 CMC, vol.6, no.2, pp.51-70, 2007

# An Adaptive Multi-resolution Method for Solving PDE's

**V. Kozulić[1], H. Gotovac[1] and B. Gotovac[1]**

**Abstract:** In this paper, we present a multi-resolution adaptive algorithm for solving problems described by partial differential equations. The technique is based on the collocation method using Fup basis functions, which belong to a class of Rvachev's infinitely differentiable finite functions. As it is possible to calculate derivation values of Fup basis functions of high degree in a precise yet simple way, so it is possible to efficiently apply strong formulation procedures. The mesh free method developed in this work is named Adaptive Fup Collocation Method (AFCM). The distribution of collocation points within the observed area is changed adaptively, depending on the character of the solution function and the accuracy criteria. The numerical procedure is designed through a method of lines (MOL). The basic characteristic of the method is an adaptive multi-resolution approach in solving problems with different spatial and temporal scales and with a desired level of accuracy using the entire family of Fup basis functions. Good performance of the proposed method is shown through the numerical examples by using a few general advection dominated problems. The results demonstrate that the method is very convenient for solving engineering problems which require extensive computational resources, especially in describing sharp fronts or high gradients and changes of narrow transition zones in space and time.

**Keywords**: Fup basis functions, compact support, method of lines, collocation method, adaptive multi-resolution approach, high gradients.

[1] Faculty of Civil Engineering and Architecture, University of Split, Split, Croatia

## 1 Introduction

Many physical and engineering problems are characterized with zones of sharp gradients that include existence of sharp interface and narrow transition zones. Numerical modeling of such processes usually encounters significant difficulties in resolving numerical oscillations and dispersion. To overcome these difficulties requires a very fine mesh or grid and small time steps, which are taxing to computational resources.

Recent numerical approaches focus on developing adaptive methods with low computational costs. The first attempt in that direction was obtained by using classical finite difference and finite element methods [Alves et al. (2002)]. The main difficulty in applying these methods was finding a stable solution at the transition between zones having different discretization. Recently, there have been many attempts to develop new adaptive procedures. Among others, special attention is paid to adaptive wavelet Galerkin methods [Beylkin and Keiser (1997), Chiavassa et al. (2002)] and collocation methods [Bertoluzza (1996), Bertoluzza and Naldi (1996), Vasilyev and Paolucci (1997), Hesthaven and Jameson (1998), Holmstrom (1999), Vasilyev and Bowman (2000), Cruz et al. (2001)]. Spline adaptive collocation methods are described in Ramachandran and Dudukovic (1984), Bhattacharya and Joseph (1998), and Wang, Keast and Muir (2004). The adaptive wavelet Galerkin methods have three potential difficulties: treatment of general boundary conditions, treatment of 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.

In recent years, a number of mesh-free methods have been developed for solving partial differen-

tial equations. Mesh-free methods can be classified into the so-called "meshless methods" and "truly meshless methods" categories. In the first category, some background mesh is still necessary for numerical integration of the weak form (e.g. element-free Galerkin method [Belytschko, Lu and Gu (1994)]). In truly meshless methods, both interpolation and integration are performed without a mesh (e.g. RBFN-based method [Kansa (1990)], meshless local Petrov-Galerkin method [(Atluri and Shen (2002), Lin and Atluri (2000)]). The power and flexibility of the MLPG approach were reported for elasto-static problems [Atluri, Han and Rajendran (2004)], and elasto-dynamic problems [Han and Atluri (2004)]. Mesh-free methods based on radial basis functions (RBFs) have recently gained much attention in many different applications in numerical analysis. Some applications using RBFs for heat transfer problems and solution of the Navier-Stokes equations were reported in [Mai-Duy (2004)], the numerical simulation of two-phase flow in porous media in [Iske and Käser (2005)], application to transport phenomena in [Šarler (2005)].

In mesh-free methods, construction of basis functions is the central issue. The domain for field variable approximation (the support domain) should be small compared with the entire problem domain (compact support). Finite basis functions which do not depend upon the type and degree of the boundary-value problem shall be selected. To avoid too much complexity in the numerical solution, it is necessary to calculate relatively simply the values of the basis functions and their derivatives as well as the scalar products of the function with itself, its derivatives, and elementary functions.

Apart from wavelets and splines, there is a relatively less known class of atomic or $R_{bf}$ basis functions (Rvachev's basis functions) [Rvachev and Rvachev (1971), Gotovac (1986)]. Atomic functions are classified between classic polynomials and spline functions. However, in practice, their application as basis functions is closer to splines or wavelets. Thus, the class of atomic functions can be regarded as splines of an infinitely high degree. In the study by Gotovac (1986), the existing knowledge on atomic functions is systemized and basis functions are transformed into a numerically applicable form. Procedures for calculation of function values are given by Gotovac and Kozulić (1999) together with an illustration of basic possibilities for their application in practice. Recently, an application of Fup basis functions has been shown (one type of atomic basis functions) in signal processing [Kravchenko et al. (2001), Zelkin et al. (2002)], for initial value problems [Gotovac and Kozulić (2002)], and in a non-adaptive collocation method for boundary value problems [Kozulić and Gotovac (2000), Gotovac et al. (2003)].

A new adaptive collocation method, developed by using Fup basis functions, is presented in this paper. The 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). Fup basis functions have good approximation properties as well as the very important property of universality [Rvachev and Rvachev (1979)], which means that the vector space of $n$-th dimension is contained within the vector space of $(n+1)$ dimension. Numerically, it means that every new added basis function will improve the solution or at least maintain the obtained accuracy. These functions support a multi-resolution analysis and adaptive collocation approach in which universality and infinite smoothness leads to an efficient solution with desirable accuracy. An application of the presented algorithm is demonstrated by a few numerical examples. The results obtained by these analyses illustrate that the proposed Adaptive Fup Collocation Method (AFCM) is well suited for solving problems with sharp fronts and for dealing with general boundary conditions and nonlinearities.

In the following sections we show a brief review of Fup basis functions, description of the adaptive numerical model with main features of the Fup collocation transform, results of numerical examples, and conclusions.

## 2 Fup basis functions

Atomic or Rvachev's basis functions – $R_{bf}$ have a compact support and they are infinitely differentiable functions [Gotovac and Kozulić (1999), Rvachev and Rvachev (1971)]. They are classified between classical polynomials and spline functions, but in practice their application as basis functions is closer to splines and wavelets.

Atomic functions, $y(.)$, 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, $\lambda$ 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 fact that atomic basis functions are exact solutions of differential-functional equations or linear combinations of these exact solutions represents a key difference from splines and wavelets which are obtained from some types of mathematical transforms.

The simplest function, which is most studied among atomic basis functions, is the $up(x)$ function (Fig. 1 and 2). 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:

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

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

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

Figure 2 shows the development of $up(x)$ function by a convolution process using an infinite number of independent uniform probability density functions (pdf) (Eq. 4). Every next convolution uses a uniform pdf with double decreasing support. It means that function $up(x)$ can also be regarded as a pdf function due to condition (3).

