MATHEMATICS OF COMPUTATION
VOLUME 60, NUMBER 202
APRIL 1993, PAGES 591–616

RUNGE-KUTTA METHODS AND LOCAL UNIFORM GRID REFINEMENT

R. A. TROMPERT AND J. G. VERWER

ABSTRACT. Local uniform grid refinement (LUGR) is an adaptive grid technique for computing solutions of partial differential equations possessing sharp spatial transitions. Using nested, finer-and-finer uniform subgrids, the LUGR technique refines the space grid locally around these transitions, so as to avoid discretization on a very fine grid covering the entire physical domain. This paper examines the LUGR technique for time-dependent problems when combined with static regridding. Static regridding means that in the course of the time evolution, the space grid is adapted at discrete times. The present paper considers the general class of Runge-Kutta methods for the numerical time integration. Following the method of lines approach, we develop a mathematical framework for the general Runge-Kutta LUGR method applied to multispace-dimensional problems. We hereby focus on parabolic problems, but a considerable part of the examination applies to hyperbolic problems as well. Much attention is paid to the local error analysis. The central issue here is a "refinement condition" which is to underly the refinement strategy. By obeying this condition, spatial interpolation errors are controlled in a manner that the spatial accuracy obtained is comparable to the spatial accuracy on the finest grid if this grid would be used without any adaptation. A diagonally implicit Runge-Kutta method is discussed for illustration purposes, both theoretically and numerically.

1. INTRODUCTION

Local uniform grid refinement (LUGR) is an adaptive grid technique for computing solutions of partial differential equations (PDEs) possessing sharp spatial transitions. Using nested, finer-and-finer, uniform subgrids, the LUGR technique refines the space grid locally around these transitions to avoid discretization on a very fine grid covering the entire domain. In this paper we examine the LUGR technique for time-dependent problems. Thus, typical solutions aimed at are those possessing sharp moving transitions, such as steep fronts, emerging layers, moving pulses, etc. For time-dependent problems, LUGR is combined with static regridding. Static regridding means that in the course of the time evolution, the space grid is adapted at discrete times.

We consider Runge-Kutta methods for the time integration and, following the method of lines approach, develop a mathematical framework for the general Runge-Kutta LUGR method. We hereby focus on parabolic problems, but

Received by the editor February 20, 1991 and, in revised form, May 4, 1992.
1991 *Mathematics Subject Classification.* Primary 65M50; Secondary 65M20.
Key words and phrases. Partial differential equations, numerical mathematics, time-dependent problems, Runge-Kutta methods, adaptive grid methods, error analysis.

0025-5718/93 $1.00 + $.25 per page

a considerable part of the discussion applies to hyperbolic problems as well. The present paper is a continuation of [12] which deals with the implicit Euler method. Here we discuss how the ideas developed in [12] are extended to the general Runge-Kutta case. As in [12], much attention is paid to the local error analysis. The central issue here is the "refinement condition", which is to underlie the refinement strategy. By obeying this condition, spatial interpolation errors are controlled in a manner such that the spatial accuracy obtained is comparable to the spatial accuracy on the finest grid if this grid would be used without any adaptation. Nonnumerical subjects, such as the data structure and the memory use, are not discussed here. These are the same as in [11]. For related earlier work on LUGR methods, we refer to Berger and Oliger [3], Gropp [6, 7], Arney and Flaherty [2], and references therein.

Section 2 is devoted to the method formulation. Here we develop the mathematical framework that enables us to give a concise description of the Runge-Kutta LUGR method. In §3 we set up a general error scheme, which is further elaborated in §§4 and 5. Section 4 briefly addresses the stability issue, while §5 is devoted to the local error analysis. Here we derive the important "refinement condition". Under a natural assumption on the Runge-Kutta method, we next prove that the "uniform in h" temporal order of the method is at least equal to the stage order. Noteworthy is that §§3–5 apply to the whole class of Runge-Kutta methods. As a result, the outcome of the analysis is of a general nature, so that for a specific Runge-Kutta method further elaboration is needed. Such an elaboration is presented in the remainder of the paper for a 3-stage diagonally implicit Runge-Kutta (DIRK) method. In §6 attention is given to the order reduction phenomenon and to the manner in which to implement the "refinement condition" for this specific method. Section 7 deals with two numerical examples in two space dimensions. Finally, we conclude the paper with §8 discussing two important matters of practical interest.

2. The general method formulation

2.1. **The Runge-Kutta method.** Consider the initial value problem for a standard ODE system,

$$\frac{d}{dt}U(t) = F(t, U(t)), \quad 0 < t \le T, \qquad U(0) = U^0. \tag{2.1}$$

The general one-step, s-stage RK scheme for the numerical solution of (2.1) is denoted by

$$U^{(i)} = U^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F(t_{n-1} + c_j \tau, U^{(j)}), \qquad 1 \le i \le s, \tag{2.2}$$

$$U^{n} = U^{n-1} + \tau \sum_{i=1}^{s} b_i F(t_{n-1} + c_i \tau, U^{(i)}), \tag{2.3}$$

where the stepsize τ may vary with n. Superscripts will refer to time, while superscripts in parentheses are used for approximations at intermediate stages. As usual, we suppose $c_i = a_{i1} + \cdots + a_{is}$. In the sequel it is convenient to combine (2.2)–(2.3) into one formula. Denote $a_{s+1i} = b_i$, $1 \le i \le s$, $U^{(s+1)} = U^n$; then

we rewrite (2.2)–(2.3) as

$$U^{(i)} = U^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F(t_{n-1} + c_j \tau, U^{(j)}), \qquad 1 \le i \le s+1. \tag{2.4}$$

2.2. **The semidiscrete problem.** Consider an initial-boundary value problem in d space dimensions,

$$u_t = L(t, u), \quad 0 < t \le T, \qquad u(\underline{x}, 0) = u_0(\underline{x}), \tag{2.5}$$

where L is supposed to be of at most second order and provided with appropriate boundary conditions on the boundary $\partial\Omega$ of the space domain Ω. The boundary is taken to be locally parallel to the coordinate axes. The function $u(\underline{x}, t)$ may be vector-valued and is supposed to exist uniquely and to be as often differentiable on $(\Omega \cup \partial\Omega) \times [0, T]$ as the numerical analysis requires.

LUGR methods use local uniform grids whose size and number usually vary in time. Therefore, LUGR methods generate a sequence of operations on vectors in vector spaces with a variable dimension. This complicates the error analysis. In [12] we got around this problem by expanding the fine grids in the mathematical formulation of the method, so that the entire domain is covered. Also here we use this "grid expansion". Temporal integration then takes place on one part of the expanded fine grid and interpolation on the other. Note that this grid expansion does not take place in the actual application but only in the mathematical formulation of the method. Nevertheless, the results of the error analysis presented remain valid for the method as applied.

Let $l \in \mathbb{N}^+$. For $k = 1, \ldots, l$ we introduce uniform space grids ω_k, where each ω_k is supposed to cover the whole of the interior domain Ω. The grid ω_k has no points on $\partial\Omega$. The grid ω_1 is called the base grid and, given this grid, ω_2 is obtained from ω_1 by bisecting all sides of all cells of ω_1, etc. With (2.5) we now associate on each ω_k a real Cauchy problem for an explicit ODE system in $\mathbb{R}^{d_k}$,

$$\frac{d}{dt} U_k(t) = F_k(t, U_k(t)), \quad 0 < t \le T, \qquad U_k(0) = U_k^0, \tag{2.6}$$

defined by a finite difference space discretization of (2.5) and its boundary conditions. Thus, U_k and F_k are vectors representing the values of grid functions defined on the grid ω_k. Each component of U_k and F_k itself is vector-valued if u is vector-valued. The boundary conditions have been worked into the semidiscrete system by eliminating semidiscrete values at $\partial\Omega$. The dimension d_k is determined by the spatial dimension, the grid spacing, and the number of PDEs. The initial vector U_k^0 for (2.6) is supposed to be exact.

In the sequel we let S_k with $\dim(S_k) = d_k$ denote the grid function space. S_k coincides with $\mathbb{R}^{d_k}$ and U_k, F_k are elements of S_k. Let $u_k(t) \in S_k$ represent the natural (nodal wise) restriction of $u(\underline{x}, t)$ to ω_k. In S_k the fully continuous problem (2.5) and the semidiscrete problem (2.6) are related by the local spatial discretization error

$$\alpha_k(t) = \frac{d}{dt} u_k(t) - F_k(t, u_k(t)), \qquad 0 \le t \le T. \tag{2.7}$$

In particular, u_k and α_k are sufficiently often differentiable with respect to t, and $\alpha_k(t)$ has the order of consistency of the finite difference scheme. Finally,

we note once more that we consider elements $u_k(t)$, $U_k(t) \in S_k$ defined on space grids ω_k which cover the entire physical domain Ω.

2.3. **The multilevel multistage RK method.** Starting at the coarse base grid ω_1, this method successively integrates on subgrids of ω_k for $k = 2, \ldots, l$ over the same time interval $[t_{n-1}, t_n]$. Characteristic for the method is that subgrids, henceforth called the integration domains, are nested and that, in a sense, on each domain a new initial-boundary value problem is solved. Required initial values are defined by interpolation from the next coarser integration domain or taken from a possibly existing one from the previous time interval. Boundary values required at internal boundaries are also interpolated from the next coarser integration domain. At each level of refinement, the domains are allowed to be disjunct and thus may consist of two or more subdomains. The nesting is continued up to a level fine enough to resolve the anticipated fine-scale structure. This means that, given ω_1, the integer l must be chosen sufficiently large. Having completed the integration on the finest, lth level integration domain, the process is repeated for the next time interval $[t_n, t_{n+1}]$ by again starting from ω_1. We note that all refined subgrids computed at forward time are kept in storage as they are used for step continuation. Further, for step continuation always the most accurate solution is used that is available.

The process described above is defined by the formulas

$$U_1^{(i)} = R_{l1} U_l^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_1(t_{n-1} + c_j \tau, U_1^{(j)}), \qquad 1 \le i \le s+1, \quad k = 1, \tag{2.8a}$$

$$U_k^{(i)} = D_k^n \left[R_{lk} U_l^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_k(t_{n-1} + c_j \tau, U_k^{(j)}) \right] + (I_k - D_k^n)[P_{k-1\,k} U_{k-1}^{(i)} + b_k^{(i)}], \qquad 1 \le i \le s+1, \quad 2 \le k \le l, \tag{2.8b}$$

where $U_k^{(s+1)} = U_k^n \in S_k$ is the approximation to $u_n(t_n)$ at the grid ω_k, $U_k^{(i)} \in S_k$ is the ith intermediate approximation at ω_k, $I_k : S_k \to S_k$ is the unit matrix, $D_k^n : S_k \to S_k$ is a diagonal matrix with entries $(D_k^n)_{ii}$ either unity or zero, $R_{lk} : S_l \to S_k$, $k = 1, \ldots, l$, is the natural restriction operator from ω_l to ω_k with $R_{ll} = I_l$, $P_{k-1\,k} : S_{k-1} \to S_k$, $k = 2, \ldots, l$, is an interpolation operator from ω_{k-1} to ω_k, and $b_k^{(i)} \in S_k$ contains time-dependent terms emanating from the physical boundary $\partial\Omega$.

