and the corresponding adaptive strategy. This procedure dynamically changes the grid and significantly reduces the computational cost.

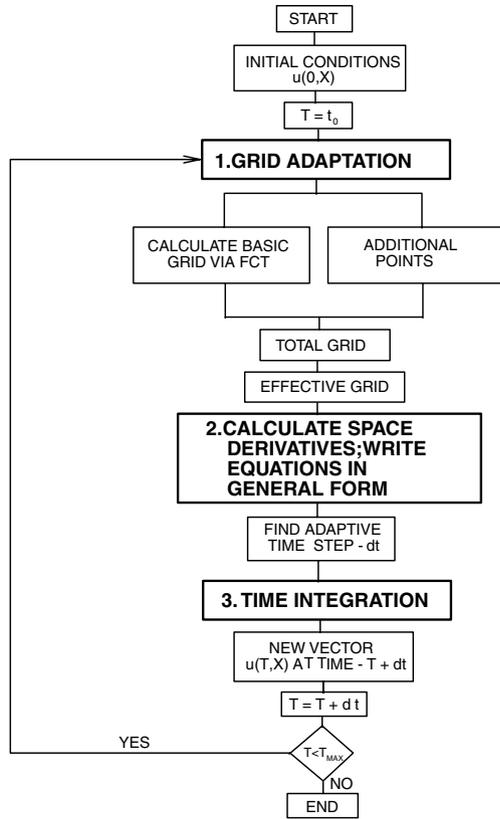


Fig. 5. Flow chart for the adaptive *Fup* collocation method.

The main part of the spatial adaptation strategy is the *Fup* collocation transform or the approximation of the solution from the initial conditions or previous time step. All FCT points are called basic points since they create the basic grid. Apart from basic points (which are related to a-priori adaptive criterion), we need additional points which enable the consistent approximation of the system dynamics (temporal solution changes) during the calculated adaptive time step Δt . Basic and additional points create the total grid needed for the description of the system dynamics from time T to time $T + \Delta t$. The basic hypothesis behind the algorithm (during time step Δt) is that the solution does not “move” outside the border of the adapted non-uniform grid. However, the total grid is not appropriate for time integration because of the repetition of some collocation points at different levels. Thus, the total grid needs to be transformed into an effective grid suitable for time integration.

The spatial grid adaptation strategy consists of commonly used steps [15,16,25,29,47]. Their modification and adjustment to the AFCM are summarized below:

- (a) Knowing the function values from the initial conditions or from previous time steps we perform the FCT solution. In this way we get the basic grid (based on a-priori adaptive criterion) required for the *Fup* approximation with a desired solution accuracy defined by threshold ε . Furthermore, we get a continuous solution and all derivatives in the form of a linear combination of the *Fup* basis functions. These basic points describe the solution at time T , but additional points are needed for the description of possible solution changes between T and $T + \Delta t$.
- (b) For each basic point x_k^j we add a certain number of additional points to the left and right at the same level ($x_{k+i}^j, i = -N_L, \dots, N_R$). These points are included to guarantee an accurate approximation of the possible movement of sharp solution features during the time step (Fig. 6a, $N_L = N_R = 2$). For advection dominated problems, the maximum allowed time step must be related to the maximum velocity in the following way [15]: $\Delta t_{\text{adapt}} = \max(N_L, N_R) \Delta x_{j_{\text{max}}} / v_{\text{max}}$ in order to guarantee that the front will not move beyond a distance $\max(N_L, N_R) \Delta x_{j_{\text{max}}}$ at the finest resolution level. Note that we chose an arbitrary number of additional points (N_L, N_R) which are directly connected with the size of the time step.
- (c) For each basic point x_k^j we add additional points at the arbitrary number (M) for higher resolution levels $\left(x_{2^l(k-N_L^U)-2^l+1}^{j+l}, \dots, x_{2^l(k+N_R^U)+2^l-1}^{j+l}, l = 1, \dots, M\right)$.

Note that parameters N_L^U and N_R^U must be less or equal to N_L and N_R , respectively, since it is impossible to add points at higher levels without the existence of corresponding points at lower levels. These points are included to guarantee an accurate approximation if the solution becomes steeper in this part of the domain during the time step (Fig. 6b, $N_L^U = N_R^U = M = 1$). For these additional points and for the number of higher resolution levels (M), there is no exact calculation due to a-priori unknown steepness of the solution during the next time step. Numerical experiments show that $M = 1$ is usually sufficient for most problems, but it generally depends on

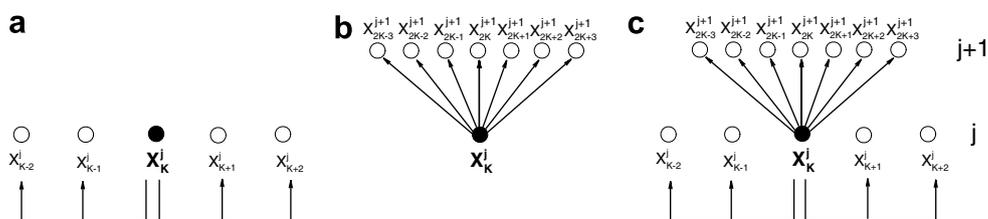


Fig. 6. Additional points at the same level (a), level above (b) and at the same and above level (c), $N_L = N_R = 2, N_L^U = N_R^U = M = 1$.

numerical and physical characteristics and should be tested for all kinds of problems [15,47]. In this paper we employ $M = 1$ as a sufficient number for very accurate modeling of the groundwater flow and transport problems. All additional points are shown in Fig. 6c.

- (d) Create the total grid by adding basic and additional points. In the case of more dependent variables (each one having its own grid) create the union of all particular grids.
- (e) The effective grid is constructed from the total grid in the following way: at the zero resolution level all collocation points belong to the effective grid, but at higher levels only odd numbered collocation points are kept. Thus, this procedure reduces the number of collocation points (approximately 50% for 1-D, 25% for 2-D and only 12.5% for 3-D problems).

Adaptive spatial strategy is directly dependent on the connection between the number of additional points and frequency of the grid adaptation. If the grid adaptation is not performed frequently, a larger number of points should be added and *vice versa*. The optimal strategy is unknown a priori because it depends on the physical and numerical side of the considered problem. Furthermore, the procedure can be utilized by different criteria for the grid adaptation. Apart from the analysis of function values, the procedure can use function derivatives or some other physical criteria (Peclet number) or a combination of different numerical and physical criteria.

3.2.2. Calculation of spatial derivatives

The time integration algorithm (DAE system 25–26) requires numerous calculations of spatial derivatives on an adaptive grid in operators F and G . An efficient algorithm needs fast and accurate calculation of spatial derivatives from the function values at collocation points. Hence we apply a standard procedure and construct the finite difference (FD) operator on an adaptive non-uniform grid [15,47]. Note that the *Fup* order is closely related to the order of the FD operator. If we use the same order for *Fup* basis functions and for the FD operator, the calculated spatial derivatives on an adaptive non-uniform grid should be very similar. A more detailed description of the calculation of the spatial derivatives is given in Appendix A.

3.2.3. Time integration

Time integration is obtained by solving the system of the differential-algebraic equations (25) and (26) with initial conditions obtained either from original initial conditions (first time step) or from the previous time step. The system (25) and (26) changes after every time step due to the applied spatial grid adaptation and contains all points from the adapted grid. During the time step, the adaptive grid and the system (25) and (26) remain unchanged.

By applying the collocation procedure to the system (25) and (26) and using the described FD operator for spatial

derivatives and backward differential formulas (BDF) for temporal derivatives, a discrete implicit form of the DAE system can be obtained and can be solved for a given time step by public domain subroutine DASPCK [4]

$$0 = \mathbf{H}\left(t_n, \mathbf{u}_n, \frac{\partial \mathbf{u}_n}{\partial t}\right) = \mathbf{H}\left(t_n, \mathbf{u}_n, \sum_{j=0}^k \alpha_j^{(1)} \mathbf{u}_{n-j}\right) \quad (27)$$

where $\alpha_j^{(1)}$ are BDF coefficients, n is the index of the current time step and k represents the order of the method.

DASPCK uses the implicit Petzold-Gear (BDF) method with a variable order (up to the fifth order) and the adaptive inner step size with variable coefficient strategy. Note that it is very important to distinguish the outer time step when the adaptive grid remains unchanged from the inner time step needed for time integrator routine DASPCK to achieve accuracy and stability. This routine is appropriate for systems of stiff equations (usually for strongly nonlinear groundwater problems) and attempts to keep the local error proportional to a user-specified tolerance [47]. Unfortunately, it does not guarantee that the global integration error is controlled and therefore this tolerance should be smaller by a few orders than the threshold ε in order to keep the global numerical accuracy closely related to the spatial approximation error (Appendix B).

Within a DASPCK subroutine, the modified Newton method is used for solving the nonlinear system (25),(26). Without a loss of generality, we can show that the Newton algorithm will have the following form when using the simple backward Euler method or the BDF method of the first-order ($k = 1$)

$$\mathbf{u}_n^{v+1} = \mathbf{u}_n^v - \left(\frac{1}{\Delta t_n} \frac{\partial \mathbf{H}}{\partial \mathbf{u}'} + \frac{\partial \mathbf{H}}{\partial \mathbf{u}}\right)^{-1} \mathbf{H}\left(t_n, \mathbf{u}_n^v, \frac{\mathbf{u}_n^v - \mathbf{u}_{n-1}}{\Delta t_n}\right) \quad (28)$$

where v presents the index of the Newton iteration. Special attention should be devoted to consistent initial conditions. This algorithm usually enables only one factorization of the finite difference Jacobian per time step, but the procedure loses the quadratic convergence properties. Despite that, numerical simulations show that this approach is relatively cheap and efficient. The Jacobian is presented by a very sparse matrix. For 1-D problems this matrix can be arranged to a banded form, but for 2-D problems it is not possible. Hence we use a sparse IMSL routine LFTXG which is implemented in the routine DASPCK. For larger problems, routine GMRES should be used, but for efficient modeling we must find the appropriate preconditioner.

