IMPACT OF COMPUTING IN SCIENCE AND ENGINEERING **4**, 195–216 (1992)

Fractional Runge–Kutta Methods with Application to Convection–Diffusion Equations

P. J. VAN DER HOUWEN AND B. P. SOMMEIJER

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

Received March 26, 1991

P. J. van der Houwen and B. P. Sommeijer, Fractional Runge–Kutta Methods with Application to Convection–Diffusion Equations, *IMPACT of Computing in Science and Engineering* **4**, 195–216 (1992).

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 relax the time step condition by applying fractional step (or operator splitting) methods based on Runge–Kutta methods. Typically, fractional step methods have a low order of accuracy. Therefore, we also discuss a variant with increased order. This version has the additional advantage that parallelism can be exploited. The methods are illustrated by application to the Burgers equation and by a comparison with the LOD method. © 1992 Academic Press, Inc.

1. INTRODUCTION

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

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

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). As an example, consider the nonlinear convection–diffusion equation

$$\frac{\partial u}{\partial t} = \epsilon \Delta u - u \nabla u. \tag{1.2}$$

0899-8248/92 $5.00

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 diffusion and convection terms, respectively.

Systems of the type (1.2) 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 the other hand, when *explicit* methods are used, stability will dictate time steps that are usually much smaller than the accuracy of the numerical approximation requires. If the diffusion term in a convection–diffusion problem such as (1.2) 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 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 [11], Marchuk [7], and Swayne [9]. 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

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

and the RK methods generated by the Butcher arrays

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

or equivalently

$$y_{n+1} = y_n + h\phi_j(f, \mathbf{e}t_n + \mathbf{c}_j h, y_n), \qquad j = 1, \ldots, k, \tag{2.2'}$$

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

$$\Phi_j(f, \mathbf{e}t_n + \mathbf{c}_j h, y_n) = \mathbf{b}_j^{\mathrm{T}} f(\mathbf{e}t_n + \mathbf{c}_j h, \mathbf{Y}),$$
$$\mathbf{Y} = y_n\mathbf{e} + hA_j f(\mathbf{e}t_n + \mathbf{c}_j h, \mathbf{Y}). \tag{2.2''}$$

By expanding the increment function Φ_j we obtain

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

$$g(t, y) := \frac{\partial f}{\partial t}(t, y), \qquad J(t, y) := \frac{\partial f}{\partial y}(t, y). \tag{2.3}$$

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

$$p = 1 \quad \text{if} \quad \mathbf{b}_j^{\mathrm{T}} \mathbf{e} = 1; \quad p = 2 \quad \text{if, in addition,} \quad \mathbf{b}_j^{\mathrm{T}} \mathbf{c}_j = \mathbf{b}_j^{\mathrm{T}} A_j \mathbf{e} = \tfrac{1}{2}. \tag{2.4}$$

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

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

(briefly, the *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 consider three types of FRK methods that differ from each other in the choice of the abscissas $\mathbf{e}t_n + \mathbf{c}_j h$ for the stage vector $\mathbf{Y}$, as given in Table 2.1.

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

The back step method uses "conventional" abscissas as in RK methods for ordinary differential equations (ODEs). It successively integrates the "fractional" equations

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	$\mathbf{c}_j := (j-1)\mathbf{e} + A_j\mathbf{e}, j = 1, \ldots, k$

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

from t_n to t_{n+1} using the preceding result as initial value. A disadvantage of these back step methods arises if Eq. (1.1) explicitly contains the time variable t. In such cases, they may suffer a drop in accuracy. This is caused by a possible lack of compatibility 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}, whereas the corresponding t-argument again runs from t_n to t_{n+1}.