Since Eq. (4) represents the exact, but not mathematically suitable expression, Rvachev (1982) and Gotovac and Kozulić (1999) provided a numerically more adequate expression for calculating function $up(x)$:

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

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

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

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 Gotovac and Kozulić (1999), and Gotovac and Kozulić (2002). Argument $(x - 0, p_1 \ldots p_k)$ in Eq. (5) is the difference between the real value of coordinate $x$ and its binary form with $k$ bits, where $p_1 \ldots p_k$ are the 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, depends upon computer 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 $\delta_k$ are the coefficients with $\pm 1$ value

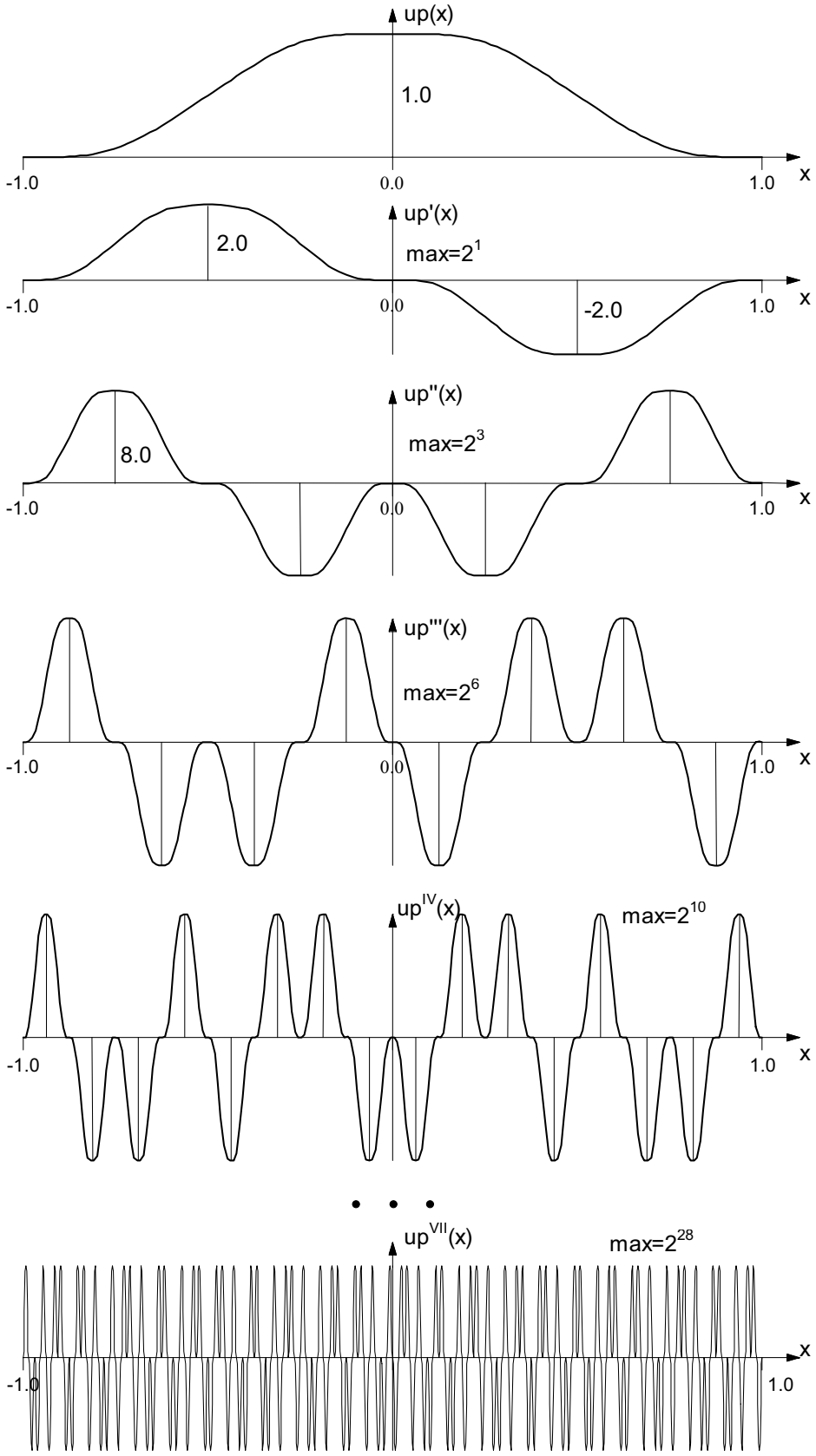


Figure 1: Function $up(x)$ and its derivatives

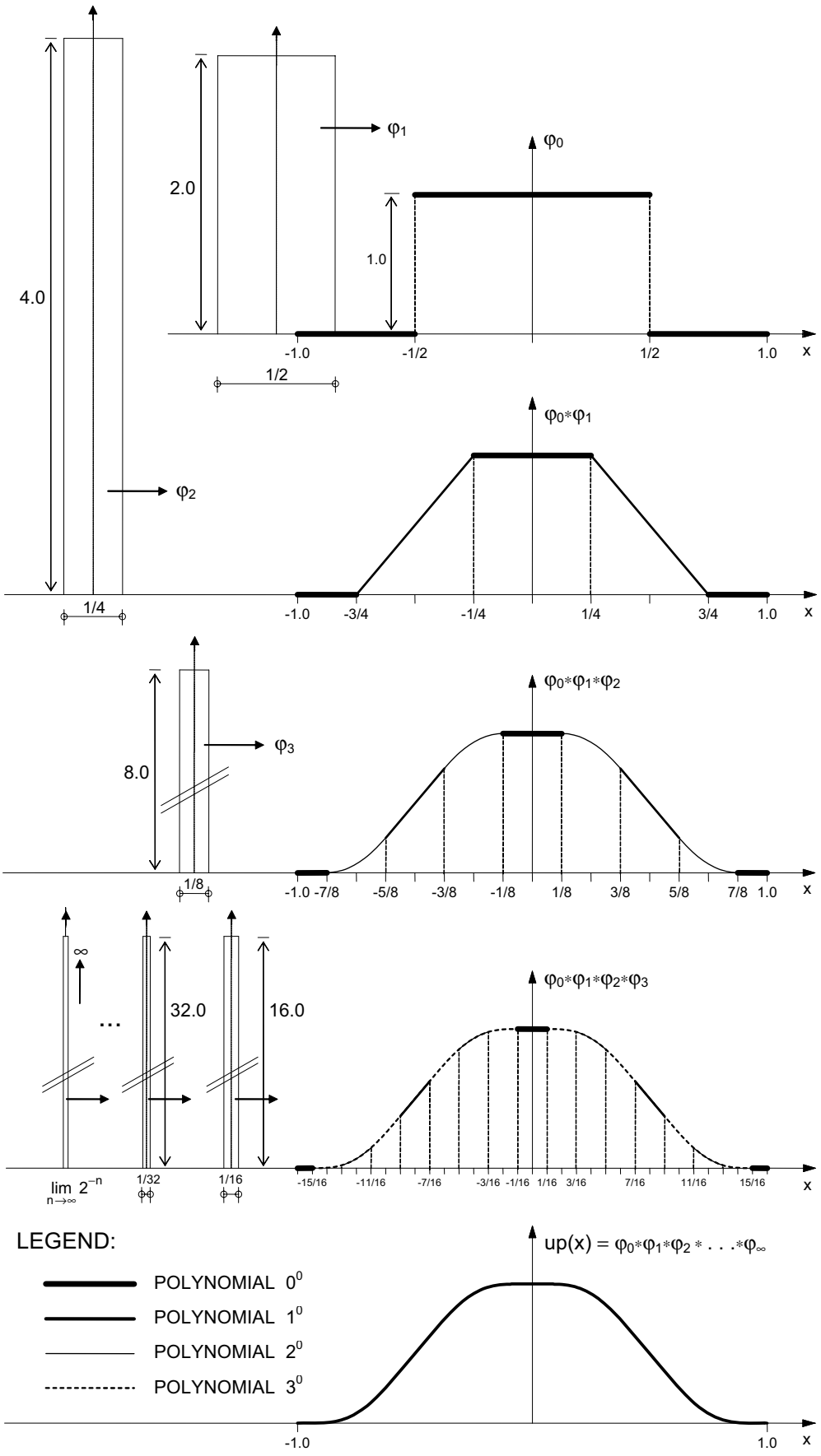


Figure 2: Function $up(x)$ development by convolution

which determine the sign of each term. They change according to the following recursive formulas:

