

A local defect correction technique for time-dependent problems

Citation for published version (APA):

Minero, R., Anthonissen, M. J. H., & Mattheij, R. M. M. (2004). *A local defect correction technique for time-dependent problems*. (CASA-report; Vol. 0425). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2004

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

A Local Defect Correction Technique for Time-Dependent Problems

R. Minero, M.J.H. Anthonissen, R.M.M. Mattheij

Eindhoven University of Technology
Department of Mathematics and Computer Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email r.minero@tue.nl

In this paper a Local Defect Correction technique for time-dependent problems is presented. The method is suitable for solving Partial Differential Equations characterized by a high activity, which is mainly located, at each time, in a small part of the physical domain. The problem is solved at each time step by means of a global uniform coarse grid and a local uniform fine grid. Local and global approximation are improved iteratively. Results of numerical experiments illustrate the accuracy, the efficiency and the robustness of the method.

1 Introduction

Solutions of Partial Differential Equations (PDEs) are often characterized, at each time, by regions where spatial gradients are quite large compared to those in the rest of the domain, where the solution presents a relatively smooth behavior. Examples are the solutions of PDEs describing shock hydrodynamics, transport in turbulent flow fields, combustion processes, etc. An efficient numerical solution of such problems requires the usage of adaptive grid techniques. In adaptive grid methods, a fine grid spacing is adopted only where the large variations occur, so that the computational effort and the memory requirements are minimized.

A large number of adaptive grid methods for time-dependent problems have been proposed in the literature. A first category includes the *moving-grid* or *dynamic-regridding* methods. In this approach, nodes are moving continuously in the space-time domain, like in classical Lagrangian methods, and the discretization of the PDEs is coupled with the motion of the grid. A few examples can be found in [9, 12, 13, 14, 18]. These methods differ in how the motion of the grid is governed. The grid is anyhow always nonuniform and the number of nodes remains constant in time. A disadvantage of this approach is that programming these methods often involves quite complicated data structures.

Another type of adaptive grid techniques is represented by the *static-regridding* methods. Here, the idea is to adapt the grid at each time by adding grid points where a high activity occurs and removing them where they are no longer needed. This process is controlled by error estimates or methods based on the measure of some characteristics of the solution (e.g. gradients, slope, etc.). In this kind of methods the number of

grid points is not constant in time. A static-regridding technique of particular interest is the Local Uniform Grid Refinement (LUGR) method, described in [16]. In this method, at each time step the problem is first solved on a global uniform coarse grid. The coarse grid solution provides approximate boundary conditions on a local uniform fine grid and the problem is then solved locally. At the end, the fine grid values are used to replace the coarse grid values in the region of refinement. The technique relies on the fact that the coarse grid solution provides artificial boundary conditions for the local problem that are accurate enough. The main advantage of LUGR is the possibility of working with uniform grids and uniform grid solvers only.

The Local Defect Correction (LDC) method we present in this paper is a static-regridding method. LDC shares with LUGR the possibility and the advantage of working with uniform grids and uniform grid solvers only. Moreover, LDC has the advantage that the fine grid solution is used not only to replace the coarse grid values in the area of refinement, but to overall improve the coarse grid approximation. This can be achieved through the so called defect correction, in which the fine grid solution is used to approximate the local discretization error. The improved coarse grid approximation defines new artificial boundary conditions for a new local problem, which in turn can correct the global solution. At each time step, LDC is thus an iterative process and, as discussed in [2] for stationary cases, its convergence is very fast. In this way, LDC does not have to rely on the accuracy of the artificial boundary condition provided by the first coarse grid approximation, turning out to be a more robust method than LUGR.

As a technique for solving elliptic problems, Local Defect Correction was first introduced in [11]. An analysis of LDC in combination with finite difference is presented in [7, 8]. The LDC technique is studied in combination with a finite volume discretization in [3, 1], and in combination with finite element in [17]. The LDC method has also been employed with different grid types: in [15] the global coarse grid is cartesian, while polar coordinates are used locally; in [10] the local fine grid is in a slantwise direction.

In this work we extend the LDC method for solving time-dependent PDEs. This paper is organized as follows: in Section 2 we describe the LDC algorithm for a time-dependent problem; in Section 3 we present the results of some numerical experiments; Section 4 is finally devoted to conclusions.

2 The LDC method for a time-dependent problem

In this section the Local Defect Correction method for time-dependent problems is presented. At a generic time step, the problem is first solved on a global coarse grid which provides artificial boundary conditions for a local fine grid solution; this is explained in Section 2.1. The local solution is then employed to improve the global approximation through the defect correction; the process can be iteratively repeated till convergence. The defect correction is the subject of Section 2.2.

2.1 A first approximation of the solution at time t_n

To formulate the LDC algorithm for a two-dimensional time-dependent problem, we consider the following equation:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{L}\mathbf{u} + \mathbf{f}, \quad \text{in } \Omega \times \Theta, \quad (1)$$

in which $u = u(\mathbf{x}, t)$ is unknown, L is a linear spatial differential operator and $f = f(\mathbf{x}, t)$ a source term. Furthermore Ω is an open rectangular subset of \mathbb{R}^2 and Θ is the time interval $(0, t_{\text{end}}]$. We close problem (1) by prescribing the Dirichlet boundary condition

$$u = \psi, \quad \text{on } \partial\Omega \times \Theta, \quad (2)$$

and the initial condition

$$u(\mathbf{x}, 0) = \varphi_0, \quad \text{in } \bar{\Omega}. \quad (3)$$

In (2) and (3), $\psi = \psi(\mathbf{x}, t)$ and $\varphi_0 = \varphi_0(\mathbf{x})$ are two known functions, while $\partial\Omega$ denotes the boundary of Ω and $\bar{\Omega} := \Omega \cup \partial\Omega$. We suppose that u , the continuous solution of (1) that satisfies (2) and (3), has for every time in Θ a high activity region that covers some (small) part of Ω . We also suppose that explicit time integration schemes are not suitable for computing a numerical approximation of u .

