INPE – National Institute for Space Research
São José dos Campos – SP – Brazil – May 16-20, 2016

A SECOND ORDER ADAPTIVE MULTIRESOLUTION LOCAL TIME SCHEME FOR EVOLUTIONARY PARTIAL DIFFERENTIAL EQUATIONS WITH LOCALISED PHYSICAL PHENOMENA

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

[1]Brazilian Institute for Space Research, São José dos Campos, Brazil, muller.lopes@inpe.br
[2]Brazilian Institute for Space Research, São José dos Campos, Brazil, margarete.domingues@inpe.br
[3]Brazilian Institute for Space Research, São José dos Campos, Brazil, odim.mendes@inpe.br
[4]Laboratoire de Mécanique, Modélisation et Procédés Propres & Centre de Mathématiques et d'Informatique, Aix-Marseille Université, Marseille, France, kschneid@cmi.univ-mrs.fr

Abstract: Techniques for numerical simulations of localised complex physical phenomena may require a lot of CPU time, making the solution unavailable in a reasonable time. To overcome this restriction, adaptive methods are used. They consist in generating an automatic mesh adapted to the solution. To improve their performance when explicit methods are used in time, a second order local time stepping scheme is proposed. It consists in using for each cell an individual time step proportional to its refinement, respecting thus the CFL condition locally. For that, synchronisation steps must be done. Some validation tests are presented, where we show that the CPU time and precision of the solutions are improved compared to standard finite volume and multiresolution schemes.

keywords: Numerical Simulation and Optimisation, Fluid-dynamics, Geophysical Nonlinear Dynamics.

1. INTRODUCTION

In order to reduce the computational time and the memory requirements for simulating evolutionary partial differential equations with localised structures, the adaptive Multiresolution (MR) method is proposed in the context of the Finite Volumes (FV) method [4]. This approach has been successfully applied to solutions which exhibit point-wise singularities, boundary layers, localized shocks, coherent vortexes typically encountered in turbulent flows [5, 7, 8].

The MR method consists in using the fast wavelet transform and a specific refinement algorithm to represent the solution using an adaptive mesh [7]. This adaptive mesh is coarser in smooth regions and finer where structures and steep gradients are present. It can be rebuilt to a regular mesh interactively with an expected error based on the chosen threshold. The computation of the numerical fluxes are the most expensive step of the FV simulation. Therefore, the minimisation of interfaces between cells implies in a faster simulation. The use of adaptive meshes reduces the computational time of the simulation because the number of iterations between cells (numerical fluxes) is reduced.

To improve the performance of MR methods, a local time stepping (LT) approach for MR methods with explicit time schemes is proposed in [1]. This technique consists in performing the time evolution of each cell on the adaptive mesh independently according with its time step requirements. Therefore, each cell must have a time step proportional to its refinement and consequently, a larger cell performs a larger time step. This approach aims to increase the computational performance of MR methods. An extensive review of the use of LT methods for space adaptive discretizations in the context of multiresolution methods is given in [2].

This work follows the idea of Gassner et al. [3], which applied the Natural Continuous Extension for Runge-Kutta

(NERK) schemes, proposed in [9], in the context of discontinuous Galerkin methods. Here, the NERK method is used in the context of MR to perform the synchronisations required for a new second order MR/LT method, named MRLT/NERK method. The proposed method is applied for solving the two-dimensional inviscid Burgers equation. The obtained results are compared successfully with the MR method and the MR/LT approach presented in [1].

2. PURPOSE

The aim of this work is to develop a new class of LT methods in the context of MR methods, using NERK schemes developed in [9]. This allows performing the synchronisations required in the intermediary steps of the Runge-Kutta (RK) methods. And thus higher order local time stepping schemes can constructed.

3. METHODS

3.1. MR methods using finite volumes

Consider the initial value problem in d dimensions completed with appropriate initial and boundary conditions. The quantity vector $\mathbf{Q}$ satisfies the evolution equation written in divergence form,

$$\frac{\partial \mathbf{Q}}{\partial t} = -\nabla \cdot \mathbf{F}(\mathbf{Q}, \nabla \mathbf{Q}) + \mathbf{S}(\mathbf{Q}), \tag{1}$$

where $\nabla \cdot \mathbf{F}(\mathbf{Q}, \nabla \mathbf{Q})$ and $\mathbf{S}(\mathbf{Q})$ denote the divergence and source term, respectively. The problem's domain is given by $(\mathbf{x}, t) \in \Omega \times [0, +\infty)$, $\Omega \subset \mathbb{R}^d$.

