Journal of Applied Nonlinear Dynamics 7(3) (2018) 287-295

Journal of Applied Nonlinear Dynamics

https://lhscientificpublishing.com/Journals/JAND-Default.aspx

An Adaptive Multiresolution Scheme with Second Order Local Time-stepping for Reaction-Diffusion Equations

Müller Moreira Lopes[1†], Margarete O. Domingues[2], Odim Mendes[3], Kai Schneider[4]

[1] Post-graduation program in Applied Computing, National Institute for Space Research, São José dos Campos, 12227-010, Brazil

[2] Associated Laboratory for Computing and Applied Mathematics, National Institute for Space Research, São José dos Campos, 12227-010, Brazil

[3] Space Geophysics Division, National Institute for Space Research, São José dos Campos, 12227-010, Brazil

[4] Institut de Mathématiques de Marseille (I2M), Aix-Marseille Université, 13453, Marseille, Cedex 13, France

Submission Info

Communicated by J.A.T. Machado
Received 24 April 2017
Accepted 5 July 2017
Available online 1 October 2018

Keywords

Numerical simulation
Fluid dynamics
Geophysical nonlinear dynamics
Multiresolution

Abstract

For adaptive multiresolution schemes we propose a local time-stepping scheme based on natural extensions of Runge–Kutta methods. We consider reaction-diffusion equations in two space dimensions and assess the precision and efficiency of the new method. The obtained results are compared with those using classical finite volume schemes on a uniform grid and multiresolution schemes with global time stepping. It is shown that both CPU time and precision of the adaptive solutions are improved.

1 Introduction

Numerical combustion is a playground for adaptive discretisations, as flame fronts are typically very thin and therefore localised in physical space. Hence local grid refinement techniques are attractive since they can speed-up the computations and reduce the memory requirements. A considerable variety of adaptive schemes can be found on the market, for instance adaptive mesh refinement, see e.g. [1], or adaptive multiresolution schemes, see [2, 3]. The idea of adaptive multiresolution methods, which goes back to the seminal work of Harten [4], is to analyse the numerical solution computed with a classical discretisation, e.g. finite volumes, and to determine which grid points can be removed. To this end, wavelet coefficients are computed and applying thresholding yields a grid perfectly adapted

†Corresponding author.
Email address: muller.lopes@inpe.br

ISSN 2164-6457, eISSN 2164-6473/$-see front materials
DOI:10.5890/JAND.2018.09.006

to the solution. For the next time-step adjacent grid points are added to account for the translation of the solution and the generation of finer scales, the latter is necessary in the case of non-linear equations. The memory and CPU time requirements of the underlying numerical schemes can be thus significantly reduced while controlling the precision of the solution. Further speed-up can be obtained using instead of a global time-stepping, which means that the time step size is determined by the finest spatial scale, a local time stepping where the time step is adapted to the local spatial scale. This last aspect means that large time steps can be used in regions of coarse grids, while small time steps are only used in regions of fine grids. For details we refer to the original papers by Müller and Striba [5] and by Domingues et al [6], where local time stepping has been introduced in the context of adaptive multiresolution methods.

The present paper introduces a new local time-stepping approach based on natural extensions of Runge–Kutta (NERK) methods [7]. The fluxes at the intermediate Runge–Kutta stages can be interpolated and synchronisation can be directly obtained. In [8] we propose a higher order extension of NERK methods and present a detailed benchmarking for one, two and three-dimensional problems.

The outline of the paper is the following. After presenting the essential components of the multiresolution method in Section 2, we briefly summarise the used Runge–Kutta methods in Section 3. Then, the ideas of local time-stepping are recalled and the proposed approach using NERK is introduced in Section 4. Numerical results for reaction-diffusion equations are given in Section 5 and in Section 6 we draw some conclusions.

