1991

P.J. van der Houwen, B.P. Sommeijer

Parallel solution of the Burgers equation

Department of Numerical Mathematics Report NM-R9104 February

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

Parallel Solution of the Burgers Equation

P.J. van der Houwen & B.P. Sommeijer

Centre for Mathematics and Computer Science
Post box 4079, 1009 AB Amsterdam, The Netherlands

When applying the method of lines to partial differential equations and using explicit methods for the time integration, the time step is usually severely restricted by stability conditions. In this paper, we focus on the Burgers equation and we try to relax the time step condition by applying fractional step (or operator splitting) methods based on Runge-Kutta methods. Furthermore, we consider parallel versions with increased order of accuracy.

1980 Mathematics Subject Classification: 65M10, 65M20
1982 CR Categories: 5.17
Key Words and Phrases: Parallelism, Burgers' equation, stability, operator splitting, fractional step method.

1. Introduction

We shall be concerned with the initial-value problem for systems of first-order differential equations of the form

$$(1.1)\qquad y'(t) = f(t, y(t)) := \sum_{j=1}^{k} f_j(t, y(t)),$$

where the Jacobian matrices of the functions f_j have different types of eigenvalue spectra. Such systems can arise if the method of lines is applied to time-dependent partial differential equations (PDEs). In particular, we shall consider the extensively studied Burgers equation

$$(1.2)\qquad \frac{\partial u}{\partial t} = \varepsilon \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x},$$

which is of convection-diffusion type. This equation, first discussed by Burgers [2], models, in first approximation, the phenomenon of turbulence and can also be considered as a simplified form of the Navier-Stokes equations (see e.g. [1]).

The spatial discretization of (1.2) gives rise to systems of the form (1.1) with k=2 and where f_1 and f_2 correspond to the convection and diffusion terms, respectively. Systems of this type require integration methods with large stability regions, preferably A-stable methods. However, in the case where (1.1) originates from two- or three-dimensional partial differential equations, the use of A-stable methods, which are necessarily implicit, leads to a huge linear algebra problem associated with the solution of the implicit relations, and consequently, to increased computational complexity. On

Report NM-R9104
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

other hand, when using explicit methods, stability will dictate the time step, which is usually much smaller than the accuracy of the numerical approximation requires. If the diffusion term in a convection-diffusion problem is negligible (i.e., the spectrum of the Jacobian matrix $\partial f/\partial y$ is purely imaginary), then in the class of explicit Runge-Kutta (RK) methods, methods are available with a relatively large *imaginary* stability boundary, that are suitable for integrating (hyperbolic) convection equations. However, if diffusion enters, then such hyperbolic time integrators may impose severe time step restrictions, because of their relatively small real stability boundaries.

In this paper, we study fractional step methods based on explicit RK methods (FRK methods) which are suitable for the time integration of convection-diffusion equations. They are related to the fractional step (or operator splitting) methods of e.g. Yanenko [10], Marchuk [6] and Swayne [8]. Usually, methods based on fractional steps are only of first order. We investigate whether the order of FRK methods can be increased, without loosing stability, by forming linear combinations of various approximations that are computed concurrently.

The analysis will be presented for the general form (1.1) but in the numerical experiments (Section 4) we return to the special case of the Burgers equation (1.2).

2. FRK methods

Consider the initial-value problem

(2.1) $$y'(t) = f(t, y(t)), \quad y(t_n) = y_n, \quad t_n \le t \le t_{n+1} := t_n + h,$$

and the RK methods generated by the Butcher arrays

(2.2) $$\begin{array}{c|c} \mathbf{c}_j & A_j \\ \hline & \mathbf{b}_j^T \end{array}, \qquad j = 1, \dots, k,$$

or equivalently

(2.2') $$y_{n+1} = y_n + h\,\Phi_j(f, \mathbf{e}t_n+\mathbf{c}_jh, y_n), \quad j = 1, \dots, k,$$

where Φ_j is the increment function of the RK method. Ignoring tensor products, this function is defined by

(2.2") $$\Phi_j(f, \mathbf{e}t_n+\mathbf{c}_jh, y_n) = \mathbf{b}_j^T f(\mathbf{e}t_n+\mathbf{c}_jh, \mathbf{Y}), \qquad \mathbf{Y} = y_n\mathbf{e} + hA_jf(\mathbf{e}t_n+\mathbf{c}_jh, \mathbf{Y}).$$

By expanding the increment function Φ_j we obtain

(2.3) $$\Phi_j(f, \mathbf{e}t+\mathbf{c}_jh, y) = \mathbf{b}_j^T\mathbf{e}\, f(t, y) + \mathbf{b}_j^T\mathbf{c}_j\, h\, g(t, y) + \mathbf{b}_j^TA_j\mathbf{e}\, h\, J(t, y)\, f(t, y) + O(h^2),$$

where

$$g(t, y) := \frac{\partial f}{\partial t}(t, y), \quad J(t, y) := \frac{\partial f}{\partial y}(t, y).$$

For the order p of the RK methods (2.2) we have that (cf. [4]):

(2.4) $\quad p = 1$ if $\mathbf{b}_j{}^T\mathbf{e} = 1$; $\quad p = 2$ if, in addition, $\mathbf{b}_j{}^T\mathbf{c}_j = \mathbf{b}_j{}^T A_j\mathbf{e} = 1/2$.