The nesting property of the integration domains is induced by the grid strategy. This strategy determines at which nodes integration or interpolation is carried out and defines the diagonal matrices D_k^n. If at a node integration is to take place, then the associated diagonal entry $(D_k^n)_{ii}$ is defined as $(D_k^n)_{ii} = 1$. For all remaining interpolation nodes, $(D_k^n)_{ii} = 0$. The nesting property itself cannot be recovered from the above formulation, as this is hidden in the actual definition of D_k^n.

The interpolation step on level $k \ge 2$ stands on its own and is represented by

$$(I_k - D_k^n) U_k^{(i)} = (I_k - D_k^n)[P_{k-1\,k} U_{k-1}^{(i)} + b_k^{(i)}], \qquad 1 \le i \le s+1. \tag{2.9}$$

The grid function $b_k^{(i)}$ plays an auxiliary role. We need to include it as boundary conditions have been worked into (2.6) (method of lines). For the analysis presented, $b_k^{(i)}$ plays no role (it contains merely time-dependent terms and does not depend on $u(\underline{x}, t)$). Likewise, the integration step on the integration domain of level k is represented by

$$D_k^n U_k^{(i)} = D_k^n \left[R_{lk} U_l^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_k(t_{n-1} + c_j \tau, U_k^{(j)}) \right], \quad 1 \le i \le s+1, \tag{2.10}$$

where, according to (2.8a), $D_1^n = I_1$. Values at, or beyond, internal boundaries needed in the function evaluation in (2.10) are defined by (2.9), for each RK stage. Hence, owing to the internal boundaries, (2.10) cannot be considered uncoupled from the interpolation (2.9). Also observe that at each grid level the integration has the fine grid solution $D_k^n R_{lk} U_l^{n-1}$ as initial function. Note that if we substitute the implicit Euler formula in (2.10), the scheme of [12] is obtained.

In (2.8) the approximations $U_k^{(i)}$ are defined on the whole of the grids ω_k and thus are also elements of S_k. Consequently, for any $k \ge 2$ interpolation is considered to take place on the whole of ω_k, which is costly. In actual application, the interpolations are therefore restricted to the nested integration domains. This point will be discussed later in the paper. For the time being, it is assumed that the numerical solutions are indeed generated as grid functions in S_k (grid expansion).

In (2.8) the number of grid levels l is fixed a priori, independent of time. In applications this fixed-level mode of operation may be inefficient. For example, if a solution steepens up in time, fewer levels are needed in the initial integration than at later times. Consequently, at early times l must be taken larger than necessary, which is not efficient. On the other hand, the solution may also become less steep, which again makes a fixed l inefficient. Obviously, the method should be capable of working with a variable l. For this variable-level mode of operation (2.8) requires a modification. Let l_{n-1}, l_n denote the number of levels from t_{n-1} to t_n and t_n to t_{n+1}, respectively. Then, for the step from t_{n-1} to t_n, (2.8) is modified to

$$U_1^{(i)} = R_{l_{n-1}1} U_{l_{n-1}}^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_1(t_{n-1} + c_j \tau, U_1^{(j)}), \quad 1 \le i \le s+1, \ k = 1, \tag{2.11a}$$

$$U_k^{(i)} = D_k^n \left[R_{l_{n-1}k} U_{l_{n-1}}^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_k(t_{n-1} + c_j \tau, U_k^{(j)}) \right] + (I_k - D_k^n)[P_{k-1\,k} U_{k-1}^{(i)} + b_k^{(i)}], \quad 1 \le i \le s+1, \ 2 \le k \le l_{n-1}, \tag{2.11b}$$

and, provided $l_n > l_{n-1}$, for $k = l_{n-1} + 1, \ldots, l_n$ we have

$$U_k^{(i)} = P_{k-1\,k} U_{k-1}^{(i)} + b_k^{(i)}, \qquad 1 \le i \le s+1. \tag{2.11c}$$

Consequently, if the number of levels should increase for use in the next step, then so-called full interpolations (2.11c) are carried out at the end of the current step, so that the required initial function, which is to be taken from the highest grid level that will be used, is always available. If $l_n \leq l_{n-1}$, then (2.11c) is omitted and nothing really changes. Full interpolation is necessary only when the solution steepens up in time. Because we will let the l_n depend exclusively on the spatial steepness, and because $\max_n\{l_n\}$ is finite, full interpolation is carried out only for a finite number of steps, uniformly in τ. Hence full interpolation cannot have a strongly diminishing effect on global accuracy. Like the matrices D_k^n, the actual choice for l_n is part of the adaptation strategy.

We conclude this section with a minor modification for certain RK methods. Above, D_k^n depends only on the step number n and the level index k, and not on the stages. There exist RK methods for which all coefficients a_{1j} are zero, trivially so for all explicit methods, but for example also for the implicit Lobatto IIIA-methods ($s = 2$ yields the familiar trapezoidal rule). If this is the case, then it is more natural to define for all grid levels the 1st stage value as

$$U_k^{(1)} = R_{lk} U_l^{n-1} \tag{2.12}$$

to avoid interpolation. This means that at stage one, D_k^n is to be replaced by the unit matrix I_k.

3. The general error scheme

To save space, (2.11) is rewritten as

$$\begin{aligned} U_k^{(i)} = {} & D_k^n \left[R_{l_{n-1}k} U_{l_{n-1}}^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_k(t_{n-1} + c_j\tau,\, U_k^{(j)}) \right] \\ & + (I_k - D_k^n)[P_{k-1\,k} U_{k-1}^{(i)} + b_k^{(i)}], \qquad 1 \leq i \leq s+1,\ 1 \leq k \leq l_n. \end{aligned} \tag{3.1}$$

Note that $D_1^n = I_1$ and $D_k^n = 0$ if $k > l_{n-1}$. Further, if $a_{1j} = 0$ $(1 \leq j \leq s)$, then D_k^n is to be replaced by I_k for $i = 1$, but only for $1 \leq k \leq l_{n-1}$. The rewriting of (2.11) into (3.1) introduces variables not existing in reality, viz., the grid functions $U_0^{(i)}$, $b_1^{(i)}$ and the operators P_{01} and $R_{l_{n-1}k}$ for $k > l_{n-1}$. Formally we can use (3.1) owing to the definition of D_k^n.

The derivation of the error scheme parallels that in [12]. Consider the perturbed scheme

$$\begin{aligned} \tilde{U}_k^{(i)} = {} & D_k^n \left[R_{l_{n-1}k} \tilde{U}_{l_{n-1}}^{n-1} + \tau \sum_{j=1}^{s} a_{ij} F_k(t_{n-1} + c_j\tau,\, \tilde{U}_k^{(j)}) \right] \\ & + (I_k - D_k^n)[P_{k-1\,k} \tilde{U}_{k-1}^{(i)} + b_k^{(i)}] + r_k^{(i)}, \\ & 1 \leq i \leq s+1,\ 1 \leq k \leq l_n, \end{aligned} \tag{3.2}$$

with the local perturbations $r_k^{(i)}$ still arbitrary. Introduce the errors

$$e_k^n = \tilde{U}_k^n - U_k^n, \quad e_k^{(i)} = \tilde{U}_k^{(i)} - U_k^{(i)}, \qquad 1 \leq i \leq s+1,\ 1 \leq k \leq l_n, \tag{3.3}$$

and subtract (3.1) and (3.2) to obtain

$$e_k^{(i)} = D_k^n \left[R_{l_{n-1}k} e_{l_{n-1}}^{n-1} + \tau \sum_{j=1}^{s} a_{ij} M_k^{(j)} e_k^{(j)} \right] + (I_k - D_k^n) P_{k-1\,k} e_{k-1}^{(i)} + r_k^{(i)}, \qquad 1 \le i \le s+1,\ 1 \le k \le l_n. \tag{3.4}$$

Here, $M_k^{(j)}$ is the integrated Jacobian matrix resulting from the use of the mean value theorem:

$$F_k(t_{n-1} + c_j\tau, \tilde{U}_k^{(j)}) - F_k(t_{n-1} + c_j\tau, U_k^{(j)}) = M_k^{(j)}(\tilde{U}_k^{(j)} - U_k^{(j)}), \tag{3.5a}$$

$$M_k^{(j)} = \int_0^1 F'(t_{n-1} + c_j\tau, \theta\tilde{U}_k^{(j)} + (1-\theta)U_k^{(j)})\, d\theta. \tag{3.5b}$$

We next introduce the Kronecker product notation. Let E_{s+1} be the unit matrix of order $s+1$ and denote $e = [1, \dots, 1]^T \in \mathbb{R}^{s+1}$. Introduce the augmented vectors

$$\mathbf{e}_k^n = [e_k^{(1)T}, \dots, e_k^{(s+1)T}]^T, \qquad \mathbf{r}_k^n = [r_k^{(1)T}, \dots, r_k^{(s+1)T}]^T \tag{3.6}$$

in the augmented space $\mathbf{S}_k = \mathbb{R}^{(s+1)d_k}$ and the matrix operators

$$\begin{aligned} &\mathbf{R}_{l_{n-1}k}: \mathbf{S}_{l_{n-1}} \to \mathbf{S}_k, \qquad \mathbf{R}_{l_{n-1}k} = E_{s+1} \otimes R_{l_{n-1}k} = \operatorname{diag}(R_{l_{n-1}k}), \\ &\mathbf{P}_{k-1\,k}: \mathbf{S}_{k-1} \to \mathbf{S}_k, \qquad \mathbf{P}_{k-1\,k} = E_{s+1} \otimes P_{k-1\,k} = \operatorname{diag}(P_{k-1\,k}), \\ &\mathbf{I}_k: \mathbf{S}_k \to \mathbf{S}_k, \qquad \mathbf{I}_k = E_{s+1} \otimes I_k = \operatorname{diag}(I_k). \end{aligned} \tag{3.7}$$

Define $\mathbf{D}_k^n: \mathbf{S}_k \to \mathbf{S}_k$, $\mathbf{D}_k^n = \operatorname{diag}(I_k, D_k^n, \dots, D_k^n)$ if $a_{1j} = 0$, $1 \le j \le s$ (cf. (2.12)), and otherwise $\operatorname{diag}(D_k^n)$. Finally, we introduce the augmented Jacobian operators

$$\mathbf{M}_k^n = \begin{pmatrix} a_{11}M_k^{(1)} & a_{12}M_k^{(2)} & \cdots & a_{1s}M_k^{(s)} & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{s1}M_k^{(1)} & a_{s2}M_k^{(2)} & \cdots & a_{ss}M_k^{(s)} & 0 \\ a_{s+1\,1}M_k^{(1)} & a_{s+1\,2}M_k^{(2)} & \cdots & a_{s+1\,s}M_k^{(s)} & 0 \end{pmatrix}, \tag{3.8}$$

$$\mathbf{Z}_k^n = \mathbf{I}_k - \tau \mathbf{D}_k^n \mathbf{M}_k^n,$$

so that (3.4) can now be written in the compact form

$$\mathbf{Z}_k^n \mathbf{e}_k^n = \mathbf{D}_k^n \mathbf{R}_{l_{n-1}k}(e \otimes e_{l_{n-1}}^{n-1}) + (\mathbf{I}_k - \mathbf{D}_k^n)\mathbf{P}_{k-1\,k}\mathbf{e}_{k-1}^n + \mathbf{r}_k^n, \qquad 1 \le k \le l_n. \tag{3.9}$$