2 Adaptive multiresolution methods using finite volumes

A two-dimensional initial value problem for a vector quantity $\mathbf{Q} = \mathbf{Q}(x,y,t)$, inside a physical domain Ω, can be written in the following divergence form as:

$$\frac{\partial \mathbf{Q}}{\partial t} = -\frac{\partial \mathbf{F}}{\partial x} - \frac{\partial \mathbf{F}}{\partial y} + \mathbf{S}, \quad \text{on } (x,y,t) \in \Omega \times [0,+\infty), \Omega \subset \mathbb{R}^2, \tag{1}$$

where the physical flux $\mathbf{F}$ depends on $\mathbf{Q}$ and its derivatives, and the source term $\mathbf{S}$ depends on $\mathbf{Q}$. The flux $\mathbf{F}$ can be decomposed into advective and diffusive contributions, where the diffusion coefficient is positive and constant. Here, those advective and diffusive terms are discretised with a McCormack scheme.

To complete this problem, an initial condition $\mathbf{Q}(x,y,t=0) = \mathbf{Q}_0(x,y)$ and appropriate boundary conditions are imposed. The computational domain is partitioned into rectangular cells using a classical finite volume discretisation in Cartesian geometry.

This partition is performed using identical rectangles $(\Omega_{ij})_{i,j\in\Lambda},\ \Lambda = \{0,\dots,i_{\max}\} \times \{0,\dots,j_{\max}\}$, where $\Omega = \cup_{i,j}\Omega_{i,j}$.

For each cell, a volume value $|\Omega_{i,j}| = \int_{\Omega_{i,j}} dx\,dy$ is computed. This value is used to compute the cell average value $\bar{\mathbf{q}}_{i,j}(t)$ for the quantity $\mathbf{Q}$ inside every $\Omega_{i,j}$ at a time instant t by,

$$\bar{\mathbf{q}}_{i,j}(t) = \frac{1}{|\Omega_{i,j}|} \int_{\Omega_{i,j}} \mathbf{Q}(x,y,t)\,dx\,dy.$$

The finite volumes formulation consists in using these average values to represent the solution in the computational domain, and then we evolve those averages in time.

In the two-dimensional case, $\Omega_{i,j}$ is given by the rectangle $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ of length $\Delta x_i \Delta y_j$, where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\Delta y_j = y_{j+1/2} - y_{j-1/2}$. The following relation used to perform the time evolution of a cell $\Omega_{i,j}$ in the finite volume formulation is obtained by integrating the Equation (1)

on the cell $\Omega_{i,j}$:

$$\frac{d\bar{\mathbf{q}}_{i,j}}{dt}(t) = -\frac{1}{\Delta x_i}\left(\bar{\mathbf{F}}_{i+1/2,j} - \bar{\mathbf{F}}_{i-1/2,j}\right) - \frac{1}{\Delta y_j}\left(\bar{\mathbf{F}}_{i,j+1/2} - \bar{\mathbf{F}}_{i,j-1/2}\right) + \bar{\mathbf{S}}_{i,j}, \tag{2}$$

where $\bar{\mathbf{F}}$ is the numerical flux. The term $\bar{\mathbf{S}}_{i,j}$ is a approximation of the source term in the cell $\Omega_{i,j}$.

Aiming for a CPU time and memory storage reduction in the adaptive computations, Harten [4] proposed the adaptive multiresolution method (MR) in the cell average context. This approach represents the solution using a hierarchy of regular grids which are dyadically nested. These grids are enumerated by its refinement level ℓ, associated with the number of cells on the regular grid. A two-dimensional grid of refinement level ℓ, denoted by Ω^{ℓ}, is composed by $2^{2\ell}$ cells.

Computationally, this sequence of nested grids is implemented using a quad-tree structure, where the nodes of level ℓ generate the grid Ω^{ℓ}. This structure can be interpreted so that a finer scale contains the same information of a coarser scale plus a sum of details between these levels. In the MR context, these details are called wavelet coefficients. To build the adaptive grid, a thresholding procedure is applied over the hierarchy of grids. For that, the tree is checked in each refinement level, starting from the finest one, for cells whose wavelet coefficients are smaller than a predetermined threshold value ε. These cells are removed from the tree. Subsequently, this procedure is repeated on the next coarser level. After this process, the adaptive grid is obtained by considering the leaves of this tree as its cells. This MR approach was recently compared with adaptive mesh refinement [9] and it was concluded that MR yields more efficient, i.e., sparser grids, for the same precision.