We define FRK methods employing k fractional steps for the k terms occurring in (1.1) by the formula

(2.5) $\quad y^{(0)} = y_n; \quad y^{(j)} = y^{(j-1)} + h\,\Phi_j(f_j, \mathbf{e}t_n{+}\mathbf{c}_jh, y^{(j-1)}),\ \ j = 1, \ldots, k; \quad y_{n+1} = y^{(k)}$

(briefly, *k-term FRK method*). Given the increment functions Φ_j (i.e., the arrays A_j and $\mathbf{b}_j$), we can specify various FRK methods by prescribing the vectors $\{\mathbf{c}_j\}$. We shall consider three types of FRK methods:

Table 2.1. FRK methods (2.5)

Method	Vectors $\mathbf{c}_j$
Back step method:	$\mathbf{c}_j := A_j\mathbf{e},\ j = 1, \ldots, k$
Zero step method:	$\mathbf{c}_1 := A_1\mathbf{e},\ \mathbf{c}_j := \mathbf{e},\ j = 2, \ldots, k$
Forward step method:	$\mathbf{c}_j := (j - 1)\mathbf{e} + A_j\mathbf{e},\ j = 1, \ldots, k$

Notice that the first fractional step in the three families of FRK methods is identical. Furthermore, if the righthand side functions in (1.1) do not explicitly depend on t, then these methods collaps to one and the same method.

The Back step method successively integrates the 'fractional' equations

$$y'(t) = f_j(t, y(t)), \quad j = 1, \ldots, k$$

from t_n to t_{n+1} using the preceding result as initial value. A disadvantage of these Back step methods arises if the equation (1.1) explicitly contains the time variable t. In such cases, they may suffer a drop of accuracy. This is caused by a possible lack of consistency of the solution values and the time-dependent terms in the equation. If all time-dependent terms can be collected into the first function f_1, then there is, in general, no loss of accuracy. This can be explained by observing that in such cases the t-dependency only plays a role in the first fractional step where $y^{(1)}$ is computed. Since in this first step, integration is performed from t_n until t_{n+1} starting with the initial value y_n, no reduction of accuracy is to be expected. However, if not all t-dependent terms can be stored in f_1 (for instance, if the boundary conditions depend on t), then the second and following fractional steps integrate again and again from t_n until t_{n+1} with initial values $y^{(j)}$ that are consistent at t_{n+1}, where they should be consistent at t_n.

The second and next fractional steps of the Zero step method sets the time variable in the righthand side evaluations employed by the RK methods equal to t_{n+1} which results in rather unconventional RK methods. However, the advantage is that possible time-dependent terms occurring in these steps are tuned to the initial value $y^{(j)}$.

The Forward step method integrates from t_n to t_{n+k} using different righthand side functions in each subinterval of length h. The advantage of this Forward step method lies in the fact that all inconsistencies of t and y values are avoided. A disadvantage of this approach is that, in the step from t_n to t_{n+1}, time-dependent terms are evaluated beyond the point t_{n+1}.

2.1. Accuracy of two-term FRK methods

We consider the accuracy of FRK methods in more details for k=2, as is the case for the Burgers equation (1.2). First, we observe that second-order accuracy of the FRK method requires that y_{n+1} satisfies the condition

$$(2.6)\qquad y_{n+1} = y_n + hf + \tfrac{1}{2}h^2\,[g + J_2f_1 + J_1f_2 + J_1f_1 + J_2f_2] + O(h^3),$$

where all functions are evaluated at $t=t_n$ and $y=y_n$ (in order to abbreviate the formulas, the arguments of the functions f, f_j, g, g_j, J and J_j will be omitted if they equal (t_n,y_n)).

For k=2, the FRK method (2.5) reduces to the simple scheme

$$(2.7)\qquad y_{n+1} = y^{(1)} + h\,\Phi_2(f_2, \mathbf{e}t_n{+}\mathbf{c}_2h, y^{(1)}),\quad y^{(1)} := y_n + h\,\Phi_1(f_1, \mathbf{e}t_n{+}\mathbf{c}_1h, y_n).$$

From (2.3) we deduce for (2.7)

$$\begin{aligned}
y^{(1)} &= y_n + \mathbf{b}_1{}^T\mathbf{e}\, hf_1 + \mathbf{b}_1{}^T\mathbf{c}_1\, h^2\, g_1 + \mathbf{b}_1{}^T\mathrm{A}_1\mathbf{e}\, h^2\, J_1\, f_1 + O(h^3),\\
y_{n+1} &= y^{(1)} + \mathbf{b}_2{}^T\mathbf{e}\, hf_2(t_n, y^{(1)}) + \mathbf{b}_2{}^T\mathbf{c}_2\, h^2\, g_2(t_n, y^{(1)}) + \mathbf{b}_2{}^T\mathrm{A}_2\mathbf{e}\, h^2\, J_2(t_n, y^{(1)})\, f_2(t_n, y^{(1)}) + O(h^3)\\
&= y_n + \mathbf{b}_1{}^T\mathbf{e}\, hf_1 + \mathbf{b}_1{}^T\mathbf{c}_1\, h^2\, g_1 + \mathbf{b}_1{}^T\mathrm{A}_1\mathbf{e}\, h^2\, J_1f_1\\
&\qquad + \mathbf{b}_2{}^T\mathbf{e}\, hf_2 + \mathbf{b}_2{}^T\mathbf{e}\; \mathbf{b}_1{}^T\mathbf{e}\, h^2\, J_2f_1 + \mathbf{b}_2{}^T\mathbf{c}_2\, h^2\, g_2 + \mathbf{b}_2{}^T\mathrm{A}_2\mathbf{e}\, h^2J_2f_2 + O(h^3).
\end{aligned}$$