In order to be solved numerically, problem (1) has to be discretized in space and time. For this purpose, in Ω we introduce a global regular coarse grid (grid size H), which we denote by Ω_H . Grid points $\partial\Omega_H$ are placed on $\partial\Omega$ too and we define $\bar{\Omega}_H := \Omega_H \cup \partial\Omega_H$. The time interval Θ is divided into $N_t \geq 1$ subintervals such that $t_{\text{end}}/N_t := \Delta t$. We also introduce $t_n := n\Delta t$, with $n = 0, 1, 2, \dots, N_t$, and by u_H^n we denote the approximation of the continuous solution u on the global coarse grid at time t_n .

Because of the local high activity of the solution, at time t_n a coarse grid approximation u_H^n might not be accurate enough to adequately represent $u^n := u(t_n)$. Therefore we want to find a more accurate (both in space and time) local approximation of u^n and eventually use it to correct and improve u_H^n . At the time interval $[t_{n-1}, t_n] \subset \Theta$, we choose for this purpose Ω_l^n , an open subset of Ω such that the high activity of u is entirely contained in Ω_l^n . In Ω_l^n the subscript l reminds us that we are dealing with a local region, while the superscript n refers to the fact that this local region is defined on the n -th time interval, namely $[t_{n-1}, t_n]$. The boundary of Ω_l^n is $\partial\Omega_l^n$ and we also define $\bar{\Omega}_l^n := \Omega_l^n \cup \partial\Omega_l^n$. The high activity may be captured by introducing a local fine regular grid (grid size $h < H$), which we denote by $\Omega_{l,h}^n$. We also introduce the grid points $\partial\Omega_{l,h}^n$ on the boundary of $\Omega_{l,h}^n$ and we define $\bar{\Omega}_{l,h}^n := \Omega_{l,h}^n \cup \partial\Omega_{l,h}^n$. The local fine grid is chosen such that $\Omega_H \cap \Omega_l^n \subset \Omega_{l,h}^n$, i.e. grid points of the global coarse grid that lie in the area of refinement belong to the local fine grid too. In this way the *factor of grid refinement* σ , defined as

$$\sigma := \frac{H}{h}, \quad (4)$$

turns out to be a positive integer. The union of coarse and fine grids defines a *composite grid*. Figure 1 shows an example of a composite grid. In that figure the coarse grid points Ω_H are marked with a cross and the grid points $\partial\Omega_H$ are denoted with an empty triangle. The region with grey background is the area of refinement Ω_l^n : in that region the fine grid points $\Omega_{l,h}^n$ are marked with a small circle, whereas the grid points $\partial\Omega_{l,h}^n$ are denoted with a small triangle. The figure also shows that the coarse grid points that lie in Ω_l^n belong to the fine grid too. In this example the factor of grid refinement σ is 3.

The time interval $[t_{n-1}, t_n]$ is further divided into $n_t \geq 1$ subintervals such that $(t_n - t_{n-1})/n_t := \delta t$. Analogously to σ , we can thus define the *factor of time refinement* as

$$\tau := \frac{\Delta t}{\delta t}. \quad (5)$$

We finally introduce $t_{n-1+k/\tau} := t_{n-1} + k\delta t$, with $k = 0, 1, 2, \dots, n_t$; with $u_{l,h,n}^{n-1+k/\tau}$ we denote an approximation of u on the local fine grid $\Omega_{l,h}^n$ at time $t_{n-1+k/\tau}$. In $u_{l,h,n}^{n-1+k/\tau}$

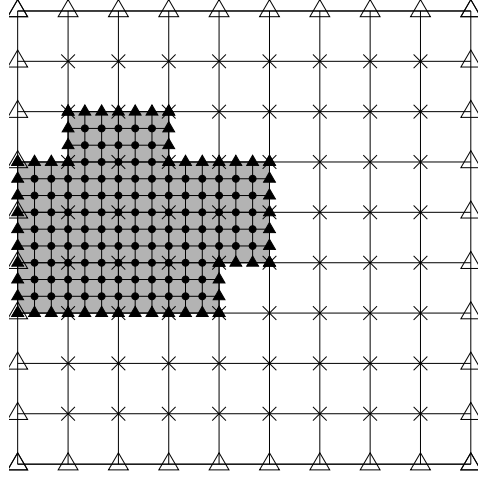


Figure 1: Example of a composite grid at a certain time t_n .

the superscript refers to the time level, while the three subscripts l , h and n remind us that we are dealing with an approximation defined on a local grid (l) with grid space h in the n -th time interval $[t_{n-1}, t_n]$. In a time-dependent problem the high activity region can both move and change its size as time proceeds, so that it can easily happen that $\Omega_{l,h}^n \neq \Omega_{l,h}^{n-1}$. There is thus a difference between $u_{l,h,n}^{n-1}$, an approximation of u^{n-1} on $\Omega_{l,h}^n$, and $u_{l,h,n-1}^{n-1}$, which is defined at the same time level but on $\Omega_{l,h}^{n-1}$.

In the sequel of this section we will first find an expression for u_H^n and secondly we will define the local problem that leads to determine $u_{l,h,n}^n$, a local better approximation of u^n .