The functionality of the grid generation procedure is justified by the fact that the wavelet coefficients have the property of being small in regions where the solution is smooth, while for the regions with steep gradients, they are larger.

However, in order to obtain better algorithms for flux computations between cells of different scales, the adapted grid has the restriction to be a graded-tree [3]. This restriction implies that the neighbour of every cell has at maximum, one refinement level of difference. This restriction implies the usage of auxiliary leaves placed as child nodes of leaves which have an interface with a finer leaf. These new leaves are called virtual leaves and are used to perform the flux computations between their parent leaf and its finer neighbour.

Due to the graded-tree restriction, the flux computations required to evolve a leaf are divided into two possible scenarios. One happens when the neighbour leaf or virtual leaf is in the same refinement level. In this scenario, the flux is computed in the same way performed for the finite volume method. The other scenario happens when the neighbour leaf is in a more refined scale. In this case, the flux computations are performed between the current cell virtual leaves and the neighbour leaf.

Once the adaptive grid is obtained, the current solution is considered well represented in this grid. However, this grid may not be suitable for the solution obtained after the time evolution. In order to guarantee that the adaptive grid is adequate for both current and future solution, the leaves with interfaces with finer leaves are refined. Then, new virtual leaves are placed where they are required. More details of the MR scheme, including the operators applied to compute the wavelet coefficients, are given in [4], details about the implementation used can be found in [3].

3 Runge–Kutta methods

The space discretisation of the initial value problem given in Equation 1 leads to a system of ordinary differential equations, which describes the evolution of each cell averages on the adaptive grid. This system has the following form:

$$\frac{d\bar{\mathbf{q}}}{dt} = f(t, \bar{\mathbf{q}}). \tag{3}$$

To perform a time evolution with time-step Δt^n, the following compact form of a two-stage second-order Runge-Kutta (RK) method is applied:

$$\bar{\mathbf{q}}^* = \bar{\mathbf{q}}^n + \Delta t^n \, f(t^n, \bar{\mathbf{q}}^n) \tag{4a}$$

$$\bar{\mathbf{q}}^{n+1} = \frac{1}{2}\bar{\mathbf{q}}^n + \frac{1}{2}\bar{\mathbf{q}}^* + \frac{1}{2}\Delta t^n f\left(t^n + \Delta t^n, \bar{\mathbf{q}}^*\right), \tag{4b}$$

This alternative formulation corresponds to a classical RK method with coefficients $c_1 = 0$, $c_2 = a_{21} = 1$ and $b_1 = b_2 = \frac{1}{2}$.

3.1 Natural extensions for Runge–Kutta methods

The Natural Extensions for Runge–Kutta methods are techniques to produce an polynomial approximation of the solution inside the interval $[t^n; t^n + \Delta t^n]$, after the time evolution. These methods use the same coefficients a_{ij} and c_i of the correspondent RK method, producing thus no extra computational cost. They were first introduced in [7].

These methods use weight polynomials β_i, based on the coefficients b_i of the correspondent RK method. A two-stage NERK method are expressed by:

$$\bar{\mathbf{q}}(t^n + \theta\Delta t^n) = \bar{\mathbf{q}}^n + \beta_1(\theta)\, k_1 + \beta_2(\theta)\, k_2,\ \theta \in (0,1], \tag{5}$$

A relation between the correspondent RK method and the proper polynomials β_i can be found in [10]. Using the coefficients for RK2 given earlier, the following polynomials are obtained:

$$\beta_1(\theta) = -\frac{1}{2}\theta^2 + \theta, \quad \beta_2(\theta) = \frac{1}{2}\theta^2. \tag{6}$$

In this work, the application of the NERK approximation is to obtain a second order solution at the time instant $t^n + \frac{1}{2}\Delta t^n$. This solution is required to perform the synchronisation required for the proposed MRLT/NERK method.