Assuming that the generating RK methods are at least first-order accurate (i.e., $\mathbf{b}_1{}^T\mathbf{e}=\mathbf{b}_2{}^T\mathbf{e}=1$), we may write

$$y_{n+1} = y_n + hf + h^2\,[\mathbf{b}_1{}^T\mathbf{c}_1\, g_1 + \mathbf{b}_2{}^T\mathbf{c}_2\, g_2 + J_2f_1 + \mathbf{b}_1{}^T\mathrm{A}_1\mathbf{e}\, J_1f_1 + \mathbf{b}_2{}^T\mathrm{A}_2\mathbf{e}\, J_2f_2] + O(h^3).$$

Using (2.6) we obtain for the local error

$$\begin{aligned}
(2.8)\qquad \tau(h) = \tfrac{1}{2}h^2\big[&(2\mathbf{b}_1{}^T\mathbf{c}_1 - 1)g_1 + (2\mathbf{b}_2{}^T\mathbf{c}_2 - 1)g_2 + (2\mathbf{b}_1{}^T\mathrm{A}_1\mathbf{e} - 1)J_1f_1 + (2\mathbf{b}_2{}^T\mathrm{A}_2\mathrm{e} - 1)J_2f_2\\
&+ J_2f_1 - J_1f_2\big] + O(h^3).
\end{aligned}$$

Thus, we can formulate:

Theorem 2.1. Let the generating RK methods be at least first-order accurate, i.e.,

$$(2.9a)\qquad \mathbf{b}_1{}^T\mathbf{e} = \mathbf{b}_2{}^T\mathbf{e} = 1.$$

Then the FRK method (2.7) has order one and its local error is given by (2.8). □

We conclude that in general second-order accuracy is not possible, irrespective the orders of the generating RK methods. However, if we impose the additional conditions

$$(2.9b)\qquad \mathbf{c}_1 = A_1\mathbf{e},\quad 2\mathbf{b}_1{}^TA_1\mathbf{e} = 2\mathbf{b}_2{}^TA_2\mathbf{e} = 1,$$

then the local error of (2.7) is given by

$$(2.8')\qquad \tau(h) = \tfrac{1}{2}h^2\big[(2\mathbf{b}_2{}^T\mathbf{c}_2 - 1)g_2 + J_2f_1 - J_1f_2\big] + O(h^3),$$

so that we achieve second-order accuracy in problems where

$$g_2 = 0,\quad J_2f_1 = J_1f_2.$$

In our experiments with Burgers' equation, it turned out that this error is rather small, so that in most cases a second-order behaviour was shown. This can be explained by observing that the magnitude of the term f_j and its Jacobian J_j corresponding to the diffusion term is considerably larger than for the convection term. Hence, by removing the term J_1f_1 from (2.8) reduces the local error sufficiently to get a second-order behaviour, in spite of the cross terms J_1f_2 and J_2f_1 (see the Subsections 4.3.1 and 4.3.2 for a numerical illustration).

2.2. Stability of k-term FRK methods

We consider linear stability with respect to the test equation

$$(2.10)\qquad y'(t) = \sum_{j=1}^{k} J_jy(t).$$

Theorem 2.2. Let the RK methods (2.2) have stability regions S_j. The FRK method (2.5) is stable (at the point t_n) with respect to the test equation (2.10) if, for j=1,...,k, the matrices hJ_j share the same eigensystem and if their eigenvalues lie in S_j.

Proof. Applying the FRK methods to the linear test equation (2.10) leads to the recursion

$$(2.11)\qquad y^{(0)} = y_n;\quad y^{(j)} = R_j(hJ_j)y^{(j-1)},\ j = 1, \dots , k;\quad y_{n+1} = y^{(k)},$$

where the R_j denote the stability functions of the RK methods (2.2). Thus, we obtain

$$(2.12)\qquad y_{n+1} = R(hJ_1,\dots,hJ_k)y_n,\qquad R(hJ_1,\dots,hJ_k) := \prod_{j=1}^{k} R_j(hJ_j)\ .$$

Let the eigenvalues of hJ_j be in S_j. Since, by assumption, the matrices hJ_j have identical eigensystems with eigenvalues within S_j, the matrices $R_j(hJ_j)$ have eigenvalues within the unit circle. Hence, the stability function R of the FRK method has its eigenvalues also within the unit circle. □

In practice, the above condition on the eigensystems of the Jacobian matrices J_j is applied to the eigensystems of the Jacobian matrices of the functions f_j. Such a condition is rather severe and can only be satisfied in a local sense by applying a normal mode analysis based on the same set of complex exponentials as eigenfunctions of the individual Jacobians $\partial f_j/\partial y$. However, in many PDE applications such a local analysis appears to be satisfactory, so that we may conclude that the Theorems 2.1 and 2.2 justify the use of FRK methods as time integration methods for (1.1).

