Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

R.A. Trompert, J.G. Verwer

Analysis of the implicit Euler local uniform grid refinement method

Department of Numerical Mathematics Report NM-R9011 July

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Analysis of the Implicit Euler Local Uniform Grid Refinement Method

R.A. Trompert and J.G. Verwer
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The subject of the paper belongs to the field of numerical solution of time-dependent partial differential equations. Attention is focussed on parabolic problems the solution of which possess sharp moving transitions in space and time, such as steep moving fronts and emerging and disappearing layers. 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 the techniques called static-regridding and local uniform grid refinement. Static-regridding means that in the course of the time evolution 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 uniformly refined grids. These uniform subgrids possess non physical boundaries and on each of these subgrids an integration is carried out. 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 PDE's. The central issue hereby is a refinement condition with a refinement 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.

1980 Mathematics subject classification: Primary:65M50. Secondary:65M20.
1987 CR Categories: G.1.8.
Key Words & Phrases: partial differential equations, numerical mathematics, time-dependent problems, adaptive grid methods, error analysis.

Note: This report will be submitted for publication elsewhere.

1. Introduction

The subject of the paper belongs to the field of numerical solution of time-dependent partial differential equations. Attention is focussed on parabolic problems the solution of which possess sharp moving transitions in space and time, such as steep moving fronts and emerging and disappearing layers. For such problems, a space grid held fixed throughout the entire time evolution can be computationally very inefficient, since, to afford an accurate approximation, such a grid would easily have to contain a very large number of nodes, particulary so for problems in two and three space dimensions. 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.

Report NM-R9011
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Our method is based on the techniques called static-regridding and local uniform grid refinement, as previously proposed by Berger and Oliger[4], Gropp[9], Arney and Flaherty[2], Trompert and Verwer[14], and others. With the term static-regridding we mean that in the course of the time evolution the space grid is adapted at discrete times. Static-regridding should be contrasted with dynamic-regridding where the space grid moves continuously in the space-time domain as, e.g., in the case of the moving finite element method. With the term local uniform grid refinement we mean that the actual adaptation of the space grid takes place using locally uniformly refined grids. Local uniform grid refinement should be contrasted with pointwise refinement which leads to truly nonuniform grids. The idea of the method can be briefly described as follows. Given a coarse base space grid and a base temporal stepsize, nested local uniform spatial subgrids are generated. These local uniform 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 base space-time grid while the fine grid results computed at forward time levels are kept in storage so that they can be used for step continuation.

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 PDE problems with almost one and the same code, assuming hereby that the grid and the associated data structure remain unchanged. In this connection we wish to note that an important part of the developement and coding of static-regridding methods like ours lies in the grid and data structure. 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 also 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[14], in fact, the grid structure, the data structure, the spatial differencing and the memory use are essentially the same. However, in the present paper we concentrate on analysis rather than on construction, while using the implicit Euler method instead of the explicit Runge-Kutta-Chebychev method for time integration. The main aim of this paper is to present a detailed local error analysis that takes into account all local temporal, spatial and interpolation errors and to prove stability and convergence for a certain class of linear and nonlinear PDE's. The central issue in this analysis is a refinement condition and a refinement 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.

The implicit Euler integration method is known to possess very favourable stability properties (see Dekker and Verwer[5]). A drawback is that this method has only order of consistency one. We wish to point out that the adaptive grid approach presented here, together with the refinement condition and the essential parts of the refinement strategy, is not restricted to implicit Euler. Most of it can be extended to a wide diversity of one-step integration methods such as splitting methods (e.g. ADI and LOD, see Hundsdorfer and Verwer[10]) and conventional Runge-Kutta and linear multistep methods. However, this is possible only at the expense of much more complicated derivations and calculations. Because we first wish to clarify the essential aspects and properties of the refinement condition and strategy for a simple time-stepping scheme, we confine ourselves in this paper to implicit Euler and postpone the treatment of alternative, more complicated methods to a later publication.

An overview of the organization and results of the paper reads as follows. Section 2 is devoted to the problem class discussed in this paper. Our starting point hereby is the method of lines, that is, we define a class of semi-discrete PDE's satisfying desirable properties for the analysis to follow (see e.g., Sanz-Serna and Verwer[13]). In Section 3 we introduce the tools and the formulation for the multilevel local uniform grid refinement method. In Section 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 present a detailed examination of the total local error with its component parts. Furthermore, in this section 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 refinement condition, we are able to prove a convergence result as if we are working on a single fixed grid. This refinement condition is further elaborated in Section 6 where it is shown how to implement it for practical use. A numerical illustration of the error analysis presented is given in Section 7. The numerical results found here are in complete agreement with this analysis. The final Section 8 briefly discusses our future research plans.

2. The problem class

In our method description and analysis we follow the method of lines approach (see Sanz-Serna and Verwer[13] and the references therein). Consider a real abstract Cauchy problem

$$u_t = L(t, u), \qquad 0 < t \leqslant T, \qquad u(\underline{x}, 0) = u^0(\underline{x}), \tag{2.1}$$

where L represents a second order partial differential operator which differentiates the unknown function $u(\underline{x},t)$ with respect to its space variable $\underline{x}$ in a space domain Ω in $\mathbb{R}$, $\mathbb{R}^2$ or $\mathbb{R}^3$. The function $u(\underline{x}, t)$ may be vector valued and boundary conditions for $0 < t \leqslant T$ are supposed to be included in the definition of L. The operator L should not differentiate with respect to the time variable t.

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

$$\frac{d}{dt}U(t) = F(t, U(t)), \qquad 0 < t \leqslant T, \qquad U(0) = U^0, \tag{2.2}$$

which is defined by a finite-difference space-discretization of (2.1). Thus, U and F are vectors in $\mathbb{R}^d$ representing the values of grid functions defined on a finite-difference space grid ω 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 we here consider an explicit ODE system which rules out mixed problems of parabolic-elliptic type as discussed in Petzold[11, 12]. The explicitness of the ODE system also rules out certain implicit boundary conditions which upon spatial discretization would convert into algebraic equations. In other words, we consider the standard approach in that boundary values have been eliminated and worked into the ODE system. However, since we here examine the implicit Euler method, the complete algorithm can be extended for application to implicit ODE/DAE systems as considered.

Our method description and error analysis are centered around the ODE system (2.2). We consider this convenient for the analysis and for the general treatment we aim at. We will now introduce some notations and a number of assumptions needed for further specifying (2.1) and (2.2). The norm symbol $\|.\|$ denotes the maximum norm on the vector space $\mathbb{R}^d$. We will use the same symbol for the induced matrix norm and throughout our analysis we will deal only with this maximum norm. We next introduce the logarithmic matrix norm $\mu[A]$ associated to the maximum norm of the real $d \times d$ matrix $A = (a_{ij})$; $\mu[A]$ is given by

$$\mu[A] = \max_i(a_{ii} + \sum_{j \neq i} |a_{ij}|). \tag{2.3}$$

The logarithmic norm is a useful tool in the stability analysis of certain models of nonlinear, stiff ODE's and semi-discrete PDE problems [5]. In this stability 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 our assumptions we make in further specifying (2.1) - (2.2). These assumptions are concerned with, respectively, the class of PDE's (2.1), the smoothness of their solutions, the choice of spatial grid and actual finite-differencing, and the stability of the semi-discrete system (2.2):

(A1) The local uniform grid refinement method and the analysis presented are in principle applicable in any number of space dimensions. However, following [14], we will consider the two-dimensional case while Ω is supposed to be the unit square, just for simplicity of presentation. With minor changes the domain is allowed to be composed of a union of rectangles with sides parallel to

the co-ordinate axes. In fact, as we will see later, refined grids normally are of this shape. In what follows, we will mostly use the notation $u(x,y,t)$, rather than $u(\underline{x},t)$.

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

(A3) Because we employ the notion of local uniform grid refinement, we will invariably use uniform space grids. Our base grid thus can be written as

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

where $h_x = 1/M$, $h_y = 1/N$ and M, N are positive integers. The spatial differencing on this uniform grid is supposed to be based on 3-point formulas of 2^{nd} order consistency. As a rule, we use central differencing for approximating 1^{st} and 2^{nd} order spatial derivatives. For boundary conditions involving 1^{st} order derivatives, the one-sided 3-point formula is used.

(A4) A constant ν exists such that $\mu[F'(t,\eta)] \leqslant \nu$ 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 local uniform grid refinement method remains applicable in situations where these assumptions do not hold or cannot be verified. On the other hand, for interesting classes of differential operators, such as the scalar, nonlinear parabolic operator

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

with standard restrictions on the functions f_i and p_i, one can prove the existence of a constant ν (see, e.g. [5]). The key point here is that ν is independent of the grid spacing and that this way one can also recover the damping property of parabolic operators through a negative logarithmic norm bound ν.

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

$$U^n = U^{n-1} + \tau F(t_n, U^n), \qquad n = 1,2,\ldots, \tag{2.6}$$

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

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

where r^n represents an arbitrary local perturbation and $\tilde{U}^{n-1}$, $\tilde{U}^n$ are perturbations to U^{n-1}, U^n. We have [5]

$$\|\tilde{U}^n - U^n\| \leqslant \frac{1}{1 - \tau\nu}\|\tilde{U}^{n-1} - U^{n-1} + r^n\|, \qquad n = 1,2,\ldots, \tag{2.8}$$

for all $\tau > 0$ satisfying $\tau\nu < 1$. Since ν is independent of the grid spacing, this stability inequality is valid uniformly in h_x and h_y. Note that 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 the stability analysis presented in Section 4.

3. The Implicit Euler Local Uniform Grid Refinement Method

3.1. Outline

We begin with an outline of the method in order to facilitate the presentation in the remainder (cf. [14]). The regridding technique we employ is based on the principle of local uniform grid refinement. Although its elaboration readily becomes complicated, the idea behind this principle is simple. Starting from a coarse, uniform base grid ω which covers the entire physical domain, finer-and-finer uniform subgrids are created in a nested manner in regions of high spatial activity which cover only part of the domain. A new initial-boundary value problem is solved at each fine subgrid and the integration takes place in a consecutive order, from coarse to fine. Each of these integrations spans the same time interval. The generation of the finer-and-finer subgrids is determined by the local refinement strategy and is continued until the spatial phenomena are described well enough by

the finest grid.