Considering the interest in the solution with $\theta = \frac{1}{2}$, used to perform the synchronizations, the following steps are computed along with the correspondent steps of the RK compact form:

$$\bar{\mathbf{q}}^*_{\theta=\frac{1}{2}} = \bar{\mathbf{q}}^n + \frac{3}{8}\Delta t^n f(t^n, \bar{\mathbf{q}}^n) \tag{7a}$$

$$\bar{\mathbf{q}}_{\theta=\frac{1}{2}} = \bar{\mathbf{q}}^*_{\theta=\frac{1}{2}} + \frac{1}{8}\Delta t^n f\left(t^n + \Delta t^n, \bar{\mathbf{q}}^*\right). \tag{7b}$$

This procedure is performed aiming for a better memory management during the computations.

4 Local time-stepping

To improve the computational efficiency of the MR methods, in [6] we proposed a local time stepping (LT) scheme for the MR context. This approach uses an independent and adapted time step for each leaf. This adapted time step is based on the spatial size of the leaf.

In explicit time schemes, the CFL condition is used to obtain a suitable time step Δt for the numerical simulation. Supposing that the leaves on the finest scale L respect the CFL condition of the problem, the LT scheme consists in performing, for each cell of level ℓ, the time evolution using the time step $\Delta t_\ell = 2^{L-\ell}\Delta t$. This approach is justified by the fact of the Courant number σ depends of the ratio between Δt_ℓ and Δx_ℓ, and due to the value σ remains the same for every refinement level.

This approach is expected to obtain greater efficiency when the solutions presents singularities, yielding highly efficient multiresolution representations, see for instance [3].

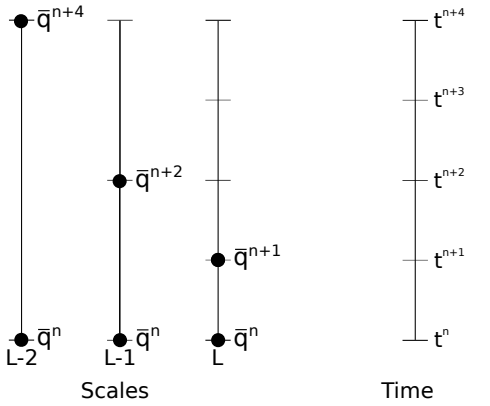

Fig. 1 Configuration of the solution after the time evolution in the LT scheme. Leaves of each refinement level performed a time step proportional to its refinement. The finer scales require a new time evolution to reach the time instant of the next coarser scale.

The scale dependent time stepping of the LT schemes is illustrated in Figure 1. In the LT schemes, not every scale is updated during an iteration of the time evolution. The coarsest scale level to be updated during the iteration n of the time evolution is defined as the minimum scale level in which the modulo operator between n and $2^{L-\ell}$ is zero. In this work, this scale is denoted by ℓ_{min}.

As shown in Figure 1, after a time evolution iteration, the finer scales require a new time evolution to reach the same time instant of the next coarser scale. However, due to the interfaces between leaves of different refinement, the information required to update the finer leaves, in this case the solution of the coarser scale at the proper time instant, is not available. In this work, we propose the use of the NERK schemes to overcome this bottleneck. The polynomial approximation in the time domain obtained by the NERK schemes is used to perform the flux computations when this situation occurs.

The proposed strategy to perform the LT scheme is, for every time evolution iteration, to compute the value of ℓ_{min} for this iteration, then perform the first RK step for every scale greater or equal ℓ_{min}, as given in Section 3. The Runge–Kutta step is performed using the correspondent value of Δt_ℓ for each scale.

Having performed this step, the values $\bar{\mathbf{q}}^*$ obtained are a first order solution at time instant $t^n + \Delta t_\ell$. Using the fluxes obtained for this evolution, the first step of the NERK solution $\bar{\mathbf{q}}^*_{\theta=\frac{1}{2}}$ is easily obtained. Also, a linear interpolation of the solution at the time instant $t^n + \frac{1}{2}\Delta t_\ell$ will be required later in the algorithm for the scales lower than L.