3. Parallel FRK methods

In this section, we discuss the possibility to employ parallel computers. A straightforward way to use parallelism in a fractional step method is to concurrently apply appropriate RK methods to each of the 'fractional differential equations' $y'(t) = f_j(t, y(t))$, $j=1,\dots,k$ and to form a linear combination of these results, that is, to apply the method

$$y_{n+1} = \sum_{j=1}^{k} a_j y^{(j)}, \qquad y^{(j)} = y_n + h_j \Phi_j(f_j, \mathbf{e}t_n+\mathbf{c}_j h_j, y_n), \quad j = 1, \dots, k, \qquad \sum_{j=1}^{k} a_j = 1 .$$

The sequential costs of this method correspond to the computational costs of the most expensive increment function, provided that k processors are available. Moreover, the coefficients a_j can be selected such that we obtain a good load balancing of the computational jobs assigned to the various processors. In this way, parallelism is used to reduce the costs of FRK methods. However, a serious disadvantage of this cheap and simple approach is that the order of the resulting approximation y_{n+1} cannot be increased beyond one.

However, by parallel application of FRK methods, it is possible to exploit parallelism to increase the order. This will be discussed for $k=2$ (as is the case for the Burgers equation):

It is possible to raise the order of FRK methods to $p=2$ by forming a linear combination of approximations computed by two *different* FRK methods. Since, on a two-processor computer, the corresponding FRK steps can be performed in parallel, the sequential costs are not increased.

In addition to the FRK method (2.7), we consider the FRK method

$$\text{(3.1)} \qquad u_{n+1} = u^{(1)} + h\Phi_1(f_1, \mathbf{e}t_n+\mathbf{d}_2 h, u^{(1)}), \quad u^{(1)} := y_n + h\Phi_2(f_2, \mathbf{e}t_n+\mathbf{d}_1 h, y_n).$$

Proceeding as in the preceding section, we find that its local error is given by (cf. (2.8))

$$\text{(3.2)} \qquad \tau^*(h) = \tfrac{1}{2}h^2\big[(2\mathbf{b}_2{}^T\mathbf{d}_1 - 1)g_2 + (2\mathbf{b}_1{}^T\mathbf{d}_2 - 1)g_1 + (2\mathbf{b}_2{}^T A_2\mathbf{e} - 1)J_2 f_2 + (2\mathbf{b}_1{}^T A_1\mathbf{e} - 1)J_1 f_1 + J_1 f_2 - J_2 f_1\big] + O(h^3).$$

Hence, the local error of $(y_{n+1} + u_{n+1})/2$ is given by

$$\tfrac{1}{2}(\tau_k + \tau^*_k) = \tfrac{1}{2}h^2\big[(\mathbf{b}_1{}^T\mathbf{c}_1 + \mathbf{b}_1{}^T\mathbf{d}_2 - 1)g_1 + (\mathbf{b}_2{}^T\mathbf{c}_2 + \mathbf{b}_2{}^T\mathbf{d}_1 - 1)g_2 + (2\mathbf{b}_1{}^T A_1\mathbf{e} - 1)J_1 f_1 + (2\mathbf{b}_2{}^T A_2\mathbf{e} - 1)J_2 f_2\big] + O(h^3).$$

This leads to the theorem:

Theorem 3.1. Let the generating RK methods satisfy the conditions

(3.3) $\mathbf{b}_1{}^T\mathbf{e} = \mathbf{b}_2{}^T\mathbf{e} = 2\mathbf{b}_1{}^T A_1\mathbf{e} = 2\mathbf{b}_2{}^T A_2\mathbf{e} = 1.$

Then the parallel FRK method $(y_{n+1} + u_{n+1})/2$ defined by {(2.7),(3.1)} is of second order if

(3.4) $\mathbf{b}_1{}^T[\mathbf{c}_1 + \mathbf{d}_2] = \mathbf{b}_2{}^T[\mathbf{c}_2 + \mathbf{d}_1] = 1.$ □

Given the FRK method (2.7), this theorem can be used in choosing the vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ in the FRK method (3.1) in order to obtain second-order accuracy. In Table 3.1, this is illustrated by giving the vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ for the FRK method (3.1) in the cases where (2.7) is, respectively, the Back step, Zero step or Forward step method of Table 2.1 with k=2.

Table 3.1. Vectors $\mathbf{c}_j$ and $\mathbf{d}_j$ yielding second-order parallel FRK methods {(2.7),(3.1)}

Method	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{d}_1$	$\mathbf{d}_2$
Parallel Back step	$A_1\mathbf{e}$	$A_2\mathbf{e}$	$A_2\mathbf{e}$	$A_1\mathbf{e}$
Parallel Zero step	$A_1\mathbf{e}$	$\mathbf{e}$	$\mathbf{0}$	$A_1\mathbf{e}$
Parallel Forward step	$A_1\mathbf{e}$	$A_2\mathbf{e} + \mathbf{e}$	$A_2\mathbf{e} - \mathbf{e}$	$A_1\mathbf{e}$

As explained in Section 2, the Forward step method has the advantage that inconsistencies of intermediate values of t and y are avoided in the case of time-dependent righthand side functions. However, in the case of the parallel Forward step method specified in Table 3.1, this advantage is lost in the computation of u_{n+1}. By choosing

(3.5) $\mathbf{d}_1 = A_2\mathbf{e}, \quad \mathbf{d}_2 = A_1\mathbf{e} + \mathbf{e},$

consistency of intermediate values of t and y is retained, but the order drops to p=1. In order to obtain again second-order accuracy, let us consider the local truncation error corresponding to (3.5). Assuming that condition (3.3) is satisfied, we have

$$\tfrac{1}{2}(\tau_k + \tau^*_k) = \tfrac{1}{2}h^2\big[g_1 + g_2\big] + O(h^3).$$