We suppose that the LDC technique has been applied in the time interval $[t_{n-2}, t_{n-1}]$, with $n > 1$. If this is the case, the following approximation of u is available on the composite grid $\bar{\Omega}_{H,h}^{n-1} := \bar{\Omega}_H \cup \bar{\Omega}_{l,h}^{n-1}$ at time t_{n-1} :

$$u_{H,h,n-1}^{n-1} := \begin{cases} u_{l,h,n-1}^{n-1}, & \text{in } \bar{\Omega}_{l,h}^{n-1}, \\ u_H^{n-1}, & \text{in } \bar{\Omega}_{H,h}^{n-1} \setminus \bar{\Omega}_{l,h}^{n-1}. \end{cases} \quad (6)$$

In (6) the numerical approximation includes the boundary condition too. The approximation $u_{H,h,n-1}^{n-1}$ is called *composite grid solution*. We notice that in the expression of the composite grid solution the local approximation is considered to be more accurate and hence defines $u_{H,h,n-1}^{n-1}$ in the common points between coarse and fine grid. We call $u_{H,H}^{n-1}$ the restriction of $u_{H,h,n-1}^{n-1}$ on Ω_H . For $n = 1$, $u_{H,H}^{n-1}$ is expressed through the initial condition (3):

$$u_{H,H}^0 = \varphi_0|_{\Omega_H}, \quad \text{in } \Omega_H. \quad (7)$$

If now the Implicit Euler scheme is applied to (1), u_H^n can be computed solving the linear system

$$\frac{u_H^n - u_{H,H}^{n-1}}{\Delta t} = L_H u_H^n + f_H^n, \quad \text{in } \Omega_H. \quad (8)$$

In (8) the Dirichlet boundary condition (2) has been incorporated in the discretized source term f_H^n . We have not been specific about the continuous operator L and the discretized operator L_H ; it may be helpful to think of L as the Laplacian operator Δ , and L_H as the standard finite differences five-point stencil approximating Δ .

We proceed defining a problem on the local fine grid $\Omega_{l,h}^n$. As already mentioned before, in a time-dependent problem it is likely that the high activity region moves and

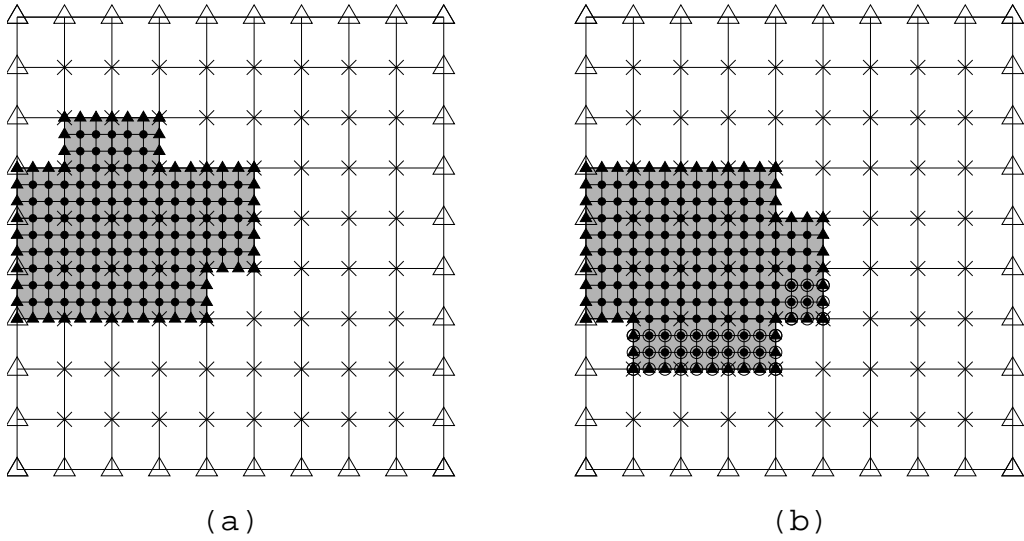


Figure 2: An example of composite grid $\bar{\Omega}_{l,h}^{n-1}$ at time t_{n-1} (a) and of composite grid $\bar{\Omega}_{l,h}^n$ at time t_n (b). The fine grid points in (b) marked with an extra circle form $\tilde{\Omega}_{l,h}^n$.

changes its size as time proceeds, so that in general $\bar{\Omega}_{l,h}^n \neq \bar{\Omega}_{l,h}^{n-1}$. This leads to the fact that a local approximation of u at t_{n-1} is directly available only at the common points between $\bar{\Omega}_{l,h}^{n-1}$ and $\bar{\Omega}_{l,h}^n$. In order to obtain it on the remaining part of $\bar{\Omega}_{l,h}^n$, i.e. on $\tilde{\Omega}_{l,h}^n := \bar{\Omega}_{l,h}^n \setminus (\bar{\Omega}_{l,h}^{n-1} \cap \bar{\Omega}_{l,h}^n)$, we introduce an operator Q_x^{n-1} that spatially interpolates $u_{H,h,n-1}^{n-1}$ on $\tilde{\Omega}_{l,h}^n$. Figure 2 represents an example in which, from t_{n-1} to t_n , the local region has changed its location, its shape and its area. The fine grid points in Figure 2-b marked with an extra circle form $\tilde{\Omega}_{l,h}^n$: in those points a fine grid approximation at time t_{n-1} is not directly available and it has to be computed through interpolation from $u_{H,h,n-1}^{n-1}$. From a practical point of view, we can imagine the operation performed by Q_x^{n-1} to be a linear interpolation. We should also notice that there was no high activity at time t_{n-1} on $\tilde{\Omega}_{l,h}^n$ (by definition $\tilde{\Omega}_{l,h}^n \cap \bar{\Omega}_{l,h}^{n-1} = \emptyset$) and therefore the interpolation takes place in a region with no high gradients. Furthermore if the time step Δt is small enough, we can imagine that the area of high activity does not move much between two consecutive time steps, so that interpolation is needed on a small region only. Once the interpolation process has taken place, we have a local approximation of u at each point of $\bar{\Omega}_{l,h}^n$ at time t_{n-1} ($n > 1$); this approximation is called $u_{l,h,n}^{n-1}$ and its expression is given by

$$u_{l,h,n}^{n-1} := \begin{cases} u_{l,h,n-1}^{n-1}, & \text{in } \bar{\Omega}_{l,h}^n \setminus \tilde{\Omega}_{l,h}^n, \\ Q_x^{n-1}(u_{H,h,n-1}^{n-1}), & \text{in } \tilde{\Omega}_{l,h}^n. \end{cases} \quad (9)$$

