

## ANALYSIS OF THE IMPLICIT EULER LOCAL UNIFORM GRID REFINEMENT METHOD\*

R. A. TROMPERT† AND J. G. VERWER†

**Abstract.** Attention is focused on parabolic problems having solutions with sharp moving transitions in space and time. An adaptive grid method is analysed that refines the space grid locally around sharp spatial transitions, so as to avoid discretization on a very fine grid over the entire physical domain. This method is based on static-regridding and local uniform grid refinement. Static-regridding means that for evolving time the space grid is adapted at discrete times. Local uniform grid refinement means that the actual adaptation of the space grid takes place using nested locally and uniformly refined grids. The present paper concentrates on stability and error analysis while using the implicit Euler method for time integration. Maximum norm stability and convergence results are proved for a certain class of linear and nonlinear partial differential equations. The central issue is a refinement condition with a strategy that distributes spatial interpolation and discretization errors in such a way that the spatial accuracy obtained is comparable to the spatial accuracy on the finest grid if this grid would be used without any adaptation. The analysis is confirmed with a numerical illustration.

**Key words.** partial differential equations, numerical mathematics, time-dependent problems, adaptive grid methods, error analysis

**AMS(MOS) subject classifications.** primary, 65M50; secondary, 65M20

**1. Introduction.** Attention is focused on parabolic problems having solutions with sharp moving transitions in space and time, such as steep fronts and disappearing layers. For such problems, a space grid held fixed throughout the entire time evolution can be computationally very inefficient. We consider an adaptive grid method that refines locally around sharp spatial transitions so as to avoid discretization on a very fine grid over the entire physical domain.

Our method is based on the techniques called static-regridding and local uniform grid refinement (LUGR), as previously proposed by Berger and Olinger [3], Gropp [6]–[8], Arney and Flaherty [2], Flaherty, Moore, and Ozturan [11], Trompert and Verwer [13], and others. Static-regridding means that for evolving time the space grid is adapted at discrete times. This should be contrasted with dynamic-regridding, where the space grid moves continuously in the space-time domain. With the term LUGR we mean that the actual adaptation of the space grid takes place using local, uniform, refined grids. LUGR should be contrasted with pointwise refinement, which leads to truly nonuniform grids. In this connection, our LUGR method bears resemblance to the fast adaptive composite grid (FAC) method [10] for elliptic equations, where the basic computational objective is to solve on an irregular grid by way of regular grids only.

The idea of the method can be briefly described as follows. Given a coarse base grid and a temporal step size, nested, local, uniform subgrids are generated. These subgrids possess nonphysical boundaries and on each of these subgrids an integration is carried out. They are generated up to a level of refinement good enough to resolve the anticipated fine scale structures. Having completed the refinement for the current base space-time grid, the process is continued to the next one while the fine grid results computed at forward time levels are kept in storage so that they can be used for step continuation.

---

\* Received by the editors August 6, 1990; accepted for publication (in revised form) March 25, 1992.

† Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, the Netherlands.

An attractive feature of the static-regridding approach is the possibility of dividing the solution process into the following computational procedures: spatial discretization, temporal integration, error estimation, regridding, and interpolation. Depending on the application, these individual procedures may range from simple or straightforward to very sophisticated. This flexibility is attractive since it makes it possible to treat different types of partial differential equation (PDE) problems with almost one and the same code, assuming that the grid and the associated data structure remain unchanged. Note that the choice of data structure is important for keeping the unavoidable overhead at an acceptable level, because at each time step grids may be created or removed, while communication between grids of adjacent levels of refinement frequently takes place.

The method we analyse in this paper has many similarities with the method constructed in Trompert and Verwer [13]. In fact, the grid and data structure, the spatial differencing, and the memory use are the same. However, in the present paper we concentrate on analysis rather than on construction, while using implicit Euler instead of the explicit Runge–Kutta–Chebyshev method for time integration. The main aim of this paper is to present a detailed error analysis and to prove stability and convergence for a certain class of PDEs. The central issue in this analysis is a refinement condition and a strategy that distributes spatial discretization and interpolation errors in such a way that the spatial accuracy obtained is comparable to the spatial accuracy on the finest grid if this grid would be used without any adaptation.

Section 2 is devoted to the problem class on which we will concentrate. In § 3 we introduce the tools and the formulation for the multilevel LUGR method. In § 4 we discuss the maximum norm stability of this method. We prove an unconditional stability result which is closely related to a maximum norm stability result of implicit Euler when applied on a single space grid. Section 5 is devoted to the error analysis. In this section we investigate the total local error with its component parts. Furthermore, here we introduce the refinement strategy underlying the so-called refinement condition. This condition enables us to control the contribution of the interpolation errors in favour of discretization errors. Due to this condition, we are able to prove a convergence result as if we are working on a single fixed grid. We further elaborate on this condition in § 6, where we show how to implement it for practical use. A numerical illustration of the error analysis is given in § 7. The numerical results found here are in complete agreement with the analysis. Finally, § 8 briefly discusses our future research plans.

**2. The problem class.** Following the method of lines approach [12], we consider a real abstract Cauchy problem

$$(2.1) \quad u_t = L(t, u), \quad 0 < t \leq T, \quad u(\underline{x}, 0) = u^0(\underline{x}),$$

where  $L$  represents a second-order partial differential operator that differentiates the (possibly vector-valued) solution  $u(\underline{x}, t)$  to its space variable  $\underline{x}$  in a space domain  $\Omega$  in  $\mathbb{R}$ ,  $\mathbb{R}^2$ , or  $\mathbb{R}^3$ . Boundary conditions are supposed to be included in the definition of  $L$ .

With (2.1) we associate a real Cauchy problem for an explicit ordinary differential equation (ODE) system in  $\mathbb{R}^d$ ,

$$(2.2) \quad \frac{d}{dt} U(t) = F(t, U(t)), \quad 0 < t \leq T, \quad U(0) = U^0,$$

which is defined by a finite-difference space discretization. Thus,  $U$  and  $F$  are vectors in  $\mathbb{R}^d$  representing grid functions on a space grid  $\omega$  covering the interior of the space

domain. Each component of  $U$  and  $F$  is vector valued if  $u$  is vector valued. The dimension  $d$  is determined by the spatial dimension, the grid spacing, and the number of PDEs in (2.1).  $F$  is determined by the type of grid, by the actual finite-difference formulas and, of course, by the precise form of  $L$  and its boundary conditions. Note that boundary values have been eliminated and worked into the ODE system. In the following, our method description and analysis are centered around this system.

Next we introduce some notations and assumptions needed for further specifying (2.1) and (2.2). The symbol  $\|\cdot\|$  denotes the maximum norm on the vector space  $\mathbb{R}^d$  or the induced matrix norm. Throughout our analysis we will deal only with the maximum norm. The symbol  $\mu[A]$  denotes the logarithmic matrix norm of the real  $d \times d$  matrix  $A = (a_{ij})$  associated with  $\|\cdot\|$ , i.e.,

$$(2.3) \quad \mu[A] = \max_i \left( a_{ii} + \sum_{j \neq i} |a_{ij}| \right);$$

$\mu[A]$  is a useful tool in the stability analysis of nonlinear, stiff ODEs and semidiscrete PDEs [4]. In this analysis, the structure of the Jacobian matrix  $F'(t, \eta) = \partial F(t, \eta) / \partial \eta$  plays a decisive role.

We are now ready to list the assumptions we make in further specifying (2.1), (2.2). These assumptions are concerned with, respectively, the class of PDEs (2.1), the smoothness of  $u$ , the choice of spatial grid and actual finite-differencing, and the stability of the semidiscrete system (2.2).

(A1) The LUGR method is applicable in any number of space dimensions. Following [13], we concentrate on the two-dimensional case, while  $\Omega$  is supposed to be the unit square. With minor changes  $\Omega$  is allowed to be composed of a union of rectangles with sides parallel to the coordinate axes. In fact, as we will see later, refined grids normally are of this shape. In what follows, we will generally use the notation  $u(x, y, t)$ , rather than  $u(\underline{x}, t)$ .

(A2) The solution  $u$  of (2.1) uniquely exists and is as smooth as the numerical analysis requires. Specifically, for our purpose it suffices that  $u$  is a  $C^2$ -function in  $t$  and a  $C^4$ -function in  $(x, y)$ .

(A3) We will invariably use uniform space grids. Thus our base grid can be written as

$$(2.4) \quad \omega = \{(x_i, y_j): x_i = ih_x, 1 \leq i \leq M-1 \text{ and } y_j = jh_y, 1 \leq j \leq N-1\},$$

where  $h_x = 1/M$ ,  $h_y = 1/N$ , and  $M, N$  are positive integers. The spatial differencing on  $\omega$  is supposed to be based on three-point formulas of second-order consistency. As a rule, we use central differencing. For boundary conditions involving first-order derivatives, the one-sided, three-point formula is used.