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

Figure 1 shows the $up(x)$ function and its derivatives. It can be observed that each derivative consists of $up(x)$ functions 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:

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

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

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

Eq. (10) is not numerically suitable for the calculation of $Fup_n(x)$ 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 $Fup_n(x)$ function values are calculated using a linear combination of displaced $up(x)$ functions:

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

where coefficient $C_0(n)$ is:

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

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 $C'_k(n)$:

$$C'_0(n) = 1, \quad \text{when } k = 0; \text{i.e. when } k > 0$$
$$C'_k(n) = (-1)^k C_{n+1}^k - \sum_{j=1}^{\min\{k;2^{n+1}-1\}} C'_{k-j}(n) \cdot \delta_{j+1} \quad (13)$$

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

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

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. 3 shows the $Fup_2(x)$ function and its first two derivatives.

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

$$x^2 = 2^{-6} \sum_{k=-1}^{2} \left(k^2 - 5/18\right) Fup_2\left(x - k/4\right). \quad (15)$$

Generally, $n+2$ $Fup_n(x)$ or $2n+1$ $up(x)$ basis functions are needed for the development of an $n$-order polynomial on a characteristic interval $2^{-n}$. This clearly shows that $Fup_n(x)$ basis functions are more suitable and efficient than $up(x)$ basis functions for numerical purposes. A more detailed discussion about Fup and generally atomic basis functions is given in Gotovac and Kozulić (1999), and Gotovac and Kozulić (2002).

## 3 Adaptive Fup Collocation Method

This section presents a complete description of the Adaptive Fup Collocation Method including all necessary steps for its implementation.

### *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

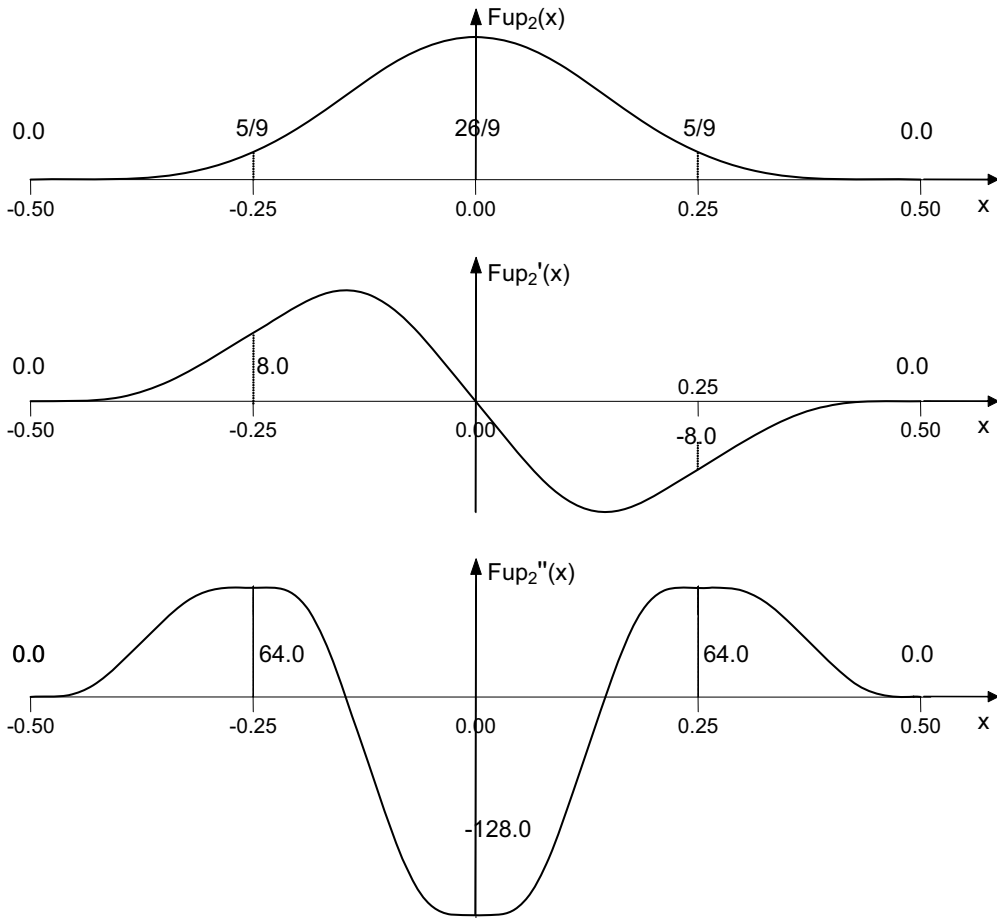


Figure 3: Function $Fup_2(x)$ and its first two derivatives

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 location 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 (Figure 1 and 3). 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 therefore it is called the Fup collocation transform. The high efficiency of the FCT lies in the transform property that 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:

$$u(x) = \sum_{j=0}^{J\to\infty} \sum_{k=-n/2}^{(2^{j_{\min}+j}+n/2)} d_k^j \varphi_k^j(x) \tag{16}$$

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

$$G^j = \left\{ x_k^j \in R : \; x_k^j = 2^{-j}k, \; k \in Z \right\}, \quad j \in Z \tag{17}$$

where $x_k^j$ are the grid collocation points. Note that an even 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. 4. 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.

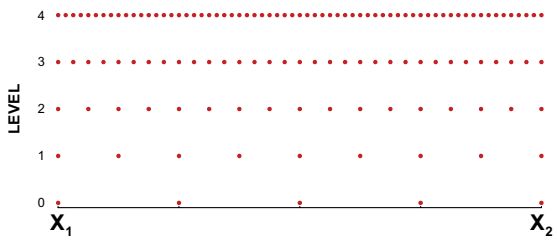


Figure 4: Example of the regular dyadic grid

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:

$$\Delta x_j = \frac{X_2 - X_1}{2^{j_{\min}+j}}. \tag{18}$$

For demonstrating FCT, consider the following test function:

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

with a relatively high threshold of $\varepsilon = 0.07$ which implies that the residual between the Fup approximation and function (19) must be less than the prescribed threshold. Other parameters are $j_{\min}$=2, $X_1$=0, $X_2$=2, and $n = 4$. Figure 5 shows the location of internal and external basis functions and corresponding collocation points. Basis functions are characterized by vertices or peaks where they have 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. 5).

The best choice for locations of the collocation points are vertices of the internal basis functions as proven numerically for splines in Prenter (1989), for wavelets in Vasilyev and Paolucci (1997), and for Fup basis functions in Gotovac and Kozulić (1999).