Hence, by adding to this parallel FRK approximation $(y_{n+1} + u_{n+1})/2$ the correction $- h^2[g_1 + g_2]/2$, we restore the second-order accuracy. This correction term can be obtained by computing

$$h\big[f(t_n, y_n) - f(t_n + \tfrac{1}{2}h, y_n)\big] = -\tfrac{1}{2}h^2 g(t_n, y_n) + O(h^3) = -\tfrac{1}{2}h^2\big[g_1 + g_2\big] + O(h^3).$$

Thus, the parallel Forward step method with correction term reads

$$
\begin{aligned}
&y_{n+1} = y^{(1)} + h\,\Phi_2(f_2,\ et_n{+}A_2eh + eh,\ y^{(1)}), \qquad y^{(1)} := y_n + h\,\Phi_1(f_1,\ et_n{+}A_1eh,\ y_n),\\
(3.6)\qquad &u_{n+1} = u^{(1)} + h\,\Phi_1(f_1,\ et_n{+}A_1eh + eh,\ u^{(1)}), \qquad u^{(1)} := y_n + h\,\Phi_2(f_2,\ et_n{+}A_2eh,\ y_n),\\
&y^*_{n+1} = \tfrac{1}{2}\big[y_{n+1} + u_{n+1}\big] + h\big[f(t_n, y_n) - f(t_n + \tfrac{1}{2}h,\ y_n)\big].
\end{aligned}
$$

It furnishes a second-order approximation y^*_{n+1} without causing inconsistencies of intermediate values of t and y.

Finally, we remark that under the conditions of Theorem 2.2 the stability function of the parallel methods discussed above is identical with that of the generating FRK method.

4. Numerical experiments

4.1. Burgers' equation

We tested the performance of the FRK methods by integrating the initial-value problem for the well known Burgers equation

$$
(4.1)\qquad \frac{\partial u}{\partial t} = \varepsilon\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} + s(x,t), \quad 0 \le x, t \le 1,
$$

where ε is a small parameter. Initial values, Dirichlet boundary conditions and the source function s(x,t) follow from the exact solution specified below:

Problem I	$u(x,t) = \exp(-x^2)\sin^2(2\pi t)$	non-vanishing spatial error
Problem II	$u(x,t) = (x - 1/2)^2 \sin^2(2\pi t)$	vanishing spatial error
Problem III	$u(x,t) = \dfrac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}}$	vanishing source term
	$A := \dfrac{x - 0.5 + 4.95t}{20\varepsilon}$, $B := \dfrac{x - 0.5 + 0.75t}{4\varepsilon}$,	$C := \dfrac{x - 0.375}{2\varepsilon}$

These problems were semidiscretized on a uniform grid with mesh size Δx using standard symmetric second-order differences. The resulting system of $N=(\Delta x)^{-1}-1$ ODEs can be represented in the form:

$$
y'(t) = f(t, y) := f_1(t,y) + f_2(t,y),
$$

$$
f_1(t,y) := \frac{\varepsilon}{(\Delta x)^2}(Dy + v_L + v_R) + \theta\, s(t), \qquad f_2(t,y) := -\frac{1}{2\Delta x}Y\,(Cy - v_L + v_R) + (1-\theta)\,s(t),
$$

$$
(4.2)\qquad v_L := \big(u(0,t), 0, \dots, 0\big)^T, \quad v_R := \big(0, \dots, 0, u(1,t)\big)^T, \quad s(t) := \big(s(j\Delta x, t)\big),
$$

$$D := \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & O \\ & & \dots & & \\ O & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}, \quad C := \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & O \\ & & \dots & & \\ O & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}, \quad Y := \mathrm{diag}(y).$$

4.2. Generating RK methods

For the generating RK methods we choose the standard fourth-order RK method (RK4) and the second-order Runge-Kutta-Chebyshev method (RKC2) proposed in [5] (see also [9]).

4.2.1. The RK4 method. RK4 requires four stages per step and is suitable for integrating convection equations because, within the class of explicit methods, it possesses an almost optimal (scaled) imaginary stability boundary. In the case of (4.2) with $\varepsilon=0$, this leads to the maximum stable time step

$$(4.3a) \qquad h_{conv} = \frac{\beta_{imag}}{\rho(\partial f_2/\partial y)} \approx \frac{\beta_{imag}\,\Delta x}{\| y_n \|_\infty}, \quad \beta_{imag} = 2\sqrt{2}\,,$$

where $\rho(\partial f_2/\partial y)$ denotes the spectral radius of $\partial f_2/\partial y$. Hence, in terms of evaluations of f_2, the overall costs to integrate the unit interval are about

$$(4.3b) \qquad N_2 := \frac{4}{h_{conv}} \approx \frac{1.4\,\|y\|_\infty}{\Delta x}\,,$$

where y denotes the averaged value of y_n in the integration interval.