(A4) A constant  $\nu$  exists such that  $\mu[F'(t, \eta)] \leq \nu$  for all  $t \in (0, T]$ ,  $\eta \in \mathbb{R}^d$ , and all grid spacings. Like (A1) and (A2), this assumption involves a restriction on the class of PDE problems. Of course, they are made only for the sake of (model) analysis. The LUGR method remains applicable in situations where these assumptions do not hold or cannot be verified. On the other hand, for interesting classes of operators, such as the scalar, nonlinear parabolic operator

$$(2.5) \quad L(t, u) = f_1(t, x, y, u, (p_1(t, x, y)u_x)_x) + f_2(t, x, y, u, (p_2(t, x, y)u_y)_y),$$

with standard restrictions on  $f_i$  and  $p_i$ , one can prove the existence of a constant  $\nu$  [4].

The inequality  $\mu[F'(t, \eta)] \leq \nu$  is to be interpreted as a stability condition, both concerning the ODE system (2.2) and its implicit Euler discretization

$$(2.6) \quad U^n = U^{n-1} + \tau F(t_n, U^n), \quad n = 1, 2, \dots,$$

where  $\tau = t_n - t_{n-1}$  is the step size and  $U^n$  is the approximation for  $U(t_n)$ . This inequality enables us to formulate the following, powerful stability result for implicit Euler. Consider the perturbed form

$$(2.7) \quad \tilde{U}^n = \tilde{U}^{n-1} + \tau F(t_n, \tilde{U}^n) + r^n, \quad n = 1, 2, \dots,$$

where  $r^n$  is an arbitrary local perturbation and  $\tilde{U}^{n-1}$ ,  $\tilde{U}^n$  are perturbations to  $U^{n-1}$ ,  $U^n$ . Then

$$(2.8) \quad \|\tilde{U}^n - U^n\| \leq \frac{1}{1 - \tau\nu} \|\tilde{U}^{n-1} - U^{n-1} + r^n\|, \quad n = 1, 2, \dots,$$

for all  $\tau > 0$  satisfying  $\tau\nu < 1$  [4]. Since  $\nu$  is independent of the grid spacing, this stability inequality is valid uniformly in  $h_x$  and  $h_y$ . For  $\nu = 0$  we have contractivity for all  $\tau > 0$ , while for  $\nu < 0$  we even have damping for all  $\tau > 0$ . A result closely related to (2.8) will be derived in § 4.

### 3. The implicit Euler local uniform grid refinement method.

**3.1. Outline.** Although its elaboration readily becomes complicated, the idea behind LUGR is simple. Starting from  $\omega$ , finer and finer uniform subgrids are created locally in a nested manner in regions of high spatial activity. These subgrids are created by bisecting sides of next coarser grid cells. A new initial-boundary value problem is solved at each subgrid, and the integration takes place in a consecutive order, from coarse to fine. Each of these integrations spans the same time interval. Required initial values are defined by interpolation from the next coarser subgrid or taken from a subgrid from the previous time step when available. Internal boundaries are treated as Dirichlet boundaries and values are also interpolated from the next coarser subgrid. The generation of subgrids is determined by the local refinement strategy and is continued until the spatial phenomena are described well enough by the finest grid.

During each time step the following operations are performed:

1. Integrate on coarse base grid.
2. Determine new finer uniform subgrid at forward time.
3. Interpolate internal boundary values at forward time.
4. Provide new initial values at backward time.
5. Integrate on subgrid using the same steplength.
6. If the desired level of refinement is reached, go to 7, else go to 2.
7. Inject fine grid values in coinciding coarser grid points.

Thus, for each time step, the computation starts at the coarse base grid using the most accurate solution available, since fine grid solution values are always injected in coinciding coarse grid points. Moreover, all subgrids are kept in storage for step continuation.

We consider the use of uniform grids attractive because uniform grids allow an efficient use of vector-based algorithms, and finite differences on uniform grids are faster and more accurate to compute than those on nonuniform grids. In this respect, the current approach is to be contrasted with pointwise refinement leading to truly nonuniform grids. Pointwise refinement techniques also require a more involved data structure [5]. On the other hand, with the LUGR method, there are nodes that exist on more than one grid at the same time, meaning that in these nodes integration takes place more than once during one time step. Hence, the total number of nodal integrations needed will be larger than on a comparable single nonuniform grid.

In [2], [3], [7] and [11] LUGR methods are examined based on noncellular refinement and truly rectangular subgrids, which may rotate and overlap to align with

an evolving fine scale structure. We avoid these difficulties. Our local subgrids do not overlap, they may be disjunct, they need not be rectangles, and the actual refinement is cellular.

**3.2. The mathematical formulation.** LUGR methods solve PDEs on the whole domain at the coarsest grid only and on a part of the domain at finer subgrids. Our method can be interpreted as a sequence of operations on vectors in  $\mathbb{R}^d$  with varying dimension  $d$ . The dimensions are time and level dependent because the number of nodes changes per level of refinement and per time step. This constitutes a problem for the formulation of the method. To bypass this difficulty, the fine grids will be expanded so that they cover the whole domain. The dimensions are then fixed per level of refinement, which facilitates the derivation of a concise mathematical formulation. We emphasise that this grid expansion is auxiliary. In actual application, only part of the expanded higher-level grids is processed.

Suppose that for a given time interval  $[0, T]$  and a given base grid,  $l$  levels are needed to describe the spatial activity of a solution sufficiently accurately when integrating over the entire time interval  $[0, T]$ . Introduce for  $k = 1, \dots, l$  the expanded uniform grids

$$(3.1) \quad \omega_k = \{(x_i, y_j) : x_i = ih_{x,k}, 1 \leq i \leq 2^{k-1}M - 1 \text{ and } y_j = jh_{y,k}, 1 \leq j \leq 2^{k-1}N - 1\},$$

where  $N$  and  $M$  are the same integers as in (2.4), and  $h_{x,k} = h_x/2^{k-1}$ ,  $h_{y,k} = h_y/2^{k-1}$ . Note that for  $k = 1$  the base grid  $\omega_1 = \omega$  given by (2.4) is recovered.

Let the generic notation for a grid function  $\eta$  defined at  $\omega_k$  be  $\eta_k$ , and let  $S_k$  denote the space of these grid functions. We then denote the semidiscrete system considered in  $S_k$  by

$$(3.2) \quad \frac{d}{dt} U_k(t) = F_k(t, U_k(t)), \quad 0 < t \leq T, \quad U_k(0) = U_k^0.$$

Note that due to the grid expansion, only a part of the components of the ODE system (3.2) is integrated for  $k > 1$  in reality.

We are now ready to formulate the implicit Euler LUGR method. The following formula defines the time step from step point  $t_{n-1}$  to  $t_n$  for  $l$  levels of refinement:

$$(3.3a) \quad U_1^n = R_{1l} U_1^{n-1} + \tau F_1(t_n, U_1^n),$$

$$(3.3b) \quad U_k^n = D_k^n [R_{lk} U_l^{n-1} + \tau F_k(t_n, U_k^n)] + (I_k - D_k^n) [P_{k-1k} U_{k-1}^n + b_k^n],$$

for  $k = 2, \dots, l$ , where

$U_k^n \in S_k^n$  is the approximation to  $u$  at  $\omega_k$  at  $t = t_n$ ,

$I_k : S_k \rightarrow S_k$  is the unit matrix,

$D_k^n : S_k \rightarrow S_k$  is a diagonal matrix with entries  $(D_k^n)_{ii}$  either unity or zero,

$R_{lk} : S_l \rightarrow S_k$  is the natural restriction operator from  $\omega_l$  to  $\omega_k$ ,  $R_{ll} = I_l$ ,

$P_{k-1k} : S_{k-1} \rightarrow S_k$  is an interpolation operator from  $\omega_{k-1}$  to  $\omega_k$ ,

$b_k^n \in S_k$  contains time-dependent terms emanating from the boundary  $\partial\Omega$ .

Specifically, the nonzero entries of  $D_k^n$  ( $2 \leq k \leq l$ ) are meant to determine that part of  $\omega_k$  where the actual integration takes place. This integration has the fine grid solution  $D_k^n R_{lk} U_l^{n-1}$  as initial function and is defined by

$$(3.4) \quad D_k^n U_k^n = D_k^n [R_{lk} U_l^{n-1} + \tau F_k(t_n, U_k^n)], \quad k = 2, \dots, l.$$

The definition of  $D_k^n$  is provided by the refinement strategy. For the time being, there is no need to further specify  $D_k^n$ . Note that the nesting property of the integration

domains is hidden in the precise definition of the matrices  $D_k^n$ . The interpolation step is defined by

$$(3.5) \quad (I_k - D_k^n)U_k^n = (I_k - D_k^n)[P_{k-1k}U_{k-1}^n + b_k^n], \quad k = 2, \dots, l,$$

where the grid function  $b_k^n$  contains various time-dependent terms occurring in physical boundary conditions. We need to include  $b_k^n$  because physical boundary conditions have been worked into the semidiscrete system. For the analysis to follow,  $b_k^n$  plays no role whatsoever.