In the zero step method, the time variable in the right-hand side evaluations employed by the RK methods is set equal to t_{n+1} in the second and next fractional steps. This 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 right-hand side functions in each subinterval of length h. Its advantage lies in the fact that all incompatibilities of t and y values are avoided. A disadvantage 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 detail for $k = 2$. First, we observe that second-order accuracy of the FRK method requires that y_{n+1} satisfy the condition

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

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

$$\begin{aligned} y_{n+1} &= y^{(1)} + h\Phi_2(f_2, \mathbf{e}t_n + \mathbf{c}_2 h, y^{(1)}), \\ y^{(1)} &:= y_n + h\Phi_1(f_1, \mathbf{e}t_n + \mathbf{c}_1 h, y_n). \end{aligned} \tag{2.7}$$

From (2.3) we deduce for (2.7)

$$\begin{aligned} y^{(1)} &= y_n + \mathbf{b}_1^{\mathrm{T}}\mathbf{e}hf_1 + \mathbf{b}_1^{\mathrm{T}}\mathbf{c}_1 h^2 g_1 + \mathbf{b}_1^{\mathrm{T}} A_1 \mathbf{e} h^2 J_1 f_1 + O(h^3), \\ y_{n+1} &= y^{(1)} + \mathbf{b}_2^{\mathrm{T}}\mathbf{e}hf_2(t_n, y^{(1)}) + \mathbf{b}_2^{\mathrm{T}}\mathbf{c}_2 h^2 g_2(t_n, y^{(1)}) \\ &\quad + \mathbf{b}_2^{\mathrm{T}} A_2 \mathbf{e} h^2 J_2(t_n, y^{(1)}) f_2(t_n, y^{(1)}) + O(h^3) \\ &= y_n + \mathbf{b}_1^{\mathrm{T}}\mathbf{e}hf_1 + \mathbf{b}_1^{\mathrm{T}}\mathbf{c}_1 h^2 g_1 + \mathbf{b}_1^{\mathrm{T}} A_1 \mathbf{e} h^2 J_1 f_1 + \mathbf{b}_2^{\mathrm{T}}\mathbf{e}hf_2 \\ &\quad + \mathbf{b}_2^{\mathrm{T}}\mathbf{e}\mathbf{b}_1^{\mathrm{T}}\mathbf{e}h^2 J_2 f_1 + \mathbf{b}_2^{\mathrm{T}}\mathbf{c}_2 h^2 g_2 + \mathbf{b}_2^{\mathrm{T}} A_2 \mathbf{e} h^2 J_2 f_2 + O(h^3). \end{aligned}$$

Assuming that the generating RK methods have $p \geqslant 1$ (i.e., $\mathbf{b}_1^{\mathrm{T}}\mathbf{e} = \mathbf{b}_2^{\mathrm{T}}\mathbf{e} = 1$), we may write

$$\begin{aligned} y_{n+1} = y_n + hf + h^2[&\mathbf{b}_1^{\mathrm{T}}\mathbf{c}_1 g_1 + \mathbf{b}_2^{\mathrm{T}}\mathbf{c}_2 g_2 + J_2 f_1 \\ &+ \mathbf{b}_1^{\mathrm{T}} A_1 \mathbf{e} J_1 f_1 + \mathbf{b}_2^{\mathrm{T}} A_2 \mathbf{e} J_2 f_2] + O(h^3). \end{aligned}$$

Using (2.6) we obtain for the local error

$$\begin{aligned} \tau(h) = \tfrac{1}{2}h^2[&(2\mathbf{b}_1^{\mathrm{T}}\mathbf{c}_1 - 1)g_1 + (2\mathbf{b}_2^{\mathrm{T}}\mathbf{c}_2 - 1)g_2 + (2\mathbf{b}_1^{\mathrm{T}}\mathbf{e} - 1)J_1 f_1 \\ &+ (2\mathbf{b}_2^{\mathrm{T}} A_2 \mathbf{e} - 1)J_2 f_2 + J_2 f_1 - J_1 f_2] + O(h^3). \end{aligned} \tag{2.8}$$

Thus, we can formulate

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

$$\mathbf{b}_1^{\mathrm{T}}\mathbf{e} = \mathbf{b}_2^{\mathrm{T}}\mathbf{e} = 1. \tag{2.9a}$$