We will outline all steps taken for advancing the solution at the coarse base grid ω at time $t = t_{n-1}$ to the same coarse base grid at time $t = t_n = t_{n-1} + \tau$. The variable τ is the temporal stepsize for the Euler method and t_{n-1} and t_n are step points. For clarity we discuss first a single level of refinement :

(1) Integrate on the coarse grid using steplength τ. Call the accepted values of u on the coarse grid at time $t = t_n$ the new coarse u-values and those at time $t = t_{n-1}$ the old coarse u-values. Both sets of values are saved.

(2) An integration step is followed by a local refinement step. Using the new coarse u-values, it is decided where the fine subgrid will be for the current time interval $t_{n-1} \leqslant t \leqslant t_n$. This is done by invoking a flagging procedure to determine untolerable cells and a clustering and buffering algorithm to distribute all untolerable cells over the fine subgrid. This grid may be disjunct. Overlapping fine subgrids are not allowed in our method and fine subgrids need not be a rectangle. The actual refinement is cellular and carried out by bisecting all sides of untolerable cells.

(3) Because together with a new grid a new initial-boundary value problem is created, initial and boundary values must be provided by interpolation. In prescribing new initial values, two different cases can occur. If a cell was refined in the previous time step from t_{n-2} to t_{n-1}, then we use the fine grid u-values already available at $t = t_{n-1}$ and interpolation is not needed. If a cell was not refined before, then we interpolate old coarse u-values. For the boundary conditions we can also distinguish two cases. When a fine grid boundary coincides with the physical boundary, then the given boundary conditions are used. Otherwise we impose Dirichlet boundary conditions at the grid interfaces where fine grid cells abut on coarse cells. Because we integrate with the one-step, one-stage implicit Euler scheme using the same steplength τ on the fine grid, boundary values are needed only at the forward time level. These are obtained from interpolating the coarse grid solution.

(4) Next the fine subgrid is integrated from t_{n-1} to t_n and the new fine grid u-values are injected in the coarse grid points, i.e., the value of u at t_n on the coarse subgrid is taken to be the value of u just

computed on the fine subgrid. Further, all new fine grid u-values are saved for use in the next time step from t_n to t_{n+1}. The solution at time t_n is now complete. If t_n coincides with the physical end time T we are finished. Otherwise, we return to step (1) and repeat the algorithm.

Multiple levels are handled in a natural, recursive fashion. After each time step of (4), taken on a grid of refinement level k, say, a regridding may take place resulting in a grid of refinement level $k+1$. In fact, the computational steps follow precisely points (2) - (4) above. Note that all fine grid results at forward time levels are kept in storage and that for step continuation the most accurate results are used that are available. It is noted that the refinement process is slightly simpler than in [14], since we here use the same steplength τ for all levels encountered within one base time step.

Inherent in the approach we have adopted is that grid information is not passed to the next time step when starting again from the base grid. This necessarily is a bit wasteful in situations where the sharp transitions move very slowly, e.g., when approaching steady state. On the other hand, the computational effort for the coarser grids normally will not be large. Further, uniform subgrids allow an efficient use of vector based algorithms and finite-difference expressions on uniform grids are more accurate and cheaper to process than on nonuniform grids. In this respect the current approach should be contrasted with pointwise refinement where arbitrary levels of refinement around any point are allowed. This pointwise refinement leads to truly non-uniform grids on which usually less points are needed than on a locally uniformly refined grid. However, an inherent drawback of a truly non-uniform grid is a more complex and expensive data structure. We refer to Gropp[7-9], Berger[3] and Ewing[6] for some further discussions on various advantages and/or disadvantages when comparing the two refinement techniques. We also note that the methods of Berger and Oliger[4] (see also Arney and Flaherty[2]) are based on noncellular refinement and truly rectangular subgrids which may overlap and rotate to align with an evolving dynamic structure.

3.2. The mathematical formulation

Local uniform grid refinement methods solve PDE's on the whole domain at the coarse base 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. It is emphasized that this grid expansion is auxiliary. In actual application only part of the expanded higher level grids is processed (integration, interpolation, injection).

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 $k = 1, \cdots, l$ the expanded uniform grids

$$\omega_k = \{(x_i, y_j): x_i = ih_{x,k},\ 1 \leqslant i \leqslant 2^{k-1}M-1 \text{ and} \tag{3.1}$$

$$y_j = jh_{y,k},\ 1 \leqslant j \leqslant 2^{k-1}N-1\},$$

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. It is assumed that this base grid is prescribed for the entire time interval $[0,T]$. Note that ω_k is obtained from ω_{k-1} simply by dividing the current grid sizes so that $\omega_k \supset \omega_{k-1}$. It is emphasized that in the analysis in the remainder the number l is independent of time.

The generic notation for a grid function η defined at the grid ω_k is η_k. Let $NPDE$ be the number of partial differential equations contained in the system (2.1). Then the dimension of the grid functions living on ω_k is $d_k = NPDE(2^{k-1}M-1)(2^{k-1}N-1)$ so that $\eta_k \in \mathbb{R}^{d_k}$. We will denote the semi-discrete system defined on the whole of ω_k by

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

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

For the communication between grids we need a restriction operator from ω_l to ω_k for $k = 1,...,l-1$ and a prolongation operator from ω_{k-1} to ω_k for $k = 2,...,l$. The restriction operator is needed for updating a solution at ω_k by a solution available at the finest level grid ω_l. The prolongation operator is needed for interpolating a solution at ω_{k-1} to the next finer grid ω_k.

Let R_{lk} denote the restriction operator from ω_l to ω_k. Specifically, $R_{lk}:\mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_k}$ is a matrix with d_l columns and d_k rows and the restriction of η_l to ω_k is carried out with the matrix-vector multiplication

$$\eta_k = R_{lk}\eta_l \tag{3.3}$$

We apply natural injection at coinciding nodes. Therefore, all the entries of all rows of R_{lk} are zero, except for one which is unity. For later use we also define $R_{ll} = I_l$, with I_l being the unit matrix of order d_l.

Let P_{k-1k} denote the prolongation operator from ω_{k-1} to ω_k. This operator $P_{k-1k}:\mathbb{R}^{d_{k-1}} \rightarrow \mathbb{R}^{d_k}$ is a matrix with d_{k-1} columns and d_k rows and the interpolation is carried out with the matrix vector multiplication

$$\eta_k = P_{k-1k}\eta_{k-1} \tag{3.4}$$

Each row of P_{k-1k} is associated to a node of ω_k. For coinciding nodes of ω_{k-1} and ω_k all entries in this row are zero except for one which is unity. Hence the interpolation leaves the values at coinciding nodes unchanged. For non-coinciding nodes the actual choice of interpolant determines the entries in this row.

We are now ready to formulate the implicit Euler local uniform grid refinement method. The following formula defines the complete time step from step point t_{n-1} to the next step point t_n involving l levels of refinement:

$$U_1^n = R_{l1}U_l^{n-1} + \tau F_1(t_n, U_1^n), \tag{3.5a}$$

$$U_k^n = D_k^n [R_{lk} U_l^{n-1} + \tau F_k(t_n, U_k^n)] + \tag{3.5b}$$

$$(I_k - D_k^n) [P_{k-1k} U_{k-1}^n + b_k^n], \qquad k = 2, ..., l.$$

Here, U_k^n denotes the approximation to $U_k(t_n)$ defined at the whole of the grid ω_k at time $t = t_n$ and D_k^n and I_k are diagonal matrices of order d_k. I_k is the unit matrix and the entries of D_k^n are either unity or zero depending whether integration takes place at a specific node or not.

Specifically, the nonzero entries of D_k^n ($2 \leqslant k \leqslant l$) are meant to determine that part of ω_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

$$D_k^n U_k^n = D_k^n [R_{lk} U_l^{n-1} + \tau F_k(t_n, U_k^n)], \qquad k = 2, ..., l. \tag{3.6}$$

The definition of D_k^n is provided by the actual local refinement strategy. For the time being there is no need to further specifying D_k^n. In this connection it is worthwile to note that the nesting property of the finer-and-finer integration domains is hidden in the precise definition of the matrices D_k^n.

The interpolation step is defined by

$$(I_k - D_k^n) U_k^n = (I_k - D_k^n) [P_{k-1k} U_{k-1}^n + b_k^n], \qquad k = 2, ..., l, \tag{3.7}$$

where the gridfunction b_k^n contains various time dependent terms occurring in physical boundary conditions. We need to include this gridfunction due to the fact that physical boundary conditions have been worked into the semi-discrete system. For the analysis presented in the remainder, b_k^n plays no role whatsoever.

The formulation (3.5) automatically comprises the interpolation of boundary values at genuine grid interfaces. This follows directly from the observation that for nodes at genuine grid interfaces, the associated diagonal entry of D_k^n is zero (there is no integration at grid interfaces). Also note that, as a result of the grid expansion, in (3.7) the interpolation is carried out for all nodal points outside the integration domain of ω_k. However, in actual application interpolation only takes place at the refined subgrids. In Section 6.2 it is shown that this does not interfere with the analysis of the remaining sec-

tions.