The spatial discretization used for Eq. 1 is a classical finite volume formulation written in the standard form, where the computational domain Ω is partitioned into cells $(\Omega_i)_{i\in\Lambda}$, $\Lambda = \{0, \ldots, i_{max}\}$ with $\Omega = \cup_i \Omega_i$.

Defining the volume of the cell by $|\Omega_i| = \int_{\Omega_i} d\mathbf{x}$, the cell-average $\bar{\mathbf{q}}_i(t)$ of a given quantity $\mathbf{Q}$ on Ω_i at time instant t is given by,

$$\bar{\mathbf{q}}_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{Q}(\mathbf{x}, t)\, d\mathbf{x}. \tag{2}$$

Considering the one-dimensional case, Ω_i is an interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ of length $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. Integrating Eq. 1 on Ω_i implies:

$$\frac{d\bar{\mathbf{q}}_i}{dt}(t) = -\frac{1}{\Delta x_i}\left(\bar{\mathbf{F}}_{i+\frac{1}{2}} - \bar{\mathbf{F}}_{i-\frac{1}{2}}\right) + \bar{\mathbf{S}}_i, \tag{3}$$

where $\bar{\mathbf{F}}$ is the numerical flux and $\bar{\mathbf{S}}_i$ is the source term of the cell Ω_i.

The adaptive multiresolution (MR) analysis for cell averages consists in a multilevel decomposition of the cell-averages of the solution on a hierarchy of nested meshes Ω^ℓ, where ℓ is the mesh refinement level. Every Ω^ℓ is a regular FV mesh, with $2^{d\ell}$ cells, where a cell at refinement level ℓ and position i is denoted by Ω_i^ℓ.

The grid generation procedure is mathematically supported by the fact that the wavelet coefficients are small in regions where the solution is smooth, while they are significant in regions where the solution exhibits steep gradients. Therefore, in practice, significant compression rates are expected only when a small number of cell-averages is present on the finest scales.

The data structure for representing the solution is organised as a dynamic graded tree [7]. It implies that every cell neighbour can have at maximum one refinement level difference. This restriction implies the use of virtual leaves for the flux computations between leaves of different refinement levels. The virtual-leaf values are predicted using a procedure to compute the wavelet coefficients. Details about the computation of the numerical fluxes between cells at different levels in MR schemes can be found in [4, 7].

3.2. Runge–Kutta methods

After spatial discretization of the initial value problem given in Equation 1, we obtain a system of ordinary differential equations, which describes the evolution of the cell average for each leaf of the adaptive mesh. These equations can be written in the following form:

$$\frac{d\bar{\mathbf{q}}}{dt} = f(t, \bar{\mathbf{q}}). \tag{4}$$

To evolve the current solution in time, a compact form second order Runge-Kutta (RK) method is applied:

$$\bar{\mathbf{q}}^* = \bar{\mathbf{q}}^n + \Delta t^n\ f(t^n, \bar{\mathbf{q}}^n) \tag{5a}$$

$$\bar{\mathbf{q}}^{n+1} = \frac{1}{2}\bar{\mathbf{q}}^n + \frac{1}{2}\bar{\mathbf{q}}^* + \frac{1}{2}\Delta t^n\, f\left(t^n + \Delta t^n, \bar{\mathbf{q}}^*\right), \tag{5b}$$

where Δt^n is the time–step used to perform the time evolution from the instant t^n to t^{n+1}.

NERK

The NERK scheme, introduced in [9], produces an approximation of the solution in the interval $[t^n; t^n + \Delta t^n]$ using the same coefficients a_{ij} and c_i as in the standard Runge-Kutta methods. The difference between NERK and RK is the use of polynomials β_i instead of the constant coefficients b_i.

The NERK method can be expressed as,

$$\bar{\mathbf{q}}(t^n + \theta\Delta t^n) = \bar{\mathbf{q}}^n + \sum_{i=1}^{s} \beta_i(\theta)\ k_i,\ \theta \in (0, 1], \tag{6}$$

where the polynomials β_i are given as a function of the coefficients b_i of the original RK method. In [6] the proper polynomials β_i can be found, which are used for the NERK method using the presented RK coefficients.