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 of the orders of the generating RK methods. However, if we impose the additional conditions

$$\mathbf{c}_1 = A_1\mathbf{e}, \qquad 2\mathbf{b}_1^{\mathrm{T}}A_1\mathbf{e} = 2\mathbf{b}_2^{\mathrm{T}}A_2\mathbf{e} = 1 \tag{2.9b}$$

(note that $\mathbf{c}_2$ is determined by the choice of the particular FRK method, see Table 2.1), then the local error of (2.7) is given by

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

so that we achieve second-order accuracy in problems where

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

In our experiments with Burgers' equation, it turned out that this error is rather small, so that in most cases second-order behavior 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, removing the term J_1f_1 from (2.8) reduces the local error sufficiently to give second-order behavior, in spite of the cross terms J_1f_2 and J_2f_1 (see Sections 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

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

Following the normal mode analysis approach commonly used in analysing the stability of PDE methods, we assume that the Jacobians J_j possess the same set of complex exponentials as eigenfunctions.

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* $j = 1, \ldots, k$, *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

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

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

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

Let the eigenvalues of hJ_j be in S_j. Then, the matrices $R_j(hJ_j)$ have eigenvalues within the unit circle. Hence, assuming that the matrices hJ_j have an equal eigensystem, 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 $\partial f_j/\partial y$ of the functions f_j. In our numerical experiments, the stability conditions derived from Theorem 2.2 appeared to be reliable.

3. Parallel FRK Methods

In this section, we discuss the introduction of parallelism in FRK methods. Two approaches are considered. The first approach aims at reducing the computational work per step, and the second approach aims at improving the order of the FRK method. Both forms of parallelism are of the so-called "parallelism across the method"-type; that is, a small number of subtasks of comparable computational length are distinguished. However, each of the subtasks involves considerable computational effort, so that many processors can be exploited within each subtask. This approach is especially relevant in the present context of PDEs, where the resulting ODEs usually have a large dimension. For example, in Section 4.2, we specify FRK methods based on *explicit* methods, for which the major part of the computational effort goes into the frequent evaluation of the right-hand side functions f_j. Obviously, the components of these functions can be evaluated concurrently, allowing for the use of many processors within each f_j-evaluation (so-called "parallelism across the problem"). Both types of parallelism are independent and a combination is easily achieved.

3.1. *Reduction of Costs per Step*

A straightforward way of exploiting parallel computers is to concurrently apply appropriate RK methods to each of the "fractional differential equations" $y'(t) = f_j(t, y(t)), j = 1, \ldots, k$, and to form a linear combination of these results:

$$y_{n+1} = \sum_{j=1}^{k} a_j y^{(j)}, \qquad \sum_{j=1}^{k=1} a_j = 1,$$

$$y^{(j)} = y_n + h_j \Phi_j(f_j, \mathbf{e}t_n + \mathbf{c}_j h_j, y_n), j = 1, \ldots, k. \tag{3.1}$$

The sequential costs of this method correspond to the computational costs of the most expensive increment function, provided that k processors are available. In this way, parallelism is used to reduce the costs of FRK methods: instead of evaluating successively k increment functions per step as in (2.5), the method (3.1) evaluates only one increment function per step. 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.

3.2. *Increasing the Order of Accuracy*

An alternative objective in exploiting parallel computers is improving the quality of the method, without increasing the sequential costs. By parallel application of several of the FRK methods introduced in the preceding sections, it is possible to exploit parallelism and at the same time to raise the order to $p = 2$. This will be discussed for two-term FRK methods, and is based on a linear combination of approximations computed by two *different* FRK methods. Since the corresponding FRK steps can be performed in parallel, the sequential costs are not increased.