4. Stability analysis

4.1. Preliminaries

Consider, on the analogy of (2.7), for $n = 1,2,....$ the perturbed scheme

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

$$\tilde{U}_k^n = D_k^n [R_{lk}\tilde{U}_l^{n-1} + \tau F_k(t_n, \tilde{U}_k^n)] + \tag{4.1b}$$

$$(I_k - D_k^n)[P_{k-1k}\tilde{U}_{k-1}^n + b_k^n] + r_k^n, \qquad k = 2,....,l,$$

with local perturbations r_k^n and introduce the errors

$$e_k^n = \tilde{U}_k^n - U_k^n, \qquad k = 1,....,l. \tag{4.2}$$

To shorten the formulas, we introduce the auxiliary quantities e_0^n, D_1^n and P_{01}: $e_0^n = 0 \in \mathbb{R}^{d_1}$, D_1^n is the unit matrix I_1 of order d_1, and P_{01} is the zero matrix. Then, by subtracting (3.5) from (4.1), we can formulate the following relations for the errors (4.2):

$$Z_k^n e_k^n = D_k^n R_{lk} e_l^{n-1} + (I_k - D_k^n)P_{k-1k}e_{k-1}^n + r_k^n, \qquad n = 1,2,.... ; \qquad k = 1,....,l, \tag{4.3}$$

where

$$Z_k^n = I_k - \tau D_k^n M_k^n, \tag{4.4}$$

with M_k^n the integrated Jacobian matrix

$$M_k^n = \int_0^1 F'(t_n, \theta\tilde{U}_k^n + (1-\theta)U_k^n)d\theta \tag{4.5}$$

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

Assuming Z_k^n to be nonsingular, we can rewrite (4.3) as

$$e_k^n = X_k^n e_{k-1}^n + \Gamma_k^n e_l^{n-1} + \phi_k^n, \qquad n = 1,2,.... ; \qquad k = 1,....,l, \tag{4.6}$$

with

$$X_k^n = (Z_k^n)^{-1}(I_k - D_k^n)P_{k-1k},$$

$$\Gamma_k^n = (Z_k^n)^{-1} D_k^n R_{lk}, \tag{4.7}$$

$$\phi_k^n = (Z_k^n)^{-1} r_k^n.$$

Note that $X_1^n = 0$ and that the operators X_k^n, Γ_k^n are associated, respectively, to the interpolation and restriction. A little inspection reveals that we can rewrite (4.6) to the standard form

$$e_k^n = G_k^n e_l^{n-1} + \psi_k^n, \qquad n = 1,2,\ldots; \qquad k = 1,\ldots,l, \tag{4.8}$$

where the amplification operators G_k^n and the local perturbation terms ψ_k^n itself are defined by a recurrence relation:

$$G_1^n = \Gamma_1^n, \tag{4.9}$$

$$G_k^n = X_k^n\, G_{k-1}^n + \Gamma_k^n, \qquad k = 2,\ldots,l,$$

and

$$\psi_1^n = \phi_1^n, \tag{4.10}$$

$$\psi_k^n = X_k^n\, \psi_{k-1}^n + \phi_k^n, \qquad k = 2,\ldots,l.$$

The error recurrence (4.8) describes the error propagation for all regridding levels. The main interest lies in the operator G_l^n and the local perturbation ψ_l^n due to the fact that coarse grid values are always updated by fine grid values. In (4.8) this is reflected by the presence of e_l^{n-1} which is the error at the finest grid at the old time value.

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

LEMMA 4.1. Let ν be the logaritmic norm value defined in assumption (A4) of Section 2. Then, for $k = 2,\ldots,l$,

$$\|(Z_k^n)^{-1}\| \leqslant \begin{cases} \dfrac{1}{1 - \tau\nu} & , \ \forall \tau\nu < 1 \text{ if } \nu > 0, \\ 1 & , \ \forall\, \tau > 0 \text{ if } \nu \leqslant 0, \end{cases} \tag{4.11}$$

while for $k = 1$

$$\|(Z_1^n)^{-1}\| \leqslant \frac{1}{1 - \tau\nu}, \quad \forall \tau\nu < 1. \tag{4.12}$$

PROOF. The result for $k = 1$ is well known due to the fact that D_1^n is the unit matrix (see [5], p.46). The premultiplication of M_k^n for $k > 1$ with the diagonal matrix 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 then can immediately deduce that for $\nu > 0$ the bound $(1 - \tau\nu)^{-1}$ still holds, whereas for $\nu \leqslant 0$ the zero rows introduce the bound 1. □

Observe that the replacement of the bound $(1 - \tau\nu)^{-1}$ by the bound 1 for $\nu < 0$ implies that in this case we do no longer exploit the damping property of implicit Euler. For the analysis to follow this is not essential since we are here merely interested in proving stability and convergence results.

Before proving a stability result for the error recurrence (4.8), we make another assumption aimed at shortening the derivations:

(A5) The logarithmic norm bound ν introduced in (A4) in Section 2 is nonpositive. This means that in the remainder we restrict ourselves to dissipative problems. This restriction is not essential; results obtained for $\nu \leqslant 0$ can be extended to the case $\nu > 0$ by inserting the bound $(1 - \tau\nu)^{-1}$ for the bound 1 any time the stability inequality $\|(Z_k^n)^{-1}\| \leqslant 1$ is used.

4.2. Stability and linear interpolation

In this section we will prove a general stability result for the multilevel adaptive grid method (3.5) that is similar to the stability result (2.8) for the Euler method applied without adaption.

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

$$\|G_k^n\| \leqslant 1, \qquad k = 1,\ldots,l, \tag{4.13}$$

$$\|\psi_k^n\| \leqslant \sum_{j=1}^{k} \|r_j^n\|, \qquad k = 1,\ldots,l, \tag{4.14}$$

and, in particular,

$$\|e_l^n\| \leqslant \|e_l^{n-1}\| + \sum_{k=1}^{l} \|r_k^n\|. \tag{4.15}$$

PROOF. Inequality (4.15) is a trivial consequence of (4.13) and (4.14). Let us first prove (4.13). This is done by induction with respect to k. Suppose $\|G_{k-1}^n\| \leqslant 1$. From (4.9) it follows that

$$\|G_k^n\| = \|X_k^n G_{k-1}^n + \Gamma_k^n\| = \|(Z_{k-1}^n)^{-1} Q_k^n\| \leqslant \|Q_k^n\|, \tag{4.16}$$

where

$$Q_k^n = (I_k - D_k^n) P_{k-1k} G_{k-1}^n + D_k^n R_{lk}. \tag{4.17}$$

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

$$\sum_j |(Q_k^n)_{ij}| = \sum_j |(R_{lk})_{ij}| = 1, \tag{4.18}$$

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

$$\sum_j |(Q_k^n)_{ij}| = \sum_j |(P_{k-1k} G_{k-1}^n)_{ij}| \leqslant \|P_{k-1k} G_{k-1}^n\| \leqslant \|P_{k-1k}\| \|G_{k-1}^n\| \leqslant 1, \tag{4.19}$$

by virtue of the induction hypothesis and the norm

$$\|P_{k-1k}\| = 1 \tag{4.20}$$

of the linear interpolation operator P_{k-1k}. Combining (4.18) and (4.19) gives $\|Q_{k+1}^n\|$ and inequality (4.13) now follows from (4.16). The induction proof is finished if we can prove that $\|G_1^n\| \leqslant 1$. This follows immediately from the observation that $G_1^n = \Gamma_1^n = (Z_1^n)^{-1} R_{l1}$.

There remains to prove (4.14). We have $\|\phi_k^n\| \leqslant \|r_k^n\|$. It then follows from (4.10) that

$$\|\psi_k^n\| \leqslant \|X_k^n\| \|\psi_{k-1}^n\| + \|r_k^n\|, \tag{4.21}$$

so that we are finished if we can prove that $\|X_k^n\| \leqslant 1$. This is trivial due to (4.20) and $\|I_k - D_k^n\| = 1$. □

The inequality (4.15) 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 (3.5) retains stability of implicit Euler through the bound $\|G_l^n\| \leqslant 1$.

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[14] we have succesfully used 4th order Lagrangian interpolation). Unfortunately, in that case we necessarily have $\|P_{k-1k}\| > 1$ with the consequence that we are not able to prove the stability 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 practical use. For the sake of completeness we yet do present it here.

From (4.8)-(4.10) we obtain,

$$e_l^n = G_l^n e_l^{n-1} + \psi_l^n, \tag{4.22}$$

where

$$G_l^n e_l^{n-1} = \sum_{j=1}^{l-1} \left(\prod_{i=l}^{j+1} X_i^n\right) \Gamma_j^n e_l^{n-1} + \Gamma_l^n e_l^{n-1}, \tag{4.23}$$

$$\psi_l^n = \sum_{j=1}^{l-1} \left(\prod_{i=l}^{j+1} X_i^n\right) \phi_j^n + \phi_l^n. \tag{4.24}$$

Let us examine expression (4.23). The sum in its righthand side stems from interpolation to level l of values obtained on grids of level $k<l$, while the single term $\Gamma_l^n e_l^{n-1}$ is due to integration on level l. Since it is the intention of our method to interpolate exclusively in low error regions, it is natural to examine a condition which implies that this integration term alone determines $\|G_l^n e_l^{n-1}\|$.

For this purpose, we first introduce an empty product convention which reads $\prod_{i=k-1}^{k} X_i^n = I_{k-1}$, for $k = 2, \ldots, l$. This convention will be used throughout the remainder of this paper. Using this empty

product convention, $\|G_l^n e_l^{n-1}\|$ can then be written as

$$\|G_l^n e_l^{n-1}\| = \| (Z_l^n)^{-1} [(I_l - D_l^n)P_{l-1l} \sum_{j=1}^{l-1} (\prod_{i=l-1}^{j+1} X_i^n) \Gamma_j^n e_l^{n-1} + D_l^n R_{ll} e_l^{n-1}] \| \tag{4.25}$$

$$= \|(Z_l^n)^{-1} q_l^n\| \leqslant \|q_l^n\|,$$

where

$$(q_l^n)_i = (P_{l-1l} \sum_{j=1}^{l-1} (\prod_{i=l-1}^{j+1} X_i^n) \Gamma_j^n e_l^{n-1})_i \tag{4.26}$$

when $(D_l^n)_{ii} = 0$, and, since $R_{ll} = I_l$,

$$(q_l^n)_i = (e_l^{n-1})_i \tag{4.27}$$

when $(D_l^n)_{ii} = 1$. Hence,

$$\|q_l^n\| = \max\{\|(I_l - D_l^n)P_{l-1l} \sum_{j=1}^{l-1} (\prod_{i=l-1}^{j+1} X_i^n) \Gamma_j^n e_l^{n-1}\|, \|D_l^n e_l^{n-1}\|\}. \tag{4.28}$$

Consequently, if

$$\|(I_l - D_l^n)P_{l-1l} \sum_{j=1}^{l-1} (\prod_{i=l-1}^{j+1} X_i^n) \Gamma_j^n e_l^{n-1}\| \leqslant \|D_l^n e_l^{n-1}\|, \tag{4.29}$$

we have again stability since then

$$\|G_l^n e_l^{n-1}\| \leqslant \|q_l^n\| \leqslant \|D_l^n e_l^{n-1}\| = \|e_l^{n-1}\|. \tag{4.30}$$

Above we have used the term 'constrained stability' to indicate that condition (4.29) cannot hold for arbitrary error vectors e_l^{n-1}, without imposing $D_l^n = I_l$. Hence, it is not well possible to explicitly enforce this condition in a practical computation, let alone the meaningfullness of such an undertaking in connection with the required computational effort and the involved matrices D_k^n. We wish to have these matrices determined exclusively by spatial accuracy requirements, as we will outline in the following sections.