Moreover, the main difficulty in transformations with localized basis functions is the special treatment of the boundary. For a complete Fup approximation in each characteristic interval $\Delta x_j$, we need $n+2$ $Fup_n(x)$ basis functions (i.e. Eq. 15) which exactly develops an $n$-order polynomial. This request is violated near the boundary if external basis functions are not used and if the level of accuracy is less than inside the domain (Fig. 5). 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 (1996) reported three possible solutions for solving this problem: 1) without external basis functions that leads to a stable but inaccurate solution, as mentioned above,

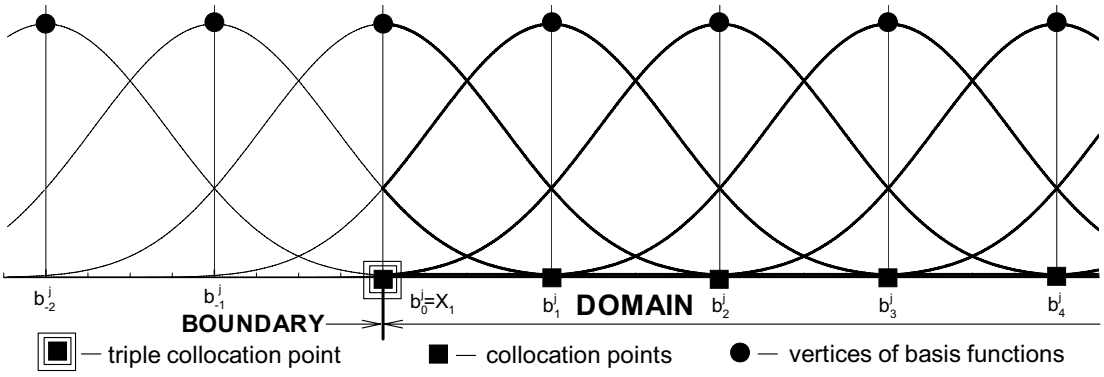


Figure 5: Vertices location of the internal and external $Fup_4(x)$ basis functions and corresponding collocation points

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 [Vasilyev and Paolucci (1997)].

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 boundary (at $X_1$ and $X_2$) as shown in Fig. 5.

The approximation for internal and external basis functions should satisfy the function values in corresponding collocation points and the first $n/2$ derivatives in 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 $\Delta x_j$ should be done in the following form with respect to a basic characteristic interval $2^{-n}$:

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

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

Figure 6 presents the adaptive multi-resolution Fup collocation transform for the chosen function (19). Figure 6a shows an adaptive grid for all levels and internal basis functions for the zero and the first level. Each 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 that is always present in the grid. The FCT satisfies function values in all collocation points and for the first two derivatives in boundary points (Figure 6b). 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 points where residual is below the prescribed threshold are dropped from the grid (Figure 6c). This procedure presents an *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 every next level are homogeneous (zero value) since they are satisfactory at the zero level.

Higher levels include only higher frequencies and

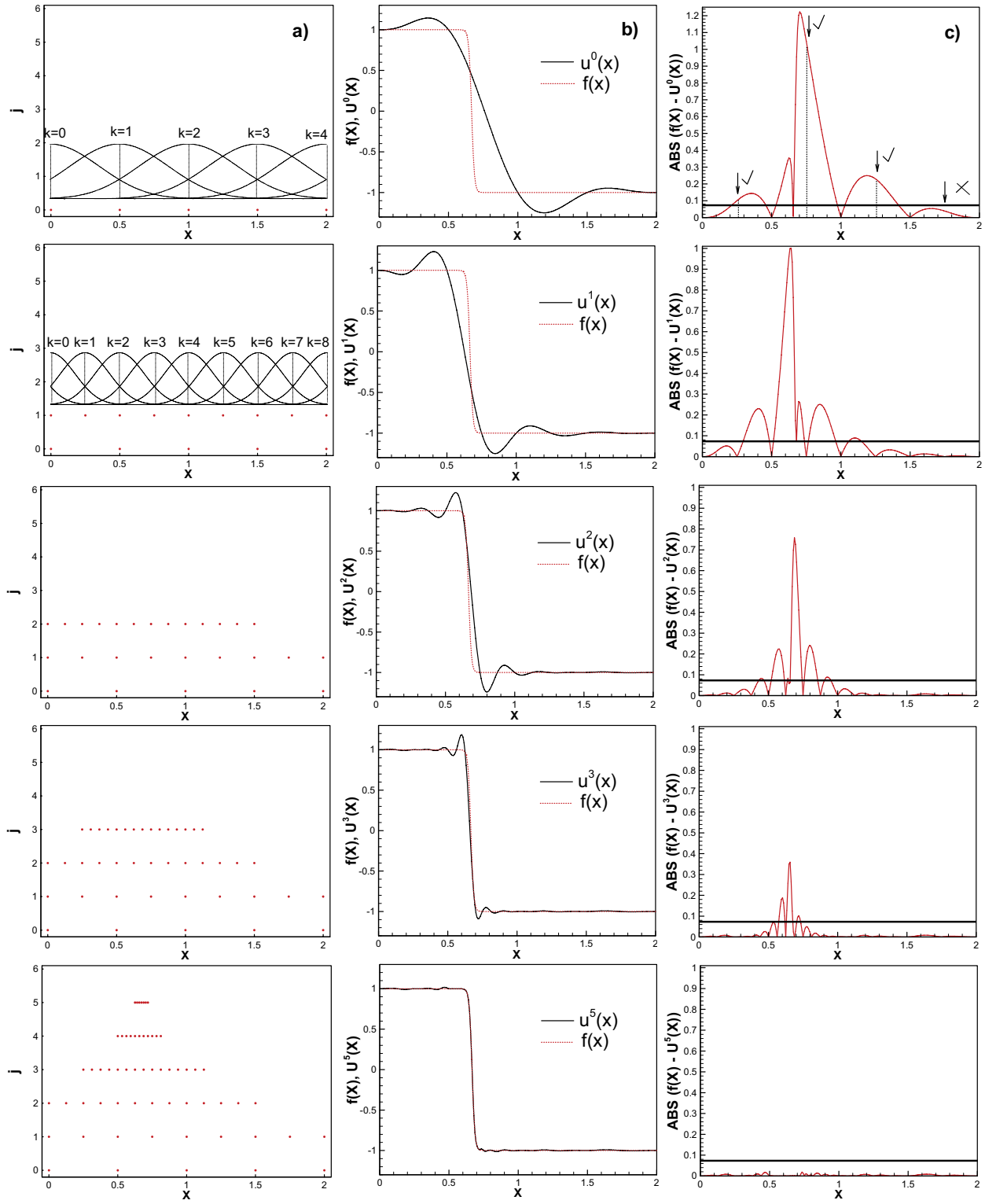


Figure 6: Multi-resolution approximation of the function (19), a) adaptive grid development and internal basis functions, b) FCT approximation and function (19), and c) *a priori* adaptive criterion for new collocation points based on residual between function (19) and their FCT approximation.

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 (Figure 6b). Finally, a residual between the 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: a) it presents an *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 b) it defines accuracy of the approximation because 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:

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

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 5). The function values are satisfied in collocation points:

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

The boundary derivatives are satisfied in points $X_1$ and $X_2$:

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

The residual vector has the following form:

$$\begin{aligned} \Delta_j(x_p^j) = f(x_p^j), &\quad p \in Z^j :\ 0 \le p \le 2^{j_{\min}+j};\\ &\quad j = 0\\ f(x_p^j) - u^{j-1}(x_p^j), &\quad p \in Z^j :\ 0 \le p \le 2^{j_{\min}+j};\\ &\quad j = 1, \ldots, J\\ \Delta_j^{(i)}(X_b) = f^{(i)}(X_b), &\quad j = 0; \quad b = 1,2\\ 0, &\quad j = 1, \ldots, J \end{aligned} \tag{24}$$

### *3.2 Adaptive numerical algorithm*

The main idea of the numerical model presented here is to employ the FCT spatial description in an adaptive algorithm for solving PDEs. Thus, the adaptive Fup collocation method was created. The main feature of the method is adaptive changes of the grid in space and time according to different spatial and temporal scales determined during the adaptation procedure. In this way, the grid follows the system dynamics.