At this stage we have defined a local grid $\bar{\Omega}_{l,h}^n$ and provided initial values $u_{l,h,n}^{n-1}$ at each of its points. If we want to solve a discrete equivalent of (1) on $\Omega_{l,h}^n \times \{t_{n-1+k/\tau}, k = 1, \dots, n_t\}$, we also have to provide conditions on the boundary of the local grid for each time $t_{n-1+k/\tau}$, $k = 1, \dots, n_t$. For $\Gamma_1^n := \partial\Omega \cap \partial\Omega_{l,h}^n$, i.e. the part of the local area's boundary which is common with the global boundary, we can appropriately use the original boundary condition (2) at any time $t_{n-1+k/\tau}$, $k = 1, \dots, n_t$. For that purpose we define $\Gamma_{1,h}^n := \Gamma_1^n \cap \partial\Omega_{l,h}^n$. As for the rest of the local area's boundary, namely $\Gamma_2^n := \partial\Omega_{l,h}^n \setminus \Gamma_1^n$, we introduce the interpolation operator in space P_x^n , that interpolates u_H^n on $\Gamma_{2,h}^n := \Gamma_2^n \cap \partial\Omega_{l,h}^n$. With P_x^n we are able to prescribe artificial Dirichlet

boundary conditions on $\Gamma_{2,h}^n$ at t_n . Since we need boundary conditions not only at t_n , but for all the $t_{n-1+k/\tau}$, with $k = 1, 2, \dots, n_t$, we define another interpolation operator $R_t^{n,k}$. The operator $R_t^{n,k}$ performs time interpolation between the time levels t_{n-1} and t_n ; in particular, $R_t^{n,k}$ interpolates between the restriction of $u_{l,h,n}^{n-1}$ on $\Gamma_{2,h}^n$, see (9), and $P_x^n(u_H^n)$. In this way, $R_t^{n,k}$ enables us to specify artificial Dirichlet boundary conditions on $\Gamma_{2,h}^n$ for every $t_{n-1+k/\tau}$, with $k = 1, 2, \dots, n_t$. We can synthetically write the boundary condition for the local problem as

$$u_{l,h,n}^{n-1+k/\tau} = \tilde{\psi}_{l,h,n}^{n-1+k/\tau}, \quad \text{on } \Gamma_{1,h}^n \cup \Gamma_{2,h}^n, k = 1, 2, \dots, n_t, \quad (10)$$

where

$$\tilde{\psi}_{l,h,n}^{n-1+k/\tau} := \begin{cases} \psi(t_{n-1+k/\tau}), & \text{on } \Gamma_{1,h}^n, k = 1, \dots, n_t, \\ R_t^{n,k} \left(u_{l,h,n}^{n-1} |_{\Gamma_{2,h}^n}, P_x^n(u_H^n) \right), & \text{on } \Gamma_{2,h}^n, k = 1, \dots, n_t. \end{cases} \quad (11)$$

From (11) we can see that the boundary condition for the local problem depends on u_H^n , the approximation of u^n that we computed on the global coarse grid solving (8).

At this moment we have provided initial values of u on $\Omega_{l,h}^n$ and boundary condition on $\partial\Omega_{l,h}^n = \Gamma_{1,h}^n \cup \Gamma_{2,h}^n$ for $t_{n-1+k/\tau}$, $k = 1, \dots, n_t$. If we now introduce a local discretization of (1), we are able to formulate a local problem from which we can compute $u_{l,h,n}^n$. The local approximation $u_{l,h,n}^n$ is regarded to be more accurate than u_H^n since it is computed using a finer grid ($h < H$) and a smaller time step ($\delta t \leq \Delta t$). If again the Implicit Euler scheme is used, the local problem that enables us to determine $u_{l,h,n}^n$ is

$$\begin{aligned} & \frac{u_{l,h,n}^{n-1+k/\tau} - u_{l,h,n}^{n-1+(k-1)/\tau}}{\delta t} \\ & = L_{l,h,n} u_{l,h,n}^{n-1+k/\tau} + f_{l,h,n}^{n-1+k/\tau}, \quad \text{in } \Omega_{l,h}^n, k = 1, \dots, n_t. \end{aligned} \quad (12)$$

Boundary condition (11) has been included in the right hand side of (12). The procedure (12) is initialized using (9) if $n > 1$, or by a proper discretization of the original initial condition (3) if $n = 1$.

We briefly summarize what we have done so far in order to compute a first approximation of u at t_n on the composite grid $\Omega_{H,h}^n$:

- u_H^n is computed performing one time step with (8);
- a local region Ω_l^n that includes u 's high activity is chosen and a fine grid $\Omega_{l,h}^n$ introduced on it;
- the time step $[t_{n-1}, t_n]$ is divided in subintervals whose length is $\delta t \leq \Delta t$;
- an approximation of u^{n-1} is provided at each grid point of $\Omega_{l,h}^n$;
- a boundary condition for the local problem (12) is provided on the border of $\Omega_{l,h}^n$ for $t_{n-1+k/\tau}$, $k = 1, \dots, n_t$;
- $u_{l,h,n}^n$, a local approximation of u^n which is considered to be more accurate than u_H^n , is computed solving (12).