The formulation (3.3a, b) automatically comprises the interpolation of boundary values at grid interfaces. This follows directly from the observation that for nodes at grid interfaces, the associated diagonal entry of  $D_k^n$  is zero (there is no integration at grid interfaces). Further, we note that (3.3) implies an order, (3.3a) is carried out for the coarse base grid and (3.3b) for  $k=2, \dots, l$  successively. Having done this, the updating will take place, meaning that  $U_k^n$  is replaced by  $R_{lk}U_l^n$  from  $k=l-1$  to 1. After this we move on to the next time step. Recall that, due to the grid expansion, in (3.3a, b) the interpolation is carried out for all nodal points outside the integration domain of  $\omega_k$ . This enables the stability and convergence analysis to be carried out in the spaces  $S_k$ . However, in actual application, interpolation only takes place at the local subgrids. In § 6.2 it is shown that this does not interfere with the analysis.

#### 4. Stability analysis.

**4.1. Preliminaries.** Consider, along the same lines as (2.7) for  $n=1, 2, \dots$ , the perturbed scheme

$$(4.1a) \quad \tilde{U}_1^n = R_{l1}\tilde{U}_l^{n-1} + \tau F_1(t_n, \tilde{U}_1^n) + r_1^n,$$

$$(4.1b) \quad \tilde{U}_k^n = D_k^n[R_{lk}\tilde{U}_l^{n-1} + \tau F_k(t_n, \tilde{U}_k^n)] + (I_k - D_k^n)[P_{k-1k}\tilde{U}_{k-1}^n + b_k^n] + r_k^n,$$

for  $k=2, \dots, l$  with local perturbations  $r_k^n$ , and introduce the errors  $e_k^n = \tilde{U}_k^n - U_k^n$ , for  $k=1, \dots, l$ . To shorten the formulas, we introduce the auxiliary quantities  $e_0^n$ ,  $D_1^n$ , and  $P_{01}$ , where  $e_0^n = 0 \in S_1$ ,  $D_1^n$  is the unit matrix  $I_1$ , and  $P_{01}$  is the zero matrix. Then, by subtracting (3.3a, b) from (4.1a, b), we get

$$(4.2) \quad Z_k^n e_k^n = D_k^n R_{lk} e_l^{n-1} + (I_k - D_k^n) P_{k-1k} e_{k-1}^n + r_k^n, \quad n = 1, 2, \dots, \quad k = 1, \dots, l,$$

where  $Z_k^n = I_k - \tau D_k^n M_k^n$  and  $M_k^n$  is the integrated Jacobian matrix

$$(4.3) \quad M_k^n = \int_0^1 F'(t_n, \theta \tilde{U}_k^n + (1-\theta)U_k^n) d\theta,$$

which results from applying the mean value theorem for vector functions.

Assuming  $Z_k^n$  to be nonsingular, we can rewrite (4.2) as

$$(4.4) \quad e_k^n = X_k^n e_{k-1}^n + \Gamma_k^n e_l^{n-1} + \phi_k^n, \quad n = 1, 2, \dots, \quad k = 1, \dots, l,$$

with

$$(4.5) \quad \begin{aligned} X_k^n &= (Z_k^n)^{-1}(I_k - D_k^n)P_{k-1k}, \\ \Gamma_k^n &= (Z_k^n)^{-1}D_k^n R_{lk}, \\ \phi_k^n &= (Z_k^n)^{-1}r_k^n. \end{aligned}$$

Note that  $X_1^n = 0$  and that the operators  $X_k^n$ ,  $\Gamma_k^n$  are associated, respectively, to the interpolation and restriction. We can rewrite (4.4) in the standard form

$$(4.6) \quad e_k^n = G_k^n e_l^{n-1} + \psi_k^n, \quad n = 1, 2, \dots, \quad k = 1, \dots, l,$$

where the amplification operators  $G_k^n$  and the local perturbation terms  $\psi_k^n$  are defined by

$$(4.7) \quad \begin{aligned} G_1^n &= \Gamma_1^n, \\ G_k^n &= X_k^n G_{k-1}^n + \Gamma_k^n, \quad k=2, \dots, l, \end{aligned}$$

$$(4.8) \quad \begin{aligned} \psi_1^n &= \phi_1^n, \\ \psi_k^n &= X_k^n \psi_{k-1}^n + \phi_k^n, \quad k=2, \dots, l. \end{aligned}$$

The error recurrence (4.6) describes the error propagation for all refinement levels. The main interest lies in the operator  $G_l^n$  and the local perturbation  $\psi_l^n$ , since coarse grid values are always updated by fine grid values. In (4.6) this is reflected by the presence of  $e_l^{n-1}$ .

The stability of the implicit Euler method in the above is contained in the following lemma.

LEMMA 4.1. *Let  $\nu$  be the logarithmic norm value defined in assumption (A4) of § 2. Then,*

$$(4.9) \quad \begin{aligned} \|(Z_1^n)^{-1}\| &\leq \frac{1}{1-\tau\nu} \quad \forall \tau\nu < 1, \quad k=1, \\ \|(Z_k^n)^{-1}\| &\leq \begin{cases} 1/(1-\tau\nu) & \forall \tau\nu < 1 \quad \text{if } \nu > 0, \\ 1 & \forall \tau > 0 \quad \text{if } \nu \leq 0, \end{cases} \quad k=2, \dots, l. \end{aligned}$$

*Proof.* The result for  $k=1$  is standard since  $D_1^n$  is the unit matrix (see [4, p. 46]). The premultiplication of  $M_k^n$  for  $k > 1$  with  $D_k^n$  has the effect that either entire rows of  $M_k^n$  are put to zero, or are left unchanged. From (2.3) we can then immediately deduce that for  $\nu > 0$  the bound  $(1-\tau\nu)^{-1}$  still holds, whereas for  $\nu \leq 0$  the zero rows introduce the bound 1.  $\square$

Observe that the replacement of the bound  $(1-\tau\nu)^{-1}$  by the bound 1 for  $\nu < 0$  implies that in this case we no longer exploit the damping property of implicit Euler. For the analysis to follow, this is no restriction since here we are merely interested in proving stability and convergence results. Specifically, the stability result we will prove is not dependent on the damping in implicit Euler. To shorten derivations, we first make another assumption.

(A5) The logarithmic norm bound  $\nu$  from (A4) is nonpositive. Hence we now restrict ourselves to dissipative problems. This is not essential; results obtained for  $\nu \leq 0$  can be extended to the case  $\nu > 0$  by inserting  $(1-\tau\nu)^{-1}$  for the bound 1 any time the stability inequality  $\|(Z_k^n)^{-1}\| \leq 1$  is used.

**4.2. Stability and linear interpolation.** In this section we will prove a general stability result for the multilevel adaptive grid method (3.3) that is similar to the stability result (2.8) for the implicit Euler method applied without adaption.

THEOREM 4.2. *Let  $\nu \leq 0$  according to (A5), and suppose that linear interpolation is used. Then, for all  $\tau > 0$  and all  $n \geq 1$ ,*

$$(4.10) \quad \|G_k^n\| \leq 1, \quad k=1, \dots, l,$$

$$(4.11) \quad \|\psi_k^n\| \leq \sum_{j=1}^k \|r_j^n\|, \quad k=1, \dots, l,$$

$$(4.12) \quad \|e_l^n\| \leq \|e_l^{n-1}\| + \sum_{k=1}^l \|r_k^n\|.$$

*Proof.* Inequality (4.12) is a trivial consequence of (4.10) and (4.11). Let us first prove (4.10). This is done by induction with respect to  $k$ . Suppose  $\|G_{k-1}^n\| \leq 1$ . From (4.7) it follows that

$$(4.13) \quad \|G_k^n\| = \|X_k^n G_{k-1}^n + \Gamma_k^n\| = \|(Z_{k-1}^n)^{-1} Q_k^n\| \leq \|Q_k^n\|,$$

where  $Q_k^n = (I_k - D_k^n) P_{k-1k} G_{k-1}^n + D_k^n R_{lk}$ .

Consider the  $i$ th row of this operator. Suppose  $(D_k^n)_{ii} = 1$ . Then

$$(4.14) \quad \sum_j |(Q_k^n)_{ij}| = \sum_j |(R_{lk})_{ij}| = 1,$$

by definition of the restriction operator  $R_{lk}$ . Next suppose  $(D_k^n)_{ii} = 0$ . Then

$$(4.15) \quad \sum_j |(Q_k^n)_{ij}| = \sum_j |(P_{k-1k} G_{k-1}^n)_{ij}| \leq \|P_{k-1k} G_{k-1}^n\| \leq \|P_{k-1k}\| \|G_{k-1}^n\| \leq 1,$$

by virtue of the induction hypothesis and the norm

$$(4.16) \quad \|P_{k-1k}\| = 1$$

of the linear interpolation operator  $P_{k-1k}$ . Combining (4.14) and (4.15) gives  $\|Q_{k+1}^n\|$ , and inequality (4.10) now follows from (4.13). The induction proof is finished if we can prove that  $\|G_1^n\| \leq 1$ . This follows immediately from the observation that  $G_1^n = \Gamma_1^n = (Z_1^n)^{-1} R_{l1}$ .