Summarizing, for higher order Lagrangian interpolants powerful stability results like those of

Theorem 4.2 are not available. 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, a condition like (4.29) is satisfied implicitly.

5. Error analysis

In this section we present a detailed examination of the total local error with its component parts. From this examination we deduce the so-called refinement condition which henceforth underlies the refinement strategy. This refinement condition enables us to control the contribution of spatial interpolation errors in favour of spatial discretization errors. Due to this condition, we are able to prove a convergent result as if we are working on a single fixed grid. Specifically, it will be proved that the usual convergence behaviour applies and that the accuracy obtained is comparable to the accuracy obtained with the finest grid if this grid would be used without any adaptation.

5.1. The local level error

Let the gridfunction $u_k(t)$ denote the pointwise restriction of the true solution $u(x,y,t)$ to the space grid ω_k. Consider the perturbed scheme (4.1). By replacing all $\tilde{U}$-values by their 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 local level error also by r_k^n:

$$r_k^n = u_k^n - D_l^n [R_{lk} u_l^{n-1} + \tau F_k(t_n, u_k^n)] - \tag{5.1}$$

$$(I_k - D_k^n)[P_{k-1k} u_{k-1}^n + b_k^n], \qquad n = 1,2,\ldots; \qquad k = 1,\ldots,l.$$

For $n \geqslant 0, u_k^n = u_k(t_n)$ and u_0^n, b_1^n are supposed to be the zero vector. (Any choice will do since $D_1^n = I_1$.)

The local level error 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. Let us first discuss the different components. They are defined in the standard way by, respectively,

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

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

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

In (5.4) the gridfunction $b_k(t)$ has the same meaning as b_k^n in (3.7). Note that $\gamma_1(t) = u_1(t)$ due to the chosen zero value for $P_{01}u_0^n + b_1^n$. This auxiliary error cannot interfere our analysis since $D_1^n = I_1$. Also note that these local errors are completely determined by the true PDE solution $u(x,y,t)$.

In view of Assumptions (A2) and (A3) made in Section 2, we have

$$\alpha_k(t) = \mathcal{O}(h_{x,k}^2) + \mathcal{O}(h_{y,k}^2), \qquad h_{x,k}, h_{y,k} \to 0, \tag{5.5}$$

where the order constants involved depend exclusively on higher order spatial derivatives of u and usually also on certain quantities of the PDE operator. Likewise, in view of Assumption (A2) it follows that $\beta_k(t) = \tau^2 C_k$ where the i^{th} component of the gridfunction C_k is the i^{th} component of $-\frac{1}{2}d^2u_k/dt^2$ evaluated at a time $t+(\kappa-1)\tau$, $0 \leqslant \kappa \leqslant 1$. If u is a C^3-function in t, then

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

where the order constants involved depend exclusively on the third temporal derivative. Let q denote the accuracy order of the (Lagrangian) interpolation. Then

$$\gamma_k(t) = \mathcal{O}(h_{x,k}^q) + \mathcal{O}(h_{y,k}^q), \qquad k = 2,\ldots,l, \tag{5.7}$$

and here the order constants involved again depend exclusively on higher spatial derivatives of u, assuming sufficient differentiability. If linear interpolation is used, then Assumption (A2) implies $q = 2$ and 2^{nd} order spatial derivatives determine the constants.

Now, using the relation $u_k^{n-1} = R_{lk} u_l^{n-1}$ for $k = 1,...,l$, an elementary substitution shows that the local level error r_k^n can be expressed as

$$r_k^n = D_k^n(\tau\alpha_k^n + \beta_k^n) + (I_k - D_k^n)\gamma_k^n, \qquad n = 1,2,.... ; \qquad k = 1,....,l. \tag{5.8}$$

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 ω_k, while $(I_k - D_k^n)\gamma_k^n$ represents the restriction of the interpolation error γ_k^n to the complement of this domain.

5.2. A crude global error bound

Denote the global discretization error by (cf. (4.2))

$$e_k^n = u_k^n - U_k^n, \qquad n = 0,1,.... ; \qquad k = 1,....,l, \tag{5.9}$$

and assume in the following that the initial errors e_k^0 are zero. For any choice of D_k^n the consistency results (5.5) - (5.8) imply

$$r_k^n = \mathcal{O}(\tau h_{x,k}^2) + \mathcal{O}(\tau h_{y,k}^2) + \mathcal{O}(\tau^2) + \mathcal{O}(h_{x,k}^q) + \mathcal{O}(h_{y,k}^q) \tag{5.10}$$

for any level k. If we suppose linear interpolation and Assumption (A5), like in Theorem 4.2, then a straightforward application of inequality (4.15) enables us to write down

$$\|e_l^n\| \leq \|e_l^{n-1}\| + \|S^n\| \leq \|S^1\| + ... + \|S^n\|, \qquad n = 1,2,.... , \tag{5.11}$$

where

$$\|S^n\| = \mathcal{O}(\tau h_{x,1}^2) + \mathcal{O}(\tau h_{y,1}^2) + \mathcal{O}(\tau^2) + \mathcal{O}(h_{x,1}^2) + \mathcal{O}(h_{y,1}^2). \tag{5.12}$$

Here the coarsest mesh width shows up due to simply adding in (4.15) all normed local level errors, including $\|r_1^n\|$. Following standard practice we thus immediately obtain at any fixed time point $t_n = n\tau$ the global error bound

$$\|e_l^n\| \leq C_1\tau + C_2(h_x^2 + h_y^2) + C_3(\frac{h_x^2}{\tau} + \frac{h_y^2}{\tau}), \tag{5.13}$$

where $h_x = h_{x,1}$, $h_y = h_{y,1}$ and C_1, C_2, C_3 are positive constants independent of stepsize and

mesh sizes. We note in passing that in the above τ is taken independent of n. Obviously, inequality (5.13) can also be derived for variable stepsizes if we put $t_n = \tau_1 + \cdots + \tau_n = n\tau$ and suppose $\tau_j = \mathcal{O}(\tau)$ as $\tau \to 0$, $n \to \infty$ with t_n fixed and $j = 1, \ldots, n$.

The first two terms in this bound are due to the temporal integration and spatial discretization. These terms will vanish if the mesh sizes and stepsize tend to zero independently of each other, thus reflecting the unconditional convergence of the implicit Euler method when applied without adaptation. On the other hand, if no relation is imposed between τ and h_x, h_y, then the third term can grow unboundedly as τ, h_x, $h_y \to 0$. This term is due to the interpolation. Hence, eventhough we have stability and consistency, this result shows that unconditional convergence cannot be hoped for.

Fortunately, this conclusion is not as bad as at it looks like. 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 necessarily must lead to a crude error bound like (5.13). In actual 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 contribution is virtually absent. This poses the task of setting up a more precise error analysis and subsequently 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 (4.8), the global error e_k^n satisfies the recurrence relation

$$e_k^n = G_k^n e_l^{n-1} + \psi_k^n, \qquad n = 1,2,\ldots; \qquad k = 1,\ldots,l, \tag{5.14}$$

where ψ_k^n is the local error defined by the recursion (cf. (4.10), (4.7))

$$\psi_1^n = (Z_1^n)^{-1} r_1^n, \tag{5.15a}$$

$$\psi_k^n = X_k^n \psi_{k-1}^n + (Z_k^n)^{-1} r_k^n, \qquad k = 2,\ldots,l. \tag{5.15b}$$

Hereby, the operators G_k^n, X_k^n and Z_k^n are supposed to be properly redefined (replace all $\tilde{U}$-values by their associated u_k-values). It is of importance to realize that the local error ψ_k^n is essentially different

from the local level error r_k^n. While the latter is associated to the single k^{th} level, ψ_k^n is associated to all levels up to this k^{th} level which follows immediately from (5.15). This recursion governs the propagation of each local level error when introducing higher-and-higher levels.