3.2.4. Extension of the algorithm for two-dimensional problems

In this section we present the extension of the AFCM for 2-D problems (related to wavelet methods [25,46,48] and the multi-resolution finite difference method by [40]). The AFCM has a very useful property which can be easily extended to a higher dimension. Generally, the algorithm keeps the identical structure and same characteristics for

any dimension. However, some parts require more detailed implementation for higher dimensions.

2-D FCT represents an extension of the 1-D algorithm and retains all properties using two-dimensional *Fup* basis functions which can be obtained from the Cartesian product of two one-dimensional *Fup* basis functions defined for each direction. The only important difference can be seen at the boundary where boundary partial derivatives are satisfied (Fig. 7). This boundary implementation is analogous to the 1-D algorithm (Fig. 3), except for four edge points where we need more equations, generally $((n + 2)/2)^2 - 1$. For every external basis function the order of partial derivative is defined by the absolute difference between the vertex index and the corresponding boundary collocation point in each direction.

The multi-resolution 2-D FCT of the function $u(x,y)$ can be presented in the following way:

$$u^j(x,y) = \sum_{j=0}^J \sum_{k,l \in Z^j} d_{k,l}^j \phi_{k,l}^j(x,y) \tag{29}$$

where j shows levels from zero to a maximum level J needed for the approximation Eq. (29) with a desired accuracy defined by a prescribed threshold ε , Z^j is the irregular grid which contains only the significant collocation points and *Fup* basis functions, n is the *Fup* order, $d_{k,l}^j$ are *Fup* coefficients, $\phi_{k,l}^j$ are *Fup* basis functions while k presents the index of collocation points at the current level for x -direction while l presents the index of collocation points at the current level for y -direction. The zero level is defined

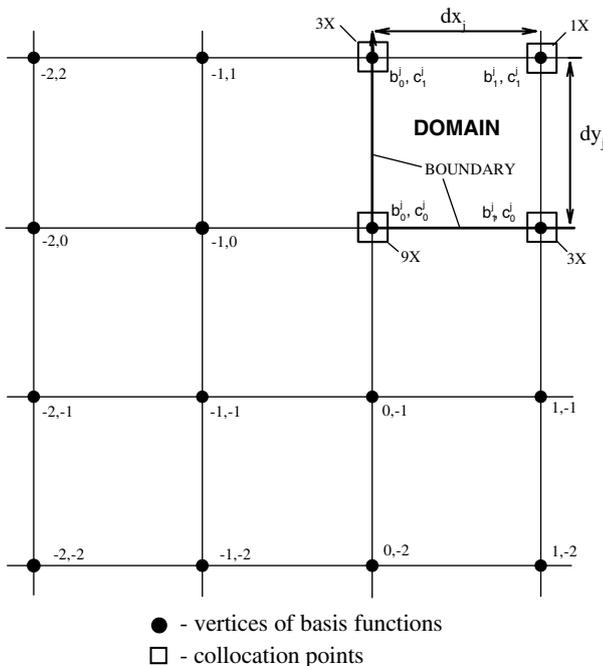


Fig. 7. Vertices location of internal and external $Fup_4(x,y)$ basis functions and the corresponding collocation points.

by a chosen resolution level $j_{\min,x}$ for the x -direction and by $j_{\min,y}$ for the y -direction. Generally, the banded linear system of equations can be obtained at each level

$$\sum_{k,l \in Z^j} d_{k,l}^j \phi_{k,l}^j(x_p^j, y_q^j) = \Delta_j(x_p^j, y_q^j) : 0 \leq p \leq 2^{j_{\min,x}+j}, 0 \leq q \leq 2^{j_{\min,y}+j} \tag{30}$$

$$\sum_{k,l \in Z^j} d_{k,l}^j \phi_{k,l}^{j(m_x, m_y)}(x_p^j, y_q^j) = \Delta_j^{(m_x, m_y)}(x_p^j, y_q^j) : (p = 0 \text{ or } p = 2^{j_{\min,x}+j} \text{ or } q = 0 \text{ or } q = 2^{j_{\min,y}+j}) \text{ and } (m_x > 0 \text{ or } m_y > 0) \tag{31}$$

where m_x and m_y are orders of the derivative in x - and y -directions, respectively. System (30) and (31) present conditions for satisfying function values within the domain and partial derivatives in the boundary points. The residual vector has the following form:

$$\Delta_j(x_p^j, y_q^j) = \begin{cases} f(x_p^j, y_q^j), & p, q \in Z^j; j = 0 \\ f(x_p^j, y_q^j) - u^{j-1}(x_p^j, y_q^j), & p, q \in Z^j; j > 0 \end{cases}$$

$$\Delta_j^{(m_x, m_y)}(x_p^j, y_q^j) = \begin{cases} f^{(m_x, m_y)}(x_p^j, y_q^j), & p, q \in Z^j; j = 0 \\ f^{(m_x, m_y)}(x_p^j, y_q^j) - u^{j-1(m_x, m_y)}(x_p^j, y_q^j), & p, q \in Z^j; j > 0 \end{cases} \tag{32}$$

The spatial adaptive strategy, the definition of basic and additional points, and the total and effective grid are the same as for the 1-D algorithm, except that all parameters have indexes related to two directions. The calculation of spatial derivatives remains the same if derivatives are defined in only one direction. For mixed partial derivatives we use a similar algorithm as in [47]. For each collocation point we define a corresponding level that belongs to the adjacent zone around that point where there are no points from higher levels and we use a regular stencil with $(n + 1)^2$ closest points. Since we use the common MOL approach, the time integration is independent of the spatial dimension (DASPK [4]).

4. Results and discussion

In this section, we present three examples for illustrating the efficiency and robustness of the *Fup* adaptive collocation method for modeling groundwater flow and transport problems. The first example is the Buckley–Leverett problem [16,33] with a moving saturation front, i.e. a strong nonlinear problem of a multiphase flow. The next two examples illustrate the strength of the method in modeling the density driven flow and transport problems, i.e. the 1-D vertical density driven flow in a sand column [50] and a well-known 2-D benchmark test case – the Henry problem [24]. In addition, Appendix B shows the additional example (Bourges equation) in order to rigorously show convergence properties of the AFCM.

4.1. The Buckley–Leverett problem

The Buckley–Leverett problem arises from the analysis of the multiphase flow which is particularly important in the oil industry. In this example we consider a two-phase flow of water and oil. Oil production is a very complex process in which numerical modeling plays a key role. The usual stimulation method consists of water being injected into wells which pushes oil under pressure to the production well. Since water and oil are two immiscible fluids, the solution is provided in terms of water saturation.

The Buckley–Leverett problem describes one of the simplest multiphase flow examples. The complexity of these processes lies in the complicated interaction between two fluids at the microscopic scale which is caused by gradient pressure fluctuations. The problem is described by the following non-dimensional saturation equation:

$$\frac{\partial s^*}{\partial t} + \frac{\partial F(s^*)}{\partial x} = v \frac{\partial}{\partial x} \left(G(s^*) \frac{\partial s^*}{\partial x} \right) \quad (33)$$

and the related functions are defined by

$$F(s^*) = \frac{s^{*2}}{s^{*2} + (1 - s^*)^2} \quad (34)$$

$$G(s^*) = 4s^*(1 - s^*) \quad (35)$$

and domain, initial and boundary conditions are:

$$v = 0.001; \quad x \in [0, 1] \quad (36)$$

$$s^*(x, 0) = \begin{cases} 1 - 3x, & 0 \leq x \leq 1/3 \\ 0, & \text{elsewhere} \end{cases} \quad (37)$$

$$s^*(0, t) = 1; \quad \frac{\partial s^*(1, t)}{\partial x} = 0 \quad (38)$$

where s^* is the non-dimensional water saturation, t and x are non-dimensional space and time coordinates, while $F(s^*)$ is a function which shows the flow ratio between two phases.

The problem is characterized by strong nonlinearities and a narrow saturation front. Moreover, the DAE system (33)–(38) is very stiff and requires short time steps. There is no analytic solution, but the problem has been solved numerically by Kurganov and Tadmor [33] and Cruz et al. [16] using a high resolution finite difference scheme. The presented AFCM uses the following parameters: $N_L^- N_R^- = 2$, $N_L^U = N_R^U = 2$, $M = 1$, $n = 2$, $\varepsilon = 10^{-3}$, $j_{\min} = 4$.

Fig. 8 shows a multi-level adaptive solution for the moving saturation front. Initial conditions are relatively simple except at one point where the saturation front has a discontinuous derivative. Therefore, six levels are needed for this simple bilinear function. At the early stages of the process the saturation front moves very slowly and at $t = 0.1$ reaches the final steepness and sharpness with eight levels and nearly 140 collocation points.

This example presents the efficiency of the method in handling strong nonlinear problems and the narrow saturation front with changes in sharpness and location both in

space and time. The adaptive grid follows the system dynamics and displays a wide range of different spatial and temporal scales that characterize this complex problem. The efficiency of the method is usually described by the compression coefficient (C_C) which is defined as a ratio between the number of collocation points in the non-adaptive and adaptive algorithm ($C_C = 2^{j_{\min} + j} / N_{AD}$). Namely, a non-adaptive algorithm would require all points at the maximum level in order to obtain the same accurate solution as the solution described by AFCM. The compression coefficient is about 30 which clearly show the efficiency of the proposed AFCM.

The method was also tested and verified (Fig. 8) by comparison, at $t = 0.2$, between solutions obtained by AFCM and those by Cruz et al. [15]. Fig. 8 shows an almost perfect match between the two solutions. Cruz et al. [15] used the adaptive multi-resolution finite difference algorithm which is essentially similar to the wavelet or *Fup* collocation methods, except that it provides a non-oscillating high-resolution FD scheme for advection terms, which is especially well suited for sharp fronts and hyperbolic problems. On the other hand, AFCM can handle advection dominated problems with very small, but non-zero dissipative or diffusive terms which stabilize the proposed numerical procedure. This comparison shows that AFCM is comparable to the mentioned high-resolution scheme. Also, the compression coefficient in recently developed adaptive discontinuous finite element methods [5,34] for similar problems (Richard equation) is between 1.5 and 5 which shows the efficiency of the AFCM. Finally, the AFCM controls classic numerical oscillations and an artificial numerical dispersion to a desirable level defined by the prescribed spatial threshold (oscillations are ten times smaller than the threshold).