There remains to prove (4.11). We have  $\|\phi_k^n\| \leq \|r_k^n\|$ . It then follows from (4.8) that

$$(4.17) \quad \|\psi_k^n\| \leq \|X_k^n\| \|\psi_{k-1}^n\| + \|r_k^n\|,$$

so that we are finished if we can prove that  $\|X_k^n\| \leq 1$ . This is trivial due to (4.16) and  $\|I_k - D_k^n\| = 1$ .  $\square$

The inequality (4.12) is the counterpart of the inequality (2.8). We may conclude from Theorem 4.2 that when implicit Euler is stable and we interpolate linearly, our multilevel adaptive grid method (3.3) retains stability of implicit Euler through the bound  $\|G_7^n\| \leq 1$ .

**4.3. Stability and higher-order interpolation.** A drawback of linear interpolation is its limited accuracy. In a genuine application, it might well be preferable to use higher-order interpolants (in [13] we successfully used fourth-order Lagrangian interpolation). Unfortunately, in this case we must have  $\|P_{k-1k}\| > 1$ , so that we are not able to prove the results of Theorem 4.2 when following the above method of proof. If  $\|P_{k-1k}\| > 1$ , then it is possible to prove (a constrained form of) stability by introducing an additional condition that underlies the intention of interpolating exclusively in low error regions. Unfortunately, this condition turns out to be of no direct practical use and is omitted here. On the other hand, numerical evidence suggests very strongly that those higher-order interpolants do not cause genuine stability problems in real application. We believe we owe this to the fact that the method interpolates in low error regions, so that, loosely speaking, this condition is satisfied implicitly.

**5. Error analysis.** We will present a detailed examination of the local error. From this we deduce the refinement condition which henceforth underlies the refinement strategy. This condition enables us to control the contribution of spatial interpolation errors in favour of spatial discretization errors. Due to this condition, we can prove a convergence result as if we are working on a single fixed grid. Specifically, it will be shown that the usual convergence behaviour applies and that the accuracy obtained is comparable to the accuracy obtained on the finest grid if this grid would be used without any adaptation.



**5.1. The local level error.** Let  $u_k(t) \in S_k$  denote the pointwise restriction of the true solution  $u(x, y, t)$  to  $\omega_k$ . Consider (4.1). By replacing all  $\tilde{U}$ -values by associated  $u_k$ -values, the local perturbation  $r_k^n$  becomes the local level error at grid level  $k$ . For convenience, we will denote this error also by  $r_k^n$ :

$$(5.1) \quad r_k^n = u_k^n - D_k^n [R_{lk} u_l^{n-1} + \tau F_k(t_n, u_k^n)] - (I_k - D_k^n) [P_{k-1k} u_{k-1}^n + b_k^n], \\ n = 1, 2, \dots, \quad k = 1, \dots, l,$$

where  $u_k^n = u_k(t_n)$  and  $P_{01}, u_0^n, b_1^n$  are auxiliary and put to zero;  $r_k^n$  contains the following local error components, the local spatial error induced by the finite-difference approximation, the local temporal error of the implicit Euler method, and the interpolation error. We first discuss these different components. They are defined in the standard way by

$$(5.2) \quad \alpha_k(t) = \frac{d}{dt} u_k(t) - F_k(t, u_k(t)) \quad (\text{spatial discretization error}),$$

$$(5.3) \quad \beta_k(t) = u_k(t) - u_k(t - \tau) - \tau \frac{d}{dt} u_k(t) \quad (\text{temporal error}),$$

$$(5.4) \quad \gamma_k(t) = u_k(t) - P_{k-1k} u_{k-1}(t) - b_k(t) \quad (\text{interpolation error}).$$

The grid function  $b_k(t)$  in (5.4) has the same meaning as  $b_k^n$  in (3.5). In the following, we assume without loss of generality that  $h_{x,k} = h_{y,k} = h_k$ . In view of assumptions (A2) and (A3) in § 2, we have

$$(5.5) \quad \alpha_k(t) = \mathcal{O}(h_k^2), \quad h_k \rightarrow 0,$$

with order constants determined by higher-order spatial derivatives of  $u$  and by PDE operator quantities. Likewise, (A2) implies  $\beta_k(t) = \tau^2 C_k$  where  $C_k = -\frac{1}{2} d^2 u_k / dt^2$  evaluated at a time  $t + (\kappa - 1)\tau$ ,  $0 \leq \kappa \leq 1$ . If  $u$  is a  $C^3$ -function in  $t$ , then

$$(5.6) \quad \beta_k(t) = -\frac{1}{2} \tau^2 \frac{d^2}{dt^2} u_k(t) + \mathcal{O}(\tau^3), \quad \tau \rightarrow 0.$$

Let  $q$  denote the accuracy order of the (Lagrangian) interpolation. Then

$$(5.7) \quad \gamma_k(t) = \mathcal{O}(h_k^q), \quad k = 2, \dots, l,$$

and here the order constants again depend exclusively on higher spatial derivatives of  $u$ , assuming sufficient differentiability. If linear interpolation is used, then assumption (A2) implies  $q = 2$  and second-order spatial derivatives determine the constants.

Now, using the relation  $u_k^{n-1} = R_{lk} u_l^{n-1}$  for  $k = 1, \dots, l$ , we can derive

$$(5.8) \quad r_k^n = D_k^n (\tau \alpha_k^n + \beta_k^n) + (I_k - D_k^n) \gamma_k^n, \quad n = 1, 2, \dots, \quad k = 1, \dots, l.$$

Note, by definition of  $D_k^n$ , that  $D_k^n (\tau \alpha_k^n + \beta_k^n)$  is the restriction of the usual local discretization error  $\tau \alpha_k^n + \beta_k^n$  to the integration domain of the grid  $\omega_k$ , while  $(I_k - D_k^n) \gamma_k^n$  represents the restriction of the interpolation error  $\gamma_k^n$  to the complement of this domain.

**5.2. A crude global error bound.** Denote the global discretization error by  $e_k^n = u_k^n - U_k^n$  and suppose  $e_k^0 = 0$ . For any choice of  $D_k^n$  the consistency results (5.5)-(5.8) imply

$$(5.9) \quad r_k^n = \mathcal{O}(\tau h_k^2) + \mathcal{O}(\tau^2) + \mathcal{O}(h_k^q).$$

If we now suppose linear interpolation and assumption (A5), then application of (4.12) yields

$$(5.10) \quad \|e_l^n\| \leq \|e_l^{n-1}\| + \|S^n\| \leq \|S^1\| + \dots + \|S^n\|, \quad n = 1, 2, \dots,$$

where  $\|S^n\| = \mathcal{O}(\tau h_1^2) + \mathcal{O}(\tau^2) + \mathcal{O}(h_1^2)$ . Here the coarsest mesh width occurs due to simply adding all normed local level errors in (4.15), including  $\|r_1^n\|$ . Following standard practice, we thus obtain at any fixed time point  $t_n = n\tau$  the global error bound

$$(5.11) \quad \|e_l^n\| \leq C_1\tau + C_2h^2 + C_3\frac{h^2}{\tau},$$

where  $h = h_1$  and  $C_1, C_2,$  and  $C_3$  are positive constants independent of step size and mesh sizes.

The first two terms are due to the temporal integration and spatial discretization. They will vanish if mesh sizes and step size tend to zero independently of each other, thus reflecting the unconditional convergence of the method when applied without adaptation. On the other hand, if no relation is imposed between  $\tau$  and  $h$ , then the third term can grow unboundedly as  $\tau, h \rightarrow 0$ . This term is due to the interpolation. Hence, even though we have stability and consistency, this result shows that unconditional convergence cannot be hoped for. Fortunately, this conclusion is not as bad as it looks. By not specifying the matrices  $D_k^n$  and, subsequently, by adding norms of the local level errors, we have simply supposed arbitrary integration domains at all levels of refinement. This must lead to a crude error bound like (5.11). In application, the computations should be organized in such a way that the interpolation only takes place in low error regions so that the interpolation error is virtually absent. This poses the task of setting up a precise error analysis and the design of a local refinement strategy aimed at a suitable selection of the matrices  $D_k^n$ .

