Pavel Kůs; Vít Dolejší

Solution of time-dependent convection-diffusion equations with the aid of higher order adaptive methods with respect to space and time

In: Jan Chleboun and Karel Segeth and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Prague, May 28-31, 2006. Institute of Mathematics AS CR, Prague, 2006. pp. 184–189.

Persistent URL: http://dml.cz/dmlcz/702835

SOLUTION OF TIME-DEPENDENT CONVECTION-DIFFUSION EQUATIONS WITH THE AID OF HIGHER ORDER ADAPTIVE METHODS WITH RESPECT TO SPACE AND TIME*

Pavel Kůs, Vít Dolejší

1. Introduction

This work deals with the solution of a scalar nonlinear convection–diffusion equation which is a model problem for a numerical simulation of viscous compressible flows. A semi-discretization with respect to the space coordinates, which is carried out with the aid of the discontinuous Galerkin method, yields a system of ordinary differential equations (ODE). Our aim is to develop and implement an efficient adaptive numerical scheme for the solution of this ODE system. We derive two stable multi-step methods of the same order of accuracy and from a difference of both approximate solutions, we estimate a local discretization error with respect to the time. Then we choose the time step in such a way, that local error is approximately equal to a given tolerance. Several numerical simulations were carried out to demonstrate the efficiency of the method.

2. Discontinuous Galerkin method

We consider the following unsteady nonlinear convection–diffusion problem: Find $u : Q_T \to I\!R$ such that

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = \varepsilon\, \Delta u \ + g \quad \text{in } Q_T, \tag{1}$$

$$u\big|_{\partial\Omega\times(0,T)} = u_D, \tag{2}$$

$$u(x,0) = u^0(x), \quad x \in \Omega. \tag{3}$$

Similarly as in the finite element method, we introduce a weak solution u of the problem

$$\frac{\mathrm{d}}{\mathrm{d}t}(u(t),v) + b(u(t),v) + a(u(t),v) = (g(t),v) \qquad \forall v \in H_0^1(\Omega), \tag{4}$$

*This work is a part of the research project MSM 0021620839 financed by the Ministry of Education of the Czech Republic and was partly supported by the Grant No. 316/2006/B-MAT/MFF of the Grant Agency of Charles University Prague.

where $(\cdot,\cdot)$ denotes the L^2-scalar product, $a(\cdot,\cdot)$ is a linear form representing the diffusive term and $b(\cdot,\cdot)$ is a nonlinear form representing the convective term. We also consider appropriate representation of initial and boundary conditions. As in the classical finite element method we use triangulation of domain Ω and a piecewise polynomial discontinuous approximation. More general, even non-convex elements with the hanging nodes are allowed. The approximate solution is sought in a space of piecewise polynomial but discontinuous functions S_h. In order to replace the inter-element continuity, we add some stabilization terms into formulation of a discrete problem. The convective term is approximated with the aid of a numerical flux, known from the finite volume method. We receive the space semidiscretization

$$\left(\frac{\partial u_h(t)}{\partial t}, \varphi_h\right) + b_h(u_h(t), \varphi_h) + a_h(u_h(t), \varphi_h) = 0 \quad \forall\, \varphi_h \in S_h, \tag{5}$$

where $a_h(\cdot,\cdot)$ and $b_h(\cdot,\cdot)$ are the discrete variants of the forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$, respectively. For more details, see [1], [2]. The relation (5) represents a system of ordinary differential equations, which must be solved by a suitable method.

3. BDF2 method

The system (5) is stiff, so we have to use an implicit method, such as backward difference formulae (BDF). In contrast to [4] where a combination of explicit and implicit schemes was employed we introduce two implicit schemes of the same order of accuracy. Using this pair of methods, we obtain two solutions and from their difference we estimate the local discretization error.

3.1. Derivation of the method

Now we shall briefly describe derivation of two n-step methods BDF2a and BDF2b for solution of a system of ordinary differential equations with an unknown function $y : (0,T) \to I\!R^m$.

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = F(t,y), \qquad y(0) = y^0\,, \tag{6}$$

where $y^0 \in I\!R^m$ and $F : (0,T) \times I\!R^m \to I\!R^m$. Let us denote by $0 = t_0 < t_1 < t_2 < \cdots < t_r = T$ the partition of the interval $(0,T)$, $\tau_k \equiv t_k - t_{k-1},\ k = 1,\dots,r$, $\theta_k = \tau_k/\tau_{k-1},\ k = 1,\dots,r$. Moreover, let y_k denote approximate value of solution $y(t_k),\ k = 0,\dots,r$.