4.2. 1-D vertical density driven flow and transport

Density driven flow and transport play a key role in many water resources problems, i.e. seawater intrusion in coastal aquifers, subsurface contaminant transport, radioactive waste disposal in salt domes, transport of geothermal energy, interaction between surface and subsurface flow, etc. others. Detailed reviews of density driven flow and transport can be found in [17,27,43]. Density variations are a fundamental property caused by pressure, concentration and/or temperature variability. Apart from the usual fluxes caused by a hydraulic gradient (advection or forced convection) and the density gradient (molecular diffusion), the density driven flow is characterized by a mechanical dispersion (velocity variations mainly caused by heterogeneity of porous media) and free convection (reverse stratification of fluids under certain conditions causes the development of fingers). The last two mixing mechanisms are very important because they reduce time scales and increase the transition zone [43]. Numerical modeling of the advection dominated problems with a small transition zone requires special treatment due to

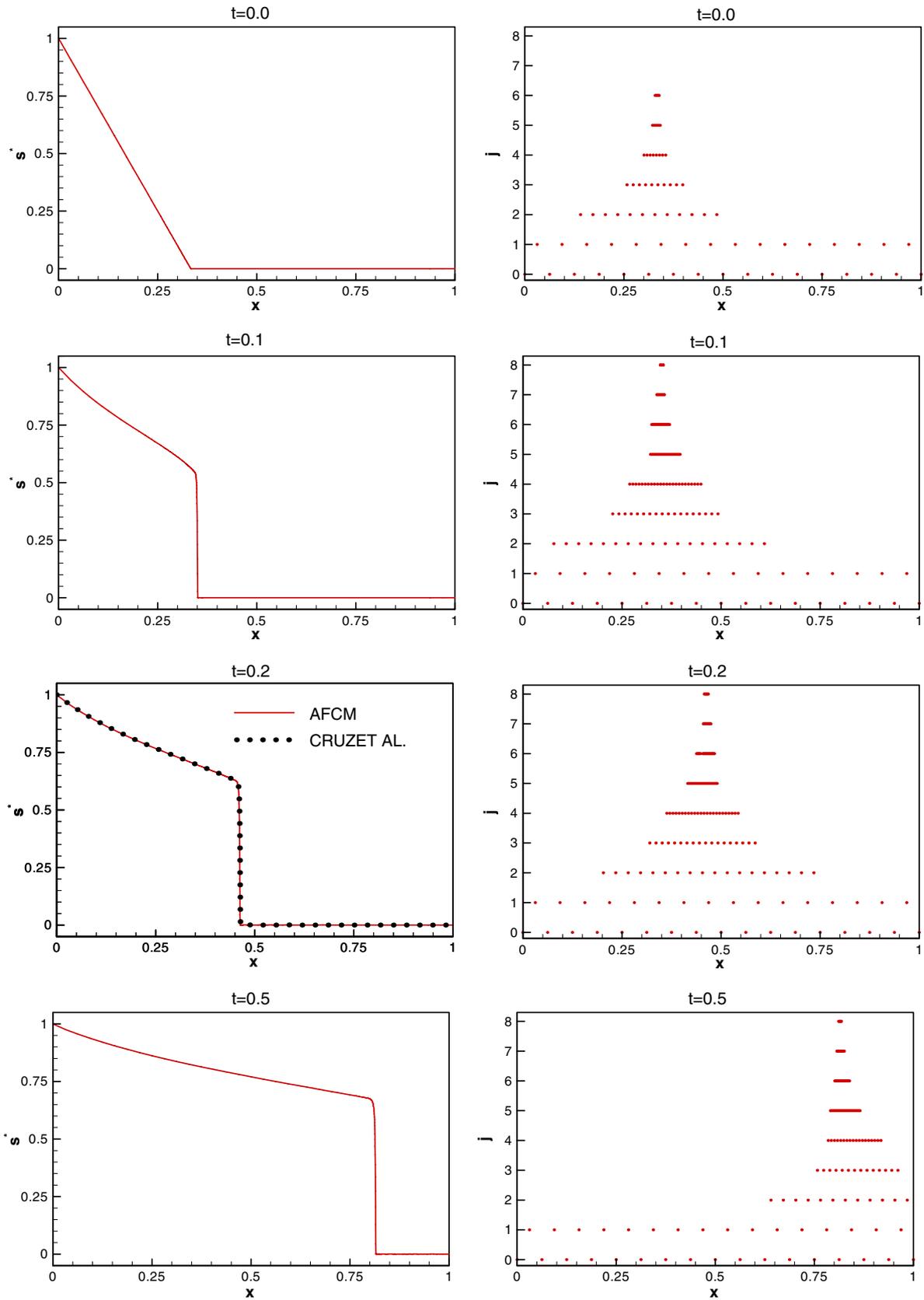


Fig. 8. Saturation front (left) and adaptive grid (right) for the Buckley–Leverett problem at four selected time steps. Comparison between AFCM and Cruz et al. [15] at $t = 0.2$.

strong nonlinearity, occurrence of numerical instability (over/under-shooting) and artificial numerical dispersion.

A full set of coupled systems of equations of density driven flow and transport is given by

$$\rho S_0^p \frac{\partial p}{\partial t} + n \beta_c^0 \frac{\partial C}{\partial t} + \nabla \cdot (\rho \mathbf{q}) = Q_P + Q_R \quad (39)$$

$$n \rho \frac{\partial C}{\partial t} + \rho \mathbf{q} \cdot \nabla C - \nabla \cdot (\rho n \mathbf{D}_H \cdot \nabla C) = (C^* - C) Q_R \quad (40)$$

where C is the concentration (M/M) of the fluid, C_s is the salt concentration (M/M), C^* is the concentration (M/M) of the injected fluid, n is porosity ($-$), ρ is density (M/L^3), p is pressure (M/T^2L), β_c^0 is the salt coefficient, \mathbf{q} is a specific discharge or Darcy velocity (L/T), Q_P is the pumping mass rate [M/L^3T], Q_R is the recharge mass rate [M/L^3T], S_0^p is the specific pressure storativity [T^2L/M] and D_H [L^2/T] is hydrodynamic dispersion tensor.

Density and specific discharge are given by a state equation and Darcy law, respectively

$$\rho = \rho_0 + \beta_c^0 (C - C_0) \quad (41)$$

$$\mathbf{q} = -\frac{\mathbf{k}}{\mu} \cdot (\nabla p - \rho \mathbf{g}) \quad (42)$$

where ρ_0 is fresh water density (M/L^3), C_0 is fresh water concentration (M/M), \mathbf{k} is the permeability tensor (L/T), μ is dynamic viscosity (M/TL), and \mathbf{g} is the gravity vector (L/T^2). A detailed discussion of the mathematical model can be found in [6,45].

In this example we consider the density driven flow in the vertical direction. Input data are given in Table 1. The results are given in a non-dimensional form

$$C^* = \frac{C}{C_s}, p^* = \frac{p}{p_0}, t^* = \frac{t}{t_0} \Rightarrow t_0 = \frac{\mu L^2}{k_0 p_0} = 10^4 (s)$$

where $p_0 = 10^5$ (Pa) is the referent pressure.

Initial and boundary conditions are given in a non-dimensional form

$$C^*(x, 0) = 0; \quad p^*(x, 0) = 1.7 - 0.7x \quad (43)$$

$$C^*(0, t) = 1 \Rightarrow t^* \leq 0.15 \quad ; \quad \frac{\partial C^*(L, t)}{\partial x} = 0 \quad (44)$$

$$C^*(0, t) = 0 \Rightarrow 0.15 < t^* \leq 0.5; \quad (45)$$

Table 1
Input data for 1-D vertical density driven flow and transport problem

| Variable | Value |
|--------------------------------|---|
| Domain length | 1 (m) |
| Salt concentration | 0.26 |
| Permeability | 10^{-12} (m^2) |
| Viscosity | 10^{-3} ($kg\ m^{-1}\ s^{-1}$) |
| Porosity | 0.2 |
| Salt coefficient | 769.23 |
| Specific pressure storativity | 2×10^{-11} ($m\ s^2\ kg^{-1}$) |
| Fresh water density | 1000 ($kg\ m^{-3}$) |
| Fresh water concentration | 0.0 |
| Longitudinal dispersion length | 0.001 (0.0001 m) |

Initial conditions are very demanding and show that the initial domain is filled by fresh water with a hydrostatic distribution. The pulse concentration boundary condition on the left side consists of the salt water source which enters the domain during the first part of the simulation. After that, the concentration is zero (fresh water). The concentration boundary condition on the right side does not present a dispersion flux boundary. Both boundary pressure conditions are constant Dirichlet boundary conditions. For this problem there is no analytic solution, but the numerical solution has been provided by Zegeling et al. [50] and Huang et al. [28] who used the r-adaptive finite difference method with a fixed number of points.

The adaptive algorithm of AFCM uses the following parameters: $N_L = N_R = 10$, $N_L^U = N_R^U = 6$, $M = 1$, $n = 4$, $\varepsilon_C = 10^{-2} - 10^{-5}$, $\varepsilon_p = 10^{-1}$, $j_{min} = 3$ where ε_C and ε_p are thresholds for concentration and pressure, respectively. The longitudinal dispersion is $\alpha_L = 0.001$ (m), while the molecular diffusion is neglected.