In (3.9) we deal with an inner and outer recursion connected, respectively, with the grid refinement index k and time stepping index n. Introduce

$$\mathbf{X}_k^n = (\mathbf{Z}_k^n)^{-1}(\mathbf{I}_k - \mathbf{D}_k^n)\mathbf{P}_{k-1\,k}, \qquad \mathbf{\Gamma}_k^n = (\mathbf{Z}_k^n)^{-1}\mathbf{D}_k^n\mathbf{R}_{l_{n-1}k}, \qquad \boldsymbol{\phi}_k^n = (\mathbf{Z}_k^n)^{-1}\mathbf{r}_k^n, \tag{3.10}$$

where $k = 1, \dots, l_n$. Note that $\mathbf{Z}_k^n = \mathbf{I}_k$, $\mathbf{X}_k^n = \mathbf{P}_{k-1\,k}$, $\mathbf{\Gamma}_k^n = \mathbf{0}$, $\boldsymbol{\phi}_k^n = \mathbf{r}_k^n$ for the full interpolation levels $k = l_{n-1}+1, \dots, l_n$. Using (3.10), we rewrite (3.9) as

$$\mathbf{e}_k^n = \mathbf{X}_k^n \mathbf{e}_{k-1}^n + \mathbf{\Gamma}_k^n(e \otimes e_{l_{n-1}}^{n-1}) + \boldsymbol{\phi}_k^n, \qquad k = 1, \dots, l_n. \tag{3.11}$$

An elementary calculation then leads to the final form

$$\mathbf{e}_k^n = \mathbf{G}_k^n(e \otimes e_{l_{n-1}}^{n-1}) + \boldsymbol{\psi}_k^n, \qquad k = 1, \ldots, l_n, \tag{3.12}$$

where $\mathbf{G}_k^n$ and $\boldsymbol{\psi}_k^n$ themselves are also defined by recursions:

$$\mathbf{G}_1^n = \boldsymbol{\Gamma}_1^n, \quad \mathbf{G}_j^n = \mathbf{X}_j^n \mathbf{G}_{j-1}^n + \boldsymbol{\Gamma}_j^n, \qquad j = 2, \ldots, k, \tag{3.13}$$

$$\boldsymbol{\psi}_1^n = \boldsymbol{\phi}_1^n, \quad \boldsymbol{\psi}_j^n = \mathbf{X}_j^n \boldsymbol{\psi}_{j-1}^n + \boldsymbol{\phi}_j^n, \qquad j = 2, \ldots, k. \tag{3.14}$$

Equation (3.12) describes the error propagation for increasing levels within one complete time step. When it is used as error recursion in time, we put $k = l_n$, as we use the highest-level approximations $U_{l_{n-1}}^{n-1}$, $U_{l_n}^n, \ldots$ for step continuation. Hence,

$$\mathbf{e}_{l_n}^n = \mathbf{G}_{l_n}^n(e \otimes e_{l_{n-1}}^{n-1}) + \boldsymbol{\psi}_{l_n}^n, \qquad n = 1, 2, \ldots, \tag{3.15}$$

is the final error scheme for the highest-level approximations. Similar as in the standard application of the RK method (single-level multistage), our main interest concerns the $(s+1)$st component vector. Note that the formulation (3.15) supposes that $U_{l_n}^n$ is taken as output rather than $U_{l_{n-1}}^n$.

4. Remarks on stability

In [12] we have presented a comprehensive analysis of the stability of the multilevel implicit Euler method. The multilevel multistage RK formulas are not so amenable to a comprehensive stability analysis. A technical difficulty arises from the property that at any RK stage, nonphysical boundary values are defined by interpolating the solution of the corresponding stage from the next coarser grid. This implies that the internal RK stages play a role in the stability analysis, even for constant-coefficient linear problems. On the other hand, we believe this role is minor, and that in applications one encounters the same step-by-step stability as on a single grid, as long as interpolation takes place in low-error regions. In this paper no further attention is paid to stability analysis. Instead, we refer to the preprint [10] for some preliminary remarks on stability and proceed with the local error analysis, which is to reveal how to define the adaptation strategy for choosing the spatial integration domains at the various refinement levels. Obviously, this is one of the main issues in the analysis, implementation and application of adaptive grid methods.

5. The local errors

5.1. **Preliminaries.** In the following, $\|\cdot\|$ denotes the conventional maximum norm. We use the maximum norm since this norm is most natural for implementing adaptation strategies. Note that $\|\cdot\|$ stands for the maximum norm in any space S_k or $\mathbf{S}_k$ under consideration, while the same symbol will be used for operators. We will examine the total local error $\boldsymbol{\psi}_k^n$ obtained by associating the local perturbations $\mathbf{r}_k^n$ with the true PDE solution. Note that the global errors $\mathbf{e}_k^n$ then become global discretization errors, viz.,

$$\mathbf{e}_k^n = \mathbf{u}_k^n - \mathbf{U}_k^n, \qquad n = 0, 1, \ldots, \ 1 \le k \le l_n. \tag{5.1}$$

For clarity, we will henceforth consistently call $\boldsymbol{\psi}_k^n$ the total local error, whereas $\mathbf{r}_k^n$ will be consistently called a residual, so as to distinguish it from $\boldsymbol{\psi}_k^n$. Note

that ψ_k^n can be interpreted as the kth-level global error after one time step starting from the true PDE solution (put $e_{l_{n-1}}^{n-1} = 0$ in (3.12)).

We have tacitly used the natural assumption that any augmented RK operator $\mathbf{Z}_k^n$ that occurs is invertible (under appropriate conditions on τ and ω_k). We thus may introduce the following bound:

$$\|(\mathbf{Z}_k^n)^{-1}\mathbf{v}\| \leq C\|\mathbf{v}\|, \quad \forall \mathbf{v} \in \mathbf{S}_k, \tag{5.2}$$

where $C \geq 1$ denotes a constant independent of τ and ω_k, while τ itself satisfies $\tau \leq \tau_0$ with τ_0 possibly depending on $\omega_{l_{n-1}}$. The constant C and stepsize bound τ_0 are assumed to take on appropriate values (C close to 1 and τ_0 not unduly restrictive). As in [12], the aim of the error analysis is to derive a refinement condition that distributes space discretization and interpolation errors in such a way that the local spatial accuracy obtained on $\omega_{l_{n-1}}$ is comparable to the local spatial accuracy if this grid would be used without any adaptation. Assuming a stable time-stepping process, this will then also be true for the global spatial accuracy.

5.2. **The local error ψ_k^n.** Replace, in the perturbed scheme (3.2), all $\tilde{U}_k^{(i)}$-values by the corresponding PDE solution values $u_k^{(i)}$. Then, in the space $\mathbf{S}_k$, the resulting residual $\mathbf{r}_k^n$ can be expressed as

$$\mathbf{r}_k^n = \mathbf{D}_k^n(\boldsymbol{\beta}_k^n + \tau\boldsymbol{\sigma}_k^n) + (\mathbf{I}_k - \mathbf{D}_k^n)\boldsymbol{\gamma}_k^n, \tag{5.3}$$

where

$$\boldsymbol{\beta}_k^n = [\beta_k^{(1)T}, \ldots, \beta_k^{(s)T}, \beta_k^{(s+1)T}]^T, \tag{5.4a}$$

$$\boldsymbol{\sigma}_k^n = (A \otimes I_k)[\alpha_k^{(1)T}, \ldots, \alpha_k^{(s)T}, \alpha_k^{(s+1)T}]^T, \tag{5.4b}$$

$$\boldsymbol{\gamma}_k^n = [\gamma_k^{(1)T}, \ldots, \gamma_k^{(s)T}, \gamma_k^{(s+1)T}]^T. \tag{5.4c}$$

The component $\beta_k^{(i)}$ is the PDE residual defined for the ith RK stage:

$$\beta_k^{(i)} = u_k(t_{n-1} + c_i\tau) - u_k(t_{n-1}) - \tau\sum_{j=1}^{s} a_{ij}\frac{d}{dt}u_k(t_{n-1} + c_j\tau). \tag{5.5}$$

The component $\alpha_k^{(i)}$ is the PDE residual defined by the semidiscretization:

$$\alpha_k^{(i)} = \frac{d}{dt}u_k(t_{n-1} + c_i\tau) - F_k(t_{n-1} + c_i\tau, u_k(t_{n-1} + c_i\tau)). \tag{5.6}$$

Following common use, $\boldsymbol{\alpha}_k^n$ and likewise $\boldsymbol{\sigma}_k^n$ and their components, will also be called local space discretization error. The matrix A represents the $(s+1)\times(s+1)$ Butcher matrix of RK coefficients a_{ij} whose $(s+1)$st column is zero. Hence, the ith component $\sigma_k^{(i)}$ of $\boldsymbol{\sigma}_k^n$ is given by $\sigma_k^{(i)} = \sum_{j=1}^{s} a_{ij}\alpha_k^{(j)}$. Finally, the component $\gamma_k^{(i)}$ is the residual defined by the interpolation,

$$\gamma_k^{(i)} = u_k(t_{n-1} + c_i\tau) - P_{k-1\,k}u_{k-1}(t_{n-1} + c_i\tau) - b_k(t_{n-1} + c_i\tau), \tag{5.7}$$

and $\gamma_k^{(i)}$ and $\boldsymbol{\gamma}_k^n$ will also be called interpolation error. Observe that any component vector

$$r_k^{(i)} = D_k^n(\beta_k^{(i)} + \tau\sigma_k^{(i)}) + (I_k - D_k^n)\gamma_k^{(i)} \tag{5.8}$$

of $\mathbf{r}_k^n$ is now determined completely by the true PDE solution $u = u(\underline{x}, t)$. Thus, $r_k^{(i)}$ can be expanded in a Taylor series, assuming sufficient differentiability.

We are now ready to determine the local error $\boldsymbol{\psi}_k^n$ defined by recursion (3.14). Assuming

$$\prod_{i=k}^{k+1} \mathbf{X}_k^n = \mathbf{I}_k, \qquad k = 1, \ldots, l_n, \tag{5.9}$$

and using (3.10) and (5.3), we get

$$\boldsymbol{\psi}_k^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} \mathbf{X}_i^n \right) (\mathbf{Z}_j^n)^{-1} [\mathbf{D}_j^n (\boldsymbol{\beta}_j^n + \tau \boldsymbol{\sigma}_j^n) + (\mathbf{I}_j - \mathbf{D}_j^n) \boldsymbol{\gamma}_j^n]. \tag{5.10}$$

A natural splitting into a spatial and a temporal local error is

$$\boldsymbol{\psi}_k^n = \boldsymbol{\psi}_{k,s}^n + \boldsymbol{\psi}_{k,t}^n, \tag{5.11}$$

where

$$\boldsymbol{\psi}_{k,s}^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} \mathbf{X}_i^n \right) (\mathbf{Z}_j^n)^{-1} [\tau \mathbf{D}_j^n \boldsymbol{\sigma}_j^n + (\mathbf{I}_j - \mathbf{D}_j^n) \boldsymbol{\gamma}_j^n], \tag{5.12}$$

$$\boldsymbol{\psi}_{k,t}^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} \mathbf{X}_i^n \right) (\mathbf{Z}_j^n)^{-1} \mathbf{D}_j^n \boldsymbol{\beta}_j^n. \tag{5.13}$$

The local space error $\boldsymbol{\psi}_{k,s}^n$ contains only contributions from the spatial approximation, viz., local space discretization errors $\boldsymbol{\sigma}_j^n$ and spatial interpolation errors $\boldsymbol{\gamma}_j^n$. The local time error $\boldsymbol{\psi}_{k,t}^n$ contains only contributions $\boldsymbol{\beta}_j^n$ from the time integration. Hence, in view of the splitting (5.11), for the spatial local error analysis we may restrict ourselves to $\boldsymbol{\psi}_{k,s}^n$ and for the temporal local error analysis to $\boldsymbol{\psi}_{k,t}^n$.