Elaborating this recursion gives, for $k = 1, \ldots, l$,

$$\psi_k^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} X_i^n\right) (Z_j^n)^{-1} r_j^n. \tag{5.16}$$

It is convenient to split ψ_k^n into its temporal and spatial part denoted by, respectively, $\psi_{k,t}^n$ and $\psi_{k,s}^n$. Hence, we write

$$\psi_k^n = \psi_{k,t}^n + \psi_{k,s}^n, \qquad k = 1, \ldots, l, \tag{5.17}$$

and it follows from (5.8) that the temporal local error is given by

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

and the spatial local error by

$$\psi_{k,s}^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} X_i^n\right) (Z_j^n)^{-1} [\tau D_j^n \alpha_j^n + (I_j - D_j^n)\gamma_j^n] \tag{5.19}$$

Let us first examine the temporal local error. Since the same temporal stepsize is used at all levels and β_k^n does not depend on mesh sizes, we have

$$\beta_k^n = R_{lk} \beta_l^n, \qquad k = 1, \ldots, l. \tag{5.20}$$

Substitution into (5.18) yields

$$\psi_{k,t}^n = \sum_{j=1}^{k} \left(\prod_{i=k}^{j+1} X_i^n\right) (Z_j^n)^{-1} D_j^n R_{lj} \beta_l^n \tag{5.21}$$

and inspection of the right-hand side operator reveals that this operator is just the amplification operator G_k^n featuring in (5.14); see the recursion (4.9). In conclusion, the temporal local error $\psi_{k,t}^n$ satisfies

$$\psi_{k,t}^n = G_k^n \beta_l^n, \qquad k = 1,....,l. \tag{5.22}$$

We next examine the spatial local error. Using the definition of X_k^n given in (4.7), we rewrite (5.19) as

$$\psi_{k,s}^n = (Z_k^n)^{-1}(I_k - D_k^n)P_{k-1k}\sum_{j=1}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)(Z_j^n)^{-1}[\tau D_j^n \alpha_j^n + (I_j - D_j^n)\gamma_j^n] + \tag{5.23}$$

$$(Z_k^n)^{-1}[\tau D_k^n \alpha_k^n + (I_k - D_k^n)\gamma_k^n]$$

$$= (Z_k^n)^{-1}[\tau D_k^n \alpha_k^n + (I_k - D_k^n)\rho_k^n], \qquad k = 1,....,l,$$

where

$$\rho_1^n = 0, \tag{5.24}$$

$$\rho_k^n = \gamma_k^n + P_{k-1k}\sum_{j=1}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)(Z_j^n)^{-1}[\tau D_j^n \alpha_j^n + (I_j - D_j^n)\gamma_j^n], \qquad k = 2,....,l.$$

In (5.23) the spatial local discretization error $D_k^n \alpha_k^n$ committed on the integration domain of grid ω_k is separated from the spatial local error part $(I_k - D_k^n)\rho_k^n$ defined outside this domain. Hence, ρ_k^n collects all spatial error contributions defined on the lower level grids ω_j $(1 \leqslant j \leqslant k-1)$ including discretization error α_j^n and interpolation error γ_j^n, together with the interpolation error γ_k^n defined on the grid ω_k. This separation enables us to formulate the so-called refinement condition that 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 due to the fact that we never return to grid points lying outside a current integration domain (integration domains are nested).

The refinement condition, which is to be interpreted as a constraint on the matrices D_k^n, thus is taken to be

$$\|(Z_k^n)^{-1}\tau D_k^n \alpha_k^n\| \geqslant \frac{1}{c}\|(Z_k^n)^{-1}(I_k - D_k^n)\rho_k^n\|, \qquad n = 1,2,.... ; \qquad k = 2,....,l, \tag{5.25}$$

where $c > 0$ denotes a threshold factor to be specified later. If (5.25) is true, then all spatial local errors $\psi_{k,s}^n$ satisfy

$$\|\psi_{k,s}^n\| \leq (1+c)\|(Z_k^n)^{-1}\tau D_k^n \alpha_k^n\| \tag{5.26}$$

and combining (5.14) with (5.17), (5.22) enables us to present the global error inequality

$$\|e_k^n\| \leq \|G_k^n\|\|e_l^{n-1}\| + \|G_k^n \beta_l^n\| + \tag{5.27}$$

$$(1+c)\|(Z_k^n)^{-1}\tau D_k^n \alpha_k^n\|, \qquad n = 1,2,\ldots; \qquad k = 1,\ldots,l.$$

The importance of the refinement condition (5.25) is reflected by the fact that in (5.27) 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. Condition (5.25) will be further elaborated in Section 6. Note that it suffices to consider (5.25) only for $k = l$, since it suffices to consider (5.26) and (5.27) for $k = l$.

We should comment on the threshold factor c. Inequality (5.27) suggests to choose c close to zero. However, if we take c very small, the effect will be that the greater part of the diagonal entries of D_k^n are put to unity in order to satisfy (5.25). This of course has the effect that the integration domains will become quite large. On the other hand, if we take c very large, then (5.25) is easier to satisfy but more levels will be needed to reach a certain desired level of accuracy. The actual choice for c may differ per problem. We will discuss this point later again (in Section 7).

5.4. Convergence and linear interpolation

If we suppose linear interpolation and Assumption (A5) like in Section 5.2, then (5.27) can be rewritten as

$$\|e_k^n\| \leq \|e_l^{n-1}\| + \|\beta_k^n\| + (1+c)\tau\|\alpha_k^n\|, \qquad n = 1,2,\ldots; \qquad k = 1,\ldots,l, \tag{5.28}$$

so that, following the same derivation as carried out for (5.13), for the highest level l the global error

bound

$$\|e_l^n\| \leqslant C_1\tau + C_2(1+c)(h_{x,l}^2 + h_{y,l}^2) \tag{5.29}$$

results where C_1 and C_2 are positive constants independent of stepsize and mesh sizes. This error bound is unconditional in the sense that it assumes no relation between stepsize and mesh sizes and, according to our above mentioned goal, the smallest mesh widths $h_{x,l}$ and $h_{y,l}$ show up. We have recovered an error bound similar to the standard error bound for implicit Euler when applied on grid ω_l without adaptation.

5.5. Convergence and higher order interpolation

As pointed out in Section 4.3, for the case of higher order Lagrangian interpolants powerful stability results like those of Theorem 4.2 are not available here. We must instead rely on the impractical condition (4.29). On the other hand, if this condition applies for any occurring gridfunctions e_l^{n-1} and β_l^{n-1} so that $\|G_l^n e_l^{n-1}\| \leqslant \|e_l^{n-1}\|$ and $\|G_l^n \beta_l^{n-1}\| \leqslant \|\beta_l^{n-1}\|$, then the improved error bound (5.29) is valid because in the derivation of the refinement condition (5.25) and the global error inequality (5.28) no a priori choice was made for the interpolants. Assuming that higher order interpolation in low error regions does not severely damages stability, as is strongly supported by our practical experience, it thus makes sense to impose the refinement condition (5.25) also in the case of higher order interpolation.

6. The refinement condition

6.1. Determining the integration domains

The refinement condition (5.25) needs first to be elaborated into a workable form before it can be implemented for determining the integration domains. To begin with we rewrite the error ρ_k^n introduced in (5.24) as

$$\rho_k^n = \gamma_k^n + P_{k-1k} \sum_{j=1}^{k-1} \Big(\prod_{i=k-1}^{j+1} X_i^n \Big) (Z_j^n)^{-1} \tau D_j^n \alpha_j^n + \tag{6.1}$$

$$P_{k-1k}\sum_{j=2}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)\,(Z_j^n)^{-1}(I_j - D_j^n)\gamma_j^n, \qquad 2 \leqslant k \leqslant l.$$

Note that the summation index j of the second sum may start with $j=2$ due to the fact that $D_1^n = I_1$. Next, we rewrite the first sum as

$$P_{k-1k}\sum_{j=1}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)\,(Z_j^n)^{-1}\tau D_j^n\alpha_j^n = \tag{6.2}$$

$$P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n\alpha_{k-1}^n +$$

$$P_{k-1k}\sum_{j=2}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)\,X_j^n(Z_{j-1}^n)^{-1}\tau D_{j-1}^n\alpha_{j-1}^n =$$

$$P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n\alpha_{k-1}^n +$$

$$P_{k-1k}\sum_{j=2}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)\,(Z_j^n)^{-1}(I_j - D_j^n)P_{j-1j}(Z_{j-1}^n)^{-1}\tau D_{j-1}^n\alpha_{j-1}^n,$$

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

$$\rho_1^n = 0, \tag{6.3}$$

$$\rho_k^n = \lambda_k^n + P_{k-1k}\sum_{j=2}^{k-1}\Big(\prod_{i=k-1}^{j+1} X_i^n\Big)\,(Z_j^n)^{-1}(I_j - D_j^n)\lambda_j^n, \qquad k=2,\ldots,l,$$

where

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

The error function λ_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 below now rests upon monitoring the error function $(Z_k^n)^{-1}(I_k - D_k^n)\rho_k^n$ occurring in (5.25) through monitoring all error functions $(I_j - D_j^n)\lambda_j^n$, $j \leqslant k$, 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 $C_3 \geqslant 1$, $C_4 \geqslant 1$ are stability constants such that

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

then

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

Hence, if the matrices D_j^n are selected such that

$$C_3(1 + \|P_{k-1k}\|(k-2)C_3C_4)\max_{2\leq j\leq k}\|(I_j - D_j^n)\lambda_j^n\| \leq \tag{6.7}$$

$$c\|(Z_k^n)^{-1}\tau D_k^n\alpha_k^n\|, \qquad k=2,....,l,$$

then the refinement condition (5.25) 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 $\|P_{k-1k}\|=1$ and also C_4 can be put equal to one so that (6.7) simplifies to

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

If (A5) does not hold or higher order interpolation is used, then the constants C_3 and C_4 may be larger than one, but not with a considerable extent. C_3 shall in general be of moderate size in view of the excellent stability behaviour of implicit Euler. Our practical experience with 4^{th} order Lagrangian interpolation is that this higher order interpolation is unlikely to yield instability problems, thus indicating that also $\|X_k^n\|$, and hence C_4, are of moderate size. These considerations make us decide to proceed with (6.8) and to use it also in situations where (A5) may be violated and/or higher order interpolation is used.

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

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

In order to satisfy this condition, estimates of

$$\lambda_k^n = \gamma_k^n + P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n \tag{6.10}$$

have to be computed. To create an extra safety margin, we replace (6.9) by the slightly more conservative condition

$$\max_{2\leqslant k\leqslant l}\|(I_k - D_k^n)\zeta_k^n\| \leqslant \frac{c}{l-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|, \tag{6.11}$$

where, componentwise, ζ_k^n is defined as

$$(\zeta_k^n)_i = |(\gamma_k^n)_i| + |(P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n)_i|. \tag{6.12}$$

Finally, we replace (6.11) by

$$\|(I_k - D_k^n)\zeta_k^n\| \leqslant \frac{c}{l-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|, \qquad k = 2,\ldots,l, \tag{6.13}$$

which is the refinement condition that will be implemented.