To be more precise, we consider, in addition to the FRK method (2.7), the FRK method

$$\begin{aligned} u_{n+1} &= u^{(1)} + h\Phi_1(f_1, \mathbf{e}t_n + \mathbf{d}_2 h, u^{(1)}), \\ u^{(1)} &:= y_n + h\Phi_2(f_2, \mathbf{e}t_n + \mathbf{d}_1 h, v_n). \end{aligned} \tag{3.2}$$

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

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

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

$$\tfrac{1}{2}(\tau_k + \tau_k^*) = \tfrac{1}{2}h^2[(\mathbf{b}_1^T\mathbf{c}_1 + \mathbf{b}_1^T\mathbf{d}_2 - 1)g_1 + (\mathbf{b}_2^T\mathbf{c}_2 + \mathbf{b}_2^T\mathbf{d}_1 - 1)g_2 + (2\mathbf{b}_1^TA_1\mathbf{e} - 1)J_1f_1 + (2\mathbf{b}_2^TA_2\mathbf{e} - 1)J_2f_2] + O(h^3).$$

This leads to the theorem

THEOREM 3.1. *Let the generating RK methods be of second order; i.e., let them satisfy the conditions*

$$\mathbf{b}_1^T\mathbf{e} = \mathbf{b}_2^T\mathbf{e} = 2\mathbf{b}_1^TA_1\mathbf{e} = 2\mathbf{b}_2^TA_2\mathbf{e} = 1. \tag{3.3}$$

Then the parallel FRK method $(y_{n+1} + u_{n+1})/2$ *defined by* $\{(2.7), (3.2)\}$ *is of second order if*

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

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 (3.2) 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 (3.2) in the cases where (2.7) is, respectively, the back step, zero step, or forward step method of Table 2.1 with $k = 2$.

As explained in Section 2, the forward step method has the advantage that incompatibilities of intermediate values of t and y are avoided in the case of time-dependent right-hand 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

$$\mathbf{d}_1 = A_2\mathbf{e}, \qquad \mathbf{d}_2 = A_1\mathbf{e} + \mathbf{e}, \tag{3.5}$$

compatibility 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

TABLE 3.1

VECTORS $\mathbf{c}_j$ AND $\mathbf{d}_j$ YIELDING SECOND-ORDER PARALLEL FRK METHODS $\{(2.7), (3.2)\}$

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	$A_1\mathbf{e}$	$A_2\mathbf{e} + \mathbf{e}$	$A_2\mathbf{e} - \mathbf{e}$	$A_1\mathbf{e}$

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

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

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

Thus, the parallel forward step method with correction term reads

$$\begin{aligned} y_{n+1} &= y^{(1)} + h\Phi_2(f_2, \mathbf{e}t_n + A_2\mathbf{e}h + \mathbf{e}h, y^{(1)}), \\ y^{(1)} &:= y_n + h\Phi_1(f_1, \mathbf{e}t_n + A_1\mathbf{e}h, y_n), \\ u_{n+1} &= u^{(1)} + h\Phi_1(f_1, \mathbf{e}t_n + A_1\mathbf{e}h + \mathbf{e}h, u^{(1)}), \\ u^{(1)} &:= y_n + h\Phi_2(f_2, \mathbf{e}t_n + A_2\mathbf{e}h, y_n), \\ y_{n+1}^* &= \tfrac{1}{2}[y_{n+1} + u_{n+1}] + h[f(t_n, y_n) - f(t_n + \tfrac{1}{2}h, y_n)]. \end{aligned} \tag{3.6}$$

It furnishes a second-order approximation y_{n+1}^* without causing incompatibilities 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 in one spatial dimension,

$$\frac{\partial u}{\partial t} = \epsilon\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} + s(x, t), \qquad 0 \leqslant x, t \leqslant 1, \tag{4.1}$$

where ϵ is a small parameter. It is well known that equations of this (singularly perturbed) type easily give rise to unwanted oscillations in the numerical solution and require a special semidiscretization (and possibly artificial viscosity) to suppress these phenomena. However, since the major aim of the present paper is to study the influence of the operator splitting on the accuracy, stability, and efficiency of the time integration, it is preferable not to interfere these aspects with possible problems originating from the spatial discretization. For that purpose, we prescribed the analytical solution from which the initial values, the Dirichlet boundary conditions, and the source function $s(x, t)$ follow. We considered the problems specified by the following exact solutions:

Problem I	$u(x,t) = \exp(-x^2)\sin^2(2\pi t)$	nonvanishing spatial error
Problem II	$u(x,t) = (x - \frac{1}{2})^2 \sin^2(2\pi t)$	vanishing spatial error
Problem III	$u(x,t) = \dfrac{0.1\,e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}}$	vanishing source term

$$A := \frac{x - 0.5 + 4.95t}{20\epsilon}, \quad B := \frac{x - 0.5 + 0.75t}{4\epsilon}, \quad C := \frac{x - 0.375}{2\epsilon}.$$

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

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

$$f_1(t, y) := \frac{\epsilon}{(\Delta x)^2}(Dy + v_{\mathrm{L}} + v_{\mathrm{R}}) + \theta s(t),$$

$$f_2(t, y) := -\frac{1}{2\Delta x} Y(Cy - v_{\mathrm{L}} + v_{\mathrm{R}}) + (1 - \theta)s(t),$$

$$v_{\mathrm{L}} := (u(0, t), 0, \ldots, 0)^{\mathrm{T}}, \qquad v_{\mathrm{R}} := (0, \ldots, 0, u(1, t))^{\mathrm{T}},$$

$$s(t) := (s(j\Delta x, t)), \tag{4.2}$$

$$D := \begin{pmatrix} -2 & 1 & & & \bigcirc \\ 1 & -2 & 1 & & \\ & & \cdots & & \\ & \bigcirc & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix},$$

$$C := \begin{pmatrix} 0 & 1 & & & \bigcirc \\ -1 & 0 & 1 & & \\ & & \cdots & & \\ & \bigcirc & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}, \qquad Y := \mathrm{diag}(y)\,.$$

4.2. *Generating RK Methods*

For the generating RK methods we choose the standard fourth-order RK method (RK4) for integrating the convection part of the equation (defined by f_2) and the second-order Runge–Kutta–Chebyshev method (RKC2) proposed in [5] (see also [10]) for integrating the diffusion part (i.e., f_1). In the next two subsections, these choices will be motivated.

4.2.1. *The RK4 Method*

The RK4 method is a widely used integrator for convection problems (see, e.g., [6]). It requires four stages per step and, within the class of explicit first- or second-order methods, it possesses an almost optimal (scaled) imaginary stability boundary. To be more precise, within the class of explicit first- or second-order four-stage methods, the maximal attainable imaginary stability boundary is $\beta_{\text{imag}} = 3$, whereas RK4 possesses $\beta_{\text{imag}} = 2\sqrt{2}$. In the case of (4.2) with $\epsilon = 0$, this leads to the maximum stable time step

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

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

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

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

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

$$q := \frac{\epsilon}{\Delta x\|y\|_\infty}. \tag{4.4}$$

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

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

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.

In conclusion, we can say that the application of RK4 is not feasible to integrate the "whole" problem (4.2), but RK4 certainly is an efficient integrator to treat the convection part.

4.2.2. *The RKC2 Method*

The second-order RKC method was constructed for integrating diffusion equations with step sizes that are merely prescribed by accuracy considerations. Stability is achieved by adapting the number of stages of the method. Similarly to RK4, RKC2 is not very suitable to integrate the "whole" problem (4.2); however, it is very efficient to treat the diffusion part of the ODE, defined by f_1. If it is used to integrate this part with step h, then the number of stages is given by (cf. [5])

$$m = 1.5\sqrt{h\rho(\partial f_1/\partial y)} \approx \frac{3\sqrt{h\epsilon}}{\Delta x}. \qquad (4.6a)$$

Setting $h = h_{\text{conv}}$, we deduce from (4.6a) and (4.3a) that the number of f_1-evaluations to integrate the unit interval is given by

$$N_1 := \frac{m}{h_{\text{conv}}} \approx \frac{3\sqrt{h_{\text{conv}}\epsilon}}{h_{\text{conv}}\Delta x} \approx 1.8\sqrt{\frac{\epsilon\|y\|_\infty}{(\Delta x)^3}}. \qquad (4.6b)$$

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. Note that these are the *minimal* numbers of evaluations of f_2 and f_1 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 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 the use of RK4.