5.3. **The local space error $\boldsymbol{\psi}_{k,s}^n$.** We rewrite $\boldsymbol{\psi}_{k,s}^n$ as

$$\begin{aligned} \boldsymbol{\psi}_{k,s}^n &= (\mathbf{Z}_k^n)^{-1} (\mathbf{I}_k - \mathbf{D}_k^n) \mathbf{P}_{k-1\,k} \\ &\quad * \sum_{j=1}^{k-1} \left(\prod_{i=k-1}^{j+1} \mathbf{X}_i^n \right) (\mathbf{Z}_j^n)^{-1} [\tau \mathbf{D}_j^n \boldsymbol{\sigma}_j^n + (\mathbf{I}_j - \mathbf{D}_j^n) \boldsymbol{\gamma}_j^n] \\ &\quad + (\mathbf{Z}_k^n)^{-1} [\tau \mathbf{D}_k^n \boldsymbol{\sigma}_k^n + (\mathbf{I}_k - \mathbf{D}_k^n) \boldsymbol{\gamma}_k^n] \\ &= (\mathbf{Z}_k^n)^{-1} [\tau \mathbf{D}_k^n \boldsymbol{\sigma}_k^n + (\mathbf{I}_k - \mathbf{D}_k^n) \boldsymbol{\rho}_k^n], \end{aligned} \tag{5.14}$$

where

$$\boldsymbol{\rho}_1^n = \mathbf{0} \quad \text{and} \quad \boldsymbol{\rho}_k^n = \boldsymbol{\gamma}_k^n + \mathbf{P}_{k-1\,k} \boldsymbol{\psi}_{k-1,s}^n, \qquad k = 2, \ldots, l_n. \tag{5.15}$$

In (5.14), the local space discretization error $\mathbf{D}_k^n \boldsymbol{\sigma}_k^n$, defined at the level k integration domain, is separated from the local spatial error part $(\mathbf{I}_k - \mathbf{D}_k^n) \boldsymbol{\rho}_k^n$ outside this domain. Note that $\boldsymbol{\rho}_k^n$ contains the level k interpolation error $\boldsymbol{\gamma}_k^n$ and the prolongated local space error $\mathbf{P}_{k-1\,k} \boldsymbol{\psi}_{k-1,s}^n$. At the full interpolation levels, (5.14) simplifies to

$$\boldsymbol{\psi}_{k,s}^n = \boldsymbol{\gamma}_k^n + \mathbf{P}_{k-1\,k} \boldsymbol{\psi}_{k-1,s}^n, \qquad k = l_{n-1} + 1, \ldots, l_n. \tag{5.16}$$

The separation of errors in (5.14) enables us to formulate the important refinement condition

$$\|(\mathbf{Z}^n_{l_{n-1}})^{-1}(\mathbf{I}_{l_{n-1}} - \mathbf{D}^n_{l_{n-1}})\boldsymbol{\rho}^n_{l_{n-1}}\| \le c\|(\mathbf{Z}^n_{l_{n-1}})^{-1}\tau\mathbf{D}^n_{l_{n-1}}\boldsymbol{\sigma}^n_{l_{n-1}}\|, \tag{5.17}$$

where $c > 0$ denotes a threshold factor to be specified later. Substitution into (5.14) yields

$$\|\boldsymbol{\psi}^n_{l_{n-1},s}\| \le (1+c)\|(\mathbf{Z}^n_{l_{n-1}})^{-1}\tau\mathbf{D}^n_{l_{n-1}}\boldsymbol{\sigma}^n_{l_{n-1}}\|. \tag{5.18}$$

Hence, apart from the factor $(1+c)$, the local space error at the finest level is bounded by the local space discretization error on its integration domain. By imposing (5.17), we have virtually removed the error contribution from interpolation committed on all levels $k \le l_{n-1}$. Inequality (5.18) is in agreement with our goal of developing an adaptation strategy that generates integration domains in such a way that the spatial accuracy obtained on the finest level is comparable to that obtained without adaptation.

The refinement condition (5.17) implies constraints on the matrices $\mathbf{D}^n_k$ for $2 \le k \le l_{n-1}$. These constraints follow from the following derivation. Let, for brevity, $l = l_{n-1}$. By a simple calculation [12], we can rewrite $\boldsymbol{\rho}^n_l$ as

$$\boldsymbol{\rho}^n_l = \boldsymbol{\lambda}^n_l + \mathbf{P}_{l-1\,l}\sum_{k=2}^{l-1}\left(\prod_{i=l-1}^{k+1}\mathbf{X}^n_i\right)(\mathbf{Z}^n_k)^{-1}(\mathbf{I}_k - \mathbf{D}^n_k)\boldsymbol{\lambda}^n_k, \tag{5.19}$$

where

$$\boldsymbol{\lambda}^n_k = \gamma^n_k\mathbf{P}_{k-1\,k}(\mathbf{Z}^n_{k-1})^{-1}\tau\mathbf{D}^n_{k-1}\boldsymbol{\sigma}^n_{k-1}, \qquad k = 2, \dots, l, \tag{5.20}$$

contains the interpolation error at level k and the prolongated spatial discretization error of level $k-1$ to k (for $k = l-1$ convention (5.9) applies). This λ-function will be used for determining the matrices $\mathbf{D}^n_k$. Let $C_I \ge 1$ be a constant such that

$$\|\mathbf{P}_{k-1\,k}\| \le C_I. \tag{5.21}$$

For linear interpolation, $C_I = 1$, while for higher-order Lagrangian interpolation, $C_I > 1$. Now,

$$\left\|\prod_{i=l-1}^{k+1}\mathbf{X}^n_i\right\| \le C_X \le (CC_I)^{l-k-1}, \tag{5.22}$$

and using (5.19), we get

$$\|(\mathbf{Z}^n_l)^{-1}(\mathbf{I}_l - \mathbf{D}^n_l)\boldsymbol{\rho}^n_l\| \le \tilde{C}\max_{2\le k\le l}\|(\mathbf{I}_k - \mathbf{D}^n_k)\boldsymbol{\lambda}^n_k\| \tag{5.23}$$

with the grid-independent constant

$$\tilde{C} = C(1 + C_I(l-2)CC_X) = C + (l-2)(CC_I)^{l-2}. \tag{5.24}$$

Hence, if for each $k = 2, \dots, l$, the matrices $\mathbf{D}^n_k$ are selected such that

$$\|(\mathbf{I}_k - \mathbf{D}^n_k)\boldsymbol{\lambda}^n_k\| \le \frac{c}{\tilde{C}}\|(\mathbf{Z}^n_l)^{-1}\tau\mathbf{D}^n_l\boldsymbol{\sigma}^n_l\|, \qquad l = l_{n-1}, \tag{5.25}$$

then the refinement condition (5.17) is satisfied. In the following, (5.25) thus replaces (5.17).

This condition says that outside any integration domain the sum of the interpolation and prolongated spatial discretization error from the previous coarser level shall be bounded by the spatial discretization error of the highest level, multiplied by $c/\widetilde{C}$. This imposes a severe restriction on the size of the interpolation and discretization errors of the lower levels. On the other hand, this restriction is natural, because, when going to a higher level within the current time step, we never return to a grid point where the solution has been interpolated (nesting property). Note that in (5.25) the temporal stepsize τ occurs. In particular, if $\tau \to 0$, then the interpolation errors will prevail and $\mathbf{D}_k^n \to \mathbf{I}_k$. Recall that we interpolate at each time step, so that interpolation errors can accumulate linearly with the number of time steps. Our refinement condition prevents this.

The refinement condition (5.25) is not applicable to the full interpolation levels since at these levels $\mathbf{D}_k^n = \mathbf{0}$. For simplicity, we now consider only one full interpolation level and note that this is sufficient for practical purposes. Using (5.16), if $l_n = l_{n-1} + 1$, we thus find, instead of (5.18),

$$\|\boldsymbol{\psi}_{l_n,s}^n\| \leq \|\boldsymbol{\gamma}_{l_n}^n\| + C_I(1+c)\tau\|(\mathbf{Z}_{l_{n-1}}^n)^{-1}\mathbf{D}_{l_{n-1}}^n\boldsymbol{\sigma}_{l_{n-1}}^n\|. \tag{5.26}$$

Recall that full interpolation occurs only in a finite number of steps, uniformly in τ. Hence, when adding all local errors for a convergence proof, assuming stability, this fact should be taken into account so as to avoid an overly pessimistic summation like

$$\sum_{j=1}^{n} \|\boldsymbol{\gamma}_{l_j}^j\| \geq \frac{T}{\tau} \min_n \|\boldsymbol{\gamma}_{l_n}^n\|. \tag{5.27}$$

With a more subtle summation, based on the finite number of full interpolations, the τ^{-1}-term is avoided.

In conclusion, by imposing the refinement condition (5.25), the local space error bounds (5.18), (5.26) are valid. In an implementation these bounds can be used to monitor the spatial accuracy, while (5.25) is then used for selecting the actual integration domains. Such an implementation is method-dependent and therefore best described for a selected method. An illustration for a DIRK method is presented below. Finally, the error bound (5.18) suggests that we choose the threshold factor c not too large. However, if we take c very small, then the effect will be that the greater part of the diagonal entries of $\mathbf{D}_k^n$ are put to unity to satisfy the refinement condition, which implies that the integration domains will become quite large.

5.4. **The local time error $\boldsymbol{\psi}_{k,t}^n$.** Since the same τ is used at all levels, and $\boldsymbol{\beta}_k^n$ does not depend on the mesh width, we have $\boldsymbol{\beta}_k^n = \mathbf{R}_{l_{n-1}k}\boldsymbol{\beta}_{l_{n-1}}^n$, so that (5.13) yields

$$\boldsymbol{\psi}_{k,t}^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} \mathbf{X}_i^n \right) (\mathbf{Z}_j^n)^{-1}\mathbf{D}_j^n\mathbf{R}_{l_{n-1}j}\boldsymbol{\beta}_{l_{n-1}}^n. \tag{5.28}$$

By comparison with the recursion (3.13) for the amplification operators $\mathbf{G}_k^n$, one can see that

$$\boldsymbol{\psi}_{k,t}^n = \mathbf{G}_k^n\boldsymbol{\beta}_{l_{n-1}}^n, \qquad 1 \leq k \leq l_{n-1}. \tag{5.29}$$

This formula shows the dependence of the local time error on the temporal residual of the finest integration level. Alternatively, we may write, similarly as for the local space error (5.14),

$$\boldsymbol{\psi}^n_{k,t} = (\mathbf{Z}^n_k)^{-1}[\mathbf{D}^n_k \mathbf{R}_{l_{n-1}k}\boldsymbol{\beta}^n_{l_{n-1}} + (\mathbf{I}_k - \mathbf{D}^n_k)\mathbf{P}_{k-1\,k}\boldsymbol{\psi}^n_{k-1,t}], \qquad k = 1, \ldots, l_{n-1}. \tag{5.30}$$

This representation gives more insight than (5.29). At each integration level we recover the local time error contribution committed on the integration domain, viz. $(\mathbf{Z}^n_k)^{-1}[\mathbf{D}^n_k \mathbf{R}_{l_{n-1}k}\boldsymbol{\beta}^n_{l_{n-1}}]$, and the prolongation of the previous local time error of the next coarser level, viz. $(\mathbf{Z}^n_k)^{-1}[(\mathbf{I}_k - \mathbf{D}^n_k)\mathbf{P}_{k-1\,k}\boldsymbol{\psi}^n_{k-1,t}]$.