Fig. 9 shows the concentration field at four selected non-dimensional times by AFCM and Zegeling et al. [50]. The AFCM solution uses four different concentration thresholds to perform the convergence test in order to obtain the unknown exact solution, while solution [50] used 100 points which are adaptively located inside the domain. We show only the concentration field because the pressure distribution remains practically unchanged during that time. This simulation can be interpreted as an injection process of denser fluid to fresh water lenses. During the injection time ($t^* \leq 0.15$), the concentration front travels to the right side and the results are similar for these two methods and all thresholds (Fig. 9a).

After the injection time, a plume is created and travels by bulk flow (1-D presentation of the plume is a wave with two fronts, Fig. 9b). However, AFCM and solution [50] are not in close agreement due to a significant difference in the backward front. The reasons are twofold: (1) Algorithm [50] considers a fixed number of points (e.g. $N = 100$). After the injection time that algorithm redistributes points around two fronts and loses accuracy and (2) a description of the shock concentration boundary condition at the start and end of the injection time causes a very steep concentration front. The second reason is supported by Fig. 10 which shows a change in the number of collocation points during the simulation time for $\varepsilon_C = 10^{-4}$. Approximately 100 points are sufficient for the entire simulation time in order to obtain a very accurate solution except at the start and end of the injection time.

Moreover, Fig. 9 presents a convergence test with four AFCM solutions which used four different concentration thresholds. We show that only the AFCM solution with the highest threshold $\varepsilon_C = 10^{-2}$ is different than the other three solutions, but oscillations are less than the prescribed threshold. The other three solutions are very similar, while solutions with $\varepsilon_C = 10^{-4}$ and $\varepsilon_C = 10^{-5}$ are practically the same which confirms that the method converges into a unique solution (up and down triangles are identical).

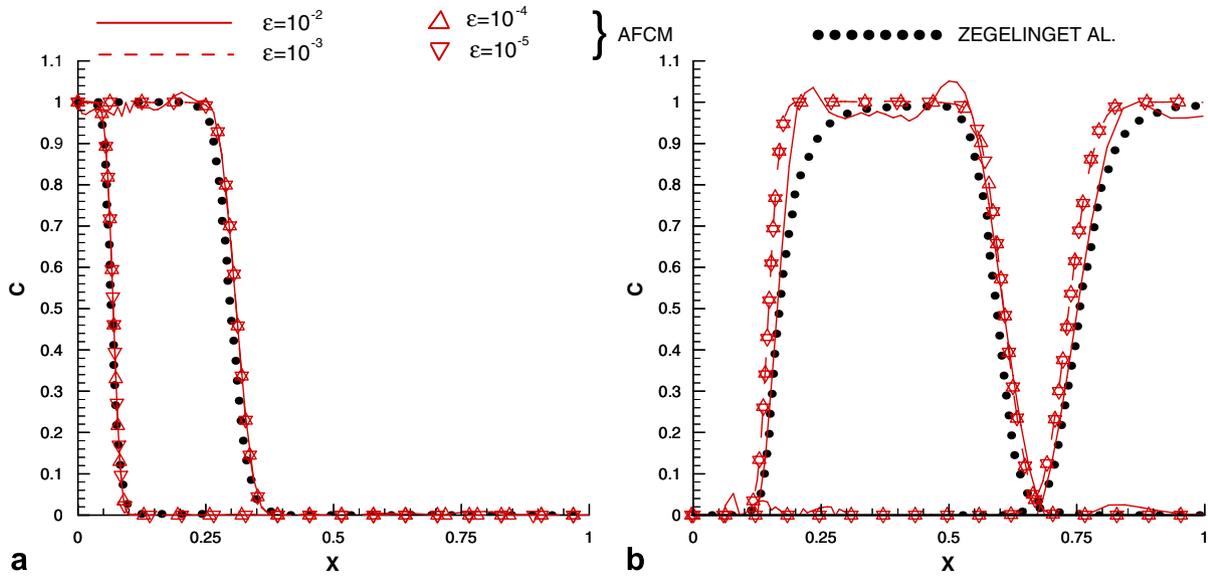


Fig. 9. Comparison between AFCM (four different thresholds) and Zegeling et al. [50] for 1-D vertical density driven flow and transport problem at (a) $t^* = 0.02$ and 0.1 and (b) $t^* = 0.2$ and 0.4 with higher dispersion ($\alpha_L = 0.001$ m) and the left pulse concentration boundary condition.

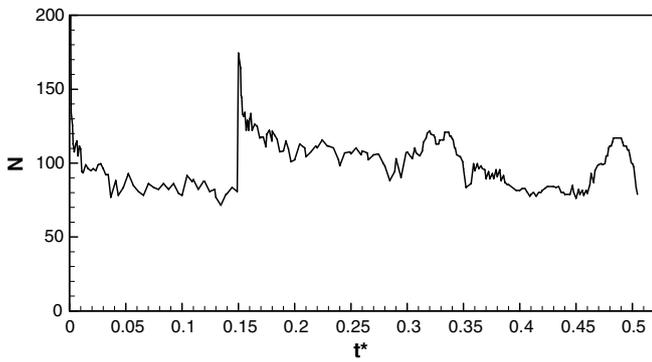


Fig. 10. Change in the number of collocation points in time in a case with higher dispersion ($\alpha_L = 0.001$ m), the left pulse concentration boundary condition and the threshold $\epsilon_C = 10^{-4}$.

Consequently, a smaller threshold includes more collocation points and resolution levels in order to obtain a more accurate solution. For four thresholds $\epsilon_C = 10^{-2}$ – 10^{-5} ,

AFCM, on the average, requires 50, 70, 100 and 140 points, respectively.

After the injection time, the backward front is steeper than the upward front due to the influence of the left concentration boundary condition. For threshold $\epsilon_C = 10^{-4}$, at $t^* = 0.16$ Fig. 11a shows that the grid follows two fronts while the backward front requires two additional resolution levels. After some time both fronts need the same resolution ($t^* = 0.30$) as shown in Fig. 11b. During the simulation time the compression coefficient fluctuates between 20 and 40.

Analyses of the density driven flow are usually closely related to the grid Peclet and Courant numbers and if they are satisfied a consistent and stable numerical simulation is possible [26]. The Peclet number should be satisfied within a transition zone and in its close neighborhood in order to avoid the occurrence of numerical instabilities and artificial dispersion. The AFCM requires only five resolution levels,

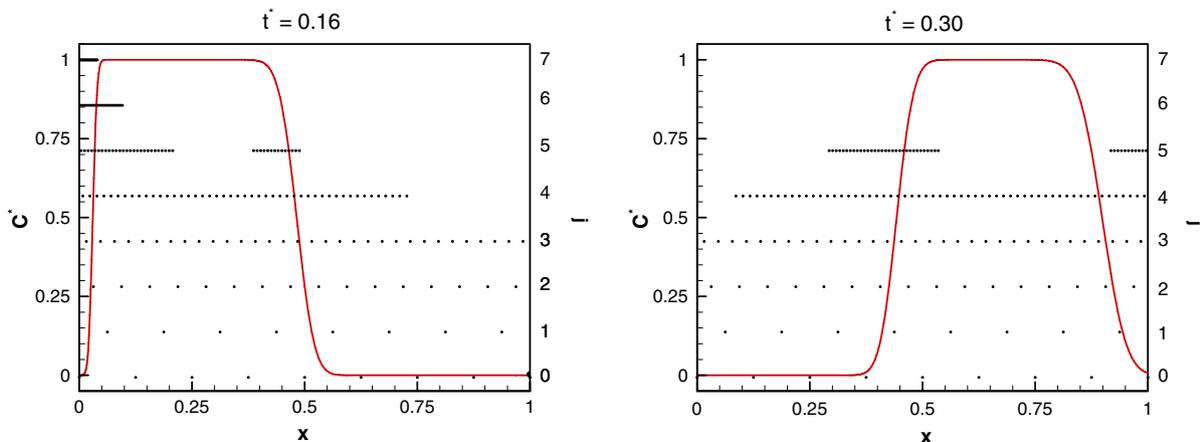


Fig. 11. Concentration field and adaptive grid for 1-D vertical density driven flow and transport problem at $t^* = 0.16$ and 0.30 with higher dispersion ($\alpha_L = 0.001$ m), the left pulse concentration boundary condition and the threshold $\epsilon_C = 10^{-4}$.

while the grid Peclet number is approximately four. However, numerical simulations with the AFCM show that the method is stable despite a higher Peclet number and that artificial dispersion is negligible with numerical oscillations below the prescribed spatial threshold. Thus, we have active control of the numerical error which is one of the key properties of the proposed adaptive *Fup* collocation method. The Courant number is always satisfied within the DASPK routine due to the stability and accuracy of this time integrator [4].

The same problem is simulated by setting fixed concentration boundary conditions on the left side with longitudinal dispersion reduced ten times ($\varepsilon_C = 10^{-4}$). In this case the transition zone is strongly decreased with a highly non-linear concentration gradient creating an even more numerically challenging problem. The presented AFCM is capable of solving this problem with a relatively high Peclet number ($Pe \sim 20$). The solution (Fig. 12) is stable and accurate (oscillations are 100 times less than the prescribed threshold $\varepsilon_C = 10^{-4}$). The transition zone is very narrow and it requires from six to eight levels and ca 175 collocation points so that the compression coefficient is greater than 40.