In order to illustrate that introducing diffusion ($\varepsilon \neq 0$) causes that the use of RK4 may be highly inefficient, let us apply RK4 to (4.2) with $\varepsilon \neq 0$. It is convenient to introduce the parameter

$$(4.4) \qquad q := \frac{\varepsilon}{\Delta x\,\|y\|_\infty}\,.$$

Now, a stable integration with RK4 requires the step size to satisfy

$$(4.5) \qquad h_{conv\text{-}diff} \leq \frac{\beta}{\rho(\partial f/\partial y)} \approx \frac{\beta\,\Delta x}{\|y\|_\infty \sqrt{1 + 16q^2}}, \qquad \beta \approx 2.6,$$

where β denotes the stability boundary of RK4 for general eigenvalue spectra in the left halfplane. Assuming that the evaluations of f_1 and f_2 are equally expensive, it follows from (4.3) and (4.5) that introducing nonzero values of ε increases the computational effort by a factor $\approx 2\sqrt{(1 + 16q^2)}$. This factor is already substantial for moderate values of the parameter q.

4.2.2. The RKC2 method. RKC2 was constructed for integrating diffusion equations with unrestricted step sizes. Stability is achieved by adapting the number of stages of the method. If it is used for integrating the diffusion part of (4.2) with step h, then the number of stages is approximately given by (cf. [5])

$$\text{(4.6a)} \qquad m = 1.2\sqrt{h\,\rho(\partial f_1/\partial y)} \approx \frac{2.4\sqrt{h\varepsilon}}{\Delta x}\ .$$

In terms of evaluations of f_1 and by setting $h{=}h_{conv}$, we deduce from (4.6a) and (4.3a) that the overall costs are

$$\text{(4.6b)} \qquad N_1 := \frac{m}{h_{conv}} \approx \frac{2.4\sqrt{h_{conv}\varepsilon}}{h_{conv}\Delta x} \approx 1.4\sqrt{\frac{\varepsilon\,\|y\|_\infty}{(\Delta x)^3}}\ .$$

Hence, the FRK based on RK4 and RKC2 using stepsizes given by h_{conv}, requires N_2 evaluations of f_2 and N_1 evaluations of f_1. Notice that these are the *minimal* number of evaluations of f_2 and f_1 in order to obtain stability for the FRK method. Assuming that the evaluations of f_1 and f_2 are equally expensive, it follows from (4.3) and (4.6) that introducing nonzero values of ε and using FRK instead of RK4 increases the computational effort by a factor $(N_1 + N_2) / (2N_2) \approx (1 + \sqrt{q}\,) / 2$. A comparison of this factor with the factor $2\sqrt{}\ (1 + 16q^2)$ derived above reveals that the use of (RK4,RKC2)-based FRK methods is much cheaper than using RK4.

4.3. Numerical results

We shall test the performance of the FRK methods by integrating the initial-value problems specified in Section 4.1. We applied a two-term FRK method with Φ_1 defined by RKC2 and Φ_2 by RK4.

The computational effort associated with the methods is measured by the total numbers of f_1- and f_2-evaluations. Notice that the *second*-order RKC2 method uses at least 2 stages.

The accuracy of the numerical results is measured by the minimal number of correct decimal digits of the components of the numerical solution at the end point t=1, i.e. by

$$cd := -\log_{10}(\,\|\text{ global error at } t{=}1\ \|_\infty\,).$$

4.3.1. Comparison of RK4 and FRK. In this subsection, we show that RK4 is an adequate method for integrating strongly convection dominated equations, that is, the stability condition on the time step is not more restrictive than the accuracy condition associated with the spatial discretization error. However, we shall also show that the FRK methods solve convection-diffusion equations much more efficiently than RK4 if the amount of diffusion increases. Choosing Problem I as a test problem and using $\Delta x{=}1/200$ for the spatial discretization, we have a spatial accuracy of 5.3 correct decimal digits. Table 4.1 lists the cd-values obtained by RK4 and the Zero step version of the FRK method.

The corresponding numbers of f_1- and f_2-evaluations are added in brackets and an unstable behaviour is indicated by $*$. This table illustrates that the time step of RK4 is dictated by accuracy for $\varepsilon \le 10^{-3}$. For $\varepsilon \ge 10^{-2}$ the stability condition is much more restrictive than the accuracy conditions. Instead, the FRK method remains stable for all ε and all h-values from this table. As a consequence, this method is able to produce solutions of realistic accuracy at the cost of a moderate computational effort.

Table 4.1. cd-values for Problem I with $\Delta x=1/200$ and $\theta=1$.

Method	ε	h=1/80	h=1/160	h=1/320	h=1/640	h=1/5800
RK4	10^{-10}	4.8 (320+320)	5.3 (640+640)			
	10^{-3}	3.9 (320+320)	5.3 (640+640)			
	10^{-2}	*	*	*	5.3 (2560+2560)	
	10^{-1}	*	*	*	*	5.3 (23200+23200)
Zero step	10^{-3}	2.6 (240+320)	3.2 (320+640)	3.8 (640+1280)	4.4 (1280+2560)	
	10^{-2}	2.8 (480+320)	3.4 (800+640)	3.9 (960+1280)	4.5 (1920+2560)	
	10^{-1}	3.1 (1440+320)	3.6 (2080+640)	4.3 (2880+1280)	4.8 (4480+2560)	

4.3.2. Mutual comparison of the FRK methods. On the basis of Problem II, we will compare the FRK versions as specified in Table 2.1. In Table 4.2 the results of the various methods are given for several values of θ. Notice that all errors are due merely to the time integration since this problem does not give rise to a spatial discretization error. We may draw two conclusions from this table: firstly, it seems recommendable to set θ equal to 1 in the operator splitting (4.2), i.e., the complete source term is added to f_1; a second conclusion is that, for $\theta=1$, the Zero step version is slightly more accurate than the two other versions.