Let $\tilde{p}$ denote the stage order of the RK method [5, 8, 9]. Using (5.2), we have

$$\|\boldsymbol{\psi}^n_{k,t}\| \le C \max\{\|\mathbf{D}^n_k \mathbf{R}_{l_{n-1}k}\boldsymbol{\beta}^n_{l_{n-1}}\|, \|(\mathbf{I}_k - \mathbf{D}^n_k)\mathbf{P}_{k-1\,k}\boldsymbol{\psi}^n_{k-1,t}\|\}. \tag{5.31}$$

Because both $\boldsymbol{\beta}^n_{l_{n-1}}$ and $\boldsymbol{\psi}^n_{1,t}$ are $O(\tau^{\tilde{p}+1})$, by definition of stage order, we thus trivially recover the usual stage-order result at all grid levels, that is,

$$\psi^n_{k,t} = O(\tau^{\tilde{p}+1}), \qquad 1 \le k \le l_{n-1}, \tag{5.32}$$

where, apart from the norm bounds C for $(\mathbf{Z}^n_k)^{-1}$ and C_I for $\mathbf{P}_{j-1\,j}$, the order constant involved depends exclusively on bounds for temporal derivatives of $u(\underline{x}, t)$ (cf. (5.5)). To recover the conventional ODE order, p say, of the RK method, the $(s+1)$st output component of $\boldsymbol{\psi}^n_{k,t}$ must be expanded. We then would also arrive at an order relation $\boldsymbol{\psi}^n_{k,t} = O(\tau^{p+1})$, but here the constant involved may depend on the negative powers of the mesh width, similarly as in existing "Method of Lines" convergence theories (see [8, 9] and the preprint [10] on the order reduction phenomenon). Finally, no integration takes place at a full interpolation level, so that

$$\boldsymbol{\psi}^n_{k,t} = \mathbf{P}_{k-1\,k}\boldsymbol{\psi}^n_{k-1,t}, \qquad l_{n-1}+1 \le k \le l_n, \tag{5.33}$$

and we thus have the same temporal order as for $\boldsymbol{\psi}^n_{k,t}$, $1 \le k \le l_{n-1}$.

6. Error analysis for a 3-stage DIRK method

By way of illustration, in this section we elaborate the local error analysis for a 3-stage DIRK method, which later on will be used for presenting numerical examples.

6.1. **The DIRK method.** The DIRK method is found in [4] and defined by the Butcher array

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2\theta & \theta & \theta & 0 \\ 1 & b_1 & b_2 & \theta \\ \hline & b_1 & b_2 & \theta \end{array} \qquad \begin{array}{l} \theta = (3+\sqrt{3})/6 \\ b_1 = 3/2 - \theta - 1/(4\theta) \\ b_2 = -1/2 + 1/(4\theta) \end{array} \tag{6.1}$$

It is strongly A-stable, has classical order $p = 3$, stage order $\tilde{p} = 2$, and uses only two effective stages (first row of coefficients is zero). Note that stage one and two define the trapezoidal rule and that stage three and four, the output stage, are identical.

6.2. **Elaboration of the local time error.** Assume, for simplicity, that the semidiscrete problem is of constant-coefficient linear type,

$$\frac{d}{dt}U_k(t) = M_k U_k(t) + f_k(t). \tag{6.2}$$

Note that the linear case reveals the essentials of the local error analysis. Also for simplicity, we put $l_{n-1} = 2$. Conclusions for the higher-level case immediately follow. Thus, our task is to examine

$$\boldsymbol{\psi}^n_{1,t} = (\mathbf{Z}^n_1)^{-1}\mathbf{R}_{21}\boldsymbol{\beta}^n_2, \qquad \boldsymbol{\psi}^n_{2,t} = (\mathbf{Z}^n_2)^{-1}[\mathbf{D}^n_2\boldsymbol{\beta}^n_2 + (\mathbf{I}_2 - \mathbf{D}^n_2)\mathbf{P}_{12}\boldsymbol{\psi}^n_{1,t}]. \tag{6.3}$$

From (5.5), (6.1) we deduce $\beta^{(1)}_2 = 0$, $\beta^{(4)}_2 = \beta^{(3)}_2$ and

$$\begin{aligned}\beta^{(2)}_2 &= -\frac{2\theta^3}{3}\tau^3\frac{d^3}{dt^3}u_2(t_{n-1}) + O(\tau^4),\\ \beta^{(3)}_2 &= \left(\frac{1}{24} - \frac{\theta}{6} - \frac{\theta^2}{3} + \frac{2\theta^3}{3}\right)\tau^4\frac{d^4}{dt^4}u_2(t_{n-1}) + O(\tau^5).\end{aligned} \tag{6.4}$$

For any $\mathbf{v}_k \in \mathbf{S}_k$ having $v^{(1)}_k = 0$, the components $w^{(i)}_k$ of $\mathbf{w}_k = (\mathbf{Z}^n_k)^{-1}\mathbf{v}_k$ satisfy $w^{(1)}_k = 0$,

$$\begin{aligned}w^{(2)}_k &= (I_k - \theta\tau D^n_k M_k)^{-1}v^{(2)}_k,\\ w^{(3)}_k &= (I_k - \theta\tau D^n_k M_k)^{-2}b_2\tau D^n_k M_k v^{(2)}_k + (I_k - \theta\tau D^n_k M_k)^{-1}v^{(3)}_k,\end{aligned} \tag{6.5}$$

and $w^{(4)}_k = w^{(3)}_k$. We note in passing that the bound (5.2) may be derived from

$$\|(I_k - \theta\tau D^n_k M_k)^{-1}v_k\| \le (1 - \theta\tau\mu)^{-1}\|v_k\|, \qquad 1 - \theta\tau\mu > 0, \tag{6.6}$$

with the logarithmic norm $\mu = \mu_\infty[D^n_k M_k]$ independent of (the mesh width of) M_k. This bound applies in all cases where implicit Euler integrates in a stable way [5, 12].

Now first put $k = 1$. In view of the foregoing we then find $\psi^{(1)}_{1,t} = 0$, $\psi^{(4)}_{1,t} = \psi^{(3)}_{1,t}$, and

$$\begin{aligned}\psi^{(2)}_{1,t} &= -\frac{2\theta^3}{3}(I_1 - \theta\tau M_1)^{-1}\tau^3 R_{21}\frac{d^3}{dt^3}u_2(t_{n-1}) + O(\tau^4),\\ \psi^{(3)}_{1,t} &= -b_2\frac{2\theta^3}{3}(I_1 - \theta\tau M_1)^{-2}M_1 R_{21}\tau^4\frac{d^3}{dt^3}u_2(t_{n-1}) + O(\tau^4).\end{aligned} \tag{6.7}$$

Using the boundedness of the operators $(I_1 - \theta\tau M_1)^{-1}$, $(I_1 - \theta\tau M_1)^{-2}\tau M_1$, for $k = 1$ we can recover the stage-order result (5.32) with $\tilde{p} = 2$. Also the classical order $p = 3$ follows from $\psi^{(4)}_{1,t}$ when interpreted as the local ODE error. However, then the order constant depends on $M_1 R_{21}(d^3/dt^3)u_2(t_{n-1}) = M_1(d^3/dt^3)u_1(t_{n-1})$. Hence, $p=3$ is meaningful only when $M_1(d^3/dt^3)u_1(t_{n-1}) = O(1)$, uniformly in the mesh width, which is the case if the third derivative is zero at $\partial\Omega$. Otherwise, the constant blows up for decreasing mesh width,

making $p = 3$ not meaningful (order reduction, see [10] for a concrete example).

Next we put the level index $k = 2$. Since also $l_{n-1} = 2$, it suffices to examine the local error of the output stage, which is calculated from (6.3) as

$$\begin{aligned}\psi_{2,t}^{(4)} = \psi_{2,t}^{(3)} &= (I_2 - \theta\tau D_2^n M_2)^{-2} b_2 \tau D_2^n M_2 [D_2^n \beta_2^{(2)} + (I_2 - D_2^n) P_{12} \psi_{1,t}^{(2)}] \\ &\quad + (I_2 - \theta\tau D_2^n M_2)^{-1} [D_2^n \beta_2^{(3)} + (I_2 - D_2^n) P_{12} \psi_{1,t}^{(3)}] \\ &= (I_2 - \theta\tau D_2^n M_2)^{-2} b_2 \tau D_2^n M_2 [D_2^n \beta_2^{(2)} + (I_2 - D_2^n) P_{12} \psi_{1,t}^{(2)}] \\ &\quad + (I_2 - \theta\tau D_2^n M_2)^{-1} (I_2 - D_2^n) P_{12} \psi_{1,t}^{(3)} + O(\tau^4).\end{aligned} \tag{6.8}$$

From the boundedness of the operators, and the results for $k = 1$, stage order $\tilde{p} = 2$ directly follows. Inspection of the various terms also reveals the classical order $p = 3$. In connection with the occurrence of internal boundaries at grid interfaces, it is of interest to again examine the possibility of order reduction.