This condition will determine the integration domain of ω_k. Let Ω_k^n denote this integration domain and recall that when a node belongs to Ω_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 the (estimated) value of $c(l-1)^{-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|$ are known and that a solution at Ω_{k-1}^n, $k \leqslant l$, has just been computed. Prior to the integration step on level k, our task is then to determine Ω_k^n. Mathematically, we thus must define D_k^n such that (6.13) is satisfied, preferably in such a way that the area of Ω_k^n is as small as possible. The actual selection of Ω_k^n is carried out by a flagging procedure which scans level-k grid points. A point is then flagged if, using appropriate estimates,

$$(\zeta_k^n)_i > \frac{c}{l-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|. \tag{6.14}$$

Hence, for such a point the corresponding diagonal entry $(D_k^n)_{ii} = 1$ and for non-flagged points we define $(D_k^n)_{ii} = 0$. This way the refinement condition (6.13) is satisfied.

In conclusion, we only interpolate if

$$|(\gamma_k^n)_i| + |(P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n)_i| \leqslant \frac{c}{l-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|. \tag{6.15}$$

Hence, the solution at a node of grid ω_k is interpolated only if the sum of the moduli of the interpolation error and the prolongated spatial discretization error multiplied with τ, is smaller than the maximum of the spatial discretization error at the finest grid multiplied with $\tau c(l-1)^{-1}$. In all other cases interpolation is avoided and an integration is carried out. No doubt this imposes a severe restriction on the size of the interpolation errors. On the other hand, this restriction is also natural because when going to a higher level within the current base time step, we never return to a grid point where the solution has been interpolated and this means that the interpolation error committed will be carried along to the next base time step. This no return is a direct consequence of the nesting property of the integration domains which we discuss next.

6.2. Restricted interpolation and the nesting property

At this point we introduce the nesting property of the integration domains. Recall that this property, being hidden in the actual definition of the matrices D_k^n, has played no role in the foregoing analysis. It is stipulated that in our actual application the nesting is enforced by means of the flagging procedure, viz. this procedure scans only level-k points lying within the previous integration domain Ω_{k-1}^n, instead of the whole of ω_k. A direct consequence is that, different from (3.7), the interpolation is carried out only for level-k points within Ω_{k-1}^n. Here we will justify this deviation caused by what we call restricted interpolation (see also the remark at the end of Section 3.2). We will make plausible that this restricted interpolation is in fact allowed by the inequality (6.15) where still interpolation over the whole of ω_k is assumed.

The error function ζ_k^n contains the vector of moduli of the components of the prolongation of $(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n$. This spatial error is defined at level $k-1$ and, by definition of D_{k-1}^n, has zero components outside Ω_{k-1}^n. Hence, these error components are all taken into account in the flagging procedure for determining Ω_k^n which scans only level-k points within Ω_{k-1}^n. For the interpolation error γ_k^n, which lives on the whole of ω_k (grid expansion), the situation is different. Because we only scan level-k points within Ω_{k-1}^n, here the deviation arises. However, the restricted interpolation is allowed

if for all level-k points outside Ω_{k-1}^n the interpolation error satisfies

$$|(\gamma_k^n)_i| \leqslant \frac{c}{l-1}\|(Z_l^n)^{-1}\tau D_l^n \alpha_l^n\|, \tag{6.16}$$

because then points outside Ω_{k-1}^n will not be flagged if the interpolation step (3.7) would be carried out on the whole of ω_k. Recall that $|(P_{k-1k}(Z_{k-1}^n)^{-1}\tau D_{k-1}^n \alpha_{k-1}^n)_i| = 0$ outside Ω_{k-1}^n. In other words, if (6.16) holds outside Ω_{k-1}^n, then the integration domains found with the restricted interpolation over Ω_{k-1}^n are equal to the domains found if the interpolation would be carried out on the whole of ω_k, which is in accordance with the method description (3.5).

Although we do not give a strict mathematical proof, the following argument shows that inequality (6.16) is very plausible with the restricted interpolation procedure. Note that without restricted interpolation (6.16) is a trivial consequence of (6.15) and that we must show that (6.16) is plausible with the restricted interpolation since this is enforced. First we recall that Ω_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 Ω_2^n inequality (6.16) is trivially satisfied. Next consider the case $k=3$. Now the interpolation is restricted to level-3 points within Ω_2^n. Since for all level-2 points outside Ω_2^n inequality (6.16) is satisfied, it is very plausible to suppose that this is also true for all level-3 points outside Ω_2^n, in view of the consistency of the interpolation (level-3 interpolation errors are smaller than level-2 errors). Further, by construction of Ω_3^n, (6.15) is satisfied for all level-3 points within Ω_2^n and outside Ω_3^n, and so is (6.16). In conclusion, it is plausible to suppose that (6.16) is satisfied for all level-3 points outside Ω_3^n when using the restricted interpolation for $k=3$. For $k=4$ and so on this argument can be repeated.

To sum up, we have made plausible that when carrying out the restricted interpolation during the solution step from t_{n-1} to t_n, the same matrices D_k^n are found as if interpolation would be carried out over the whole of the expanded grids. This means that in theory the multilevel method (3.5) is applied with the matrices D_k^n determined by the local refinement strategy of paragraph 6.1. Interpolation on the whole of the expanded grids is of course possible but also very costly and, as shown above, redundant.

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 and that 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. Implementation aspects

Before presenting results of actual numerical tests, we first discuss a number of implementation aspects and remarks of practical interest. These mainly serve to show how the refinement condition is implemented and are furthermore helpful in judging the experiments. We do not discuss the very technical issues of the data structure and memory use (see [14]).

(I) The fixed-level mode.

In the foregoing error analysis the number of grid levels l is supposed to be a priori chosen and fixed for all times. For a chosen base grid and an appropriate value of l, which need both be selected on the basis of spatial accuracy requirements, at each base time step the flagging procedure then will generate $l-1$ integration domains according to the refinement condition (6.13). This mode of operation is called the fixed-level mode. The convergence experiments reported here are performed in this fixed-level mode.

(II) The flagging procedure.

On top of the flagging procedure implementing (6.14) some safety measures have been built. Any node for which (6.14) 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 coarse or four fine meshwidths is used around any untolerable node. Hence the minimal number of nodes in a column or row of any subgrid is nine, assuming that the minimal number of internal points in a row or column of the base grid is three. Near boundaries, physical and internal

ones, the buffering slightly differs. Although in theory this buffering could be omitted, in practice it is wise to create a buffer zone around untolerable nodes because the estimation of higher spatial derivatives featuring in α_k^n and γ_k^n is prone to inaccuracies due to numerical differencing.

After the flagging procedure, a cluster algorithm groups all untolerable nodes together to form the newly defined integration domain. It is noted that possible subdomains of the new domain cannot overlap and that we do not connect subdomains which are lying close together since this leads to extra bookkeeping. For the same reason the local refinement does not distinguish between co-ordinate directions. This necessarily leads to some waste of points if a high gradient region aligns with a co-ordinate direction (cf. [14], Figure 8.2).

The threshold factor c occurring in (6.14) must be specified. In view of the global error result (5.28), c should preferably be taken sufficiently small so as to achieve that the spatial accuracy obtained is indeed nearly equal to the spatial accuracy obtained without any adaptation. On the other hand, the smaller c will be chosen, the more points will be flagged and hence the safer the local grid refinement will be (the extreme choice $c=0$ would imply global refinement). This means that c is available as a tuning parameter for the local grid refinement. In the experiments reported here we simply use the default value $c=1$.

(III) The estimators for α_k^n, γ_k^n and Z_{k-1}^n.

The use of (6.14) requires estimates of spatial interpolation and discretization errors. For $2 \leqslant k \leqslant l$ we must estimate interpolation error γ_k^n and the prolongated 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 accurately.

Consider the local spatial discretization error

$$\alpha_k^n = \frac{d}{dt} u_k(t)|_{t_n} - F_k(t_n, u_k(t_n)) \tag{7.1}$$

and let p be the order of consistency (in this paper $p=2$). The estimation technique we apply is very common and based on the use of a second spatial discretization of a higher order $\tilde{p}$. Let $\tilde{F}$ denote the associated operator and

$$\tilde{\alpha}_k^n = \frac{d}{dt}u_k(t)|_{t_n} - \tilde{F}_k(t_n, u_k(t_n)) \tag{7.2}$$

the associated local error. Elimination of $\frac{d}{dt}u_k(t)|_{t_n}$ from (7.1) yields

$$\alpha_k^n = \tilde{F}_k(t_n, u_k(t_n)) - F_k(t_n, u_k(t_n)) + \tilde{\alpha}_k^n. \tag{7.3}$$

If $\tilde{p} \geqslant p+1$, then we may neglect $\tilde{\alpha}_k^n$ for estimation purposes so as to obtain

$$\alpha_k^n \approx \tilde{F}_k(t_n, u_k(t_n)) - F_k(t_n, u_k(t_n)) \tag{7.4}$$

as an asymptotically correct estimator for α_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 4^{th} order accuracy (standard symmetrical differences), while at nodes adjacent to physical or internal boundaries 3^{rd} order accuracy is realized (standard one-sided differences).

The benefit of using uniform grids is also reflected in the estimation for the interpolation error γ_k^n (cf. (5.4)). Our algorithm provides two options, viz. standard linear interpolation of 2^{nd} order and standard Lagrangian interpolation of 4^{th} order. In the case of linear interpolation, we thus need to compute spatial derivatives u_{xx}, etc., while in the 4^{th} order case spatial derivatives like u_{xxxx} show up. For both options the error estimation procedure can be straightforwardly implemented (to save space we omit details here).