Before performing the second RK step, a tree refreshing procedure is required. This procedure updates the values of the internal nodes with their respective solutions $\bar{\mathbf{q}}^*$. The projected solutions are used to update the solution of the virtual leaves to a time instant compatible with the next RK step.

In order to perform the tree refreshing process, its important to notice that due to the different time steps in each scale, when the solution $\bar{\mathbf{q}}^*$ is projected to a lower scale, this value has a time instant different of the solution in this scale, requiring some interpolations or extrapolation.

In this step, the projection from a finer to a coarser level produces a solution at time instant $t^n + \frac{1}{2}\Delta t_\ell$, this solution is saved and extrapolated to instant $t^n + \Delta t_\ell$, producing a solution used as an approximation for $\bar{\mathbf{q}}^*$. This procedure is repeated recursively until scale ℓ_{min}.

After the projection, the virtual leaves are predicted from the scale ℓ_{min} until the finest scale using the solutions at time instant $t^n + \frac{1}{2}\Delta t_\ell$, which corresponds to the time instant where the flux computations of the leaves in the next finer level will be performed.

Having performed the tree update, the second RK step is level-wise computed from the finest scale to the scale ℓ_{min}. The flux computations are performed in the current scale and the second RK step is executed. However, to perform the flux computations in the next coarser scale, the solution of the

current scale is required in the same time instant.

To obtain this solution, a first order extrapolation is computed using the fluxes obtained during the RK steps. This extrapolation, possible due to the coefficients $a_{11} = b_1 = 1$, is performed as:

$$\bar{\mathbf{q}}_{\ell+1}\left(t^n + 2\Delta t_{\ell+1}\right) = \bar{\mathbf{q}}^*_{\ell+1} + \Delta t_{\ell+1} f\left(t + \Delta t_{\ell+1};\, \bar{\mathbf{q}}^*_{\ell+1}\right). \tag{8}$$

The internal nodes have their solutions extrapolated to this time instant using:

$$\bar{\mathbf{q}}_{\ell+1}\left(t^n + 2\Delta t_{\ell+1}\right) = 2\bar{\mathbf{q}}^*_{\ell+1} - \bar{\mathbf{q}}^n_{\ell+1}. \tag{9}$$

This extrapolated solution is projected onto the next coarser level and the virtual leaves of the current level have their solution updated to perform the flux computations for the next coarser scales. This procedure is repeated recursively until the scale $\ell_{\min}$.

After the second RK step, another tree refreshing procedure is performed to project the second order solution onto the internal nodes. For that, the second order solution of the leaves are projected onto the next coarser scale as a approximation for $\bar{\mathbf{q}}_{\theta=\frac{1}{2}}$ in the coarser scale. Using the NERK equations and the already known values, the solution at the time instant equivalent to the leaves of the same scale is reconstructed.

After the time evolution iteration, the grid adaptation can be performed. However, due to the different time stepping, only the scales to be updated in the next time evolution iteration are regrided.

5 Results and discussions

In this section, the results of the proposed MRLT/NERK method are presented in comparison with the MR, MRLT methods described in [6] and the traditional FV method.

The different schemes are applied to a reaction-diffusion model. This model considers the ignition of the initial spark creating a flame which propagates outwards under adiabatic conditions. The propagation speed is associated with the reaction rate of the flame. This speed is slowed down during time by heat losses due to radiation. The density of the gas and other thermodynamic properties are assumed to be constant. For a detailed description of this model we suggest the work of Bockhorn et al. [11] and references therein. This model is also used as an example of adaptive multiresolution methods applied to simulations of partial differential equations in the work of Roussel et al. [3] and Domingues et al. [6].