2.2 Defect correction

The crucial part of the LDC algorithm is how the local solution $u_{l,h,n}^n$ is used to improve the global approximation u_H^n through an approximation of the *local discretization error* or *defect*. The defect d_H^n is defined as

$$d_H^n := \frac{u^n|_{\Omega_H} - u^{n-1}|_{\Omega_H}}{\Delta t} - L_H u^n|_{\Omega_H} - f_H^n, \quad \text{in } \Omega_H. \quad (13)$$

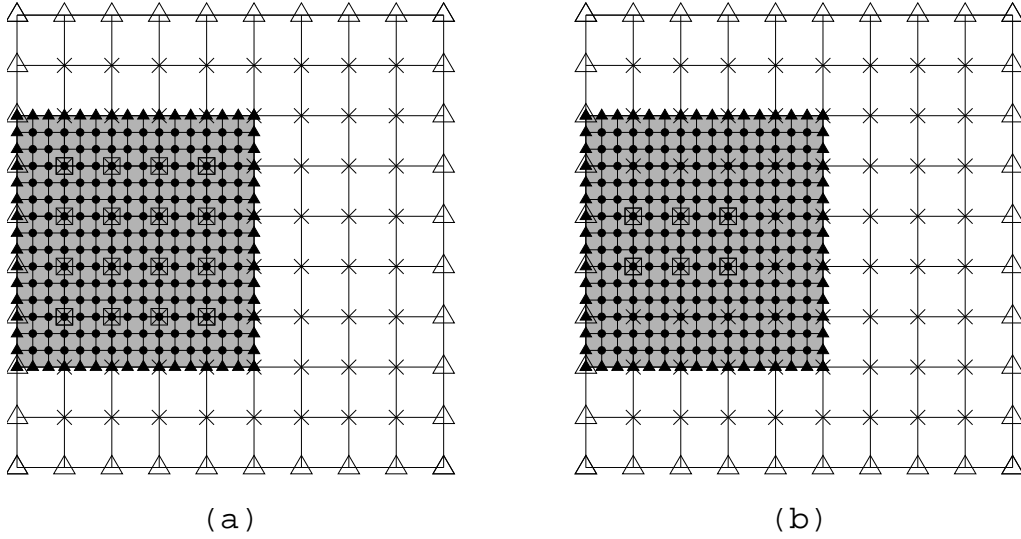


Figure 3: An example of composite grid at time t_n with no safety region (a) and with safety region of depth 1 (b).

In (13) we substituted the projection on Ω_H of the continuous solution u into the discretization scheme (8). If we would know the values of the defect d_H^n , we could use them to find a better approximation of u^n on the coarse grid. This could be achieved by putting d_H^n on the right hand side of (8). However, since we do not know the exact solution of our partial differential equation, we cannot calculate the values of d_H^n . What we can do, though, is to use the more accurate local approximation $u_{l,h,n}^n$ to get an estimation \tilde{d}_H^n of d_H^n on $\Omega_{l,H}^n := \Omega_H \cap \Omega_{l,h}^n$

$$d_H^n \approx \tilde{d}_H^n := \begin{cases} \frac{u_{l,h,n}^n - u_{H,H}^{n-1}}{\Delta t} - L_H u_{l,h,n}^n - f_H^n, & \text{on } \Omega_{l,H}^n, \\ 0, & \text{on } \Omega_H \setminus \Omega_{l,H}^n. \end{cases} \quad (14)$$

Previous results for stationary problems (see [2, 11]) show that it may be beneficial to estimate the defect not at every point of $\Omega_{l,H}^n$, but in a subset of $\Omega_{l,H}^n$ only. In particular, nodes in $\Omega_{l,H}^n$ lying close to the interface Γ_2^n should be excluded. This leads to the introduction of a subset $\Omega_{l,H,\text{def}}^n$ of $\Omega_{l,H}^n$, separated from the interface Γ_2^n by a so called *safety region*. In Figure 3 the grid points where an approximation \tilde{d}_H^n of the actual defect d_H^n is computed are marked with a square. In Figure 3-a $\Omega_{l,H,\text{def}}^n \equiv \Omega_{l,H}^n$ and there is no safety region. In Figure 3-b a safety region of depth 1 has been introduced: an approximation of \tilde{d}_H^n is not computed in the first layer of $\Omega_{l,H}^n$ next to Γ_2^n .

The estimate \tilde{d}_H^n of the local discretization error is finally placed on the right hand side of scheme (8):

$$\frac{u_{H,1}^n - u_{H,H}^{n-1}}{\Delta t} = L_H u_{H,1}^n + f_H^n + \tilde{d}_H^n, \quad \text{in } \Omega_H. \quad (15)$$

The solution of (15) leads to a better approximation of u^n on the global grid Ω_H . In (15) we have called this new approximation $u_{H,1}^n$, where the new subscript is used to distinguish the new approximation from the previous one u_H^n , from now on referred as $u_{H,0}^n$. Once $u_{H,1}^n$ is computed, we are able to define new boundary conditions, see (11), for a new local problem on $\Omega_{l,h}^n$ and this triggers an iterative procedure which is formalized

in Algorithm 1. As for the global grid solution, in Algorithm 1 an extra subscript is added to number the different approximations computed locally; the same is done for the defect term.

Algorithm 1 (LDC algorithm for a time-dependent problem)

FOR LOOP, $n = 1, 2, \dots, N_t$

INITIALIZATION

- Compute a global approximation $u_{H,0}^n$ solving problem (8).
- Choose a region of refinement Ω_l^n , introduce a fine grid $\Omega_{l,h}^n$ on it, divide the time step $[t_{n-1}, t_n]$ in $n_t \geq 1$ subintervals.
- Compute an initial condition for the local problem; for $n = 1$ use a discretization of the original initial condition (3), otherwise do as in (9).
- Compute a boundary condition for the local problem as in (11).
- Compute a local approximation $u_{l,h,n,0}^n$ solving the local problem (12).