Distinguishing local error components outside and inside the integration domain, we can write

$$(I_2 - D_2^n)\psi_{2,t}^{(4)} = (I_2 - D_2^n) P_{12} \psi_{1,t}^{(4)}, \tag{6.9a}$$

$$\begin{aligned}D_2^n \psi_{2,t}^{(4)} &= (I_2 - \theta\tau D_2^n M_2)^{-2} b_2 \tau D_2^n M_2 [D_2^n \beta_2^{(2)} + (I_2 - D_2^n) P_{12} \psi_{1,t}^{(2)}] \\ &\quad + [(I_2 - \theta\tau D_2^n M_2)^{-1} - I_2](I_2 - D_2^n) P_{12} \psi_{1,t}^{(4)} + O(\tau^4).\end{aligned} \tag{6.9b}$$

Apart from the interpolation, the outside local error (6.9a) is completely determined by level-1 properties, so that a reduction at level 1 will also be felt at level-2 components outside the integration domain. The reduction will also be felt inside the level-2 integration domain, since (6.9b) depends on internal boundary values computed at level 1. An interesting question is whether the internal boundaries will cause order reduction in case the physical one does not. To examine this question, we henceforth suppose that no reduction will take place at $\partial\Omega$ and thus assume the additional boundary condition $M_k(d^3/dt^3)u_k(t_{n-1}) = O(1)$, uniformly in the mesh width. Then $\psi_{1,t}^{(4)} = O(\tau^4)$, so that (6.8) yields

$$\begin{aligned}\psi_{2,t}^{(4)} &= (I_2 - \theta\tau D_2^n M_2)^{-2} b_2 \tau D_2^n M_2 [D_2^n \beta_2^{(2)} + (I_2 - D_2^n) P_{12} \psi_{1,t}^{(2)}] + O(\tau^4) \\ &= -b_2 \frac{2\theta^3}{3} (I_2 - \theta\tau D_2^n M_2)^{-2} \tau^4 D_2^n M_2 \\ &\quad * \left[D_2^n \frac{d^3}{dt^3} u_2(t_{n-1}) + (I_2 - D_2^n) P_{12} \frac{d^3}{dt^3} u_1(t_{n-1}) \right] + O(\tau^4).\end{aligned} \tag{6.10}$$

Substitution of the interpolation error (5.7),

$$\gamma_2(t_{n-1}) = u_2(t_{n-1}) - P_{12} u_1(t_{n-1}) - b_2(t_{n-1}), \tag{6.11}$$

yields

$$\psi_{2,t}^{(4)} = b_2 \frac{2\theta^3}{3} (I_2 - \theta\tau D_2^n M_2)^{-2} \tau^4 D_2^n M_2 (I_2 - D_2^n) \frac{d^3}{dt^3} \gamma_2(t_{n-1}) + O(\tau^4). \tag{6.12}$$

We note in passing that the additionally imposed boundary condition implies "homogeneity in boundary conditions", causing the 3rd derivative of $b_2(t)$ to

vanish. From (6.12) we now deduce that if

$$\tilde{\gamma} \equiv D_2^n M_2 (I_2 - D_2^n) \frac{d^3}{dt^3} \gamma_2(t_{n-1}) = O(1), \tag{6.13}$$

uniformly in the mesh width, then $\psi_{2,t}^{(4)} = O(\tau^4)$ uniformly in the mesh width.

Hence, assuming that at the physical boundary no order reduction takes place, an important conclusion is that the internal boundaries do not cause order reduction if the interpolation condition (6.13) holds. Fortunately, in applications this condition is easily satisfied. Sufficient is that

$$\|M_2\| \left\| \frac{d^3}{dt^3} \gamma_2(t_{n-1}) \right\| = O(1), \tag{6.14}$$

which says that the accuracy order of the interpolation should be greater than or equal to the spatial order of the differential operator (not to be confused with the order of consistency of the difference operator). For example, for second-order in space problems it suffices to use simple linear interpolation.

6.3. **Elaboration of the refinement condition.** Given a specific integration method, the general refinement condition (5.25) needs to be simplified for practical use. Two main simplifications can be distinguished:

(i) The first has to do with the augmented form. Working with (5.25) requires computing in $\mathbf{S}_k$, which is expensive. Consequently, (5.25) is better replaced by an appropriate approximating condition in S_k, preferably connected with the output stage. It is always possible to carry this out, since the refinement condition is concerned with spatial errors. Apart from various multiplying bounded operators, these errors are similar over the stages.

Consider (5.20), (5.25). First we replace the Jacobian $M_k^{(i)}$ occurring in $\mathbf{Z}_k^n$ by an approximation M_k constant over the stages. M_k is taken to be the (approximate) Jacobian, computed at the beginning of the time step. M_k is available as it is also used in the iterative Newton process for solving the implicit relations. Second, the augmented spatial error $\boldsymbol{\sigma}_k^n$ is approximated as

$$\boldsymbol{\sigma}_k^n = \begin{pmatrix} & & 0 & & \\ \theta\alpha_k^{(1)} & + & \theta\alpha_k^{(2)} & & \\ b_1\alpha_k^{(1)} & + & b_2\alpha_k^{(2)} & + & \theta\alpha_k^{(3)} \\ b_1\alpha_k^{(1)} & + & b_2\alpha_k^{(2)} & + & \theta\alpha_k^{(3)} \end{pmatrix} \approx \begin{pmatrix} 0 \\ 2\theta\alpha_k^{(3)} \\ \alpha_k^{(3)} \\ \alpha_k^{(3)} \end{pmatrix}. \tag{6.15}$$

Note that we here truncate $O(\tau)$-terms and that $\alpha_k^{(3)} = \alpha_k^n = \alpha_k(t_n)$. Next, by using (6.5), the nontrivial components of the spatial error function $\mathbf{w}_k = (\mathbf{Z}_k^n)^{-1} \mathbf{D}_k^n \boldsymbol{\sigma}_k^n$ are approximated by

$$\begin{aligned} w_k^{(2)} &\approx 2\theta (I_k - \theta\tau D_k^n M_k)^{-1} D_k^n \alpha_k^n, \\ w_k^{(4)} = w_k^{(3)} &\approx (I_k - \theta\tau D_k^n M_k)^{-1} \\ &\quad \cdot (2b_2\theta (I_k - \theta\tau D_k^n M_k)^{-1} \tau D_k^n M_k + I_k) D_k^n \alpha_k^n. \end{aligned} \tag{6.16}$$

At each of the stages we recover a proportionality with the local space discretization error $D_k^n \alpha_k^n$. This justifies to select one particular stage. We choose the approximation

$$w_k^{(4)} \approx (1 - 2b_2)(I_k - \theta\tau D_k^n M_k)^{-1} D_k^n \alpha_k^n, \tag{6.17}$$

which avoids two forward-backward substitutions and is based on

$$[2b_2\theta(I_k - \theta\tau D_k^n M_k)^{-1}\tau D_k^n M_k + I_k]D_k^n\alpha_k^n \approx (1-2b_2)D_k^n\alpha_k^n. \tag{6.18}$$

In first approximation, (6.18) is exact if $D_k^n\alpha_k^n$ is taken to be an eigenvector belonging to the maximal eigenvalue. On the other hand, the operator in (6.18) is bounded, which justifies this step.

We can now replace the constituents of the regridding condition by their counterparts in S_k:

$$\begin{aligned}&\|(I_k - D_k^n)\lambda_k^n\| \\ &\quad\le \frac{c}{l-1}\|(1-2b_2)\tau(Z_l^n)^{-1}D_l^n\alpha_l^n\|, \qquad k = 2, \dots, l = l_{n-1},\end{aligned} \tag{6.19}$$

$$\lambda_k^n = \gamma_k^n + (1-2b_2)\tau P_{k-1\,k}(Z_{k-1}^n)^{-1}D_{k-1}^n\alpha_{k-1}^n, \tag{6.20}$$

$$Z_k^n = I_k - \theta\tau D_k^n M_k. \tag{6.21}$$

Observe that $\|(I_k - D_k^n)\lambda_k^n\| = \|(\mathbf{I}_k - \mathbf{D}_k^n)\boldsymbol{\lambda}_k^n\| + O(\tau)$. The choice $l-1$ for the constant $\widetilde{C}$ is exact in case of linear interpolation, provided $C \le 1$ (see (5.24), (5.2)). We will use $\widetilde{C} = l-1$ also in other situations and note that, apart from the constant $1-2b_2$, condition (6.19) is completely identical to the regridding condition found for the implicit Euler method in [12].

(ii) The second simplification has to do with the nesting property and restricted interpolation. Once at level $k-1$ the integration is completed, (6.19) is used to select the integration domain for level k. This selection process is carried out by the so-called flagging procedure, which scans level-k points and flags those points for which (6.19) is violated to be placed within the new domain. Our mathematical framework prescribes that the scan be carried out on the whole of ω_k, as the interpolation error γ_k^n is defined on the whole of ω_k. This, of course, is time-consuming. We therefore apply restricted interpolation, meaning that the interpolation is restricted to level-k points lying within the $(k-1)$st integration domain. Subsequently, the scan is also restricted to the $(k-1)$st integration domain. In this way the nesting of the integration domains is enforced. In [12] it is shown that restricted interpolation leads to (nearly) the same integration domains as found with full interpolation; hence full interpolation is truly redundant. Finally, the flagging procedure contains some safety measures (buffering) which enhances the reliability of the restricted interpolation. This procedure also implements numerical estimators for γ_k^n, $(Z_{k-1}^n)^{-1}D_{k-1}^n\alpha_{k-1}^n$, and $\|(Z_l^n)^{-1}D_l^n\alpha_l^n\|$. To save space, we again refer to [12].

7. Numerical examples

We will illustrate the effect of the simplified refinement condition (6.19) of the DIRK method (6.1). Recall that, in theory, this condition guarantees local space errors at most equal to the maximum of the local space error on the finest grid when used without adaptation, up to a certain grid-independent constant (arising, e.g., from transferring the refinement condition to S_k and estimating $\widetilde{C}$ by $l-1$). Hence, assuming stability, our theory dictates that the usual convergence behavior of the discretization method applied without adaptation will be maintained.

Two examples are presented, both 2D. The first serves to illustrate the above claim on convergence. This problem is solved using the "fixed-level mode of operation". The second serves to illustrate the performance of the method when applied in the "variable-level mode of operation". This mode is advocated if the solution shape strongly changes in time, e.g., when steep layers emerge at later times and at earlier times large gradients are absent. In such situations it is important that new levels are created in time in order to preserve accuracy. On the other hand, new levels should not be created too early for efficiency.

7.1. **Example problem** I. The equation is linear and parabolic and given by (Adjerid and Flaherty [1])

$$u_t = u_{xx} + u_{yy} - u_x - u_y + f(x, y, t), \qquad 0 < x, y < 1, \quad t > 0. \tag{7.1}$$

The initial and Dirichlet boundary conditions and forcing f are adjusted to

$$u(x, y, t) = 1 - \tanh(25(x - t) + 5(y - 1)). \tag{7.2}$$

This solution is a skew wave propagating through the domain from left to right. The wave starts near the left boundary and approaches the right boundary at approximately $t = 0.8$. We integrate over the time interval $[0, 0.6]$. This problem is suitable to subject the LUGR method to a convergence test.

The spatial discretization is based on second-order symmetric differences. Simple linear interpolation is used and the constant c, introduced in the refinement condition, is put equal to one. Four computations were performed using, respectively, 1, 2, 3, and 4 levels. The mesh width in both x- and y-direction of the base grid is 0.05. During a computation the stepsize τ is fixed. However, when adding a level, we simultaneously halve τ. Because the stage order of the DIRK method is 2, like the order of the spatial discretization, per computation a gain factor of approximately 4 should then be found for the total global errors. To compare the accuracy with the accuracy obtained on a single uniform grid, we have also solved the problem in the standard way using the same values for τ and the mesh width of the finest level. The values of τ and the mesh width in space are always such that the space error dominates. For illustration purposes this is necessary, since otherwise no valid conclusion can be drawn on the performance of the spatial refinement condition.

TABLE 7.1 Example problem I. Maxima of global errors computed at the finest available level. Comparison with the accuracy obtained on a single uniform grid

τ	no. of levels	single grid	t 0.3	0.6
0.1	1	20×20	0.17319	0.17401
0.05	2	40×40	0.02728 0.02789	0.02815 0.02810
0.025	3	80×80	0.00624 0.00680	0.00716 0.00684
0.0125	4	160×160	0.00177 0.00168	0.00174 0.00169

The results of the computations are contained in Table 7.1. We see that the LUGR solutions converge according to the theory and, also, that these solutions are as accurate as the standard, uniform grid solutions. In view of the simplifications of §6.3, this correspondence in accuracy is striking. We should note, though, that in the actual flagging procedure some safety measures have been incorporated, like buffering. Buffering of course helps in keeping the LUGR accuracy close to the standard accuracy. Figure 7.1 shows the grids of the 2-, 3-, and 4-level computations at two different times. Note that the grids align with the wave front and become larger for smaller τ, in accordance with (6.19).