In a two-dimensional setting, the equations can be written in the following form,

$$\frac{\partial T}{\partial t} = \nabla^2 T + \omega - s \tag{10a}$$

$$\frac{\partial Y}{\partial t} = \frac{1}{Le}\nabla^2 Y - \omega \tag{10b}$$

where $T(x,y)$ is a dimensionless temperature normalised between 0 (unburned gas) and 1 (burned gas), Y is the dimensionless partial mass of the unburnt gas (in the case of $Le = 1$ and $s = 0$, we have $Y = 1 - T$), the flame velocity is given by $v_f = \int \omega\, dxdy$ and the reaction rate ω is given by:

$$\omega(T,Y) = \frac{Ze^2}{2Le} Y \exp\left[\frac{Ze(T-1)}{1+\tau(T-1)}\right], \tag{11}$$

where the Zeldovich number Ze is a dimensionless activation energy, τ is the burnt-unburnt temperature ratio and the heat loss s due to radiation is $s(T) = \kappa[(T + \tau^{-1} - 1)^4 - (\tau^{-1} - 1)^4]$, where κ is a dimensionless radiation coefficient.

Table 1 Errors, CPU time and memory compression ($t_f = 10.0$, $\varepsilon = 10^{-2}$).

	Method	L^1 Error ($\times 10^{-3}$)			CPU Time	Memory
		T	Y	ω	(%FV)	(%FV)
$L = 8$	MR/RK2	1.625	1.624	5.101	106.0	11.9
	MRLT/RK2	1.629	1.628	5.196	94.5	11.8
	MRLT/NERK2	1.660	1.659	5.194	33.8	11.8
$L = 9$	MR/RK2	0.771	0.770	2.277	46.9	4.8
	MRLT/RK2	0.666	0.665	2.083	41.5	4.8
	MRLT/NERK2	0.765	0.764	2.251	13.7	4.8
$L = 10$	MR/RK2	0.309	0.308	0.650	15.3	1.5
	MRLT/RK2	0.170	0.170	0.463	13.5	1.5
	MRLT/NERK2	0.301	0.300	0.625	4.1	1.5

For the simulations, we use $\kappa = 0.1$. The following initial condition is described using polar coordinates:

$$T(r,0) = \begin{cases} 1, & \text{if } r \le r_0 \\ \exp(1 - \frac{r}{r_0}), & \text{if } r > r_0 \end{cases} \qquad Y(r,0) = \begin{cases} 0, & \text{if } r \le r_0 \\ 1 - \exp[Le(1 - \frac{r}{r_0})], & \text{if } r > r_0, \end{cases} \tag{12}$$

where the initial radius of the flame front is given by $r_0 = 2$ and the value $r = \sqrt{X^2 + Y^2}$, with $X = \frac{1}{2}x\cos(\theta) + y\sin(\theta)$ and $Y = \frac{1}{2}x\sin(\theta) - y\cos(\theta)$.

In these simulations, we use Neumann boundary conditions, a McCormack scheme for the flux computations, $Ze = 10$, $\tau = 0.64$, $Le = 0.3$, $\theta = \frac{\pi}{6}$, a threshold factor $\varepsilon = 0.01$ and a courant number $\sigma = 0.2$ until the time instant $t_f = 10.0$. The errors in the L^1 norm, CPU time and memory usage in relation to the FV methods are summarised in Table 1.

To compute the errors, we use a reference solution computed with the FV method with a third order RK method using a grid with 1024^2 grid points, corresponding to a refinement level of $L = 10$. For the CPU time and memory usage, the methods are compared with the corresponding FV computations with a second order RK method.

The reference solution for this problem and the difference between this reference and the solutions of the analysed methods with their respective final adaptive grids, using $L = 10$, are given in Figure 2.

Considering the experiments performed in this work, the MRLT method given in [6] produced the lowest L^1 errors. However, this method does not present a significant gain in CPU time in comparison to the traditional MR method. In contrast, the proposed MRLT/NERK method presented a better solution in relation to the MR methods in the finest grids. Nevertheless, this method obtained a significant gain in computational time.