We wish to emphasize that in spite of its simplicity standard linear interpolation may become disadvantageous due to the low order of accuracy. Inspection of the various terms in (6.14) suggests to compare the following order relations:

$$\tau(P_{k-1k}(Z_{k-1}^n)^{-1}D_{k-1}^n\alpha_{k-1}^n)_i = \tau(\mathcal{O}(h_{x,k}^2) + \mathcal{O}(h_{y,k}^2)), \tag{7.5a}$$

$$\|(Z_l^n)^{-1}\tau D_l^n\alpha_l^n\| = \tau(\mathcal{O}(h_{x,l}^2) + \mathcal{O}(h_{y,l}^2)), \tag{7.5b}$$

$$(\gamma_k^n)_i = \mathcal{O}(h_{x,k}^2) + \mathcal{O}(h_{y,k}^2), \qquad 2^{\text{nd}} \text{ order linear,} \tag{7.5c}$$

$$(\gamma_k^n)_i = \mathcal{O}(h_{x,k}^4) + \mathcal{O}(h_{y,k}^4), \qquad 4^{\text{th}} \text{ order Lagrangian.} \tag{7.5d}$$

Essential for this comparison is that in the discretization terms (7.5a), (7.5b) the stepsize τ is contained. Consequently, it is the interpolation error that readily will govern the local refinement if τ is very small, and particularly so when the interpolation is linear. The comparison is clearly in favour of the 4^{th} order interpolation. For example, if τ, $h_{x,k}$, $h_{y,k}$ are of comparable size, then the comparison based on (7.5a) and (7.5d) predicts that the interpolation error will hardly play a role in the local refinement. Instead, now it is the local space discretization error that will govern the local refinement. It should be emphasized, though, that in this comparison the size of the unkown order constants has not been taken into account. This means that in actual application the comparison between the various terms may work out differently.

Our numerical example problem is linear. Hence the operator Z_{k-1}^n featuring in $(Z_{k-1}^n)^{-1} D_{k-1}^n \alpha_{k-1}^n$ is simply the implicit Euler matrix that is also used to compute the solution vector $D_{k-1}^n U_{k-1}^n$. The computation of the error term $(Z_{k-1}^n)^{-1} D_{k-1}^n \alpha_{k-1}^n$ then can be carried out in the same way as that of the implicit Euler approximation (solution of a linear system using the available factored form of the matrix).

There remains to describe how we estimate the righthand side term $\|(Z_l^n)^{-1} D_l^n \alpha_l^n\|$ of (6.14) for $2 \leqslant k \leqslant l-1$. Here we exploit the asymptotics of the spatial discretization. Since the meshwidth of level k is half that of level $k+1$, we make invoke the relation

$$\|(Z_l^n)^{-1} D_l^n \alpha_l^n\| \approx 2^{-p(l-k)} \|(Z_k^n)^{-1} D_k^n \alpha_k^n\|, \qquad l \geqslant k+1, \tag{7.6}$$

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

(IV) The use of implicit Euler.

Due to its favourable stability properties, implicit Euler is a very robust integration method. A

drawback is that this method is able to provide only limited accuracy in time due to its consistency order one. Another important point of practical interest is the implicitness. Implicitness leads to overhead costs for solving systems of linear or nonlinear algebraic equations. In the more common method of lines approach, based on sophisticated stiff ODE solvers, one invariably attempts to reduce this overhead by using old Jacobian matrices over considerable numbers of integration steps with the same stepsize. This way the costs of the numerical algebra can be reduced easily and no doubt the practical experience with these solvers is, in this respect, excellent. However, for a multilevel local refinement algorithm implicitness has a larger impact.

There are two points worth mentioning. The first point is that at any base time step refinement takes place at different levels, resulting in a different Jacobian per level the order of which usually differs per base time step. This virtually impedes the profitable use of old Jacobians, unless it is decided not to adapt grids at any base time step, but instead per prescribed number of these steps. We consider this as part of an overall strategy that can easily be placed on top of the existing one. In our experiments we adapt grids at any base time step since our main aim with these 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 any base time step, just for efficiency reasons. The second point is that the Jacobians do not posses a regular band structure, since the integration domains Ω_k^n normally have an irregular shape. Unlike the first, this point is intrinsic to the local refinement method. In our current research code the Harwell sparse matrix solver MA28 is used. This solver is well suited to cope with the structures we meet, but is rather time consuming for the present application. It is likely that standard iterative methods can be applied at lower costs. Once more, our experiments focus on illustrating the foregoing error analysis and important efficiency aspects have been set aside.

7.2. The example problem

This test example is hypothetical and due to Adjerid and Flaherty[1]. We have also used it in a convergence experiment in [14]. The equation is the linear parabolic model equation

$$u_t = u_{xx} + u_{yy} + f(x,y,t), \qquad 0<x,y<1, \qquad t>0, \tag{7.7}$$

and the initial function at $t=0$, the Dirichlet boundary conditions for $t>0$ and the source term f are selected so that the exact solution is

$$u(x,y,t) = exp[-80((x - r(t))^2 + (y - s(t))^2)], \tag{7.8}$$

where

$$r(t) = \frac{1}{4}[2 + sin(\pi t)], \qquad s(t) = \frac{1}{4}[2 + cos(\pi t)]. \tag{7.9}$$

This solution is a cone that is initially centered at (½,¾) and that symmetrically rotates around the center (½,½) of the domain in a clockwise direction. The speed of rotation is constant and one rotation has a period of 2. This problem is not a very difficult one in the sense that the spatial gradients of the solution are not very large, that is, the cone is not very steep. However, the problem is suitable to subdue the local grid refinement method to a convergence test. Observe that the semi-discrete version of this problem satisfies the dissipativity assumption (A5).

7.3. Convergence experiments

We have carried out two identical convergence experiments. In the first experiment linear interpolation has been used and in the second 4^{th} order Lagrangian. In both experiments the solution is computed four times over the time interval $0 \leqslant t \leqslant 2$, always using a uniform 10 x 10 base grid and a constant stepsize τ in time. In the first computation $l=1$, in the second $l=2$ and so on. Since per computation the smallest meshwidth is halved, τ is simultaneously decreased with the factor 2^2 in view of the 1^{st} order of implicit Euler. Hence, in line with our convergence analysis, per computation the maximal global error should also decrease with the factor 2^2.

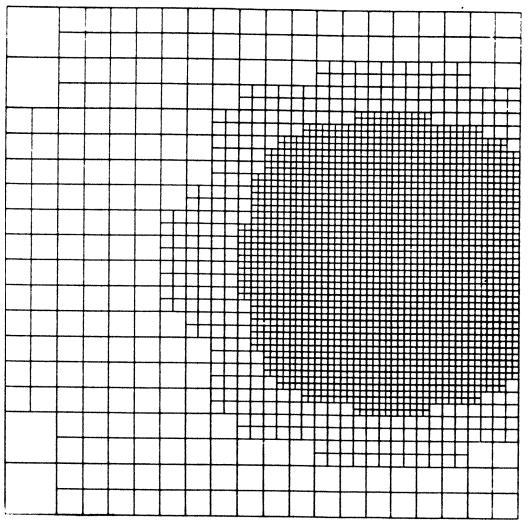
τ	no. of levels	interpolation	t 0.50	1.00	1.50	2.00
2.00000	1					0.16447
0.50000	2	linear	0.03232	0.03214	0.03213	0.03213
		4th order	0.03169	0.03149	0.03148	0.03148
0.12500	3	linear	0.01127	0.01127	0.01127	0.01127
		4th order	0.01133	0.01133	0.01133	0.01133
0.03125	4	linear	0.00259	0.00259	0.00259	0.00259
		4th order	0.00276	0.00276	0.00276	0.00276

TABLE 7.1. Maxima of global errors restricted to the base grid.

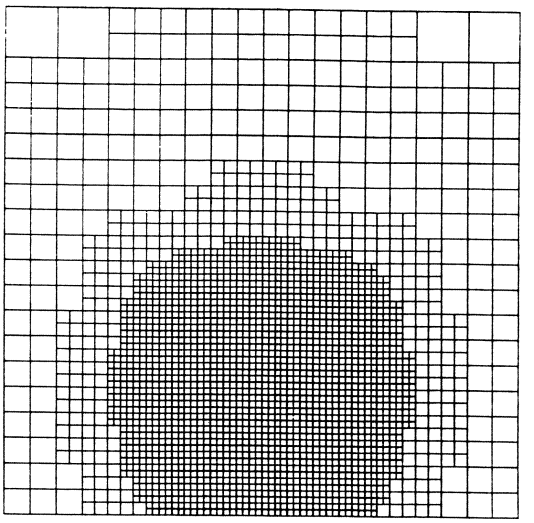
τ	no. of levels	interpolation	single grid	t 0.50	1.00	1.50	2.00
2.00000	1		10x10				0.16447
0.50000	2	linear		0.03876	0.03890	0.03891	0.03891
		4th order		0.03929	0.03945	0.03946	0.03946
			20x20	0.03865	0.03881	0.03882	0.03882
0.12500	3	linear		0.01369	0.01369	0.01369	0.01369
		4th order		0.01376	0.01376	0.01376	0.01376
			40x40	0.01389	0.01389	0.01389	0.01389
0.03125	4	linear		0.00340	0.00340	0.00340	0.00340
		4th order		0.00359	0.00359	0.00359	0.00359
			80x80	0.00347	0.00347	0.00347	0.00347

TABLE 7.2. Maxima of global errors restricted to the finest domain.
Comparison with errors on a standard uniform grid.