ITERATION, $w = 1, 2, \dots$

- Use $u_{l,h,n,w-1}^n$ to compute an approximation $\tilde{d}_{H,w-1}^n$ of the local discretization error as in (14).
- Compute a more accurate global approximation $u_{H,w}^n$ solving a modified global problem as in (15).
- Use $u_{H,w}^n$ to update the boundary condition for the local problem.
- Compute a local solution $u_{l,h,n,w}^n$ with updated boundary condition.

END ITERATION ON w

- Call $u_{l,h,n}^n$ and u_H^n the latest solutions that have been found on the local and global grid respectively (remove the last subscript); the solution at time t_n (including boundary condition) is:

$$u_{H,h,n}^n := \begin{cases} u_{l,h,n}^n, & \text{in } \bar{\Omega}_{l,h}^n, \\ u_H^n, & \text{in } \bar{\Omega}_H \setminus \bar{\Omega}_{l,h}^n. \end{cases} \quad (16)$$

END FOR LOOP ON n

In Algorithm 1 the iteration on w is called *LDC iteration*. Each LDC iteration consists in the entire recomputation of the time step Δt . For a good performance in solving the transient problem, it is thus desirable than only a small number of LDC iterations are needed at each time step. However, as it happens for stationary problems, the convergence of the LDC iterations is very fast and it generally requires only one iteration.

When presenting the LDC method for time-dependent problems we have used the Implicit Euler scheme for time discretization. We should notice here that this is not restrictive and that other implicit methods for time discretization (e.g. Runge-Kutta schemes) might be applied as well. Moreover we are not constrained to use the same scheme for the global and for the local grid.

A final consideration regards the choice of the local region Ω_l^n . Many methods have been proposed in the literature of adaptive methods and in principle one can use any kind of criterion which is suitable for one's specific application. In [5, 6, 4], for example,

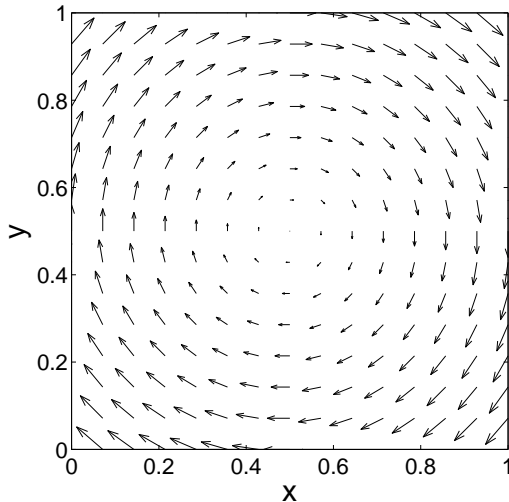


Figure 4: Plot of the velocity field $\mathbf{v} = (y - 0.5, -x + 0.5)$.

a positive weight function w_{ij} is introduced to determine which coarse grid boxes $B_{ij} := (x_i, x_{i+1}) \times (y_j, y_{j+1})$ require refinement. The weight function w_{ij} , which is in practice an indicator of the solution roughness, is computed on each box B_{ij} from the gradient of the coarse grid approximation. After that a smoothing filter, an averaging and a normalization procedure are applied. At the end the boxes B_{ij} for which $w_{ij} > \epsilon$ are labelled for refinement. The threshold value ϵ is a user-specified parameter typically ranging from 1.5 to 3. The method is described in further detail in [5, 6, 4] and it will be used in one of the examples in Section 3.

3 Numerical experiments

In this section we present two numerical examples. The first one, Section 3.1, is a 2D numerical experiment in which the LDC technique is compared with a standard uniform grid solver. We show that LDC can achieve the same accuracy as the uniform grid solver, while requiring the computation of a smaller number of unknowns and being thus a more efficient method. In the second example, Section 3.2, a 1D problem is solved both with LDC and LUGR. We illustrate the robustness of the LDC method showing that when LUGR fails, LDC can still lead to accurate results.

3.1 Example 1: a 2D convection-diffusion problem

In this section we present the results of a 2D numerical experiment. We choose $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$ and $\Theta = (0, 2]$, and we solve the following problem

$$\begin{cases} \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \lambda \Delta u, & \text{in } \Omega \times \Theta, \\ u = \exp\left(\frac{(x - 0.3)^2 + (y - 0.3)^2}{10^{-2}}\right), & \text{in } \Omega, t = 0, \\ u = 0, & \text{on } \partial\Omega \times \Theta, \end{cases} \quad (17)$$

where $\mathbf{v} = \mathbf{v}(x, y) = (y - 0.5, -x + 0.5)$, see Figure 4, and $\lambda = 10^{-4}$. We observe that

Grid and time step			ϵ_∞		Total number unknowns		
H	Δt	$\sigma = \tau$	LDC ₁	Unif.	LDC ₁ (coarse + fine)	Unif.	$\frac{\text{Unif.}}{\text{LDC}_1}$
H_0	Δt_0	3	$3.6 \cdot 10^{-1}$	$3.7 \cdot 10^{-1}$	$7.2 \cdot 10^3 + 3.1 \cdot 10^4$	$1.0 \cdot 10^5$	2.7
		5	$2.8 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$7.2 \cdot 10^3 + 1.4 \cdot 10^5$	$4.9 \cdot 10^5$	3.3
		7	$2.3 \cdot 10^{-1}$	$2.3 \cdot 10^{-1}$	$7.2 \cdot 10^3 + 4.0 \cdot 10^5$	$1.7 \cdot 10^6$	3.3
$\frac{H_0}{2}$	$\frac{\Delta t_0}{4}$	3	$1.7 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$1.2 \cdot 10^5 + 4.4 \cdot 10^5$	$1.7 \cdot 10^6$	3.0
		5	$1.1 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$1.2 \cdot 10^5 + 2.0 \cdot 10^6$	$7.9 \cdot 10^6$	3.7
		7	$8.6 \cdot 10^{-2}$	$8.6 \cdot 10^{-2}$	$1.2 \cdot 10^5 + 5.6 \cdot 10^6$	$2.2 \cdot 10^7$	3.8
$\frac{H_0}{4}$	$\frac{\Delta t_0}{16}$	3	$5.4 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$	$2.0 \cdot 10^6 + 6.8 \cdot 10^6$	$2.7 \cdot 10^7$	3.1
		5	$3.3 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	$2.0 \cdot 10^6 + 3.2 \cdot 10^7$	$1.3 \cdot 10^8$	3.8
		7	$2.4 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$2.0 \cdot 10^6 + 8.7 \cdot 10^7$	$3.5 \cdot 10^8$	3.9