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 the Fup collocation transform and corresponding spatial adaptive strategy. Time integration is obtained by solving the system of differential-algebraic equations written in a general form suitable for many engineering problems:

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

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

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. Equation (25) represents time–dependent partial differential equations which describe time evolution of the solution while algebraic equation (26) presents boundary conditions (Dirichlet, Neumann, or Chauchy mixed type). Figure 7 presents a flow chart of the adaptive Fup collocation method. In this paper, we mainly focus on the spatial approximation, while temporal integration is solved using the classic multistep routines.

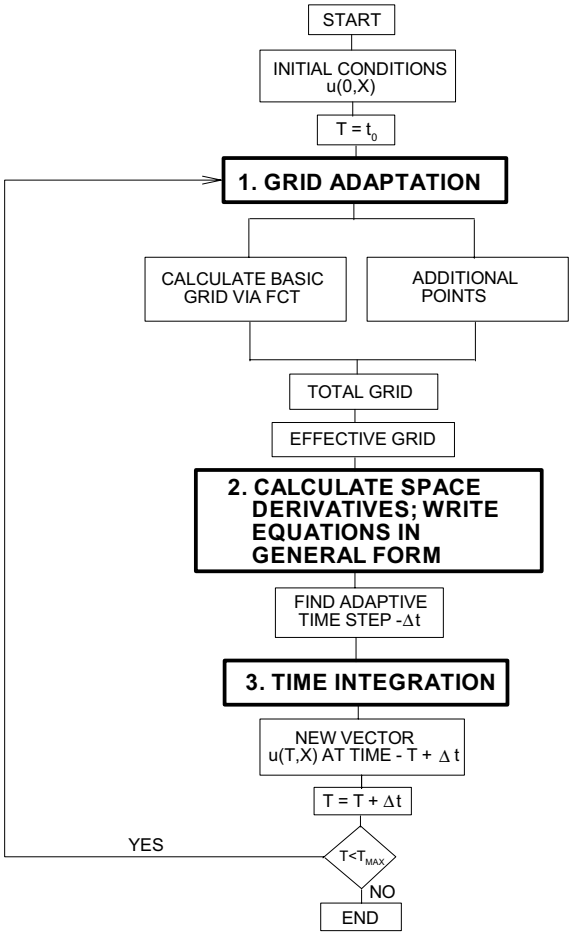


Figure 7: Flow chart for the AFCM

The general numerical algorithm consists of three commonly used basic steps [Cruz et al. (2001), Vasilyev and Bowman (2000), Wang et al. (2004)]:

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

Subsequently, the above steps are described in detail.

### *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. 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 an *a priori* adaptive criterion), we need additional points which enable consistent approximation of the system dynamics (temporal solution changes) during the calculated adaptive time step $\Delta t$. 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 the time step $\Delta t$) 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 [Cruz at al. (2003), Hesthaven and Jameson (1998), Vasilyev and Bowman (2000)]. 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 obtain the basic grid (based on the *a priori* adaptive criterion) required for the Fup approximation with a desired solution accuracy defined by threshold $\varepsilon$. 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 ($x_{k+i}^j$, $i=-N_L,\ldots,N_R$). These points are included to guarantee an

accurate approximation of a possible movement of sharp solution features during the time step. For advection dominated problems, the maximum allowed time step must be related to the maximum velocity in the following way [Cruz, Mendes and Magalhes (2001)], $\Delta t_{adapt} = \max(N_L, N_R)\Delta x_{j_{\max}}/v_{\max}$, in order to guarantee that the front will not move beyond a distance $\max(N_L, N_R)\Delta x_{j_{\max}}$ at the finest resolution level. Note that we chose an arbitrary number of additional points $(N_L, N_R)$ which are directly connected with 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 $(x^{j+l}_{2^l(k-N_L^U)-2^l+1}, \ldots, x^{j+l}_{2^l(k+N_R^U)+2^l-1}, \quad l = 1, \ldots, M)$. 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. 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 numerical and physical characteristics and should be tested for all kinds of problems [Cruz, Mendes and Magalhes (2001), Vasilyev and Bowman (2000)]. In this paper, we employ $M = 1$ as a sufficient number for very accurate modeling of the advection dominated problems.

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, but 25% for 2-D and only 12.5% for 3-D problems).

This adaptive 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 requires 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 [Cruz, Mendes and Magalhes (2001), Vasilyev and Bowman (2000)]. 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.

### *3.2.3 Time integration*

Time integration is obtained by solving the system of the differential – algebraic Eqs. (25)-(26) with initial conditions obtained either from original initial conditions (first time step) or from the previous time step. After each time step, the system (25)-(26) changes as a result of the applied spatial grid adaptation and contains all points from the adapted grid. During the time step, the adaptive grid and the system (25)-(26) remain unchanged.

By applying the collocation procedure to the system (25)-(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, which can be solved for a given time step by public domain subroutine DASPK [Ascher and

Petzold (1998)]:

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

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

DASPK 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 a very important to distinguish the outer time step when the adaptive grid remains unchanged from the inner time step needed for time integrator routine DASPK to achieve accuracy and stability. This routine is appropriate for systems of stiff equations (usually for strongly nonlinear groundwater problems) and it attempts to keep the local error proportional to a user-specified tolerance [Vasilyev and Bowman (2000)]. Unfortunately, it does not guarantee that the global integration error is controlled; this is the reason why this tolerance should be smaller by a few orders than threshold $\varepsilon$, in order to keep the global numerical accuracy closely related to the spatial approximation error.

Within a DASPK subroutine, the modified Newton method is used for solving the nonlinear system (25)-(26). Without a loss of generality, we can show the Newton algorithm in case of the simple backward Euler method or the BDF method of the first order $(k = 1)$:

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

where $\nu$ 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 in 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 DASPK.

## 4 Numerical examples

Two examples are presented in this section to illustrate efficiency of the adaptive Fup collocation method. The first example is Burgers equation with small viscosity. The results show the ability of the method in modeling stationary increased shocks. The second example presents double wave propagation which illustrates the strength of the method in handling a pure hyperbolic coupled system of wave equations.

### *4.1 Burgers equation*

Burgers equation results from an application of the Navier–Stokes equation to unidirectional flow without pressure gradient and small viscosity. The solution is characterized with one dimensional shock that is stationary in space, but increases in time. The problem is described by the following equation, and initial and boundary conditions:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} \tag{29}$$

$$u(x,0) = -\sin(\pi x) \tag{30}$$

$$u(\pm 1, t) = 0 \tag{31}$$

where $u$ is dimensionless velocity, while the domain and viscosity are defined by: $x \in [-1,1]$; $\nu = 10^{-2}/\pi$.

Initial conditions are very simple and monotonic. Dirchlet boundary conditions are homogeneous. Analytic and numerical solutions of this problem obtained via a wavelet collocation method can be

found in Vasilyev and Paolucci (1997):

$$u(x,t) = -\left[\int_{-\infty}^{\infty} \sin(\pi(x-\eta)) \cdot \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \exp\left(-\frac{\eta^2}{4vt}\right) d\eta\right] \Bigg/ \left[\int_{-\infty}^{\infty} \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \cdot \exp\left(-\frac{\eta^2}{4vt}\right) d\eta\right]. \quad (32)$$

The adaptive algorithm uses the following parameters: $N_L = N_R = 2$, $N_L^U = N_R^U = 0$, $M = 1$, $n = 6$, $\varepsilon = 10^{-5}$, $j_{\min} = 4$. Figure 8 shows time evolution of the solution. In the initial stages of the process, the solution is not demanding and requires a relatively large time step. Approximately after time $0.7/\pi$, shock sharpness increases rapidly and needs higher levels and frequencies. After time $1.3/\pi$, shock reaches the maximum steepness and grid requirement.