Table 4.2. cd-values for Problem II with $\Delta x=1/200$ and $\varepsilon=10^{-2}$.

FRK method	θ	h=1/20	h=1/40	h=1/80	h=1/160	h=1/320
Back step	1	1.7	2.2	2.7	3.3	3.9
Zero step	1	2.2	2.7	3.2	3.8	4.3
Forward step	1	1.8	2.3	2.9	3.6	4.5
Back step	0.5	1.3	1.5	1.8	2.2	2.7
Zero step	0.5	1.4	1.6	1.9	2.3	2.8
Forward step	0.5	1.4	1.7	2.0	2.3	2.8
Back step	0	0.9	1.3	1.5	1.9	2.4
Zero step	0	0.9	1.3	1.6	2.0	2.5
Forward step	0	1.1	1.4	1.7	2.0	2.5

4.3.3. Sequential versus Parallel FRK methods. It is of interest to compare the accuracies of the sequential and parallel versions of the FRK methods (we recall that both versions have the same sequential costs). When applied to the Problems I and II, the results produced by the parallel and sequential FRK methods differ only marginally (the observed differences in the cd-values are at most 0.4 and in many cases much less). With respect to the order behaviour of the sequential versions, we conclude from the Tables 4.1 and 4.2 a performance which indicates an order larger than the theoretical order 1 (notice that a p-th order behaviour means an increase of the cd-value by $\log_{10}(2^p) \approx 0.3*p$ on halving the step size is). In particular, for the (sequential) Zero step method applied to Problem I we observe a second-order behaviour, similar to the parallel version. For Problem II, all sequential methods show approximately order 2 for $\theta=1$, and for the smaller θ-values the resulting order varies between 1 and 2.

There are however problems for which the parallel versions behave markedly more accurate than the sequential counterparts. An example is Problem III. The solution of this test example, which has also been discussed in [7] and [3], developes, for small values of ε, a steep shock wave, which moves across the spatial domain. Following [7] , we set $\varepsilon=0.003$ in our test. For this ε-value, a plot of the time evolution of the exact solution can be found in [7]. To present this particular solution with reasonable accuracy on a uniform grid, we need an extremely fine spatial mesh. Therefore, we choose $\Delta x=1/800$, resulting in a spatial accuracy of 2.9 correct digits.

Table 4.3 presents the results of the sequential and parallel FRK methods. There is no need to distinguish between the various versions, since they produced the same accuracy. We remark that θ is not relevant because, for this problem, the source term vanishes identically.

Table 4.3. cd-values for Problem III with $\Delta x=1/800$ and $\varepsilon=0.003$.

Method	h=1/320	h=1/640	h=1/1280	h=1/2560	h=1/5120
Sequential FRKversions	2.0	2.3	2.5	2.7	2.8
Parallel FRKversions	2.8	2.9			

We see that the parallel variants yield approximately the maximally obtainable accuracy on this spatial grid using the largest stable step size, whereas the sequential versions need much smaller step sizes to let the time-integration error be negligible with respect to the spatial error.

Furthermore, we mention some results from [7], where this problem has been integrated using a variable-step, variable-order linear multistep (LM) method (viz. the GEARB package). Extrapolating their results obtained with 200 and 400 mesh points, this integrator would require (on a grid with 800 points) approximately 450 time steps if the *implicit* LM methods (i.e., BDF methods) are used and approximately 12000 time steps if it uses the *explicit* LM methods (i.e., Adams predictor-corrector pair). Since these methods require (in the avarage) at least 1.5 (full) f-evaluation per step, it is clear that the parallel FRK methods solve this problem more efficiently. Especially, if we take into account that the implicit LM method has a lot of additional work per step, like evaluating and decomposing Jacobian matrices and solving linear systems.

4.3.4. A further modification of the FRK methods. As we have seen in the experiments described in the preceeding subsections, it is the convection term which limits the maximal stable step size of the (RKC2,RK4)-based FRK method. The reason is, of course, the fixed (and rather small) stability boundary of the RK4 method, whereas the RKC2 method has a dynamic stability boundary, simply by adapting its number of stages to the step size required. In particular, all the aforementioned experiments show that the maximal step size was determined by the RK4 method and RKC2 was given the same step size. In many of the above tests, this resulted in a rather modest m-value (the number of stages) for RKC2. This is a relatively inefficient use of the RKC2 since its stability boundary (and hence its maximal stable step size) increases *quadratically* with m (cf. (4.6a)). Hence, per stage, RKC2 is able to proceed the integration over a distance which is linear in m.

Consequently, a more efficient use of the capabilities of RKC2 is to select a step size merely on the basis of accuracy, and to adapt the number of stages in RKC2 to obtain stability. If this step h happens to be too large for RK4, then it is divided by an integer, say M, such that h/M is a stable step size for RK4 and this 'convection integrator' is applied M times to bridge the step h, taken by RKC2.

To be more precise, this FRK* method is defined by (cf. (2.7))

$$\begin{aligned} & y^{(0)} = y_n + h\,\Phi_1(f_1, \mathbf{e}t_n+\mathbf{c}_1 h, y_n), \\ (2.7') \qquad & y^{(j)} = y^{(j-1)} + \frac{h}{M}\Phi_2(f_2, \mathbf{e}t_n+\mathbf{c}_{2,j}\, h, y^{(j-1)}), \quad j = 1, \dots, M; \\ & y_{n+1} = y^{(M)}, \end{aligned}$$

where, again, Φ_1 and Φ_2 are associated with RKC2 and RK4, respectively.