Table 1: Results of the 2D numerical experiment. In the table: $H_0 = 1/20$, $\Delta t_0 = 0.2$.

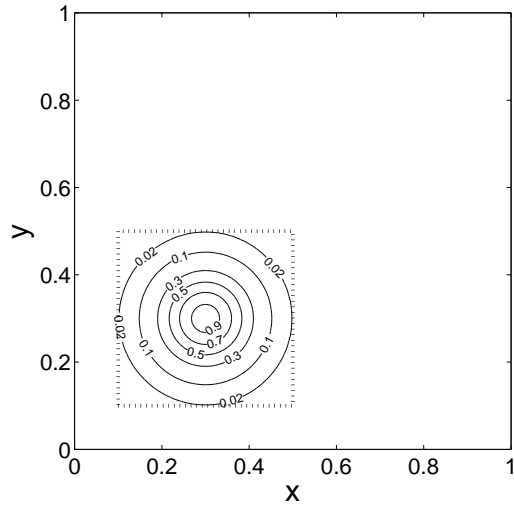
in (17) $L \equiv \lambda \Delta - \mathbf{v} \cdot \nabla$; L is thus a convection-diffusion operator. Figure 5 shows the contour plots of u for different values of time. The solution of problem (17) has at each time a region of high activity that covers a limited part of Ω . Figure 5 also shows (dotted line) the location of the local region Ω_1^n in one of our LDC runs.

Problem (17) is solved using LDC with different values of H , Δt , σ and τ (see Table 1). In all the LDC runs, one and only one LDC iteration is performed at each time step Δt ; to recall this, in Table 1 we write LDC₁, where the subscript indicates the number of LDC iterations at each time step. In all our runs the spatial discretization is performed using Finite Difference; in particular the second order centered differences scheme is applied both on the global and on the local grid. The time discretization is performed using the first order Implicit Euler scheme both globally and locally. The position and the size of the local region are determined at each time step using the already mentioned algorithm which is described in [5, 6, 4]; we choose a threshold value $\epsilon = 3$. For simplicity of implementation in all our runs the local region Ω_1^n has always a rectangular shape. Furthermore we use no safety region when computing the defect. The operators Q_x^{n-1} , P_x^n and $R_t^{n,k}$ are chosen to perform linear interpolation for every n . As a measure of the accuracy of the numerical solution found at $t_{\text{end}} = 2$ using LDC, we compute the infinity norm

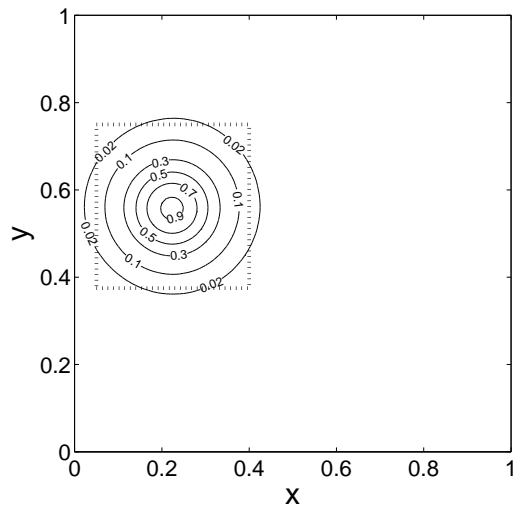
$$\epsilon_\infty := \max \left(u_{H,H}^{N_t} - u(t_{\text{end}})|_{\Omega_H} \right). \quad (18)$$

In (18) $u_{H,H}^{N_t}$ is the restriction on the coarse grid of the composite grid solution at the final time, while $u(t_{\text{end}})|_{\Omega_H}$ is the projection on Ω_H of a reference solution that we computed once on a uniform grid using a very small grid size h_{ref} and a very small time step δt_{ref} . The two parameters h_{ref} and δt_{ref} are chosen in such a way that they are a few times smaller than the fine grid size $h = H/\sigma$ and the time step $\delta t = \Delta t/\tau$ used on the most accurate of our LDC runs; in practice we have $h_{\text{ref}} = 1/2000$ and $\delta t_{\text{ref}} = 2 \cdot 10^{-4}$.