4.3. The Henry problem

The Henry problem is a 2-D standard benchmark test case for variable density flow and transport [27]. This test case presents seawater intrusion into a coastal confined aquifer and has become one of the common benchmark problems. It is characterized by a relatively wide transition zone (except at the seaward boundary), a significant constant molecular diffusion and unrealistic constant Dirichlet seaward concentration boundary conditions. The Henry solution [24] is called the Henry semi-analytic solution although it is a pure numerical solution. Segol [42] revised the Henry solution and discovered some errors in his original work. In the last three decades many scientists have attempted to solve the Henry problem with varying success. Some of the most important solutions were obtained

by Segol et al. [41], Frind [18] and Voss and Souza [45]. The main problems in these classical finite element solutions are velocity and flux discontinuity between elements. A very accurate and stationary finite difference solution was presented by Croucher and O’Sullivan [14] with a very rigorous convergence study. Recently, new advanced finite element solutions have been achieved by a discontinuous Galerkin formulation by Kolditz et al. [30], Ackerer et al. [1], and Bues and Oltean, [11], which enable flux continuity. The non-adaptive *Fup* collocation method for solving the Henry problem was presented by Gotovac et al. [22]. The applied collocation algorithm has solved the consistent velocity approximation and flux continuity (essentially for density driven flow problems) due to localized and infinitely derivable *Fup* basis functions. The method did not require typical space discretization and numerical integration, while the basic solution (pressure, concentration) and all their derivatives (velocity, other fluxes) were obtained with the same level of accuracy (according to polynomial approximation). The adaptive procedure of AFCM additionally improves numerical modeling of such processes due to resolved spatial and temporal scales using an optimal grid. In this example we have showed the new adaptive non-steady Henry solution and the application of this method to the 2-D density driven problems. For the best of our knowledge, this is the first adaptive Henry solution.

The definition of the Henry problem is presented in Fig. 13 and Table 2. Voss and Souza [45] formulated equivalent effective parameters (Table 2). The AFCM adaptive solution was obtained with the following parameters: $N_{LX} = N_{RX} = 1$, $N_{LY} = N_{RY} = 1$, $N_{LX}^U = N_{RX}^U = N_{LY}^U = N_{RY}^U = 0$, $M = 1$, $j_{\min x} = 3$, $j_{\min y} = 2$, $J = 8$, $n = 2$, $\varepsilon_C = 0.0001$, $\varepsilon_p = 1.0$.

4.3.1. Henry original boundary condition

The original Henry problem was defined by a relatively large molecular diffusion $D^* = 18.8571 \times 10^{-6}$ (m²/s) and a constant seaside boundary condition $C = C_s$. Moreover, Henry [24] used constant dispersion (a more realistic

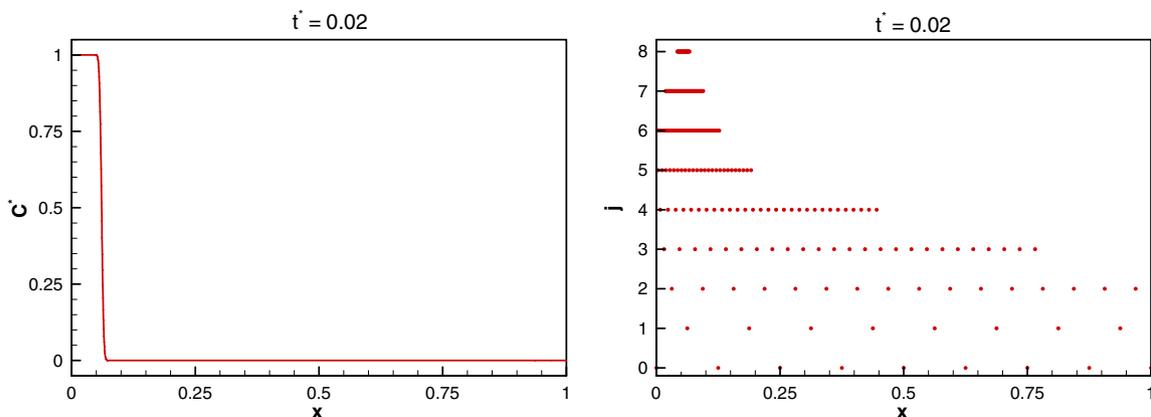


Fig. 12. Concentration field (left) and adaptive grid (right) for 1-D vertical density driven flow and transport problem at $t^* = 0.02$ with reduced dispersion ($\alpha_L = 0.0001$ m), the left constant concentration boundary condition and the threshold $\varepsilon_C = 10^{-4}$.

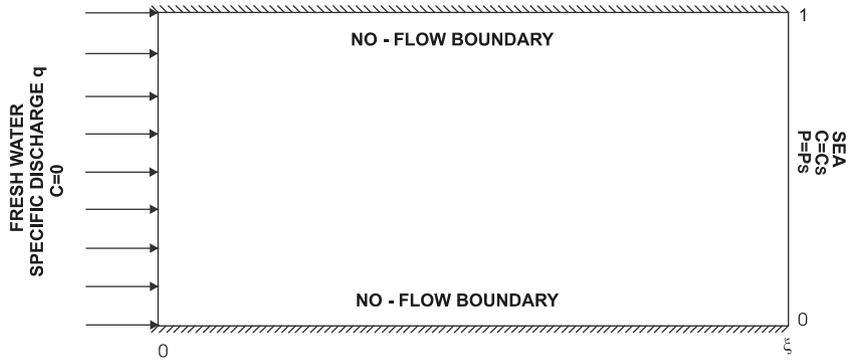


Fig. 13. Illustration of the Henry problem, considered domain and boundary conditions.

Table 2
Other input data for Henry problem

| Variable | Value |
|--------------------------------|--|
| Domain length | 2×1 (m) |
| Salt concentration | 0.03571 |
| Permeability | 10^{-9} (m ²) |
| Viscosity | 10^{-3} (kg m ⁻¹ s ⁻¹) |
| Porosity | 0.35 |
| Salt coefficient | 700 |
| Left inflow specific discharge | 6.6×10^{-5} (m/s) |
| Fresh water density | 1000 (kg m ⁻³) |
| Fresh water concentration | 0.0 |
| Molecular diffusion | 18.8571×10^{-6} (6.6×10^{-6} m ² /s) |

approach is defined with a velocity dependent dispersion tensor) due to a dimensionless formulation used in the Fourier–Galerkin approach. The seaside fixed boundary condition is not realistic because the mixed outflow water cannot have sea water concentration. This inconsistent physical formulation implies a high concentration gradient and a narrow transition zone near the outflow segment of the seaside boundary. A comparison between the AFCM solution and the revised Henry solution by Segol [42] is in very close agreement (Fig. 14).

The following simulation presents a modified Henry problem with a reduced dispersion ($D^* = 6.6 \times 10^{-6}$ m²/s). Fig. 15 shows the development of an adaptive solution in

time. Initially, the domain is filled with fresh water. At the beginning of the process, due to a high salt influx, a very narrow transition zone is created, especially in the outflow zone (large concentration gradient). At $t = 200$ (s) there is a great difference between the lower and upper parts of the transition zone. The upper part requires a maximum of eight resolution levels, while the lower part requires only four levels. Furthermore, the saltwater wedge intrudes inland through the bottom of the aquifer and increases the lower part of the transition zone. In the upper part, i.e. in the outflow zone, there are eight maximum resolution levels during the entire simulation due to a high concentration gradient and outflow velocity (relatively small outflow segment). During the first 1000 (s) the saltwater wedge intrudes relatively quickly, while after 3000 (s) its movement is significantly reduced. In the outflow zone, the maximum velocity is 2.18×10^{-3} (m/s). Generally, the adaptive grid requires more points and levels in the outflow zones, the narrow transition zones and around the stagnation point of the saltwater wedge (i.e. $t = 3600$ s).

After 6000 (s) the solution changes very slowly and after 12,000 (s) a dynamic equilibrium between the flows of fresh water and salt water are reached. The efficiency of AFCM is illustrated by the compression coefficient staying around 500. After 6000 (s) the number of collocation points is ca 4000. This means that the non-adaptive algorithm for the same accuracy within the entire domain requires ca.

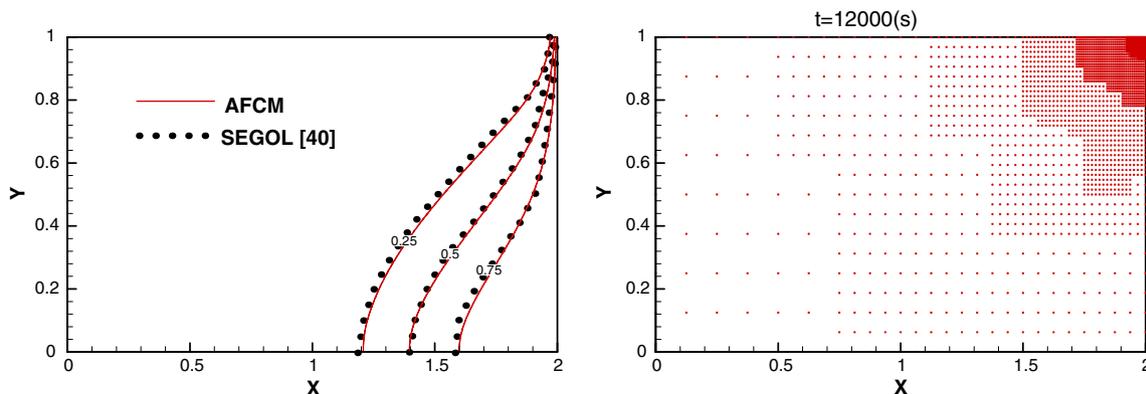


Fig. 14. Comparison between solutions by AFCM and Segol [40] and adaptive grid for the original Henry problem.