**5.3. Local and global errors.** According to (4.6), the global error  $e_k^n$  satisfies the recurrence relation

$$(5.12) \quad e_k^n = G_k^n e_k^{n-1} + \psi_k^n, \quad n = 1, 2, \dots, \quad k = 1, \dots, l,$$

where  $\psi_k^n$  is the local error defined by recursion (cf. (4.8), (4.5))

$$(5.13) \quad \begin{aligned} \psi_1^n &= (Z_1^n)^{-1} r_1^n, \\ \psi_k^n &= X_k^n \psi_{k-1}^n + (Z_k^n)^{-1} r_k^n, \quad k = 2, \dots, l. \end{aligned}$$

The operators  $G_k^n, X_k^n,$  and  $Z_k^n$  are supposed to be redefined (replace all  $\tilde{U}$ -values by associated  $u_k$ -values). Note that  $\psi_k^n$  is essentially different from the local level error  $r_k^n$ . While  $r_k^n$  is associated with the single  $k$ th level,  $\psi_k^n$  is associated with all levels up to this  $k$ th level according to (5.13). This recursion governs the propagation of each local level error when introducing higher and higher levels. Elaborating, it gives, for  $k = 1, \dots, l,$

$$(5.14) \quad \psi_k^n = \sum_{j=1}^k \left( \prod_{i=k}^{j+1} X_i^n \right) (Z_j^n)^{-1} r_j^n.$$

Next we split  $\psi_k^n$  into its temporal and spatial part denoted by, respectively,  $\psi_{k,t}^n$  and  $\psi_{k,s}^n$ :

$$(5.15) \quad \psi_k^n = \psi_{k,t}^n + \psi_{k,s}^n, \quad k = 1, \dots, l,$$

and it follows from (5.8) that  $\psi_{k,t}^n$  and  $\psi_{k,s}^n$  are given, respectively, by

$$(5.16) \quad \psi_{k,t}^n = \sum_{j=1}^k \left( \prod_{i=k}^{j+1} X_i^n \right) (Z_j^n)^{-1} D_j^n \beta_j^n,$$

$$(5.17) \quad \psi_{k,s}^n = \sum_{j=1}^k \left( \prod_{i=k}^{j+1} X_i^n \right) (Z_j^n)^{-1} [\tau D_j^n \alpha_j^n + (I_j - D_j^n) \gamma_j^n].$$

Let us first examine  $\psi_{k,t}^n$ . Since  $\beta_k^n$  does not depend on mesh sizes, we have  $\beta_k^n = R_{ik}\beta_i^n$ . Substitution into (5.16) then yields

$$(5.18) \quad \psi_{k,t}^n = \sum_{j=1}^k \left( \prod_{i=k}^{j+1} X_i^n \right) (Z_j^n)^{-1} D_j^n R_{ij} \beta_i^n,$$

and we see that this operator is just the amplification operator  $G_k^n$  featured in (5.12); see the recursion (4.7). In conclusion,  $\psi_{k,t}^n$  satisfies

$$(5.19) \quad \psi_{k,t}^n = G_k^n \beta_l^n, \quad k = 1, \dots, l,$$

We next examine  $\psi_{k,s}^n$ . Using the definition of  $X_k^n$  given in (4.5), we rewrite (5.17) as

$$(5.20) \quad \begin{aligned} \psi_{k,s}^n &= (Z_k^n)^{-1} (I_k - D_k^n) P_{k-1k} \sum_{j=1}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} [\tau D_j^n \alpha_j^n + (I_j - D_j^n) \gamma_j^n] \\ &\quad + (Z_k^n)^{-1} [\tau D_k^n \alpha_k^n + (I_k - D_k^n) \gamma_k^n] \\ &= (Z_k^n)^{-1} [\tau D_k^n \alpha_k^n + (I_k - D_k^n) \rho_k^n], \quad k = 1, \dots, l, \end{aligned}$$

where

$$(5.21) \quad \rho_k^n = \gamma_k^n + P_{k-1k} \psi_{k-1,s}^n, \quad k = 2, \dots, l,$$

and  $\rho_1^n = 0$ . In (5.20) the spatial local discretization error  $D_k^n \alpha_k^n$  committed on the integration domain of grid  $\omega_k$  is separated from the spatial local error part  $(I_k - D_k^n) \rho_k^n$  defined outside this domain. Hence,  $\rho_k^n$  collects all spatial error contributions defined on the grids  $\omega_j$  ( $1 \leq j \leq k-1$ ), including discretization error  $\alpha_j^n$  and interpolation error  $\gamma_j^n$ , together with  $\gamma_k^n$  on  $\omega_k$ . This separation enables us to formulate a refinement condition which ensures that when a new grid level is introduced, the spatial local accuracy outside its integration domain will be smaller than or equal to the spatial accuracy on the integration domain itself. This distribution of local space errors is desirable, as we never return to grid points lying outside a current integration domain.

The refinement condition constrains the matrices  $D_k^n$ , and is taken to be

$$(5.22) \quad \|(Z_k^n)^{-1} \tau D_k^n \alpha_k^n\| \leq \frac{1}{c} \|(Z_k^n)^{-1} (I_k - D_k^n) \rho_k^n\|, \quad n = 1, 2, \dots, \quad k = 2, \dots, l,$$

where  $c > 0$  is a constant specified in § 6. If (5.22) is true, then all errors  $\psi_{k,s}^n$  satisfy

$$(5.23) \quad \|\psi_{k,s}^n\| \leq (1+c) \|(Z_k^n)^{-1} \tau D_k^n \alpha_k^n\|,$$

and combining (5.12) with (5.15), (5.19) enables us to present the global error inequality

$$(5.24) \quad \|e_k^n\| \leq \|G_k^n\| \|e_l^{n-1}\| + \|G_k^n \beta_l^n\| + (1+c) \|(Z_k^n)^{-1} \tau D_k^n \alpha_k^n\|, \\ n = 1, 2, \dots, \quad k = 1, \dots, l,$$

The importance of the refinement condition (5.22) is reflected by the fact that in (5.24) the interpolation error contribution has been removed. This is in agreement with our goal of developing a local refinement strategy that generates refined subgrids such that the accuracy obtained on the final finest grid is comparable to the accuracy obtained if this finest grid would be used without adaptation. We will elaborate on condition (5.22) in § 6. Note that it suffices to consider (5.22) only for  $k = l$ , since it suffices to consider (5.23) and (5.24) for  $k = l$ .

**5.4. Convergence and linear interpolation.** Assuming linear interpolation and assumption (A5), as in § 5.2, (5.24) can be rewritten as

$$(5.25) \quad \|e_k^n\| \leq \|e_l^{n-1}\| + \|\beta_k^n\| + (1+c) \tau \|\alpha_k^n\|, \quad n = 1, 2, \dots, \quad k = 1, \dots, l,$$

Hence, following the same derivation as carried out for (5.11), for the highest level  $l$  the global error bound

$$(5.26) \quad \|e_l^n\| \leq C_1 \tau + C_2(1+c)h_l^2$$

results where  $C_1$  and  $C_2$  are positive constants independent of step size and mesh sizes. This bound is unconditional in the sense that it assumes no relation between step size and mesh sizes and, according to our goal, the smallest mesh width  $h_l$  occurs. We have recovered an error bound similar to the standard error bound for implicit Euler when applied on a single grid.

**5.5. Convergence and higher-order interpolation.** As pointed out in § 4.3, for the case of higher-order Lagrangian interpolants, a powerful stability result like that of Theorem 4.2 is not available. However, assuming that higher-order interpolation in low error regions does not severely damage stability, as is strongly supported by our practical experience, it is natural to impose the refinement condition (5.22) also in the case of higher-order interpolation. Note that in the derivation of (5.22) no a priori choice was made for the interpolants.