3.3. MR: Local time stepping

The LT approach consists in using an adapted time step for each leaf individually. The time step of each cell is proportional to its spatial size. The CFL condition implies a time step Δt for the most refined level L, and in the LT scheme, the time step used in a cell of level ℓ is:

$$\Delta t_\ell = 2^{L-\ell}\Delta t. \tag{7}$$

Due to the different time stepping of the LT scheme, the solution after the first RK step is in a different time instant for each refinement level. Hence a synchronization procedure is required in time before the flux computations for the second RK step can be executed. Before this step, as in the MR method, a tree refreshing process to project the solution from the leaves to its parent cells is necessary [7]. However, in LT schemes, each scale has a different time instant, requiring a linear extrapolation in order to perform those projections.

The proposed solution to perform the second RK step uses a linear interpolation in time to obtain the solution of coarser leaves in the same time instant as for the finer levels. Once the finest level is evolved, a linear extrapolation is used to obtain an approximation at the time instant of the following coarser scale. This procedure is recursively repeated for every coarser scale to be evolved in this iteration of the time evolution.

After performing the second RK step, another tree refreshing process is required. In this case, the second order solution of the leaves is projected onto its parent cell. However, this solution is in half of the parent cell's time step. To extrapolate this solution to the correct time instant without losing the second order, a relation between Equation 6 with parameters $\theta = \frac{1}{2}$ and 1 are used. The solution obtained with $\theta = \frac{1}{2}$ is also used, in the coarser scales, to compute the numerical fluxes of the first RK step of the finer scales in order to perform its second time evolution, reaching thus the same time instant.

4. RESULTS

In this section we present results of the proposed MRLT/NERK method and compare with the MR, MRLT methods described in [1] and the traditional FV method. The different schemes are applied to the 2D Burgers equation, which is a non-linear PDE that can be considered as a simple model for turbulence. The governing equation reads,

$$\frac{\partial Q}{\partial t} + \frac{1}{2}\left[\frac{\partial Q^2}{\partial x} + \frac{\partial Q^2}{\partial y}\right] = 0, \tag{8}$$

for $(x, y) \in (0, 1) \times (0, 1)$ and $t > 0$. Eq. 8 is completed by the following initial condition:

$$Q(x, y) = \sin(2\pi x)\sin(2\pi y), \tag{9}$$

and the imposed boundary conditions are,

$$Q(x, 0) = Q(x, 1) = Q(0, y) = Q(1, y) = 0. \tag{10}$$

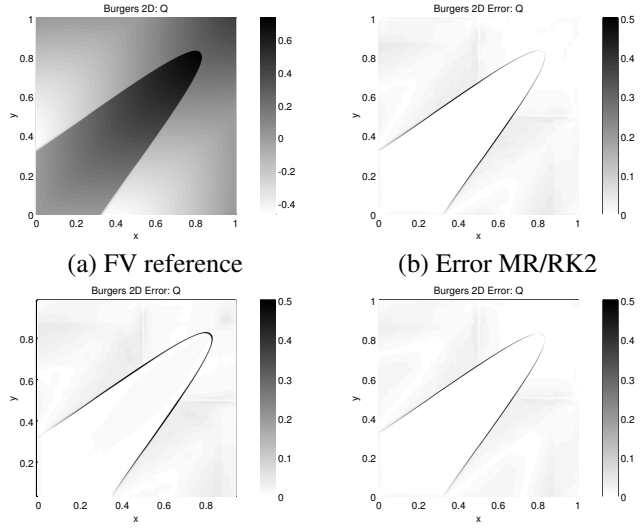
All simulations are performed with CFL $\sigma = 0.5$ and the threshold value $\epsilon = 0.01$ until the time instant $t_f = 0.9$. The errors in the L^1 norm are summarized in Table 1. The solutions computed with the adaptive methods are compared with an FV/RK3 reference solution computed on a uniform grid with $L = 11$ (corresponding to 2048^2 grid points). The CPU time and memory for adaptive computations are compared with the corresponding FV/RK2 computations on a uniform grid.

Table 1 – Errors, CPU time and memory compression

	Method	L^1 Error ($\times 10^{-2}$)	CPU Time (%FV)	Memory (%FV)
$L = 8$	MR/RK2	1.376	68.4	19.5
	MRLT/RK2	1.646	67.8	19.5
	MRLT/NERK2	1.163	40.3	24.2
$L = 9$	MR/RK2	1.241	34.7	8.9
	MRLT/RK2	1.546	32.5	8.9
	MRLT/NERK2	1.019	16.7	10.3
$L = 10$	MR/RK2	1.180	14.7	4.1
	MRLT/RK2	1.495	13.5	4.1
	MRLT/NERK2	0.962	6.7	4.4

Note: All adaptive computations use $\epsilon = 10^{-2}$; final time: $t_f = 0.9$ on an Intel Core$^{TM}i7$ CPU 2.67GHz. FV/RK2 CPU time: 2.0min ($L = 8$); 15.7min ($L = 9$); and 2.3h ($L = 10$).