The effect of using the FRK* method is that we can take larger steps than would have been possible with the FRK methods. Notice, that the total number of f_2-evaluations over the whole integration interval is the same for both methods, but the total number of f_1-evaluations will be less for FRK*. This is particularly advantageous if f_1 is expensive. Since we concluded from our previous experiments that choosing $\theta=1$ (i.e., adding a source term to f_1) is recommendable, it is very likely that f_1 is rather expensive indeed.

Usually, the accuracy will decrease, simply because we use a larger step size but this can be a desirable situation in cases where the maximal stable step size in the FRK methods still yields too much precision at the costs of considerable computational effort. In fact, we can say that the FRK* method is an adaptive scheme which can treat any timestep.

We have repeated the experiments described in the Subsections 4.3.1 and 4.3.3, using the Zero step version, that is $\mathbf{c}_1 = A_1\mathbf{e}$ and $\mathbf{c}_{2,j} = \mathbf{e}$, j=1,...,M (cf. Table 2.1). Similar to the FRK methods, we can also construct parallel, second-order FRK* methods by reversing the sequence of the Φ_1- and Φ_2-applications in (2.7'). Thus, this scheme reads

$$u^{(0)} = y_n,$$

$$(3.1') \qquad u^{(j)} = u^{(j-1)} + \frac{h}{M}\Phi_2(f_2, \mathbf{e}t_n+\mathbf{d}_{1,j}h, u^{(j-1)}), \quad j = 1, \dots, M;$$

$$u_{n+1} = u^{(M)} + h\,\Phi_1(f_1, \mathbf{e}t_n+\mathbf{d}_2h, u^{(M)}),$$

and, finally, a second-order approximation is obtained by setting $y^*_{n+1} = (y_{n+1} + u_{n+1})/2$. For the Zero step variant, $\mathbf{d}_{1,j} = 0$, j=1,...,M and $\mathbf{d}_2 = A_1\mathbf{e}$.

We have applied this method to Problem I and used step sizes which are equal or larger than those used in Subsection 4.3.1. Table 4.4 contains the results.

Table 4.4. cd-values for Problem I obtained with the sequential and parallel FRK* method, Δx=1/200, ε=0.1 and θ=1

h	M	# f_1-evaluations	# f_2-evaluations	Seq. Zero step	Par. Zero step
1/80	1	18*80	320	3.1	3.1
1/40	2	25*40	320	2.5	2.5
1/20	4	36*20	320	1.9	1.9
1/10	8	50*10	320	1.2	1.2

We see that if one is satisfied with a global error of 10^{-2} (i.e., cd=2), then a step size h ≈ 1/20 can be used which is not possible in the FRK methods. As a consequence, the number of f_1-evaluations drops from 1440 to 720 (cf. Table 4.1). Furthermore, we observe for this problem, similar to the FRK methods, that the parallel version does not improve the accuracy of the sequential version since the latter also shows a second-order behaviour.

We also repeated the integration of Problem III. The results are collected in Table 4.5. Similar to the situation for the FRK methods (see Subsection 4.3.3) we see that, for this problem, the parallel version is superior to the sequential variant. Again, the FRK* method offers the possibility to choose the step size on the basis of accuracy considerations without being restricted by stability.

Table 4.5. cd-values for Problem III obtained with the sequential and parallel FRK* method, Δx=1/800 and ε=0.003

h	M	# f_1-evaluations	# f_2-evaluations	Seq. Zero step	Par. Zero step
1/320	1	7*320	1280	2.0	2.8
1/160	2	9*160	1280	1.7	2.2
1/80	4	13*80	1280	1.1	1.3
1/40	8	18*40	1280	0.6	0.7

References

[1] **Ames, W.F. (1965)**: Non-linear partial differential equations in engineering, Academic Press, New York.

[2] **Burgers, J.M. (1948)**: A mathematical model illustrating the theory of turbulence, Advances in Applied Mechanics **1**, pp. 171-199.

[3] **Evans, D.J. & Abdullah, A.R. (1984)**: The group explicit method for the solution of Burgers' equation, Computing **32**, pp. 239-253.

[4] **Hairer, E., Nørsett S.P. & Wanner, G. (1987)**: Solving ordinary differential equations I, Nonstiff problems, Springer-Verlag, Berlin.

[5] **Houwen, P.J. van der & Sommeijer, B.P. (1980)**: On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values, ZAMM **60**, pp. 479-485.

[6] **Marchuk, G.I. (1981)**: Methods of numerical mathematics, second edition, Springer-Verlag, New York.

[7] **Sincovec, R.F. & Madsen, N.K. (1975)**: Software for nonlinear partial differential equations, ACM Trans. Math. Software **1**, pp. 232-260.

[8] **Swayne, D.A. (1987)**: Time dependent boundary and interior forcing in locally one-dimensional schemes, SIAM J. Sci. Comp. **8**, pp. 755-767.

[9] **Verwer, J.G., Hundsdorfer, W.H. & Sommeijer, B.P. (1990)**: Convergence properties of the Runge-Kutta-Chebyshev method, Numer. Math. **57**, pp. 157-178.

[10] **Yanenko, N.N. (1971)**: The method of fractional steps, Springer-Verlag, Berlin.