#### 6. The refinement condition.

**6.1. Determining the integration domains.** Condition (5.22) first needs to be elaborated into a workable form before it can be implemented for determining the integration domains. To begin with, we rewrite the error  $\rho_k^n$  as

$$(6.1) \quad \begin{aligned} \rho_k^n = & \gamma_k^n + P_{k-1k} \sum_{j=1}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} \tau D_j^n \alpha_j^n \\ & + P_{k-1k} \sum_{j=2}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} (I_j - D_j^n) \gamma_j^n, \quad 2 \leq k \leq l. \end{aligned}$$

Next, we rewrite the first sum as

$$(6.2) \quad \begin{aligned} P_{k-1k} \sum_{j=1}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} \tau D_j^n \alpha_j^n = & P_{k-1k} (Z_{k-1}^n)^{-1} \tau D_{k-1}^n \alpha_{k-1}^n \\ & + P_{k-1k} \sum_{j=2}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} (I_j - D_j^n) \\ & \cdot P_{j-1j} (Z_{j-1}^n)^{-1} \tau D_{j-1}^n \alpha_{j-1}^n, \end{aligned}$$

and substitute this expression into (6.1). It then follows that  $\rho_k^n$  can be written as

$$(6.3) \quad \rho_k^n = \lambda_k^n + P_{k-1k} \sum_{j=2}^{k-1} \left( \prod_{i=k-1}^{j+1} X_i^n \right) (Z_j^n)^{-1} (I_j - D_j^n) \lambda_j^n, \quad k=2, \dots, l,$$

where

$$(6.4) \quad \lambda_j^n = \gamma_j^n + P_{j-1j} (Z_{j-1}^n)^{-1} \tau D_{j-1}^n \alpha_{j-1}^n, \quad j=2, \dots, l.$$

The error function  $\lambda_j^n$  contains the interpolation error at level  $j$  and the prolongation of the spatial discretization error of level  $j-1$  to level  $j$ . The derivation now rests upon monitoring the error  $(Z_k^n)^{-1} (I_k - D_k^n) \rho_k^n$  occurring in (5.22) through monitoring all errors  $(I_j - D_j^n) \lambda_j^n$ ,  $j \leq k$ , occurring in (6.3). The idea is to select the matrices  $D_j^n$  such that the error functions  $(I_j - D_j^n) \lambda_j^n$  become sufficiently small. This makes sense because if  $C_3$  and  $C_4$  are stability constants such that

$$(6.5) \quad \|(Z_j^n)^{-1}\| \leq C_3, \quad \left\| \prod_{i=k-1}^{j+1} X_i^n \right\| \leq C_4,$$

then

$$(6.6) \quad \|(Z_k^n)^{-1}(I_k - D_k^n)\rho_k^n\| \leq C_3(1 + \|P_{k-1k}\|(k-2)C_3C_4) \max_{2 \leq j \leq k} \|(I_j - D_j^n)\lambda_j^n\|.$$

Hence, if for  $k = 2, \dots, l$ , the matrices  $D_k^n$  are selected such that

$$(6.7) \quad C_3(1 + \|P_{k-1k}\|(k-2)C_3C_4) \max_{2 \leq j \leq k} \|(I_j - D_j^n)\lambda_j^n\| \leq c \|(Z_k^n)^{-1}\tau D_k^n \alpha_k^n\|,$$

then the refinement condition (5.22) is satisfied.

In general, the stability constants  $C_3$  and  $C_4$  are unknown. However, if the dissipativity assumption (A5) is satisfied, then the constant  $C_3 \leq 1$ . Furthermore, if we use linear interpolation, then (4.16) applies and  $C_4$  also can be set equal to one, so that (6.7) simplifies to

$$(6.8) \quad \max_{2 \leq j \leq k} \|(I_j - D_j^n)\lambda_j^n\| \leq \frac{c}{k-1} \|(Z_k^n)^{-1}\tau D_k^n \alpha_k^n\|, \quad k = 2, \dots, l.$$

If assumption (A5) does not hold or if higher-order interpolation is used, then  $C_3$  and  $C_4$  may be larger than one, but not by a considerable amount.  $C_3$  shall generally be of moderate size in view of the excellent stability behaviour of implicit Euler. Our practical experience with fourth-order Lagrangian interpolation is that higher-order interpolation is unlikely to yield instability problems, thus indicating that  $\|X_k^n\|$ , and hence  $C_4$ , are also of moderate size. That is why we proceed with (6.8) and also use it in situations where (A5) may be violated and/or higher-order interpolation is used.

In application, it suffices to impose (5.22) for  $k=l$  only, so that (6.8) can be replaced by

$$(6.9) \quad \max_{2 \leq k \leq l} \|(I_k - D_k^n)\lambda_k^n\| \leq \frac{c}{l-1} \|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|.$$

In order to satisfy this condition, estimates of  $\lambda_k^n$  have to be computed. Therefore, to create an extra safety margin, we replace (6.9) by the slightly more conservative condition

$$(6.10) \quad \|(I_k - D_k^n)\zeta_k^n\| \leq \frac{c}{l-1} \|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|, \quad k = 2, \dots, l,$$

where, componentwise,  $\zeta_k^n$  is defined as

$$(6.11) \quad (\zeta_k^n)_i = |(\gamma_k^n)_i| + |(P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n)_i|.$$

Condition (6.10) will determine the integration domain of  $\omega_k$ . Let  $\Omega_k^n$  be this integration domain and recall that when a node belongs to  $\Omega_k^n$ , the corresponding diagonal entry of  $D_k^n$  is equal to one and zero otherwise. Suppose that the maximal level number  $l$  and  $c(l-1)^{-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|$  are known and that a solution at  $\Omega_{k-1}^n$ ,  $k \leq l$ , has just been computed. Prior to the integration step on level  $k$ , our task is then to determine  $\Omega_k^n$ . That is, we must define  $D_k^n$  such that (6.10) is satisfied and in such a way that the area of  $\Omega_k^n$  is as small as possible. The actual selection of  $\Omega_k^n$  is carried out by a flagging procedure that scans level- $k$  grid points. A point is flagged if, using appropriate estimates,

$$(6.12) \quad (\zeta_k^n)_i > \frac{c}{l-1} \|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|.$$

Hence, for such a point the corresponding diagonal entry  $(D_k^n)_{ii} = 1$ , and for nonflagged points we define  $(D_k^n)_{ii} = 0$ . Thus the refinement condition (6.10) is satisfied.

In conclusion, the solution at a node of grid  $\omega_k$  is interpolated only if a corresponding component of  $\zeta_k^n$  is smaller than the maximum of the spatial discretization error at the finest grid multiplied with  $\tau c(l-1)^{-1}$ . Otherwise, integration is carried out at this node. No doubt this imposes a severe restriction on the size of the interpolation errors. 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, which means that the interpolation error will be carried along to the next time step. The fact that we do not return is a direct consequence of the nesting property of the integration domains, which we will discuss next.

**6.2. Restricted interpolation and the nesting property.** We now introduce the nesting property of the integration domains. Recall that this property, being hidden in the definition of the matrices  $D_k^n$ , has played no role in the foregoing analysis. We stipulate that in application the nesting is enforced by the flagging procedure; in other words, this procedure scans only level- $k$  points lying within the previous integration domain  $\Omega_{k-1}^n$ . A direct consequence is that, unlike (3.5), the interpolation is carried out only for level- $k$  points within  $\Omega_{k-1}^n$ . Here we will justify the deviation due to this restricted interpolation. We will argue that the restricted interpolation is in fact allowed by the inequality (6.10), where interpolation over the whole of  $\omega_k$  is still assumed.

Consider the error  $(Z_{k-1}^n)^{-1} \tau D_{k-1}^n \alpha_{k-1}^n$ , contained in  $\zeta_k^n$ . This spatial error is defined at level  $k-1$  and, by definition of  $D_{k-1}^n$ , has zero components outside  $\Omega_{k-1}^n$ . Hence, all its prolonged components are taken into account in the flagging procedure for determining  $\Omega_k^n$ . For the interpolation error  $\gamma_k^n$ , which lives on the whole of  $\omega_k$  (grid expansion), the situation is different. However, restricted interpolation is allowed if for all level- $k$  points outside  $\Omega_{k-1}^n$ , the interpolation error satisfies

$$(6.13) \quad |(\gamma_k^n)_i| \leq \frac{c}{l-1} \|(Z_l^n)^{-1} \tau D_l^n \alpha_l^n\|,$$

because then points outside  $\Omega_{k-1}^n$  will not be flagged if the interpolation step (3.5) would be carried out on the whole of  $\omega_k$ . In other words, if (6.13) holds outside  $\Omega_{k-1}^n$ , then the integration domains found with the restricted interpolation over  $\Omega_{k-1}^n$  are equal to the domains found if the interpolation would be carried out on the whole of  $\omega_k$ , which is in accordance with the method description (3.3a, b).

The following argument shows that inequality (6.13) is very plausible with the restricted interpolation procedure. First we recall that  $\Omega_1^n$  coincides with the entire physical domain. Hence for  $k=2$  there is no restricted interpolation, so that for all level-2 points outside  $\Omega_2^n$ , inequality (6.13) is trivially satisfied. Next consider the case  $k=3$ . Now the interpolation is restricted to level-3 points within  $\Omega_2^n$ . Since for all level-2 points outside  $\Omega_2^n$  inequality (6.13) is satisfied, we are justified in supposing that this is also true for all level-3 points outside  $\Omega_2^n$ , in view of the consistency of the interpolation (level-3 interpolation errors are smaller than level-2 errors). Further, by construction of  $\Omega_3^n$ , (6.10) is satisfied for all level-3 points within  $\Omega_2^n$  and outside  $\Omega_2^n$ , and so is (6.16). In conclusion, we may suppose that (6.13) is satisfied for all level-3 points outside  $\Omega_3^n$  when using the restricted interpolation for  $k=3$ . For  $k=4$  and so on this argument can be repeated.