This example is a classical nonlinear benchmark test that shows ability to handle a shock which is fixed in space, but rapidly changes steepness in time. The shock is very narrow due to small viscosity and needs higher resolution levels, especially in later time steps. The grid follows all changes in solution structure during time simulation.

An efficiency of the algorithm is illustrated by a compression coefficient that is the ratio between a number of collocation points of the non–adaptive uniform grid at the maximum resolution level and a number of collocation points of the adaptive grid. The compression ratio is very high in the initial time steps due to lower solution demand. However, with increase of the shock steepness, the number of levels and collocation points increases progressively and therefore the compression ratio decreases. In later time steps, the solution becomes very stiff and needs an implicit Gear BDF method for efficient time integration, although in initial stages the problem is non-stiff; it requires that the compression ratio drop down to six.

The comparison between analytical and numerical solutions shows the accuracy of the proposed algorithm (Fig. 9). Numerical error is larger outside of the shock and approximately similar within the whole domain, which presents one of the natural advantages of the method. In this example, numerical error is larger than the threshold $\varepsilon$, but keeps the same order of the accuracy.

### 4.2 Double pulse propagation

Wave processes that can be described as pure hyperbolic problems are one of the most demanding tasks in the numerical analysis. We consider two waves that travel in opposite directions and interact to each other. This example has general meaning and shows the ability of the method to solve the coupled system of hyperbolic equations with a reaction term. The double pulse propagation problem is described by the following system of equations and initial and boundary conditions [Alves et al. (2002)]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -\frac{\partial u}{\partial x} + 10uv; \\ \frac{\partial v}{\partial t} &= +\frac{\partial v}{\partial x} - 10uv \end{aligned} \quad (33)$$

$$\begin{aligned} u(x,0) &= \begin{Bmatrix} 1, & 0.3 < x < 0.4 \\ 0, & \text{elsewhere} \end{Bmatrix}; \\ v(x,0) &= \begin{Bmatrix} 1, & 0.6 < x < 0.7 \\ 0, & \text{elsewhere} \end{Bmatrix} \end{aligned} \quad (34)$$

$$\begin{aligned} u(0,t) &= \quad v(1,t) = \quad 0; \\ \frac{\partial u(1,t)}{\partial x} &= \quad \frac{\partial v(0,t)}{\partial x} = \quad 0 \end{aligned} \quad (35)$$

where $u$ and $v$ are dimensionless wave velocities, while the domain is defined by $x \in [0,1]$.

The initial conditions are very demanding because they consist of two pulses or waves and four practically discontinuous fronts. The boundary conditions are mixed (Dirichlet and Neumann type) while the Neumann type corresponds to an outflow boundary for each wave. The problem was solved with an adaptive multi-resolution approach by Alves et al. (2002).

The adaptive algorithm uses the following parameters: $N_L = N_R = 2$, $N_L^U = N_R^U = 2$, $M = 1$,

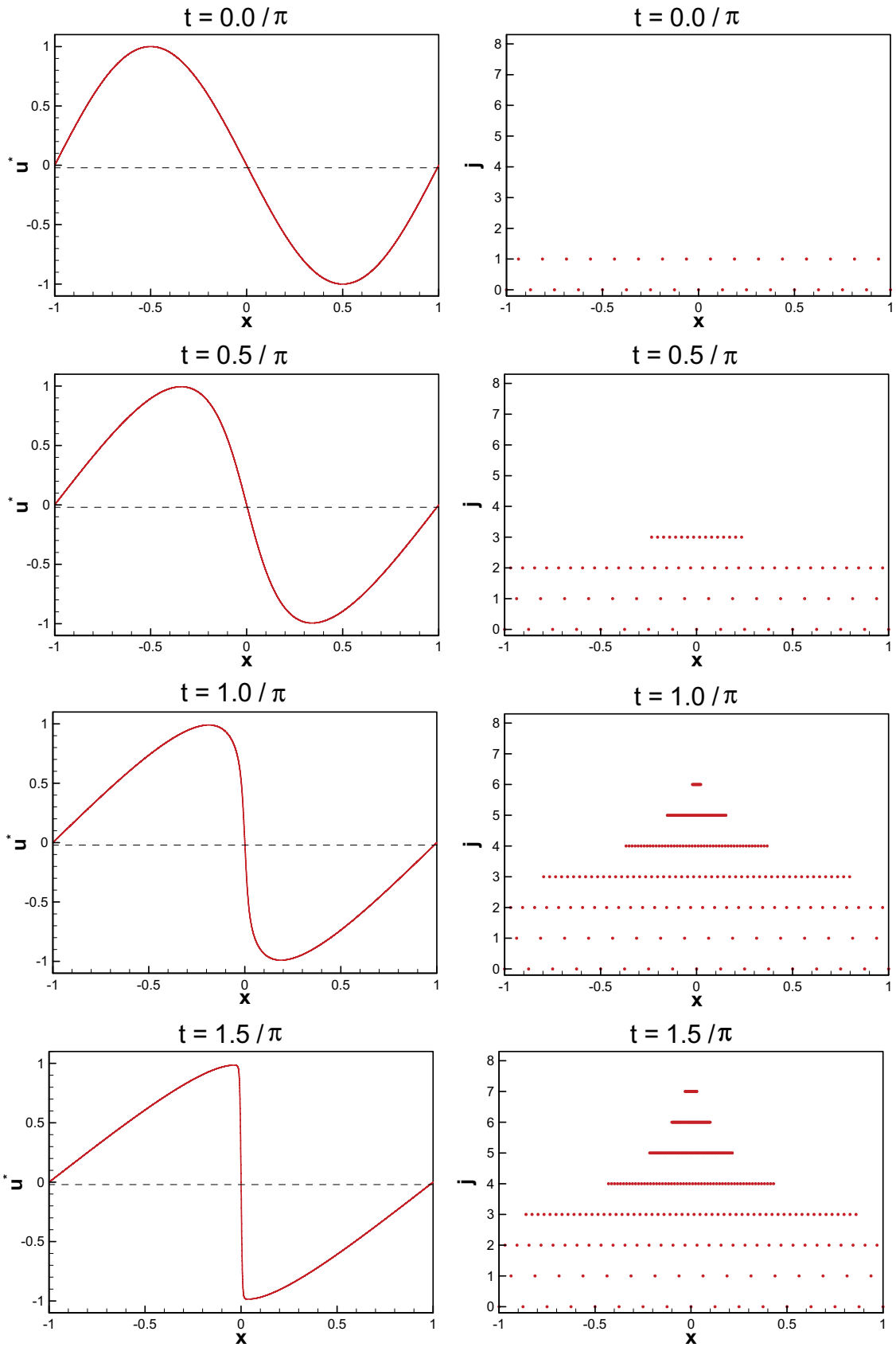


Figure 8: Numerical solution of the problem I: velocity (left) and adapted grid (right)