6 Conclusion

This work presented a new second order local time-stepping approach using NERK schemes in the context of adaptive multiresolution methods. These schemes are used to obtain the intermediary values and some interpolations required to perform the time evolution with local time stepping in higher orders. This work focused on the implementation of the proposed methodology for second order time evolution. Some numerical results are presented for two-dimensional reaction-diffusion equations. In general, the proposed method obtained a significant gain in CPU time, mainly when finer grids are used. The solution obtained with the proposed methods presented very similar aspects compared with the solutions obtained in MR methods and observed in the error analysis. The MRLT/NERK methods

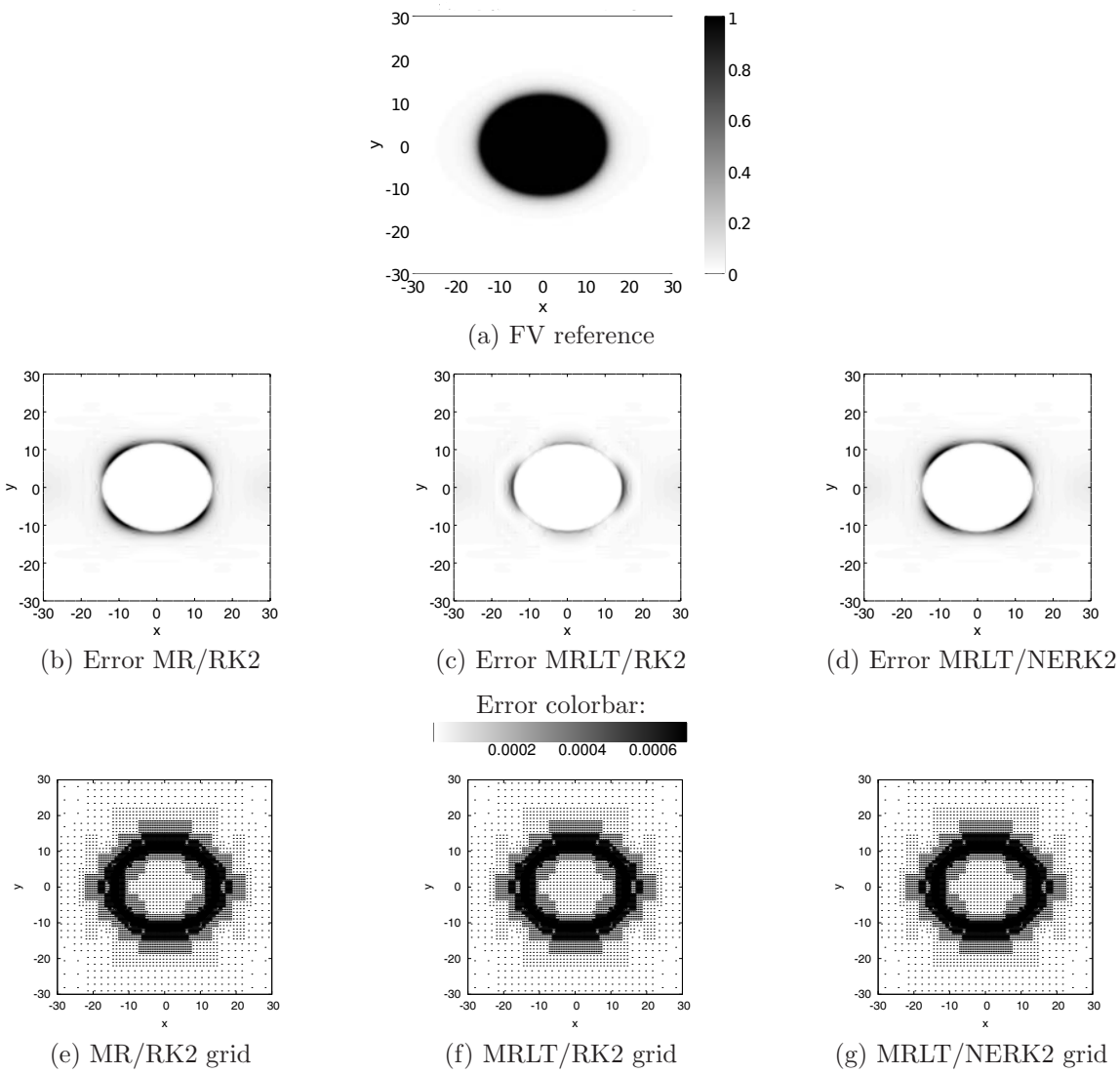

Fig. 2 2D reaction-diffusion equations: FV/RK3 reference solution at $t = 10.0$ for the variable T (a), and the modulus of the difference between this reference and the solutions obtained using the methods MR/RK2 (b), MRLT/RK2 (c), and MRLT/NERK2 (d). The corresponding final adaptive grids are given in (e), (f) and (g) respectively.

showed advantageous over the MR and MRLT approaches due to the significant reduction in CPU time.