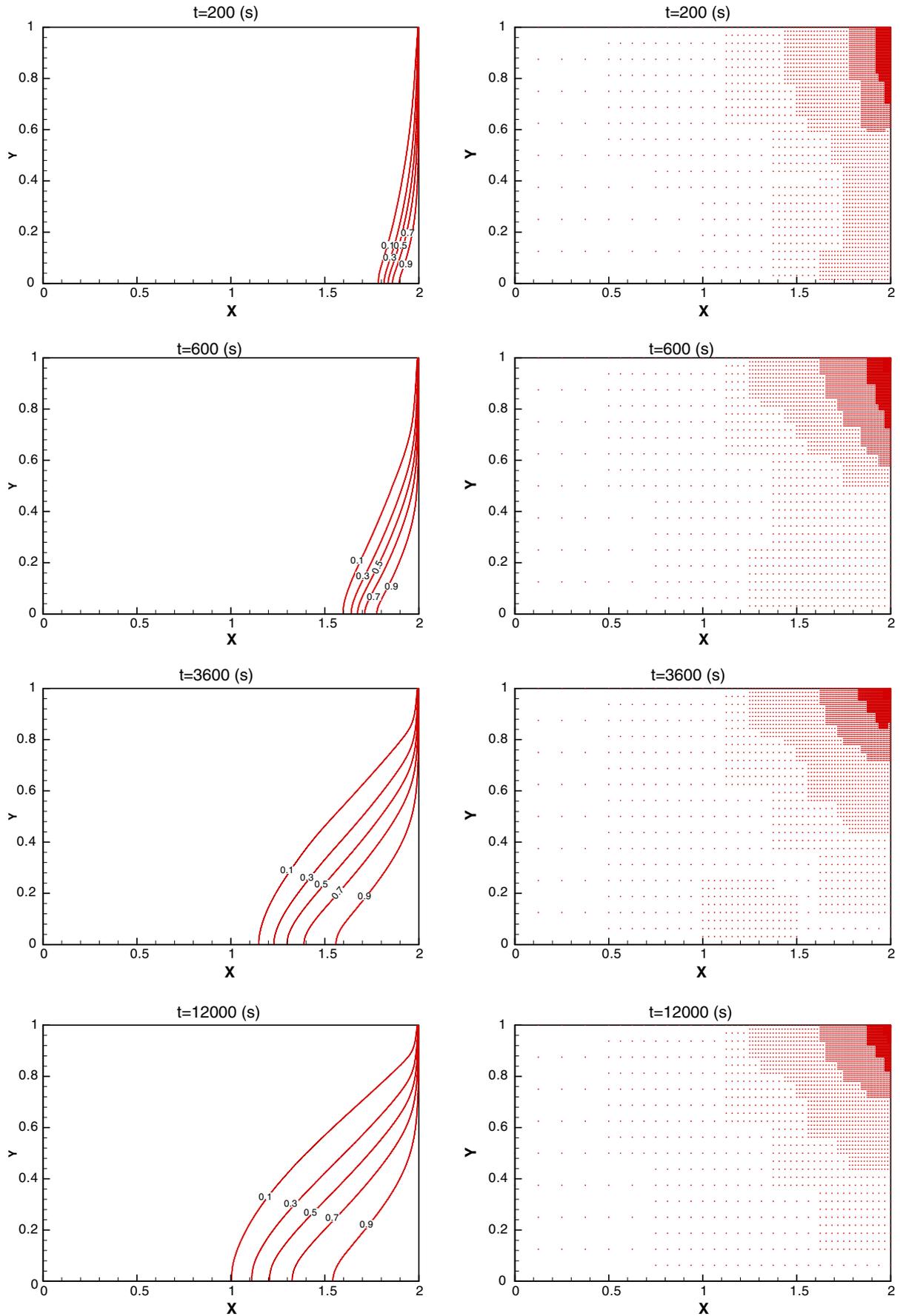


Fig. 15. Modified Henry problem with a reduced dispersion ($D^* = 6.6 \times 10^{-6} \text{ m}^2/\text{s}$): concentration field (left) and adaptive grid (right).

2,000,000 collocation points. Furthermore, after 6000 (s) the number of collocation points and resolution levels remains practically unchanged. The steady state solution is very similar, as expected, to the non-adaptive collocation solution [22].

4.3.2. Mixed boundary conditions

The Henry original and unrealistic boundary seaside condition should be replaced by a stronger physical condition. Voss and Souza [45] and Bear [6] discussed the seaside boundary condition formulation and its numerical implementation in more detail. In this paper the mixed boundary condition is implemented in the collocation procedure.

Segol et al. [41] defined the mixed boundary condition so that the upper 20% of the seaside boundary is zero dispersion flux boundary (Neumann type), while elsewhere they kept the original Dirichlet boundary condition. Comparison between the AFCM and Frind [18] solutions demonstrated their close agreement (Fig. 16), especially near the boundary where solutions are comparable. A final AFCM solution needs less than 3000 collocation points resulting in a very high compression coefficient of about 800. Moreover, the seaside boundary formulation has a main disadvantage due to the fixed outflow boundary that creates

an unrealistic high concentration gradient. It is necessary to formulate the auto-consistent boundary condition where the numerical algorithm can automatically find the dividing point. In the finite element this formulation was presented by Voss and Souza [45] and Galeati et al. [19].

The auto-consistent boundary condition, with respect to the boundary flux orientation, has been implemented into the time integration routine DASPK [4]. For those points where the flux is oriented inland (inflow segment), AFCM introduces the Cauchy boundary condition (flux continuity through the boundary); otherwise it uses the Neumann boundary condition (outflow segment) as before.

Fig. 17 shows a comparison between AFCM and SUTRA [44] solutions for the auto-consistent boundary condition and reduced dispersion. The solutions near the seaward boundary are in close agreement which show successful implementation of this type of boundary condition to the AFCM. Differences between solutions in the other parts of the domain are not comparable due to the coarse finite element mesh in SUTRA simulation with only 200 elements. In this case, the entire transition zone is relatively wide and there is no requirement for higher resolution levels and frequencies (Fig. 17). Only the central part of the

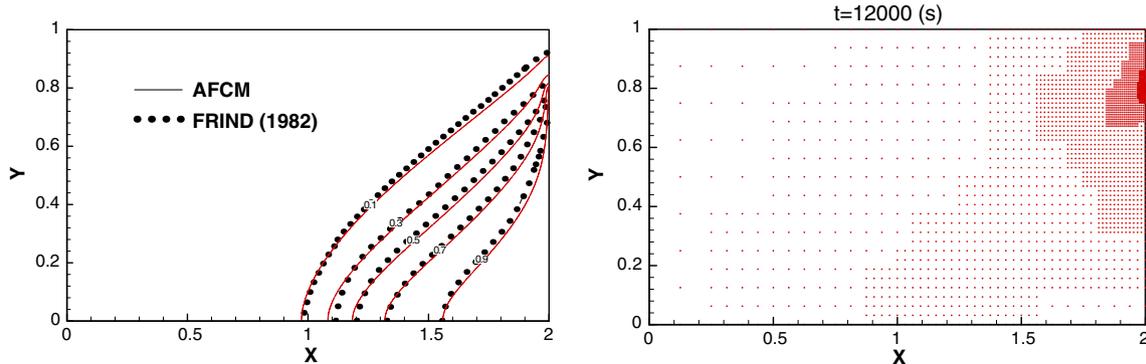


Fig. 16. Comparison between solutions by AFCM and Frind [17] and adaptive grid for the modified Henry problem with a mixed seaside boundary condition and reduced dispersion ($D^* = 6.6 \times 10^{-6} \text{ m}^2/\text{s}$).

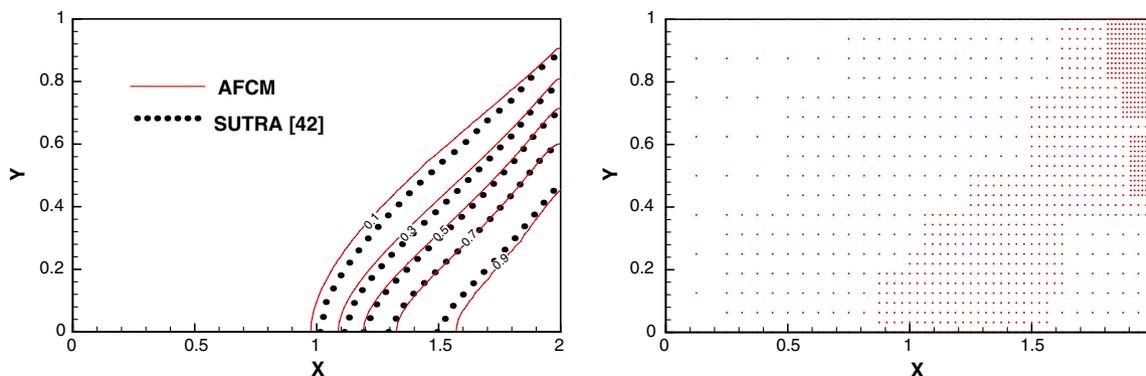


Fig. 17. Comparison between solutions by AFCM and Sutra [42] and adaptive grid for the modified Henry problem with an auto-consistent seaside boundary condition and reduced dispersion ($D^* = 6.6 \times 10^{-6} \text{ m}^2/\text{s}$).

transition zone needs three levels; the outflow boundary requires four levels, while the other parts of the domain are very smooth. This solution required only 1000 collocation points.

5. Conclusions

In this paper we presented a multi-resolution adaptive *Fup* approach applicable to groundwater flow and transport problems with sharp gradients, fronts and narrow transition zones. AFCM enables adaptive multi-resolution evolution of the system dynamics resolving different spatial and temporal scales with a desired level of accuracy using the entire family of *Fup* basis functions. AFCM has been tested and verified on a few groundwater flow and transport problems, three 1-D problems with sharp fronts (the Buckley–Leverett problem, the vertical density driven flow problem and an additional Bourges example) and the standard 2-D density driven benchmark test case—the Henry problem. The results demonstrate that AFCM is robust and efficient having significant advantages over conventional numerical methods. The key characteristics of AFCM can be summarized as follows:

- (a) Accurate solution with a global numerical error closely related to the prescribed spatial threshold which implies active control of numerical oscillations and dispersion.
- (b) Resolved wide range of spatial and temporal scales (particularly for complex nonlinear groundwater problems) on a nearly optimal adaptive grid.
- (c) Reduced computational efforts due to space and time adaptation.
- (d) Classic grid discretization and numerical integration are avoided.
- (e) The solution (i.e. concentration, pressure, etc.) and all its derivatives (i.e. velocity and other fluxes) are obtained with the same level of accuracy due to properties of the localized *Fup* basis functions which imply a consistent and continuous velocity and flux approximation. This is particularly important for complex groundwater flow and transport problems.