Figure 1 shows the reference FV/RK3 solution obtained on a fine grid (L=11), and the difference between this reference and the solutions obtained with the methods MR/RK2, MRLT/RK2 and MRLT/NERK2, respectively.

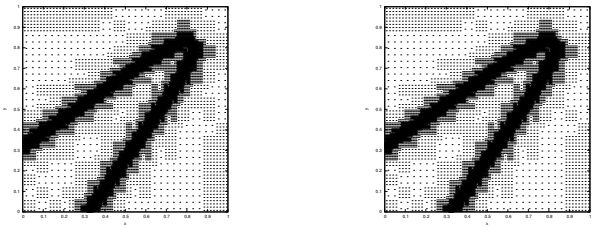


(a) FV reference (b) Error MR/RK2

(c) Error MRLT/RK2 (d) Error MRLT/NERK2

Figure 1 – FV/RK3 reference solution at $t = 0.9$ for the 2D Burgers equation (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 adaptive meshes of the three methods are very similar, i.e. locally refined around the shock, and Figure 2 shows them for MR/RK2 and MRLT/NERK.



(a) MR/RK2 adaptive mesh (b) MRLT/NERK2 adaptive mesh

Figure 2 – Adaptive meshes obtained with MR/RK2 (a) and MRLT/NERK2 (b) methods

5. DISCUSSION

In all MR cases the errors presented in Table 1 are comparable, nevertheless, the MRLT/NERK2 exhibits smaller L_1 errors than MR and specially the MRLT/RK2 case. We can also observe in Figure 1 that the maximum differences occur in the propagation front, where the MRLT/NERK2 case yields a better result than MRLT/RK2. The proposed methods achieve a gain of about 50% in CPU time in relation to the MR [7] and MRLT [1] methods with a better precision than the MR technique for $L = 9$ and 10. For $L = 8$, due to the higher number of used cells on the adaptive mesh, the gain is 41%, but still expresses an improvement in the results compared with the other MR techniques. In all cases the memory increases for the MRLT/NERK2, however this becomes less pronounced as L increases.

6. CONCLUSION

The presented results show that with a small of amount of additional memory requirements the proposed method yields an improvement in CPU time and precision compared to traditional MR methods we tested here. Hence we can recommend this local time stepping approach MRLT/NERK2 for future studies where its properties will be investigated in detail.

ACKNOWLEDGMENTS

The authors would like to thank CAPES for the master scholarship and CNPq for the scholarship of MMSL (project $140626/2014-0$), moreover we thank the Brazilian agencies CNPq (project $306038/2015-3$) and FAPESP for financial support. We also thank V. Menconi and A. K. F. Gomes for his helpful computational assistance, and O. Roussel for providing the original Carmen code.

References

[1] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. J. Comput. Phys., 227(8):3758-3780, 2008.

[2] M. O. Domingues, S. M. Gomes, O. Roussel and K. Schneider. Adaptive multiresolution methods. ESAIM: Proc., Vol. 34, p. 1-96, 2011.

[3] G. J. Gassner, F. Hindenlang, and C.-D. Munz. A Runge-Kutta based discontinuous Galerkin method with time accurate local time stepping. Adv. Comput. Fluid Dyn., 2:95-118, 2011.

[4] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Commun. Pur. Appl. Math., 48:1305-1342, 1995.

[5] S. Müller. Adaptive multiscale schemes for conservation laws, volume 27 of Lect. Notes Comput. Sci. Eng. Springer, Heidelberg, 2003.

[6] B. Owren and M. Zennaro. Derivation of efficient, continuous, explicit Runge-Kutta methods. SIAM. J. Sci. Stat. Comp., 13(6):1488-1501, 1992.

[7] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptative multiresolution algorithm for parabolic PDEs. J. Comput. Phys., 188:493-523, 2003.

[8] K. Schneider and O. Vasilyev. Wavelet methods in computational fluid dynamics. Annu. Rev. Fluid Mech., 42, 473-503, 2010.

[9] M. Zennaro. Natural continuous extensions of Runge-Kutta methods. Math. Comput., 46(173):119-133, 1986.