4.3. *Reference Method*

A well-known fractional step method is the LOD scheme due to Yanenko [11]. This implicit method nicely fits into the FRK framework if we allow the generating RK method to be implicit. As a matter of fact, the LOD scheme is based on the implicit backward Euler method and can be formulated as (cf. (2.5))

$$\begin{aligned} y^{(0)} &= y_n, \\ y^{(1)} &= y^{(0)} + hf_1(t_n + h, y^{(1)}), \\ y_{n+1} := y^{(2)} &= y^{(1)} + hf_2(t_n + h, y^{(2)}). \end{aligned} \tag{4.7}$$

This first-order LOD scheme has been proposed to circumvent the expensive linear algebra involved in the case of higher-dimensional PDEs. Therefore, the splitting of the right-hand side function $f(t, y)$ into the k functions $f_j(t, y)$ corresponds to the k spatial dimensions of the PDE. In this paper, we use (4.7) as a reference method in testing a two-dimensional extension of problem (4.1) (cf. Section 4.4.5). We remark that the LOD scheme is unconditionally stable, but is rather expensive since in every step nonlinear relations have to be solved.

4.4. *Numerical Results*

We 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. Note 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

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

4.4.1. *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 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 unstable behavior is indicated by $*$. This table illustrates that the time step of RK4 is dictated by accuracy for $\epsilon \leqslant 10^{-3}$. For $\epsilon \geqslant 10^{-2}$ the stability condition is much more restrictive than the accuracy conditions. Instead, the FRK method remains

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)	

TABLE 4.2
CD-VALUES FOR PROBLEM II WITH $\Delta x = 1/200$ AND $\epsilon = 10^{-2}$

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

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.

4.4.2. *Mutual Comparison of the FRK Methods*

On the basis of Problem II, we 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. Note 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: first, it seems advisable to set θ equal to 1 in the operator splitting (4.2), i.e., to add the complete source term to f_1; second, for $\theta = 1$, the zero step version is slightly more accurate than the two other versions.

4.4.3. *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 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 even less). With respect to the order behavior of the sequential versions, Tables 4.1 and 4.2 indicate a performance an order larger than the theoretical order 1 (note that pth order behavior means an increase of the cd-value by $\log_{10}(2^p) \approx 0.3 * p$ on halving the step size). In particular, for the (sequential) zero step method applied to Problem I we observe second-order behavior, 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 accurately than the sequential counterparts. An example is Problem III. The solution of this test example, which has also been discussed in [8, 3], develops, for small values of ϵ, a steep shock wave, which moves across the spatial domain. Following [8], we set $\epsilon = 0.003$ in our test. For this ϵ-value, a plot of the time evolution of the exact solution can be found in [8]. 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.

We see that the parallel variants yield approximately the maximal 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 [8], 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 *implicit* LM methods (i.e., BDF methods) were used and approximately 12,000 time steps if it used *explicit* LM methods (i.e., Adams predictor–corrector pair). Since these methods require (in the average) at least 1.5 (full) f-evaluations 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, such as evaluating and decomposing Jacobian matrices and solving linear systems.