**6.3. Implementation aspects.** On top of the flagging procedure implementing (6.12), a safety measure has been built. Any node for which (6.12) is true is flagged together with its eight neighbours. Next, to create an extra buffer, all sides of cells with at least one flagged corner node are bisected. This means that a buffer zone of two mesh widths is used around any intolerable node. Near boundaries, physical and

internal ones, the buffering differs slightly. Although in theory this buffering could be omitted, in practice it is wise to create a buffer zone around intolerable nodes because the estimation of higher spatial derivatives contained in  $\alpha_k^n$  and  $\gamma_k^n$  is prone to inaccuracies. After the flagging procedure, a cluster algorithm groups all flagged nodes together to form the newly defined integration domain.

The parameter  $c$  in (6.12) must be specified. In view of result (5.25),  $c$  should be taken small so that the spatial accuracy obtained is indeed nearly equal to the spatial accuracy obtained without adaptation. In fact, the smaller  $c$  is, the more points will be flagged and hence the safer the local refinement will be ( $c=0$  implies global refinement). On the other hand, when  $c$  is too large, it can occur that space errors are large and refinement is necessary but no nodes are flagged because (6.12) is satisfied at every node. Hence,  $c$  is available as a tuning parameter. In the experiments in § 7 we have simply put  $c = 1$ .

Estimates of spatial interpolation and discretization errors are required. For  $2 \leq k \leq l$  we must estimate the interpolation error  $\gamma_k^n$  and the prolonged spatial discretization error  $P_{k-1k}(Z_{k-1}^n)^{-1}D_{k-1}^n\alpha_{k-1}^n$ . Further, an estimate of the spatial discretization error  $(Z_l^n)^{-1}D_l^n\alpha_l^n$  committed at the final  $l$ th level must be available at all lower levels. Because we use local uniform grids, the estimation of these errors can be realized cheaply and easily. Consider the error  $\alpha_k^n$  (cf. (5.2)) and let  $p$  be the order of consistency (in this paper  $p = 2$ ). The estimation we apply is based on the use of a second spatial discretization operator  $\tilde{F}$  of a higher-order  $\tilde{p}$ . After some elementary calculations we obtain the approximation

$$(6.14) \quad \alpha_k^n \approx \tilde{F}_k(t_n, u_k(t_n)) - F_k(t_n, u_k(t_n))$$

as an asymptotically correct estimator for  $\alpha_k^n$ . The benefit of using uniform grids now lies in the fact that  $\tilde{F}$  is easily constructed. At internal nodes our  $\tilde{F}$  provides fourth-order accuracy (standard symmetrical differences), while at nodes adjacent to physical or internal boundaries third-order accuracy is realized (standard one-sided differences). The benefit of using uniform grids is also reflected in the estimation for the error  $\gamma_k^n$  (cf. (5.4)). So far we have implemented Lagrangian interpolation of second (linear) and fourth order. For the second-order interpolation we need to estimate spatial derivatives  $u_{xx}$ , etc., while in the fourth-order case spatial derivatives like  $u_{xxxx}$  appear. For both cases the estimation is straightforward.

We emphasise that, in spite of its simplicity, linear interpolation may become disadvantageous due to the low order of accuracy. Inspection of the various terms in (6.12) suggests a comparison between the following order relations:

$$(6.15a) \quad \tau(P_{k-1k}(Z_{k-1}^n)^{-1}D_{k-1}^n\alpha_{k-1}^n)_i = \mathcal{O}(\tau h_{k-1}^2),$$

$$(6.15b) \quad \|\tau(Z_l^n)^{-1}D_l^n\alpha_l^n\| = \mathcal{O}(\tau h_l^2),$$

$$(6.15c) \quad (\gamma_k^n)_i = \mathcal{O}(h_k^2), \quad \text{second-order Lagrangian (linear),}$$

$$(6.15d) \quad (\gamma_k^n)_i = \mathcal{O}(h_k^4), \quad \text{fourth-order Lagrangian.}$$

In the discretization terms the step size  $\tau$  is contained. Consequently, it is the interpolation error that may govern the refinement if  $\tau$  is very small, and particularly so when the interpolation is linear. The comparison is clearly in favour of the fourth-order interpolation.

To estimate the right-hand side term  $\|(Z_l^n)^{-1}D_l^n\alpha_l^n\|$  of (6.12) for  $2 \leq k \leq l-1$ , we exploit the asymptotics. Since the mesh width of level  $k$  is half that of level  $k+1$ , we

thus invoke

$$(6.16) \quad \|(Z_l^n)^{-1} D_l^n \alpha_l^n\| \approx 2^{-p(l-k)} \|(Z_k^n)^{-1} D_k^n \alpha_k^n\|, \quad l \geq k+1,$$

for  $k = 1, 2, \dots$ . In theory it suffices to do this only for  $k = 1$ , but since for larger values of  $k$  this estimation will become more and more accurate, it is done for every  $k$ .

Finally, we will make a few remarks about the approximations (6.14) and (6.16). Our method, like every other adaptive grid method, is designed to solve PDEs with steep solutions. Yet (6.14) and (6.16) underlie asymptotics, which means that they are only accurate if the solution is sufficiently smooth on the grid in use. This constitutes a problem for LUGR methods, because these methods estimate errors on coarse grids. Nevertheless, if in practice the estimated error is not that accurate, it might still give a good indication of where the spatial error is large and where it is not and, specifically, the estimated error might still be in the same order of magnitude as the exact error, in which case the implemented refinement strategy based upon (6.14) will still work. In our experience so far, this is indeed the case. We believe this is due to the fact that estimation (6.16) is carried out for finer and finer subgrids with an increasing accuracy which partly remedies the problem. However, if solutions become very steep, it might be necessary to improve the implementation of the refinement condition (5.22).

**7. Numerical example.** This section is devoted to an illustration of the foregoing error analysis. Our goal here is to numerically illustrate that by imposing the refinement condition, the usual order behaviour is recovered. At the same time, the spatial accuracy obtained is comparable to the spatial accuracy on the finest grid if this grid would be used without adaptation.

**7.1. The issue of implicitness.** We use the implicit Euler method for time integration. In connection with implicitness, two points are worth mentioning. The first is that at any time step refinement takes place at different levels, resulting in a different Jacobian per level whose order usually varies. This impedes the profitable use of old Jacobians (like in sophisticated stiff ODE solvers), unless it is decided not to adapt grids at every time step, but instead per prescribed number of steps. We consider this as part of an overall strategy that can easily be placed on top of the existing one. We adapt grids at every time step since our main aim with the experiments is to illustrate the convergence analysis together with the refinement strategy. However, when dealing with real applications, it is most likely to be more advantageous to omit adaptation at every time step, just for efficiency reasons. The second point is that the Jacobians do not possess a regular band structure, since the integration domains  $\Omega_k^n$  normally have an irregular shape. Unlike the first, this point is intrinsic to the local refinement method. In the experiments reported here, the Harwell sparse matrix solver MA28 has been used. This solver is well suited to coping with the structure we meet, but is rather time consuming for the present application. It is likely that standard iterative methods can be applied at lower costs.

**7.2. The example problem.** The problem is hypothetical and due to [1]. The equation is the linear parabolic equation

$$(7.1) \quad u_t = u_{xx} + u_{yy} + f(x, y, t), \quad 0 < x, y < 1, \quad t > 0,$$

and the initial function, the Dirichlet boundary conditions, and the source  $f$  are selected so that the exact solution is

$$(7.2) \quad u(x, y, t) = \exp[-80((x-r(t))^2 + (y-s(t))^2)],$$



where  $r(t) = \frac{1}{4}[2 + \sin(\pi t)]$  and  $s(t) = \frac{1}{4}[2 + \cos(\pi t)]$ . This solution is a cone that is initially centered at  $(\frac{1}{2}, \frac{3}{4})$  and that symmetrically rotates around  $(\frac{1}{2}, \frac{1}{2})$  in a clockwise direction with a constant speed. We have used this problem to subdue our refinement method to a convergence test. Observe that the semidiscrete version of this problem satisfies the dissipativity assumption (A5).

**7.3. Convergence experiments.** We have carried out two identical convergence experiments. In the first, linear interpolation was used and in the second, fourth-order Lagrangian was used. In both the solution is computed four times over the interval  $0 \leq t \leq 2$ , using a uniform  $10 \times 10$  base grid and a constant time step size  $\tau$ . In the first computation  $l=1$ , in the second  $l=2$ , and so on. Since per computation the smallest mesh width is halved,  $\tau$  is simultaneously decreased by  $2^2$  in view of the first order of implicit Euler. Hence, in line with our analysis, per computation the maximal global error should also decrease by  $2^2$ .