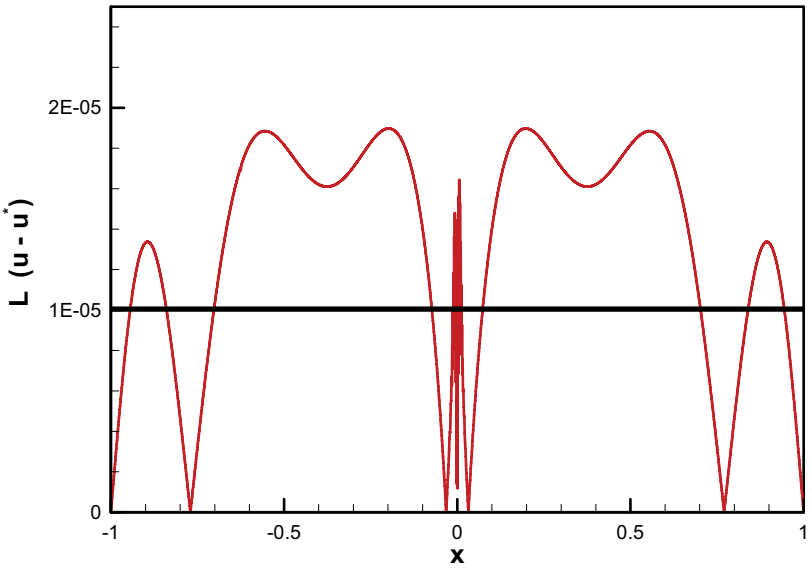


Figure 9: Difference between numerical ($u^*$) and exact ($u$) solution at time $t = 1.5/\pi$

$n = 2$, $\varepsilon = 10^{-3}$, $j_{\min} = 4$, $j_{\max} = 11$. We prescribe maximum level because initial conditions describe abrupt velocity fronts that cannot be approximated accurately with the given threshold. Figure 10 shows time evolution of the wave process. At first, waves only go in opposite directions. The reaction term is zero while waves do not overlap and interact with each other. Because of the opposite reaction term, one wave exits higher with a positive slope, while the second wave exits lower with a negative slope through the interaction zone. After the interactive period, waves travel to the opposite outflow boundary and exit outside the domain.

The adaptive algorithm uses a nearly constant number of collocation points (around 250-260 points). A slight reduction of this number is obtained during the interactive period, and means that the compression ratio is around seven. The numerical solution obtained by the AFCM is practically the same as the solution obtained by Alves et al. (2002).

Solving this relatively simple physical process is not trivial. If we use the proposed numerical algorithm, unphysical and artificial oscillations around fronts occur. The classical central oriented FD operator for the advective term is not adequate for pure hyperbolic problems. In these problems, the method should adopt high–resolution bounded FD schemes. Here we use the SMART high–resolution scheme for an advective term defined by Alves et al. (2002) that has a boundedness property, but introduces some artificial diffusion. The scheme uses four closest points and corresponds to the $Fup_2(x)$ basis functions. Because of the high accuracy of the SMART scheme and the adaptive algorithm, artificial diffusion can be neglected for practical purposes. If we have even a small but stabile diffusive term, as is the case in advection dominated problems, use of bounded high–resolution schemes is not necessary because unphysical oscillations can be reduced within the range of the threshold. This example can serve as the upper limit of the possible and physical based advection dominated problems.

## 5 Conclusions

The main purpose of this work is to present an adaptive multi-resolution method for solution of PDE's and to demonstrate its ability in describing problems with sharp gradients, fronts, and narrow transition zones that change in space and time. The technique is based on the collocation method using infinitely differentiable Fup basis functions. The numerical procedure is designed through a method of lines. Space discretization

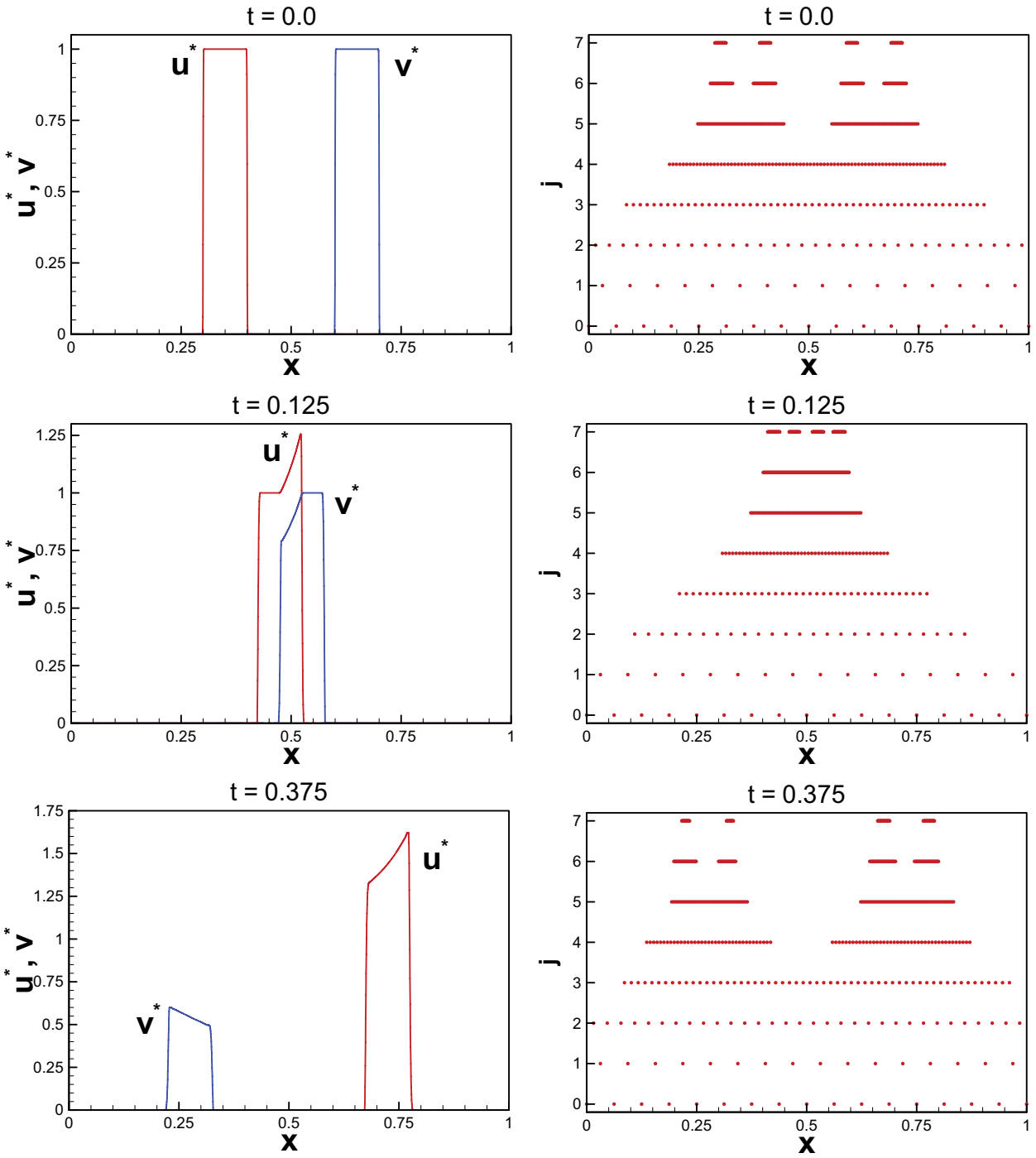


Figure 10: Numerical solution of the problem II: velocity (left) and adapted grid (right)

and grid adaptation on a dyadic grid is obtained by the Fup collocation transform, while time integration is reduced to solving a system of DAE equations. The AFCM enables adaptive multi-resolution evolution of system dynamics resolving different spatial and temporal scales with a desired level of accuracy using the entire family of the Fup basis functions.