First method is derived from the Taylor formula in t_k. We express values of solution in t_{k-1},…,t_{k-n}. When we neglect higher order terms, we obtain a system of n equations with unknown approximate solutions y_k,…,y_{k-n} and derivatives $y'(t_k)$,…,$y^{(n)}(t_k)$. By eliminating higher order derivatives we obtain method BDF2a :

$$\sum_{i=0}^{n} \alpha_i y_{n-i} = \tau_k F_k \tag{7}$$

The second method can be derived similarly from the Taylor formula in t_{k-1}

$$\sum_{i=0}^{n} \bar{\alpha}_i \bar{y}_{n-i} = \tau_k F_{k-1}. \tag{8}$$

This method is explicit and therefore not suitable for the solution of stiff problems. So we define the method BDF2b as a linear combination of schemes (7) and (8) by

$$\sum_{i=0}^{n} \hat{\alpha}_i \hat{y}_{n-i} = \hat{\gamma}_0 \tau_k F_k + \hat{\gamma}_1 \tau_k F_{k-1}. \tag{9}$$

3.2. Error estimation

From the Taylor formula we also get an estimation of the local discretization error for the BDF2a and BDF2b methods in the form

$$\begin{aligned} e_k \equiv y(t_k) - y_k &\approx f_1(\tau_k, \ldots, \tau_{k-n+1}) y^{(n+1)}(t_k), \\ \hat{e}_k \equiv y(t_k) - \hat{y}_k &\approx f_2(\tau_k, \ldots, \tau_{k-n+1}) y^{(n+1)}(t_{k-1}). \end{aligned} \tag{10}$$

Now let us assume that $y^{n+1}(t_k) \approx y^{n+1}(t_{k-1})$. From (10) we eliminate the term $y^{n+1}(\cdot)$ and after substituton we obtain a computable expression for the local discretization error depending on both approximate solutions only. Therefore we have

$$e_k \approx \delta(y_k - \hat{y}_k), \tag{11}$$

$$\hat{e}_k \approx \hat{\delta}(y_k - \hat{y}_k). \tag{12}$$

We can also combine our two solutions to obtain final solution of a higher order of accuracy by

$$\breve{y}_k = \hat{\delta} y_k - \delta \hat{y}_k, \tag{13}$$

whose order of convergence is equal to $n + 1$. In [3], we computed coefficients for $n = 1, 2, 3$ and verified stability of the proposed methods.

4. Full space–time discretization

By a direct application of an implicit method to the semi-discrete problem (5), we obtain a system of nonlinear algebraic equations at each time step, which is expensive to solve. Therefore we use a semi–implicit approach, where the linear terms are treated implicitly, whereas the nonlinear ones explicitly. For the nonlinear terms we employ an explicit higher order extrapolation. Then we obtain the scheme

$$\begin{aligned} \frac{1}{\tau_k} \left(\sum_{l=0}^{n} \alpha_l u_h^{k-l}, v_h \right) + \gamma_0 a_h(u_h^k, v_h) + \gamma_0 b_h \left(\sum_{l=1}^{n} \beta_l u_h^{k-l}, v_h \right) \\ + \gamma_1 a_h(u_h^{k-1}, v_h) + \gamma_1 b_h(u_h^{k-1}, v_h) = 0 \quad \forall\, v_h \in S_h . \end{aligned} \tag{14}$$

5. Adaptive choice of time step

An important feature of modern numerical algorithms is the adaptivity, i.e., their ability to estimate the local discretization error during execution and adapt a time step in such a way, that the local discretization error is under a given tolerance. Thus, at each time step, we estimate the local discretization error and on the basis of this estimation we choose the next time step. In order to ensure an efficiency of the method the local discretization error at each time step should be approximately equal to the given tolerance TOL. Let us denote by EST the estimate of the local error. Since the order of convergence of the method is equal to $n+1$, we have

$$\mathrm{EST} = C\tau_k^{n+1}. \tag{15}$$

We want to find a time step $\bar{\tau}_k$ such that

$$\mathrm{TOL} = C\bar{\tau}_k^{n+1}. \tag{16}$$

Therefore we define the next time step by

$$\bar{\tau}_k = \tau_k \sqrt[n+1]{\frac{\mathrm{TOL}}{\mathrm{EST}}}. \tag{17}$$

If EST is much larger than TOL, we reject the last time step and compute it again using $\bar{\tau}_k$ instead of τ_k. Otherwise we accept the last time step and compute the next one using $\tau_{k+1} := \bar{\tau}_k$.

6. Numerical results

6.1. Orders of convergence

We investigate the experimental orders of convergence of the presented numerical schemes. We carried out numerical experiments for an ordinary differential equation having the exact solution in the form

$$y = \frac{\mathrm{e}^{\alpha t} - 1}{\mathrm{e}^{\alpha} - 1} \tag{18}$$

on interval $t \in [0,1]$ with $\alpha = 500$. The following table contains the computational errors for the one, two, and three-step BDF and the corresponding orders of convergence.

n	10^{-2}	10^{-3}	10^{-4}	10^{-5}	order
1	1.53×10^{0}	2.07×10^{-2}	2.08×10^{-4}	2.08×10^{-6}	1.96
2	1.04×10^{0}	3.20×10^{-3}	3.45×10^{-6}	3.47×10^{-9}	2.84
3	8.06×10^{-1}	6.31×10^{-4}	7.65×10^{-8}	1.23×10^{-11}	3.64

We observe the order of convergence $n+1$ since we used a combination of two different methods of order n. However, this procedure can not be used in case of the scalar convection-diffusion equation. Not only we do not obtain more accurate solution, but combination of two solutions of order n is even worse. So we have to use one of our two methods of order n.