The presented multi-resolution adaptive approach implemented in AFCM provides a different modeling opportunity in many areas of complex groundwater flow and transport problems. The current areas of research include the extension of AFCM to a simultaneous space–time *Fup* approach with a complex domain. Furthermore, we intend to explore the possibility of applying AFCM to groundwater flow and transport problems in heterogeneous porous media characterized by a high permeability variance.

Acknowledgements

This research was supported and funded by Croatian Ministry of Science under project No. 4-0083-121; <http://www.gradst.hr/cer>.

The authors are grateful to Reviewers for their constructive and detailed comments which resulted in the improved version of the paper.

Appendix A

For each collocation point in the irregular adaptive grid, n closest collocation points are selected and the spatial derivatives are calculated by using the Taylor series expansion (for instance $n = 4$):

$$\begin{bmatrix} (x_1 - x_0) & \frac{(x_1 - x_0)^2}{2!} & \frac{(x_1 - x_0)^3}{3!} & \frac{(x_1 - x_0)^4}{4!} \\ (x_2 - x_0) & \frac{(x_2 - x_0)^2}{2!} & \frac{(x_2 - x_0)^3}{3!} & \frac{(x_2 - x_0)^4}{4!} \\ (x_3 - x_0) & \frac{(x_3 - x_0)^2}{2!} & \frac{(x_3 - x_0)^3}{3!} & \frac{(x_3 - x_0)^4}{4!} \\ (x_4 - x_0) & \frac{(x_4 - x_0)^2}{2!} & \frac{(x_4 - x_0)^3}{3!} & \frac{(x_4 - x_0)^4}{4!} \end{bmatrix} \cdot \begin{bmatrix} u_0^{(1)} \\ u_0^{(2)} \\ u_0^{(3)} \\ u_0^{(4)} \end{bmatrix} = \begin{bmatrix} u_1 - u_0 \\ u_2 - u_0 \\ u_3 - u_0 \\ u_4 - u_0 \end{bmatrix} \tag{A.1}$$

Instead of calculating the above system for each collocation point, it is possible to find a general expression in the form of function values:

$$u_0^{(i)} = \sum_{j=0}^n \alpha_j^{(i)} u_j \tag{A.2}$$

where $\alpha_j^{(i)}$ are weight coefficients which depend on the distance between collocation points. For the first two derivatives, which are sufficient for most groundwater flow and transport problems, we get these coefficients in a closed form by using $h_i = x_i - x_0$

$$\alpha_0^{(i)} = - \sum_{j=1}^n \alpha_j^{(i)} \tag{A.3}$$

$$\alpha_j^{(1)} = (-1)^n \frac{\prod_{k=1, k \neq j}^n h_k}{\prod_{k=1, k \neq j}^n (h_j - h_k^*)}; \quad h_k^* = \begin{cases} h_k & \Rightarrow k \neq j \\ 0 & \Rightarrow k = j \end{cases}; \tag{A.4}$$

$j = 1, \dots, n$

$$\alpha_j^{(2)} = (-2)(-1)^n \frac{\sum_{l=1, l \neq j}^n \prod_{k=1, k \neq j, k \neq l}^n h_k}{\prod_{k=1, k \neq j}^n (h_j - h_k^*)}; \quad h_k^* = \begin{cases} h_k & \Rightarrow k \neq j \\ 0 & \Rightarrow k = j \end{cases}; \tag{A.5}$$

$j = 1, \dots, n$

It is possible to write system (25),(26) in the general form for the adaptive time step and for the adaptive non-uniform grid where spatial derivatives depend only on function values. Note that the calculation of the above weight coefficients is not needed for each collocation point because a lot of points belonging to the same resolution level also have the same finite difference stencil. It is unsatisfactory mostly in points where the FD stencil contains points from different resolution levels or in points located near the boundary.

Appendix B

The Bourges equation results from the application of the Navier–Stokes equations to unidirectional flow without pressure gradient and small viscosity. It is classical nonlinear benchmark due to the existence of the analytic solution included in order to show more general convergence properties of AFCM. In addition, difference between adaptive and non-adaptive approach and dependence of the threshold and absolute error are demonstrated. Bourges equation with initial and boundary conditions is

$$\frac{\partial u}{\partial t} = v \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} \tag{B.1}$$

$$u(x, 0) = -\sin(\pi x) \tag{B.2}$$

$$u(\pm 1, t) = 0 \tag{B.3}$$

where u is the dimensionless velocity, while space–time domain and viscosity are defined by

$$x \in [-1, 1]; \quad t \in [0, 1.5/\pi]; \quad v = 10^{-2}/\pi$$

Initial conditions are very simple and monotonic. Dirichlet boundary conditions are homogeneous. Analytic solution can be found in [47]:

$$u^*(x, t) = - \frac{\int_{-\infty}^{\infty} \sin(\pi(x-\eta)) \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \exp\left(-\frac{\eta^2}{4vt}\right) d\eta}{\int_{-\infty}^{\infty} \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \exp\left(-\frac{\eta^2}{4vt}\right) d\eta} \tag{B.4}$$

Adaptive algorithm uses the following parameters: $N_L = N_R = 1$, $N_L^U = N_R^U = 0$, $M = 1$, $j_{\min} = 4$, $\Delta t_{\text{adapt}} = 0.025$. Fig. B.1a shows time evolution of the analytic solution. It is characterized with one dimensional shock that is stationary in space, but changes in time. During initial stages, the solution is smooth, but after $t = 0.7/\pi$ the shock becomes strongly steeper and numerically demanding. After $t = 1.3/\pi$ shock reaches the maximum steepness.

Fig. B.1b–d shows convergence properties of AFCM in terms of L_{\max} norm which presents maximum absolute error between analytic and numerical solution in space and time. Fig. B.1b shows dependence of the total CPU time (P4 M processor, 2.2 GHz) and L_{\max} norm for both, adaptive and non-adaptive case and for different order of Fup basis functions. Generally, a significant amount of the computational work is reduced in the adaptive case. For the same level of accuracy, adaptive algorithm requires approximately from 100 to 2000 times less CPU time than non-adaptive algorithm. Fig. B.1b also demonstrates that

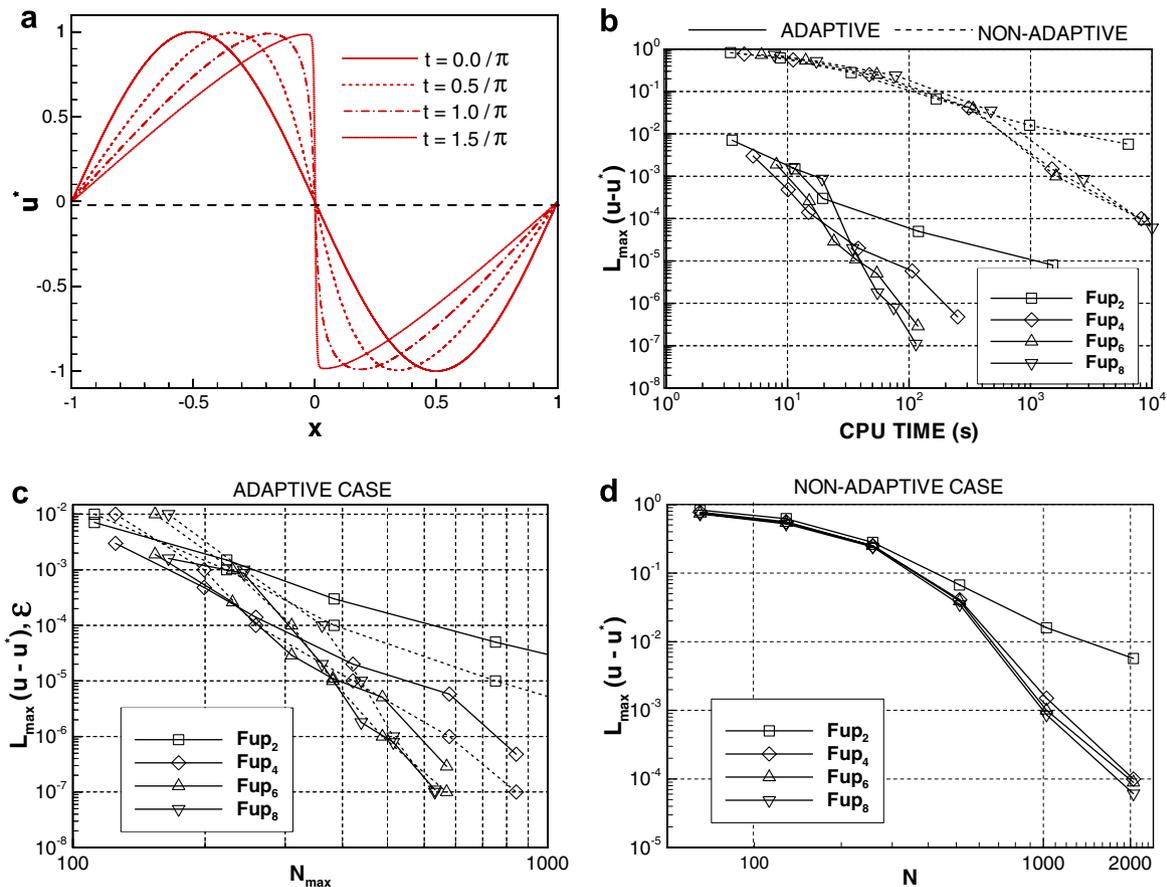


Fig. B.1. AFCM convergence properties using Bourges equation example: (a) analytic solution of Eq. (B.4), (b) total CPU time versus absolute error between numerical and analytic solution for adaptive (solid lines) and non-adaptive case (dashed lines), (c) maximum number of collocation points required versus absolute error (solid lines) and threshold (dashed lines) for adaptive case and (d) number of collocation points versus absolute error for non-adaptive case.

for higher absolute error (e.g. 10^{-3}) the choice of *Fup* order is not important. However, by decreasing the absolute error accuracy ($<10^{-3}$) the choice of *Fup* order becomes significant in terms of CPU time.