4.4.4. *A Further Modification of the FRK Methods*

As we have seen in the experiments described in the preceding subsections, it is the convection term which limits the maximal stable step size of the

TABLE 4.3

CD-VALUES FOR PROBLEM III WITH $\Delta x = 1/800$ AND $\epsilon = 0.003$

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

TABLE 4.4

CD-VALUES FOR PROBLEM I OBTAINED WITH THE SEQUENTIAL AND PARALLEL FRK* METHODS, $\Delta x = 1/200$, $\epsilon = 0.1$, AND $\theta = 1$

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

(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 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))

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

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. As a matter of 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 4.4.1 and 4.4.3, using the zero step version, that is $\mathbf{c}_1 = A_1\mathbf{e}$ and $\mathbf{c}_{2,j} = \mathbf{e}, j = 1, \dots, M$ (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

$$\begin{aligned} u^{(0)} &= y_n, \\ u^{(j)} &= u^{(j-1)} + \frac{h}{M}\Phi_2(f_2, \mathbf{e}t_n + \mathbf{d}_{1,j}h, u^{(j-1)}), \qquad j = 1, \dots, M, \\ u_{n+1} &= u^{(M)} + h\Phi_1(f_1, \mathbf{e}t_n + \mathbf{d}_2 h, u^{(M)}), \end{aligned} \tag{3.1′}$$

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, $\mathbf{d}_{1,j} = 0, j = 1, \dots, M$ 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 4.4.1. Table 4.4 contains the results.

We see that if one is satisfied with a global error of 10^{-2} (i.e., cd = 2), then a step size $h \approx 1/20$ 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, similarly to the FRK methods, that the parallel version does not improve the accuracy of the sequential version since the latter also shows second-order behavior.

TABLE 4.5
CD-VALUES FOR PROBLEM III OBTAINED WITH THE SEQUENTIAL AND PARALLEL FRK* METHODS, $\Delta x = 1/800$ AND $\epsilon = 0.003$

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

TABLE 4.6a

CD-VALUES FOR THE ZERO STEP FRK METHOD WITH $\theta = 1$, APPLIED TO PROBLEM IV WITH $\Delta x = \Delta y = 10^{-2}$ AND $\epsilon = 0.1$

$h = 1/60$	$h = 1/120$	$h = 1/160$
2.4 (900 + 240)	2.9 (1320 + 480)	3.1 (1440 + 640)

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 4.4.3) we see that, for this problem, the parallel version is superior to the sequential variant. Again, the FRK* method offers the possibility of choosing the step size on the basis of accuracy considerations without being restricted by stability.

4.4.5. *Two-Dimensional Experiments*

Finally, we test the *sequential* FRK methods on the basis of a two-dimensional form of the Burgers equation and compare their behavior with the LOD method as defined in Section 4.3. For that purpose, we consider the PDE (cf. (4.1))

$$\frac{\partial u}{\partial t} = \epsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - u\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) + s(x, y, t), \qquad 0 \leqslant t \leqslant 1, \qquad (4.1')$$

defined on the unit square. Similar to the approach followed in solving the one-dimensional variant, the initial values, the Dirichlet boundary conditions, and the inhomogeneous term $s(x, y, t)$ follow from the exact solution, which is chosen to be of the form

Problem IV $\quad u(x, y, t) = \exp(-x^2 - 2y^2)\sin^2(2\pi t)$

Problem V $\quad u(x, y, t) = [(x - 0.5)^2 + 0.5y^2]\sin^2(2\pi t).$

TABLE 4.6b

CD-VALUES FOR THE LOD METHOD, APPLIED TO PROBLEM IV WITH $\Delta x = \Delta y = 10^{-2}$ AND $\epsilon = 0.1$

$h = 1/80$	$h = 1/160$	$h = 1/320$	$h = 1/640$	$h = 1/1280$
1.9 (352 + 370)	2.2 (630 + 638)	2.5 (1165 + 1216)	2.8 (2128 + 2219)	3.1 (3830 + 3834)

TABLE 4.7a
CD-VALUES FOR THE ZERO STEP FRK METHOD WITH $\theta = 1$, APPLIED TO PROBLEM V WITH $\Delta x = \Delta y = 10^{-2}$ AND $\epsilon = 0.01$

$h = 1/40$	$h = 1/80$	$h = 1/120$
2.2 (240 + 160)	2.4 (400 + 320)	3.3 (480 + 480)

Again, since the solutions are smooth in space and time, we used the standard second-order semidiscretization on a uniform grid. In all tests we set $\Delta x = \Delta y = 10^{-2}$, resulting in a system of ODEs of dimension 99^2.