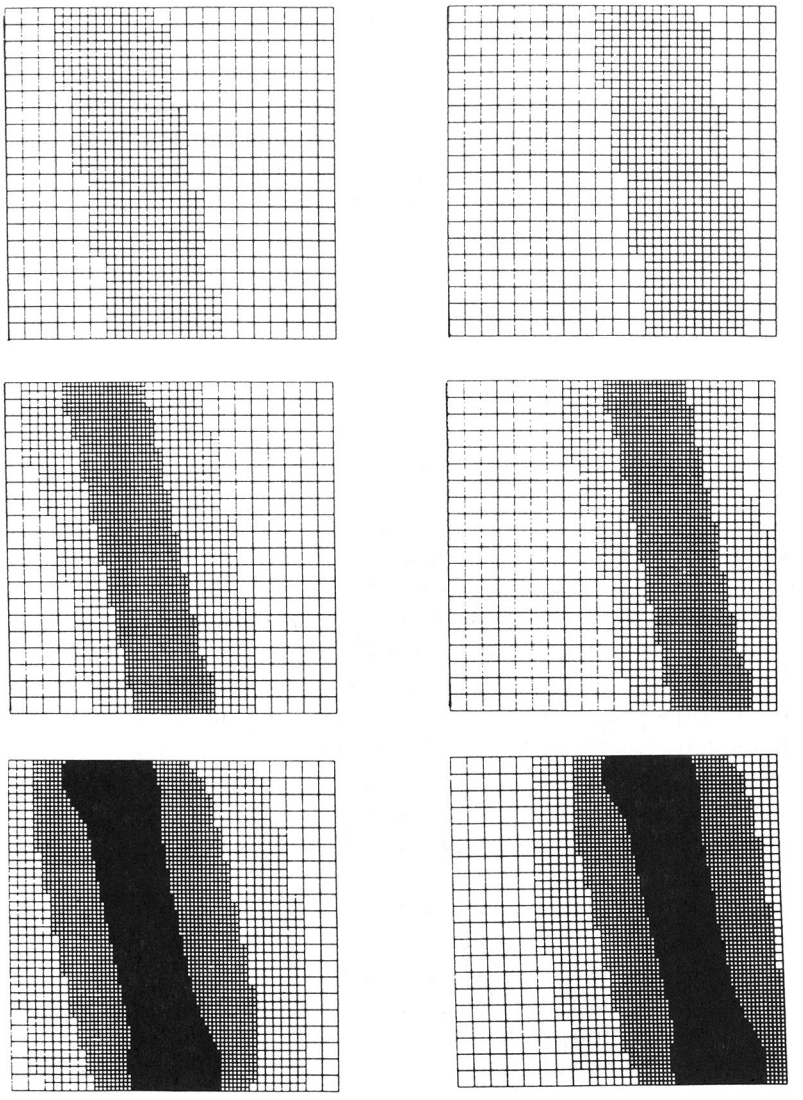


FIGURE 7.1. Example problem I. Grids of the 2-, 3-, and 4-level computations at $t = 0.3$ and $t = 0.6$

7.2. **Example problem** II. The equation is again linear and parabolic,

$$u_t = u_{xx} + u_{yy} + f(x, y, t), \qquad 0 < x, y < 1, \quad t > 0. \tag{7.3}$$

The initial and Dirichlet boundary conditions and forcing f are adjusted to

$$u(x, y, t) = 1 - \tanh(100[(x - 0.5)^2 + (y - 0.5)^2 - t + 0.025]). \tag{7.4}$$

This solution rapidly varies its shape and serves to illustrate the "variable-level mode of operation". At $t = 0$ the solution is almost zero over the entire domain. As time elapses, it steepens up at $[0.5, 0.5]$, developing a circular wave front. This front starts to propagate towards the boundaries when $u(0.5, 0.5, t) \approx 2$ and during the propagation the front becomes steeper. When the front has passed a point (x, y), the solution $u(x, y, t)$ approximates the value 2. We solve the problem over the time interval $[0, 0.1]$, which is sufficiently large to see all phenomena happen.

The refinement condition (6.19) tells us where to integrate on a finer level. When using the "fixed-level mode of operation" this suffices. When using the "variable-level mode of operation", we also need a criterion to decide when to change the number of levels. A natural thing to do is to associate this criterion with the spatial local error value. In the present experiment we employ the numerical spatial local error expression as used in (6.19). Within each base time step we monitor the number of grid levels with the criterion

$$(1 + c)(1 - 2b_2)\tau \| (Z_k^n)^{-1} D_k^n \alpha_k^n \| < \tau \, \mathrm{TOL}, \tag{7.5}$$

where TOL represents a tolerance value. Starting with $k = 1$, this inequality is checked after each level integration. If it is violated, then k is increased by 1. Otherwise it is decided that enough levels have been introduced and l_{n-1} is assigned the current value for k. Hence, the idea is to select l_{n-1} in such a way that the local error expression in (7.5) is kept close to $\tau \, \mathrm{TOL}$.

We will encounter a few full interpolations. The full interpolation error is neglected in (7.5). We justify this heuristic decision with the observation that full interpolation can take place only in a small number of steps (see also §5). However, to remain on the safe side, we now use 4th-order Lagrangian interpolation instead of 2nd-order linear. It is obvious that full interpolation must not diminish the quality of the approximations, since otherwise the estimation of the discretization and interpolation errors used by the refinement condition is jeopardized. The full interpolation should also not interfere with the estimation of the number of levels needed in the step to follow. Therefore, the additional errors stemming from full interpolation have to be restricted in some manner. In the present experiment, 4th-order interpolation has turned out to work satisfactorily.

The actual experiment with problem (7.3)–(7.4) concerns one run over the time interval $[0, 0.1]$. The constant c of the refinement condition is again put equal to 1. The stepsize $\tau = 0.001$ is kept constant. The value of 0.001 is sufficiently small to guarantee that spatial effects dominate. The mesh width in both x- and y-direction of the base grid is 0.05 and the tolerance parameter $\mathrm{TOL} = 50$. Results are collected in Tables 7.2–7.3 and Figure 7.2. For a subset of time points, including those where a new grid level is added, Table 7.2 shows the course of the number of grid levels and the maximum of the global error measured at the finest available grid. Note that while the circular wave front

TABLE 7.2 Example problem II. Maxima of global errors computed at the finest grid at various time points, including those where a new grid is introduced

t	no. of levels	global error
0.01	1	0.01074
0.017		0.03171
0.018	2	0.01222
0.02		0.01117
0.03		0.01612
0.039		0.02523
0.04	3	0.01392
0.05		0.01493
0.06		0.01668
0.07		0.02168
0.072		0.02136
0.073	4	0.01289
0.08		0.01191
0.09		0.00722
0.1		0.00713

develops, the algorithm keeps the error at a fairly constant level, which is in line with the idea behind the error monitor (7.5).

The pictures contained in Figure 7.2 (see pp. 613–615) illustrate that the grids accurately reflect the circular wave front form (symmetry), showing that the refinement condition, which tells us where to refine, works as anticipated. On the other hand, the number of levels needed is not always computed optimally. This happens, e.g., at $t = 0.04$ and $t = 0.073$ time points, where a new grid level is used for the first time. The grid pictures show that at these time points the new fine grid almost completely overlaps the existing one, indicating that the new fine grid is introduced too late (the solution steepens up). Fortunately, Table 7.2 shows that this small deficiency does not diminish the accuracy for evolving time. Also note that at later points of time this phenomenon disappears (see $t = 0.05$ and $t = 0.1$). This is of course what should happen in view of the ever increasing solution gradients.

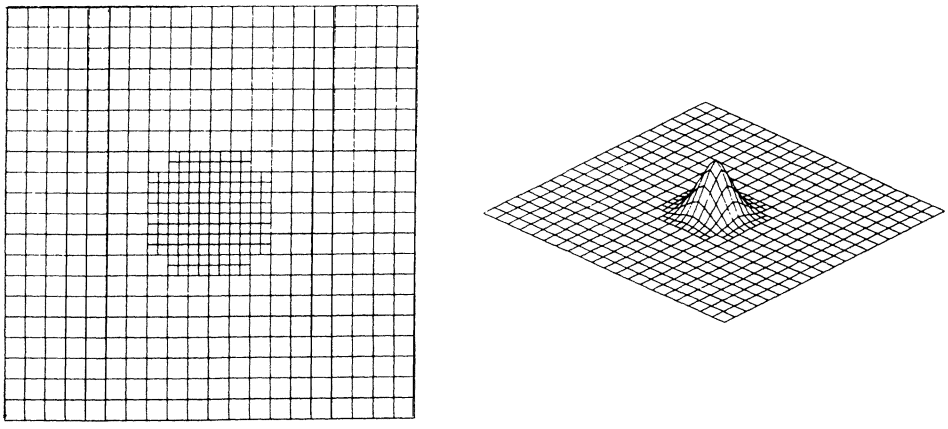
The precise origin of this small deficiency is not clear. The error introduced by the full interpolation can play a role here (this error is not monitored by (7.5)). More likely is, however, that it emanates from the lack of asymptotics at the coarser grids. This lack of asymptotics is inherent to any monitoring process that starts on coarse grids and therefore very difficult to overcome. To provide insight into the asymptotics for the estimator of (7.5), we have added Table 7.3. This table shows the exact, analytical values for (7.5) with their estimated numerical values at time points just before and after the introduction of a new grid level. First, we see that at corresponding levels before and after the listed time points the numerical estimations are in fairly good agreement

with one another, even on the coarse base grid. This supports the conclusion that the full interpolation is sufficiently accurate as to not interfere with the selection of number of levels. Second, there is excellent agreement between the exact and numerical values on the fine grids. However, particularly at later times, the coarse grid values are not in good agreement with one another. This means that we are outside the asymptotic regime, and this is likely to cause some disturbances in the selection of the right number of levels. We wish to emphasize once more that in spite of this lack of asymptotics, the overall behavior of the algorithm is very satisfactory.