Fig. B.1c–d shows dependence between the number of collocation points and L_{\max} norm for adaptive and non-adaptive case and for different order of *Fup* basis functions. More collocation points and higher order of basis functions cause higher accuracy, especially for absolute error smaller than 10^{-3} . We can see similar behavior as before such that adaptive case requires from 6 to 12 times less collocation points than the non-adaptive case for the same level of accuracy (this present compression coefficient- C_R). Finally, Fig. B.1c also presents dependence of the absolute error and threshold. Smaller thresholds include more collocation points and higher accuracy. It is a very important to notice that threshold and absolute error always have the same order of magnitude for all *Fup* basis functions. This additional example clearly presents that threshold directly defines accuracy of AFCM. Moreover, threshold defines error indicator and a-priori adaptive criterion in AFCM as direct measure of the difference between numerical and exact solution, even when exact solution is not known. Notice that absolute error is strictly less than the prescribed threshold for higher *Fup* basis functions or higher chosen thresholds.

References

- [1] Ackerer P, Younes A, Mose R. Modeling variable density flow and solute transport in porous medium: 1. Numerical model and verification. *Transport Porous Media* 1999;35(3):345–73.
- [2] Alves MA, Cruz P, Mendes A, Magalhes FD, Pinho FT, Oliveira PJ. Adaptive multiresolution approach for solution of hyperbolic PDE's. *Comput Methods Appl Mech Eng* 2002;191:3909–28.
- [3] Andričević R, Cvetković V. Relative dispersion for solute flux in aquifers. *J Fluid Mech* 1998;361:145–74.
- [4] Ascher U, Petzold L. *Computer methods for ordinary differential equations and differential-algebraic equations*. Berkeley: Society for Industrial and Applied Mathematics; 1998. 297p.
- [5] Bause M, Knabner P. Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods. *Adv Water Resour* 2004;27:565–81.
- [6] Bear J. Conceptual and mathematical modeling. In: Bear J, Cheng AHD, Sorek S, Quazar D, Herrera I, editors. *Seawater intrusions in coastal aquifers*. Dordrecht: Kluwer Publishers; 1999. p. 127–61.
- [7] Bertoluzza S. Adaptive wavelet collocation method for the solution of Burgers equation. *Transport Theory Statist Phys* 1996;25:339–52.
- [8] Bertoluzza S, Naldi G. A wavelet collocation method for the numerical solution of PDEs. *Appl Comput Harmonic Anal* 1996;3:1–9.
- [9] Beylkin G, Keiser JM. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *J Comput Phys* 1997;132:233–59.
- [10] Bhattacharya A, Joseph B. Simulation of fixed – bad gas solid reactors using an adaptive spline collocation method. *Comp Chem Eng* 1998;12(4):351–2.
- [11] Bues MA, Oltean C. Numerical simulations for saltwater intrusions by the mixed hybrid finite element method and discontinuous finite element method. *Transport Porous Media* 2000;40(2): 171–200.
- [12] Chiaavassa G, Guichaoua M, Liandrat J. Two adaptive wavelet algorithms for nonlinear parabolic differential equations. *Comput Fluids* 2002;31:467–80.
- [13] Class H, Helmig R, Bastian P. Numerical simulation of non-isothermal multiphase multicomponent processes in porous media. An efficient solution technique. *Adv Water Resour* 2002;25:533–50.
- [14] Croucher AE, O'Sullivan MJ. The Henry problem for saltwater intrusion. *Water Resour Res* 1995;31(7):1809–14.
- [15] Cruz P, Mendes A, Magalhes FD. Using wavelets for solving PDEs: and adaptive collocation method. *Chem Eng Sci* 2001;56: 3305–9.
- [16] Cruz P, Alves MA, Magalhes FD, Mendes A. Solution of hyperbolic PDEs using a stable adaptive multiresolution method. *Chem Eng Sci* 2003;58:1777–92.
- [17] Diersch H-JG, Kolditz O. Variable density flow and transport in porous media: approaches and challenges. *Adv Water Resour* 2002;25:899–944.
- [18] Frind E. Simulation of long-term transient density-dependent transport in groundwater. *Adv Water Resour* 1982;5:73–88.
- [19] Galeati G, Gambolati G, Neumann SP. Coupled and partially coupled Eulerian–Lagrangian model of fresh-water–seawater mixing. *Water Resour Res* 1992;28:149–65.
- [20] Gotovac B, Kozulić V. On a selection of basis functions in numerical analyses of engineering problems. *Int J Eng Model* 1999;12(1–4):25–41.
- [21] Gotovac B, Kozulić V. Numerical solving the initial value problems by R_{bf} basis functions. *Struct Eng Mech* 2002;14(3):263–85.
- [22] Gotovac H, Andričević R, Gotovac B, Kozulić V, Vranješ M. An improved collocation method for solving the Henry problem. *J Contam Hydrol* 2003;64:129–49.
- [23] Helming R. *Multiphase flow and transport processes in the subsurface*. Berlin: Springer; 1997. p. 367.
- [24] Henry HR. Effects of dispersion on salt encroachment in coastal aquifers. In: *Sea water in coastal aquifers*. US Geol Surv Water Supply Pap 1613-C, 1964; p.70–84.
- [25] Hesthaven JS, Jameson LM. A wavelet optimized adaptive multi-domain method. *J Comput Phys* 1998;145:280–96.
- [26] Holmström M. Solving hyperbolic PDEs using interpolation wavelets. *J Sci Comput* 1999;21:405–20.
- [27] Holzbecher E. *Modeling density-driven flow in porous media*. Berlin: Springer; 1998. p. 198.
- [28] Huang W, Zheng L, Zhan X. Adaptive moving mesh methods for simulating one-dimensional groundwater problems with sharp moving fronts. *Int J Numer Methods Eng* 2002;54:1579–603.
- [29] Jameson LM. A Wavelet-optimized, very high order adaptive grid and order numerical method. *Siam J Sci Comput* 1998;19(8):1980–2013.
- [30] Kolditz O, Ratke R, Diersch HJG, Zielke W. Coupled groundwater flow and transport: 1. Verification of variable-density flow and transport models. *Adv Water Resour* 1998;21:27–46.
- [31] Kozulić V, Gotovac B. Numerical analysis of 2-D problems using *Fup* basis functions. *Int J Eng Mod* 2000;13(1–2):7–18.
- [32] Kravchenko VF, Basarab MA, Perez-Meana H. Spectral properties of atomic functions used in digital signal processing. *J Commun Technol Electron* 2001;46:494–511.
- [33] Kurganov A, Tadmor E. New high resolution central scheme for nonlinear conservation laws and convection–diffusion equations. *J Comput Phys* 2000;160:241–82.
- [34] Miller CT, Abhishek C, Farthing MN. A spatially and temporally adaptive solution of Richards' equation. *Adv Water Resour* 2006;29:525–45.
- [35] Prenter PM. *Splines and variational methods*. New York: Wiley; 1989. p. 323.
- [36] Ramachandran PA, Duduković MP. Solution by triple collocation for periodic operation of heat regenerators. *Comp Chem Eng* 1984;8(6):377–88.
- [37] Rvačev VL, Rvačev VA. Pro odnu finitnu funkciju. *DAN URSR Ser A (in Russian)* 1971;6:705–7.
- [38] Rvačev VL. *Teorija R-funkcij i nekotorija jeje prilozenija (in Russian)*. Naukova dumka Kiev; 1982. p. 551.

- [39] Queazada C, Clement TP, Lee KK. Generalized solution to multi-dimensional multi-species transport equations coupled with a first-order reaction network involving distinct retardation factors. *Adv Water Resour* 2004;27:507–20.
- [40] Santos JC, Cruz P, Alves MA, Oliviera PJ, Magalhes FD, Mendes A. Adaptive multiresolution approach for two dimensional PDEs. *Comp Methods Appl Mech Eng* 2004;193:405–25.
- [41] Segol G, Pinder GF, Gray WG. A Galerkin-finite element technique for calculating the transient position of the saltwater front. *Water Resour Res* 1975;11(2):343–7.
- [42] Segol G. *Classic groundwater simulations-proving and improving numerical models*. Englewood Cliffs, PTR: Prentice-Hall; 1994. p. 530.
- [43] Simmons CT, Fenstemaker TR, Sharp JM. Variable-density groundwater flow and solute transport in heterogeneous media: approaches, resolutions and future challenges. *J Contam Hydrol* 2001;52(1–4): 245–75.
- [44] Voss CI. A finite-element simulation model for saturated-unsaturated fluid-density-dependent groundwater flow with energy transport or chemically-reactive single-species solute transport. *US Geol Surv Water Resour Invest* 1984; p. 409 [Rep 84-4369].
- [45] Voss CI, Souza WR. Variable density flow and solute transport simulation of regional aquifers containing a narrow freshwater-saltwater transition zone. *Water Resour Res* 1987;26:2097–106.
- [46] Vasilyev OV, Paolucci S. A fast adaptive wavelet collocation algorithm for multidimensional PDEs. *J Comput Phys* 1997;125:16–56.
- [47] Vasilyev OV, Bowman C. Second Generation wavelet collocation method for the solution of the partial differential equations. *J Comput Phys* 2000;165:660–93.
- [48] Vasilyev OV. Solving the multidimensional problems with localized structures using the second generation wavelets. *Int J Comput Fluid Dyn* 2003;17(2):151–68.
- [49] Wang R, Keast P, Muir P. A High order global spatially adaptive collocation method for 1-D parabolic PDEs. *Appl Numer Math* 2004;50:239–60.
- [50] Zegeling PA, Werver JG, van Eijkeren JKM. Application of moving grid method to a class of 1-D brine transport problems. *Int J Numer Methods Fluids* 1992;15(2):175–91.
- [51] Zelkin EG, Kravchenko VF, Basarab MA. Interpolation of signals with a Finite spectrum by Fourier transform of atomic functions and application of this interpolation to Antenna synthesis problems. *J Commun Technol Electron* 2002;47:413–20.