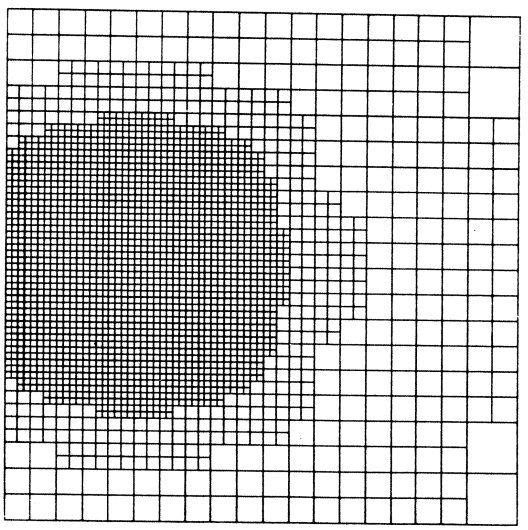
For a sequence of time points Table 7.1 shows for the two experiments the maxima of the global errors restricted to the base grid. This table clearly reveals the expected order behaviour. The errors of the $l=4$ runs are about a factor 4 smaller than the corresponding errors of the $l=3$ runs. Note that 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. At this point we emphasize that in spite of the relatively large values for τ, the spatial error dominates the global errors shown in



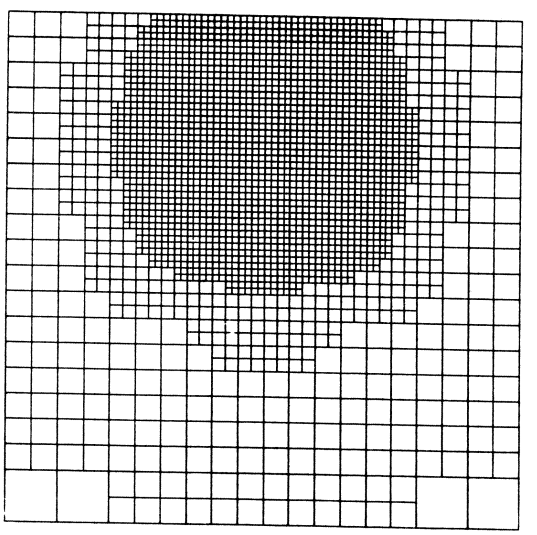
$t = 0.5$



$t = 1.0$



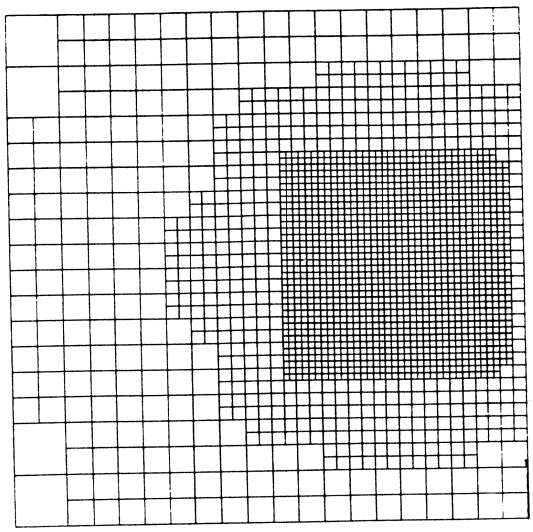
$t = 1.5$



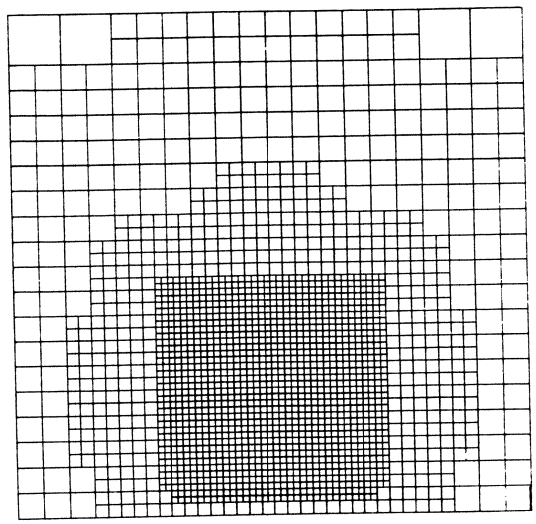
$t = 2.0$

FIGURE 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.

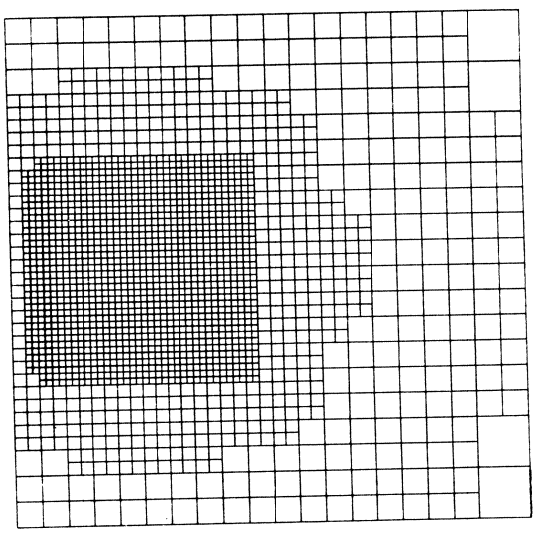
this table. For example, using $\tau = 0.125$ instead of $\tau = 0.5$ in the $l = 2$ run, the same global errors are found (they deviate in the 3rd or 4th decimal digit). In other words, conclusions on the spatial error



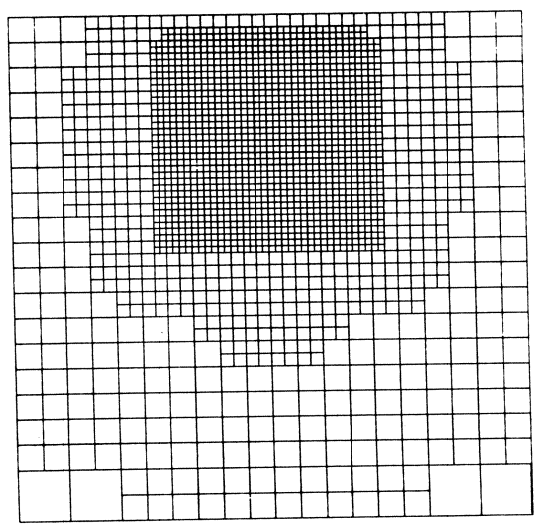
$t = 0.5$



$t = 1.0$



$t = 1.5$



$t = 2.0$

FIGURE 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.

behaviour induced by the local refinement algorithm can be drawn from this table.

Table 7.2 shows the maxima of the global errors restricted to the finest integration domain in use. This table also contains the maxima of the correponding errors for this grid used without adaptation

(uniform). Note the striking correspondence between the errors. These results convincingly show, at least 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 if this grid is used without any adaptation. The threshold factor $c = 1$ apparently has no influence on the error. We owe this to the fact that the refinement condition has been derived from errors bounds.

The use of the two different interpolants is expressed in the slightly different integration domains shown in Figures 7.1 and 7.2. As expected, at the higher levels linear interpolation gives rise to somewhat larger domains. This means that the use of linear interpolation is more expensive. Note that with linear interpolation the shape of the cone is more accurately reflected in the domains than with 4^{th} order interpolation. As a rule, 4^{th} 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

Noteworthy is that in spite of the overhead due to the regridding and error estimation, in the $l = 4$ run with linear interpolation our FORTRAN research code spent approximately 85% of the total elapsed CPU time in the integration routine (1620 sec. measured on a SUN/SPARC station 1). Only 2% was spent by the regridding routines and 6% by the estimation of the spatial discretization errors together with the involved back solves. The high percentage spent by the integration routine is due to using the sparse matrix solver MA28 whose application is no doubt expensive here. Needless to say that for real applications the efficiency of the time stepping scheme in connection with the spatial adaptation and static regridding should be a point of serious concern. As mentioned before, in the successful convergence experiments discussed above this issue has been set aside.

In our future research in this field we plan to pay serious attention to time stepping efficiency. Using the refinement strategy of this paper as a starting point, the plan is to examine the application

of methods possessing a higher order in time. Natural candidates belong to the class of Runge-Kutta and linear multistep 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 [10]) 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 in fixed-level mode, but to have also the possibility to vary the number of levels. This might be useful for the computation of solutions which, for example, steepen up in time like the combustion problem in [14]. For such problems, the application of a variable number of levels should be combined with the use of variable temporal stepsizes. Preferably, the complete adaptation then should 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.

ACKNOWLEDGEMENTS

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

REFERENCES

1. S. ADJERID and J.E. FLAHERTY (1988). A local Refinement Finite Element Method for Two Dimensional Parabolic Systems, *SIAM J. Sci. Stat Comput.*, 9, 792-881.
2. D.C. ARNEY and J.E. FLAHERTY (1989). An Adaptive Local Mesh Refinement Method for Time-Dependent Partial Differential Equations, *Appl. Numer. Math.*, 5, 257-274.
3. M.J. BERGER (1986). Data Structures for Adaptive Grid Generation, *SIAM J. Sci. Stat Comput.*, 7, 904-916.
4. M.J. BERGER and J. OLIGER (1984). Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *J. Comput. Phys.*, 53, 484-512.
5. K. DEKKER and J.G. VERWER (1984). *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, Amsterdam-New York-Oxford.
6. R.E. EWING (1989). Adaptive Grid Refinement for Transient Flow Problems, in *Adaptive Methods*

for Partial Differential Equations, ed. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD, J.D. VASILAKIS, SIAM Publications, Philadelphia.

7. W.D. GROPP (1980). A Test of Moving Mesh Refinement for 2D-Scalar Hyperbolic Problems, *SIAM J. Sci. Comput.*, 1, 191-197.

8. W.D. GROPP (1987). Local Uniform Mesh Refinement with Moving Grids, *SIAM J. Sci. Stat. Comput.*, 8, 292-304.

9. W.D. GROPP (1987). Local Uniform Mesh Refinement on Vector and Parallel Processors, in *Large Scale Scientific Computing*, 349-367, ed. P. DEUFLHARD, B. ENGQUIST, Birkhauser Series Progress in Scientific Computing.

10. W.H. HUNDSDORFER and J.G. VERWER (1989). Stability and Convergence of the Peaceman-Rachford ADI Method, *Math. Comp.*, 53, 81-101.

11. L.R. PETZOLD (1987). Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *Appl. Numer. Math.*, 3, 347-360.

12. L.R. PETZOLD (1987). *Adaptive Moving Grid Strategies for One-Dimensional Systems of Partial Differential Equations,* Preprint UCLR-96190, Lawrence Livermore National Laboratory.

13. J.M SANZ-SERNA and J.G. VERWER (1989). Stability and Convergence at the PDE/Stiff ODE Interface, *Appl. Numer. Math.*, 5, 117-132.

14. R.A. TROMPERT and J.G. VERWER (1989). *A Static-Regridding Method for Two Dimensional Parabolic Partial Differential Equations,* Report NM-R8923, Centre for Mathematics and Computer Science, Amsterdam (to appear in Appl. Numer. Math.).