Further we consider the scalar convection–diffusion equation (1) with the exact solution

$$\bar{u} = x(1-x)y(1-y)\frac{e^{\alpha t}-1}{e^{\alpha}-1} \tag{19}$$

on $[0,1]\times[0,1]$ and time interval $t\in[0,1]$. The computational errors and the order of convergence are shown in the following table.

	10^{-1}	5×10^{-2}	10^{-2}	5×10^{-3}	order
$n=1$	3.18×10^{-1}	1.48×10^{-1}	2.74×10^{-2}	1.34×10^{-2}	1.05
$n=2$	1.14×10^{-1}	3.49×10^{-2}	1.42×10^{-3}	2.63×10^{-4}	2.02
$n=3$	8.15×10^{-2}	1.05×10^{-2}	2.58×10^{-4}	9.31×10^{-5}	2.28

We observe, that the numerical order of convergence in this case is approximately n and it corresponds to the expected one. However, for the case $n=3$, the order is 2.28 only, which is caused by the fact that the solution depends on both time and space discretization and its order of convergence is $O(h^p+\tau^n)$. Hence, if τ^n is so small that h^p has nonnegligible influence then further increase of order of accuracy in time has no effect.

6.2. Efficiency of the adaptive strategy

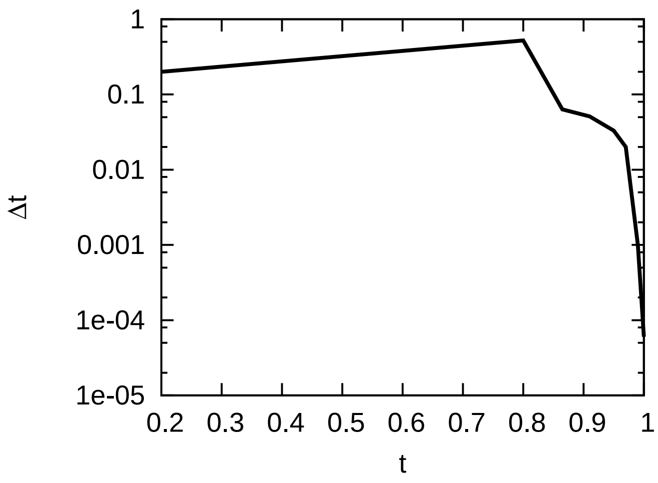
In this section we compare the efficiency of the methods using a constant and adaptive time step. We compared how many time steps are needed to obtain solution with prescribed accuracy.

6.2.1. Ordinary differential equations

First we carried out experiments for the ordinary differential equation with the exact solution (18). The following table shows the numbers of time steps necessary to obtain solution with errors 10^{-2} to 10^{-6}.

		10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
	$n=1$	1375	4474	14365	45790	143641
constant	$n=2$	642	1425	3197	6972	15110
	$n=3$	410	855	1586	2879	5222
	$n=1$	34	81	241	965	2520
adaptive	$n=2$	26	36	65	145	266
	$n=3$	24	29	43	70	108

We observe that the adaptive method is more effective. The differential equation is chosen in such a way, that the exact solution is almost constant in the major part of the interval. However, at the end of the interval the solution grows very steeply. So the major part of the interval can be done with few steps, which adaptive method allows. The following figure shows the lengths of time steps with respect to the time.



We observe that the time step is quite long at the beginning of the interval, while at the end it is shortening rapidly.

6.2.2. Scalar convection–diffusion equations

Further we consider the scalar equation (1) with the exact solution in the form (19). The numbers of iterations, which are needed to obtain solution with errors 10^{-1}, 10^{-2} and 10^{-3}, are in the following table, which verifies the efficiency of the adaptive strategy.

		10^{-1}	10^{-2}	10^{-3}
constant	$n = 1$	7	98	> 10000
	$n = 2$	5	27	> 10000
	$n = 3$	4	18	8973
adaptive	$n = 1$	9	29	1335
	$n = 2$	6	11	650
	$n = 3$	5	9	321

References

[1] V. Dolejší and M. Feistauer: *Error estimates of the discontinuous Galerkin method for nonlinear nonstationary convection-diffusion problems.* Numer. Funct. Anal. Optim. **26**, 2005, 2709–2733.

[2] V. Dolejší, M. Feistauer, and V. Sobotíková: *A discontinuous Galerkin method for nonlinear convection–diffusion problems.* Comput. Methods Appl. Mech. Engrg., **19**, 2005, 2709–2733.

[3] P. Kůs: *Solution of convection–diffusion equations with adaptive methods of higher order in space and time.* Master's thesis, Charles University Prague, Faculty of Mathematics and Physics, 2006, (in Czech).

[4] P. Lötstedt, S. Söderberg, A. Ramage, and L. Hemmingsson-Frändén: *Implicit solution of hyperbolic equations with space-time adaptivity.* BIT **42**, 2002, 134–158.