Table 7.1 shows the maxima of global errors restricted to the finest integration domain in use. This table also contains the maxima of the errors for the corresponding grid used without adaptation. The table clearly reveals the expected order behaviour. The errors of the  $l=4$  runs are about a factor four smaller than the corresponding errors of the  $l=3$  runs. Note that there is hardly a difference between the corresponding errors, showing that, as anticipated by our strategy, the choice of interpolant has no notable influence on the error. We emphasise that, in spite of the relatively large values for  $\tau$ , the spatial error dominates the global errors shown in this table. For example, using  $\tau=0.125$  instead of  $\tau=0.5$  in the  $l=2$  run, the same global errors are found (they deviate in the third or fourth decimal digit). In other words, conclusions on the spatial error behaviour induced by the local refinement algorithm can be drawn from this table. These results convincingly show, for the current example problem, that the use of the refinement condition ensures that the spatial accuracy obtained is very much comparable to the spatial accuracy on the finest grid if this grid is used without any adaptation. Finally we note that the choice  $c=1$  apparently has no influence on the error. We owe this to the fact that the refinement condition has been derived from error bounds and is thus conservative.

The use of the two different interpolants is expressed in the slightly different integration domains shown in Figs. 7.1 and 7.2. As expected, at the higher levels linear

TABLE 7.1

*Maxima of global errors restricted to the finest domain. Comparison with errors on a standard uniform grid.*

$\tau$	No. of levels	Interpolation	Single grid	$t$			
				0.50	1.00	1.50	2.00
2.00000	1		$10 \times 10$				0.16447
0.50000	2	linear fourth order		0.03876	0.03890	0.03891	0.03891
				0.03929	0.03945	0.03946	0.03946
			$20 \times 20$	0.03865	0.03881	0.03882	0.03882
0.12500	3	linear fourth order		0.01369	0.01369	0.01369	0.01369
				0.01376	0.01376	0.01376	0.01376
			$40 \times 40$	0.01389	0.01389	0.01389	0.01389
0.03125	4	linear fourth order		0.00340	0.00340	0.00340	0.00340
				0.00359	0.00359	0.00359	0.00359
			$80 \times 80$	0.00347	0.00347	0.00347	0.00347

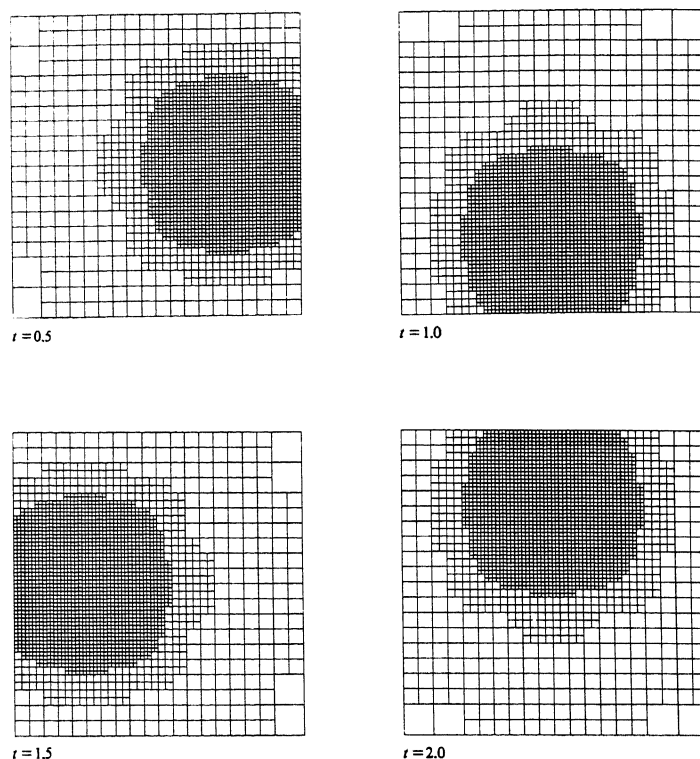


FIG. 7.1. Linear interpolation. Integration domains for the  $l=4$  run at four different times. The size of the integration domains decreases only slowly with the number of levels. This is due to the fact that the cone is not very steep. At the end time,  $t=2.0$ , the number of nodes amounts to 121, 425, 813, and 1917, respectively.

interpolation gives rise to somewhat larger domains, showing that linear interpolation is more expensive. As a rule, fourth-order interpolation is to be preferred, as it leads to smaller domains. Note that for both interpolants the moving domains accurately reflect the symmetric rotation of the cone, which once again nicely illustrates the reliability of the implemented refinement condition with the various estimators.

**8. Final remarks and future plans.** In our future research we plan to pay more attention to time-stepping efficiency. Using the refinement strategy of this paper as a starting point, we plan to examine the application of methods possessing a higher order in time. Natural candidates belong to the class of Runge–Kutta methods. It should be stressed, though, that fully implicit methods can only be of serious advantage if the numerical algebra issue can be satisfactorily solved. In this connection splitting methods of the ADI and LOD type (see [9]) may therefore provide an attractive alternative to fully implicit ones, although they are usually less accurate in time. Another point of serious practical concern is to apply methods not only using an a priori chosen number of levels, but to also have the possibility to vary the number of levels. This might be useful for the computation of solutions that, for example, steepen in time, like the combustion problem in [13]. For such problems, the application of a variable number of levels should be combined with the use of variable temporal step sizes. Preferably, the complete adaptation should then be monitored by estimators of temporal and spatial errors in such a way that there is a balance between the two which aims at minimizing the waste of computing time.

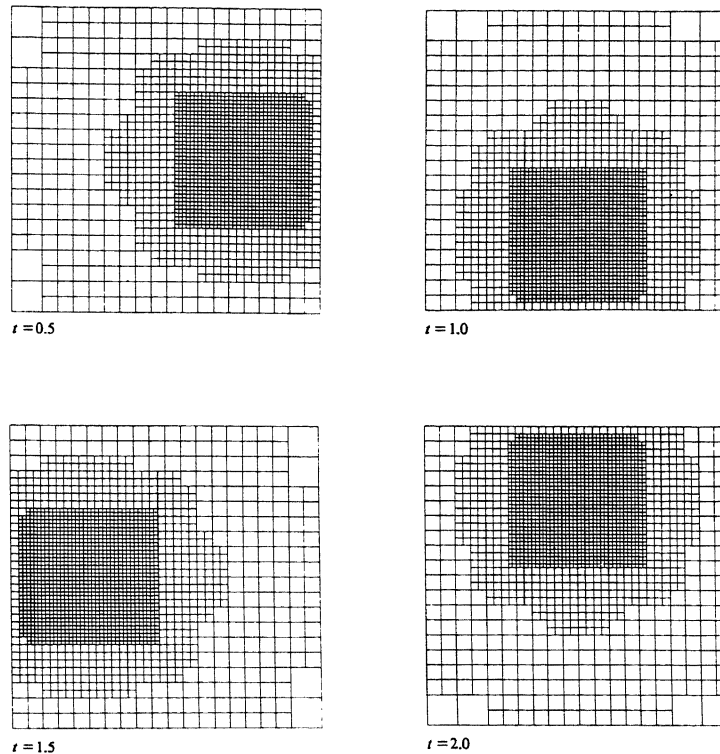


FIG. 7.2. Fourth-order interpolation. Integration domains for the  $l=4$  run at four different times. At the end time,  $t=2.0$ , the number of nodes amounts to 121, 425, 813, and 1361, respectively.

**Acknowledgment.** We would like to thank Willem Hundsdorfer for his careful reading of the manuscript.

#### REFERENCES

- [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), pp. 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), pp. 257-274.
- [3] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484-512.
- [4] K. DEKKER AND J. G. VERWER, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, Amsterdam, New York, Oxford, 1984.
- [5] R. E. EWING, *Adaptive grid refinement for transient flow problems*, in Adaptive Methods for Partial Differential Equations, J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1989.
- [6] W. D. GROPP, *A test of moving mesh refinement for 2D-scalar hyperbolic problems*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 191-197.
- [7] ———, *Local uniform mesh refinement with moving grids*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 292-304.
- [8] ———, *Local uniform mesh refinement on vector and parallel processors*, in Large Scale Scientific Computing, P. Deuffhard and B. Engquist, eds., Birkhäuser Series in Progress in Scientific Computing, Birkhäuser-Verlag, Basel, 1987, pp. 349-367.
- [9] W. H. HUNSDORFER AND J. G. VERWER, *Stability and convergence of the Peaceman-Rachford ADI method*, Math. Comp., 53 (1989), pp. 81-101.

- [10] S. MCCORMICK AND J. W. THOMAS, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., 46 (1986), pp. 439-456.
- [11] J. E. FLAHERTY, P. K. MOORE, AND C. OZTURAN, *Adaptive overlapping grid methods for parabolic systems*, in Adaptive Methods for Partial Differential Equations, J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1989.
- [12] J. M. SANZ-SERNA AND J. G. VERWER, *Stability and convergence at the PDE/stiff ODE interface*, Appl. Numer. Math., 5 (1989), pp. 117-132.
- [13] R. A. TROMPERT AND J. G. VERWER, *A static-regridding method for two dimensional parabolic partial differential equations*, Appl. Numer. Math., 8 (1991), pp. 65-90.