For the FRK algorithm we selected the zero step version with $\theta = 1$, since this variant turned out to be most efficient (cf. Section 4.4.2). In the implementation of the LOD scheme, we used a modified Newton process to solve the (nonlinear) implicit relations in each step. For that purpose, the Jacobian matrices corresponding to each horizontal and vertical grid line were reevaluated every timestep. The results for Problem IV obtained for various (constant) stepsizes can be found in Tables 4.6a and 4.6b. Again, we list the cd-values and, in parentheses, the required numbers of f_1- and f_2-evaluations.

Note that, in the case of the LOD method, the f_1- and f_2-evaluations correspond to the one-dimensional part of the total derivative function in the x- and y-directions, respectively.

In comparing both types of methods, we should realize that they are quite different in nature, i.e., explicit versus implicit. The most important consequence of the implicitness of the LOD scheme is, of course, that apart from the f_1- and f_2-evaluations this scheme needs as many times to solve a tridiagonal linear system (or, more precisely, 99 tridiagonal systems, each of dimension 99). Moreover, the LOD scheme needs to evaluate (and store) the Jacobian matrices. This is in contrast with the explicit FRK methods, where the only computational intensive work consists in the evaluations of the right-hand side functions. Assuming that all f_1- and f_2-evaluations in both types of methods are approximately equally expensive and taking into account the aforementioned considerations with respect to the LOD method, it is clear that Tables 4.6a,b show superior behavior for the FRK methods.

TABLE 4.7b
CD-VALUES FOR THE LOD METHOD, APPLIED TO PROBLEM V WITH $\Delta x = \Delta y = 10^{-2}$ AND $\epsilon = 0.01$

$h = 1/80$	$h = 1/160$	$h = 1/320$	$h = 1/640$	$h = 1/1280$
2.1 (391 + 412)	2.4 (678 + 730)	2.7 (1232 + 1249)	3.0 (2173 + 2355)	3.3 (3832 + 3835)

We performed a similar test for Problem V and collected the results in Tables 4.7a and 4.7b.

For this example we see that the FRK method needs fewer right-hand side evaluations to yield the same global accuracy. Taking into account that the LOD method requires much additional computational effort, we may conclude that the FRK method is by far superior.

REFERENCES

1. W. F. Ames (Ed.), *Non-linear Partial Differential Equations in Engineering,* Vol. 1. Academic Press, New York (1965).
2. J. M. Burgers, A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1,** 171–199 (1948).
3. D. J. Evans and A. R. Abdullah, The group explicit method for the solution of Burgers' equation. *Computing* **32,** 239–253 (1984).
4. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations. I. Nonstiff Problems.* Springer-Verlag, Berlin (1987).
5. P. J. van der Houwen and B. P. Sommeijer, On the internal stability of explicit, m-stage Runge–Kutta methods for large m-values. *Z. Angew. Math. Mech.* **60,** 479–485 (1980).
6. A. Jameson, W. Schmidt, and E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. In: *Proceedings of the AIAA 14th Fluid and Plasma Dynamics Conference, June 23–25, 1981, Palo Alto, CA,* AIAA paper 81-1259.
7. G. I. Marchuk, *Methods of Numerical Mathematics,* 2nd ed. Springer-Verlag, New York, (1981).
8. R. F. Sincovec and N. K. Madsen, Software for nonlinear partial differential equations. *ACM Trans. Math. Software* **1,** 232–260 (1975).
9. D. A. Swayne, Time dependent boundary and interior forcing in locally one-dimensional schemes. *SIAM J. Sci. Comput.* **8,** 755–767 (1987).
10. J. G. Verwer, W. H. Hundsdorfer, and B. P. Sommeijer, Convergence properties of the Runge–Kutta–Chebyshev method. *Numer. Math.* **57,** 157–178 (1990).
11. N. N. Yanenko, *The Method of Fractional Steps.* Springer-Verlag, Berlin (1971).