The main advantages of the proposed mesh-free method are: (1) the possibility of resolving a wide range of spatial and temporal scales on near optimal adaptive grid, (2) reduced computational efforts due to the space and time adaptation, (3) predefined mesh and numerical integration are avoided, and (4) an accurate numerical solution with global numerical error is closely related to the prescribed spatial threshold. That implies active control of numerical errors, especially numerical oscillations and dispersion.

The numerical model is tested and verified with a few general advection dominated problems. Results show that new adaptive Fup collocation method is well suited for dealing with general boundary conditions and nonlinearities. Further work on this research is now being conducted in order to extension the algorithm to 2D problems with complex domain and simultaneous space-time Fup approach.

## References

**Alves, M. A.; Cruz, P.; Mendes, A.; Magalhes, F. D.; Pinho, F. T.; Oliviera, P. J.** (2002): Adaptive multi-resolution approach for solution of hyperbolic PDE's. *Comp. Methods Appl. Mech. Eng.*, vol. 191, pp. 3909-3928.

**Ascher, U.; Petzold, L.** (1998): *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Society for Industrial and Applied Mathematics, Berkeley.

**Atluri, S. N.; Han, Z. D.; Rajendran, A. M.** (2004): A new implementation of the meshless finite volume method, through the MLPG "mixed" approach. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6(6), pp. 491-514.

**Atluri, S. N.; Shen, S.** (2002): The meshless local Petrov-Galerkin (MLPG) method: A simple & lesscostly alternative to the finite element and boundary element methods. *CMES: Computer Modeling in Engineering & Sciences,* vol. 3(1), pp. 11-52.

**Belytschko, T.; Lu, Y. Y.; Gu, L.** (1994): Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229-256.

**Bertoluzza, S.** (1996): Adaptive wavelet collocation method for the solution of Burgers equation. *Transport Theory and Statistical Physics*, vol. 25, pp. 339-352.

**Bertoluzza, S.; Naldi, G.** (1996): A Wavelet Collocation Method for the Numerical Solution of PDEs. *Applied & Computational HarmonicAnalysis*, vol. 3, pp. 1-9.

**Beylkin, G.; Keiser, J. M.** (1997): On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases. *J. Comput. Physics*, vol. 132, pp. 233-259.

**Bhattacharya, A.; Joseph, B.** (1988): Simulation of fixed – bad gas solid reactors using an adaptive spline collocation method. *Comp. Chem. Eng.*, vol. 12(4), pp. 351-352.

**Chiavassa, G.; Guichaoua, M.; Liandrat, J.** (2002): Two adaptive wavelet algorithms for nolinear parabolic differential equations. *Int. J. Comput. Fluids*, vol. 31, pp. 467-480.

**Cruz, P.; Mendes, A.; Magalhes, F. D.** (2001): Using wavelets for solving PDEs: an adaptive collocation method. *Chemical Eng. Science*, vol. 56, pp. 3305-3309.

**Cruz, P.; Alves, M. A.; Magalhes, F. D.; Mendes, A.** (2003): Solution of hyperbolic PDEs using a stable adaptive multi-resolution method. *Chem. Eng. Science*, vol. 58, pp. 1777-1792.

**Gotovac, B.** (1986): *Numerical modelling of engineering problems by smooth finite functions*, (in Croatian), Ph.D. Thesis, Faculty of Civil Engineering, University of Zagreb, Zagreb.

**Gotovac, B.; Kozulić, V.** (1999): On a selection of basis functions in numerical analyses of engineering problems. *Int. J. Eng. Model.*, vol. 12(1-4), pp. 25-41.

**Gotovac, B.; Kozulić, V.** (2002): Numerical

solving the initial value problems by $R_{bf}$ basis functions. *Int. J. Struct. Eng. Mech.*, vol. 14(3), pp. 263-285.

**Gotovac, H.; Andricevic, R.; Gotovac, B.; Kozulić, V.; Vranješ, M.** (2003): An improved collocation method for solving the Henry problem. *J. Contaminant Hydrology*, vol. 64, pp. 129-149.

**Han, Z. D.; Atluri, S. N.** (2004): A Meshless Local Petrov-Galerkin (MLPG) approaches for solving 3-dimensional elasto-dynamics. *CMC: Computers, Materials & Continua*, vol. 1(2), pp. 129-140.

**Hesthaven, J. S.; Jameson, L. M.** (1998): A wavelet Optimized Adaptive Multi – domain method. *J. Comput. Physics*, vol. 145, pp. 280-296.

**Holmstrom, M.** (1999): Solving hyperbolic PDEs using interpolation wavelets. *Scientific Computing*, vol. 21, pp. 405-420.

**Iske, A.; Käser, M.** (2005): Two-phase flow simulation by AMMoC, an adaptive meshfree method of characteristics. *CMES: Computer Modeling in Engineering & Sciences,* vol. 7(2), pp. 133-148.

**Kansa, E. J.** (1990): Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics-II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, vol. 19(8-9), pp. 147-161.

**Kozulić, V.; Gotovac, B.** (2000): Numerical analysis of 2-D problems using Fup basis functions. *Int. J. Eng. Model.*, vol. 13(1-2), pp. 7-18.

**Kravchenko, V. F.; Basarab, M. A.; Perez-Meana, H.** (2001): Spectral properties of atomic functions used in digital signal processing. *Commun. Tech. Elect.*, vol. 46, pp. 494-511.

**Li, S.; Petzold, L.** (2004): Design of new DASPK for sensitivity analysis. *Internet communication*.

**Lin, H.; Atluri, S. N.** (2000): Meshless local Petrov-Galerkin (MPLG) method for convection-diffusion problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 1(2), pp. 45-60.

**Mai-Duy, N.** (2004): Indirect RBFN method with scattered points for numerical solution of PDEs. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6(2), pp. 209-226.

**Prenter, P. M.** (1989): *Splines and Variational Methods*, Wiley, New York.

**Ramachandran, P. A.; Dudukovic, M. P.** (1984): Solution by triple collocation for periodic operation of heat regenerators. *Comp. Chemical Eng.*, vol. 8(6), pp. 377-388.

**Rvachev, V. L.; Rvachev, V. A.** (1971): On a Finite Function. *DAN URSR*, A(6), pp. 705-707.

**Rvachev, V. L.; Rvachev, V. A.** (1979): *Non-classical methods for approximate solution of boundary problems*, (in Russian), Naukova dumka, Kiev.

**Rvachev, V. L.** (1982): *Theory of R-functions and their application*, (in Russian), Naukova dumka, Kiev.

**Šarler, B.** (2005): A radial basis function collocation approach in computational fluid dynamics. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7(2), pp. 185-193.

**Vasilyev, O. V.; Paolucci, S.** (1997): A fast adaptive wavelet collocation algorithm for multidimensional PDEs. *J. Comput. Physics*, vol. 125, pp. 16-56.

**Vasilyev, O. V.; Bowman, C.** (2000): Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations. *J. Comput. Physics*, vol. 165, pp. 660-693.

**Vasilyev, O. V.** (2003): Solving the multidimensional problems with localized structures using the second generation wavelets. *Comput. Fluid Dynamics*, vol. 17(2), pp. 151-168.

**Wang, R.; Keast, P.; Muir, P.** (2004): A High order global spatially adaptive collocation method for 1-D parabolic PDEs. *Applied Num. Math.*, vol. 50, pp. 239-260.

**Zelkin, E. G.; Kravchenko, V. F.; Basarab, M. A.** (2002): Interpolation of signals with a Finite spectrum by Fourier transforms of atomic functions and application of this interpolation to Antenna synthesis problems. *Commun. Tech. Elect.*, vol. 47, pp. 413-420.