Acknowledgements

The authors thank the Brazilian agencies CAPES, CNPq (140626/2014-0, 306038/2015-3, 307083/2017-9), FAPESP (2015/50403-0, 2015/25624-2), FINEP/CT-INFRA (01120527-00) for financial support. The authors are indebted to Prof. C.-D. Munz who motivated the use of NERK method for local time stepping. We thank Dr. Olivier Roussel for developing the original Carmen Code and fruitful scientific discussions. We also thank Eng. V. E. Menconi and Dr. A. K. F. Gomes for their helpful computational assistance. MD thankfully acknowledges financial support from ECM, France. KS acknowledges financial support from the ANR-DFG project AIFIT (Grant 15-CE40-0019), and the Pacific Institute for Mathematical Sciences, Banff, Canada, for hospitality.

References

[1] Deiterding, R. (2011), Block-structured adaptive mesh refinement - theory, implementation and application, *ESAIM: Proceedings*, **34**, 97-150.

[2] Müller, S. (2003), Adaptive multiscale schemes for conservation laws, volume 27 of *Lecture Notes in Computational Science and Engineering*, Springer: Heidelberg.

[3] Roussel O., Schneider K., Tsigulin, A., and Bockhorn, H. (2003), A conservative fully adaptative multiresolution algorithm for parabolic PDEs, *Journal of Computational Physics*, **188**, 493-523.

[4] Harten, A. (1995), Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, *Communications on Pure and Applied Mathematics*, **48**, 1305-1342.

[5] Müller, S. and Striba, Y. (2007), Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, *Journal of Scientific Computing*, **30**(3), 493-531.

[6] Domingues, M.O., Gomes, S.M., Roussel, O., and Schneider, K. (2008), An adaptive multiresolution scheme with local time stepping for evolutionary PDEs, *Journal of Computational Physics*, **227**(8), 3758-3780.

[7] Zennaro, M. (1986), Natural continuous extensions of Runge-Kutta methods, *Mathematics of Computation*, **46**(173), 119-133.

[8] Moreira Lopes, M., Domingues, M.O., Schneider, K., and Mendes, O., High-order local time-stepping for adaptive multiresolution methods, *unpublished*.

[9] Deiterding, R., Domingues, M.O., Gomes, S.M., and Schneider, K. (2016), Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations, *SIAM: Journal of Scientific Computing*, **38**(5), S173-S193.

[10] Owren, B. and Zennaro, M. (1992), Derivation of efficient, continuous, explicit Runge-Kutta methods, *SIAM. Journal on Scientific and Statistical Computing*, **13**(6), 1488-1501.

[11] Bockhorn, H., Fröhlich, J., and Schneider, K. (1999), An adaptive two-dimensional wavelet-vaguelette algorithm for the computation of flame balls, *Combustion Theory and Modelling*, **3**, 177-198.

[12] Domingues, M.O., Gomes, S.M., Roussel, O., and Schneider, K. (2011), Adaptive multiresolution methods, *ESAIM: Proceedings*, **34**, 1-96.

[13] Schneider, K. and Vasilyev, O. (2010), Wavelet methods in computational fluid dynamics, *Annual Review of Fluid Mechanics*, **42**, 473-503.

[14] Vermiglio, R. and Zennaro, M. (1993), Multistep natural continuous extensions of Runge-Kutta methods: the potential for stable interpolation, *Applied Numerical Mathematics*, **12**(6), 521-546.

[15] Winters, A.R and Kopriva, D.A. (2014), High-order local time stepping on moving DG spectral element meshes, *Journal of Scientific Computing*, **58**(1), 176-202.