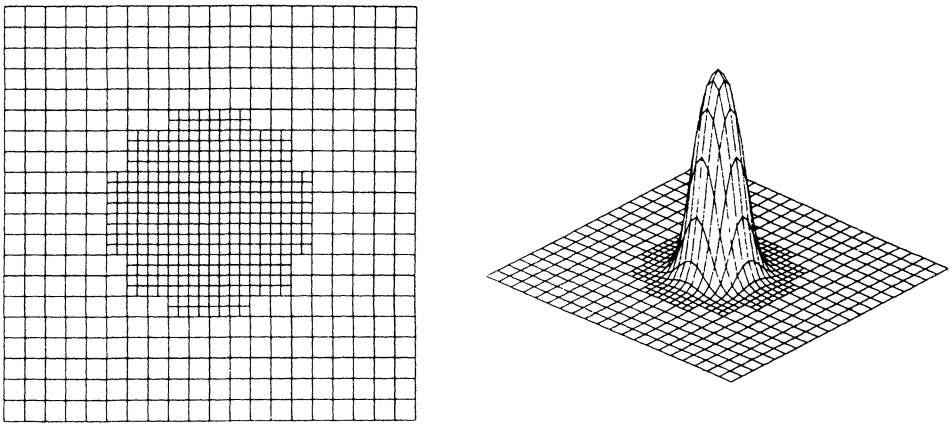
TABLE 7.3 Example problem II. Exact values and numerical estimates of the spatial local error expression (7.5). Note that the stepsize $\tau = 0.001$ is included in these values

t	level	approx.	exact
0.017	1	0.04616	0.05307
0.018	1	0.05156	0.05793
	2	0.01246	0.01468
0.039	1	0.12511	0.28075
	2	0.04678	0.05579
0.04	1	0.13972	0.33177
	2	0.05084	0.06329
	3	0.01495	0.01467
0.072	1	0.31630	1.48171
	2	0.18625	0.30880
	3	0.04685	0.05130
0.073	1	0.34144	1.39739
	2	0.18078	0.29537
	3	0.05088	0.05367
	4	0.01683	0.01382

Let us conclude with a remark on the choice of TOL, in connection with the discrepancy between the value TOL = 50 and the global accuracy shown in Table 7.2. A discrepancy like this is unavoidable, owing to damping of global errors. Note that we have a parabolic problem and that the DIRK method mimics the damping property of the parabolic operator (strong A-stability). Part of the discrepancy may also originate from cancellation between temporal and spatial terms. This damping of global errors, and possible cancellation, has not been taken into account in our error analysis, which focuses on local errors,



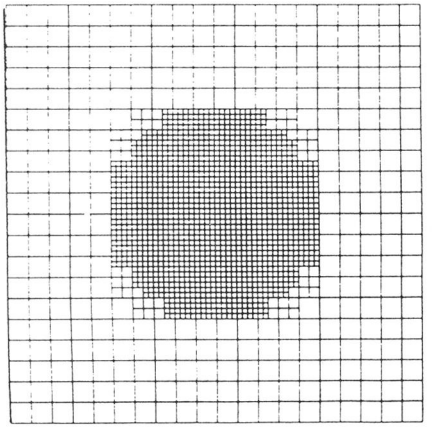
$t = 0.018$

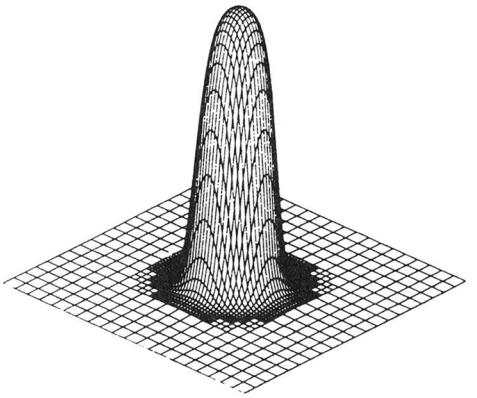


$t = 0.03$

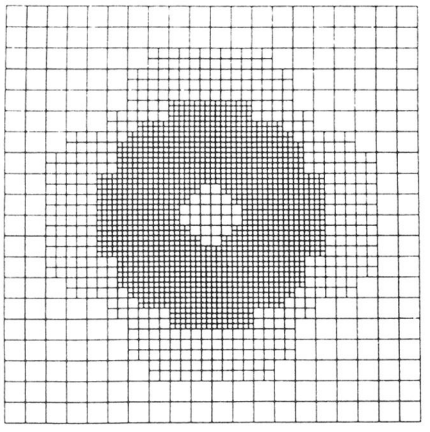
FIGURE 7.2. Example problem II. The course of the local uniform grids and the computed solution of (7.3)

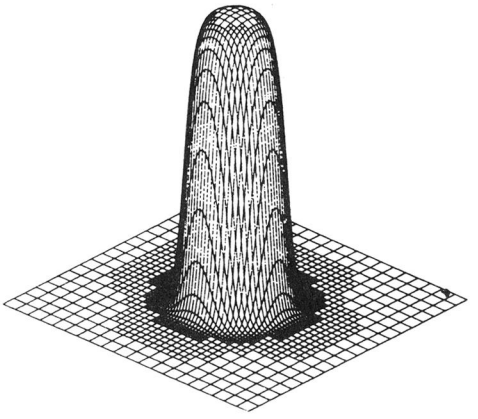
in particular on local error bounds. For precise estimation purposes our analysis is simply too general. On the other hand, the present example once more shows that local error bounds like (7.5) can be much too conservative (the simplified form is not essential for the present discussion). Consequently, for application, local error expressions like (7.5) are better interpreted as error monitors. In connection with grid selection purposes, our practical experience is that with this interpretation the (simplified) spatial local error expression is reliable and works very satisfactorily.





$t = 0.04$





$t = 0.05$

FIGURE 7.2. (continued)

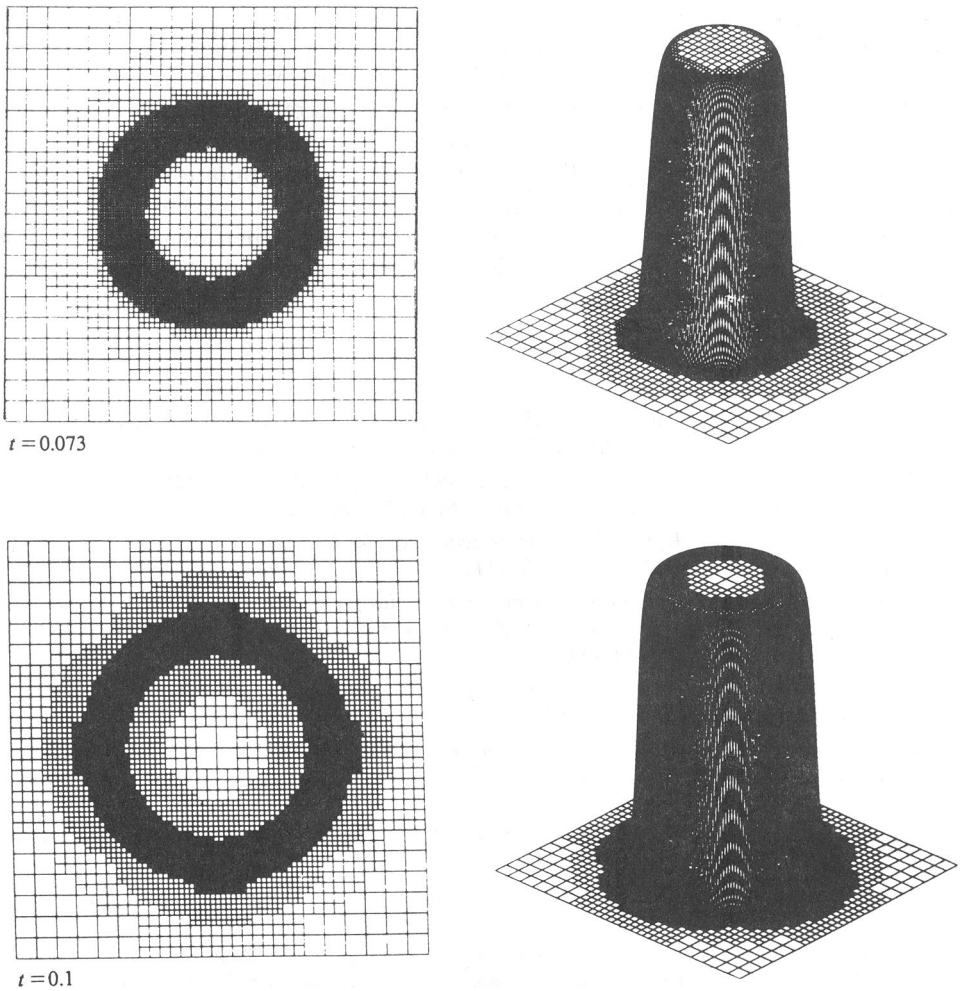


FIGURE 7.2. (continued)

8. EFFICIENCY OF TIME-STEPPING

An important subject for future research is that of efficiency of the time-stepping scheme itself when combined with the LUGR technique. Two important issues not addressed in this work concern the use of variable time steps and the solution of the arising systems of linear and nonlinear algebraic equations, in case of an implicit scheme. Straightforward use of variable time step algorithms, as successfully applied in single-grid method of lines computations, gives rise to problems since approximations obtained with an LUGR method are always difficult to numerically differentiate in time. The reason for this is that some of the components are obtained from a numerical integration, some from interpolation or injection. The resulting "nonsmoothness" is then felt when computing higher temporal derivatives. More precisely, the higher temporal derivatives are estimated in a rough way, resulting in disturbances in the stepsize selection (see also [11]). In our experience, smoothing or filtering procedures provide only a partial remedy here.

Concerning the second issue, by the nature of the LUGR approach approximations are computed in varying dimensions, even within one base time step. For DIRK or alternative implicit methods this obviously implies that the numerical algebra effort required in solving systems of algebraic equations becomes highly important. In the numerical experiments reported here, we have paid no attention to the efficiency of the numerical algebra computations and simply used an available sparse matrix technique (same as in [12]). This technique, however, is known to result in a considerable overhead when used in the solution of time-dependent problems. It is most likely that sophisticated iterative solution procedures will be much more effective.

BIBLIOGRAPHY

1. S. Adjerid and J. E. Flaherty, *A local refinement finite element method for two dimensional parabolic systems*, SIAM J. Sci. Statist. Comput. **9** (1988), 792–811.
2. D. C. Arney and J. E. Flaherty, *An adaptive local mesh refinement method for time-dependent partial differential equations*, Appl. Numer. Math. **5** (1989), 257–274.
3. M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys. **53** (1984), 484–512.
4. M. Crouzeix and P. A. Raviart, *Approximation des problèmes d'évolution.* 1: *Etude des méthodes linéaires a pas multiples et des méthodes de Runge-Kutta*, unpublished lecture notes, Université de Rennes, France.
5. K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North-Holland, Amsterdam-New York-Oxford, 1984.
6. W. D. Gropp, *Local uniform mesh refinement on vector and parallel processors*, Large Scale Scientific Computing (P. Deuflhard and B. Engquist, eds.), Progr. Sci. Comput., Birkhäuser, Boston, 1987, pp. 349–367.
7. ——, *Local uniform mesh refinement with moving grids*, SIAM J. Sci. Statist. Comput. **8** (1987), 292–304.
8. J. M. Sanz-Serna and J. G. Verwer, *Stability and convergence at the PDE/stiff ODE interface*, Appl. Numer. Math. **5** (1989), 117–132.
9. J. M. Sanz-Serna, J. G. Verwer, and W. H. Hundsdorfer, *Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations*, Numer. Math. **50** (1987), 405–418.
10. R. A. Trompert and J. G. Verwer, *Runge-Kutta methods and local uniform grid refinement*, Report NM-R9022, Centre for Mathematics and Computer Science, Amsterdam, preprint.
11. ——, *A static-regridding method for two dimensional parabolic partial differential equations*, Appl. Numer. Math. **8** (1991), 65–90.
12. ——, *Analysis of the implicit Euler local uniform grid refinement method*, SIAM J. Sci. Statist. Comput. **14** (1993).

CENTRE FOR MATHEMATICS AND COMPUTER SCIENCE, P.O. BOX 4079, 1009 AB AMSTERDAM, THE NETHERLANDS