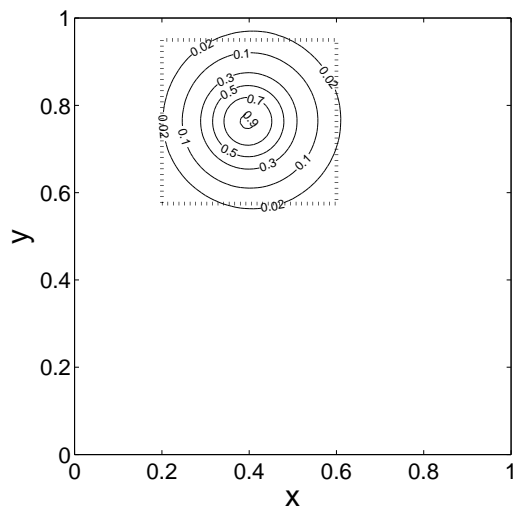
The results of each of the LDC runs are compared with the numerical solution found solving problem (17) on a single global uniform grid with the same discretization schemes. The grid size in each uniform grid run is the same as the fine grid size $h = H/\sigma$ of the corresponding LDC run; the same holds for the time step. Also for the single uniform grid runs we measure the infinity norm (18). Results in Table 1 show that in all the cases we considered, LDC is able to achieve the same accuracy as the uniform grid solver. Of course LDC requires less computational effort than the uniform grid solver and it is a more efficient technique since the fine grid and the small



(a) u at $t = 0.0$



(b) u at $t = 1.0$



(c) u at $t = 2.0$

Figure 5: Contour plots of the solution of problem (17) and position (dotted line) of Ω_1^n at different times.

time step are used only where it is needed. To give a rough idea of the complexity of the two methods, in Table 1 we also report the total number of unknowns that have to be computed to solve problem (17). For the uniform grid solver the total number of unknowns is calculated as the product of the number of grid points and the number of time steps. For LDC this product is given separately for the coarse grid and for the sum of all the local problems; the numbers in Table 1 already take into account that in LDC₁ each time step Δt has to be repeated twice (computation of a first approximation at t_n plus one defect correction). From the table we can see that in our example, when we use LDC, we have to compute a total number of unknowns which is about three times less than when the uniform grid solver is adopted. The gain increases when a higher factor of grid refinement is used.

To conclude, this example shows that LDC can achieve the same accuracy as a uniform grid solver that uses the same grid size and time step as in the LDC local problem. Yet, LDC is a more efficient method than the uniform grid solver: in LDC the fine grid is adaptively placed only where it needs to be and this guarantees a saving in the total number of unknowns that have to be computed to solve the problem.

3.2 Example 2: a 1D convection-diffusion problem

In this section we present the results of a 1D numerical experiment aimed at showing the robustness of the LDC method. We choose $\Omega = (0, 2)$ and $\Theta = (0, 1]$, and we solve the partial differential equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = f, \quad \text{in } \Omega \times \Theta. \quad (19)$$

We notice that in (19)

$$L \equiv -\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}. \quad (20)$$

Clearly L is thus a convection–diffusion operator. The initial condition, the Dirichlet boundary conditions and the source term f are chosen in such a way that the exact analytical solution of the problem is

$$u(x, t) = \left(\tanh(100(x - 1/8 - t)) + 1 \right) (1 - e^{-2t}). \quad (21)$$

At time $t > 0$ the exact solution (21) has a region of high activity around point $x_a = 1/8 + t$ (see Figure 6).

The problem above is solved using both LDC and LUGR. Equation (19) is discretized in space using Finite Differences; in particular, both globally and locally a second order centered differences scheme is adopted. The time discretization is performed with the first order Implicit Euler scheme both on the global and on the local grid. Like in the previous example, in all our runs we measure the infinity norm

$$\epsilon_\infty := \max \left(u_{H,H}^{N_t} - u(t_{\text{end}})|_{\Omega_H} \right), \quad (22)$$

where $u_{H,H}^{N_t}$ is the computed numerical solution at $t_{\text{end}} = 1$, while $u(t_{\text{end}})|_{\Omega_H}$ is the projection on the coarse grid of the exact solution (21) at the final time. In our experiments the local region Ω_l^n has a constant width; at each time step Ω_l^n is chosen in such a way that its left and right bounds coincide with coarse grid points and x_a lies in the middle of Ω_l^n . In this way there is no need to define the operator P_x^n . We only have to define Q_x^n and $R_t^{n,k}$: we choose them to perform linear interpolation in space and time respectively. In all our runs in this section the safety region has depth 1.

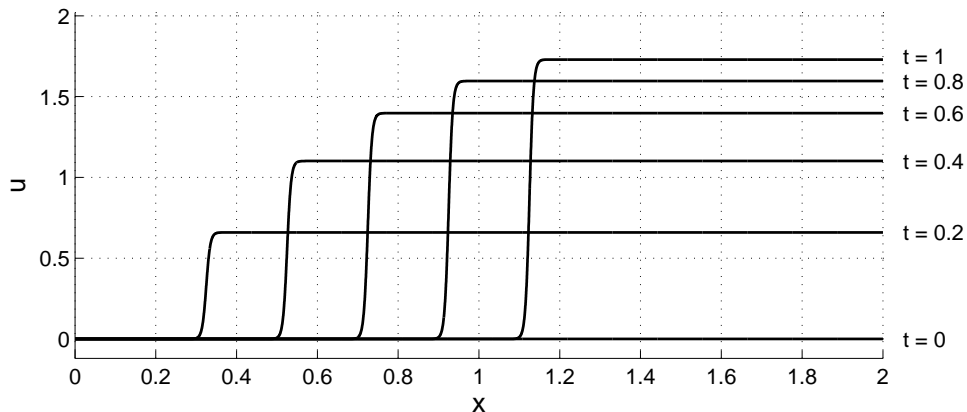


Figure 6: Plot of the exact solution (21) at different values of time t . At each time $t > 0$, u has a region of high activity around point $x_a = 1/8 + t$.

	ϵ_∞	N_g	N_l
LUGR	$5.2 \cdot 10^1$	1.49= 49	1.75= 75
LDC ₁	$1.9 \cdot 10^0$	2.49= 98	2.75=150
LDC ₂	$1.0 \cdot 10^{-1}$	3.49=147	3.75=225
LDC ₃	$3.4 \cdot 10^{-2}$	4.49=196	4.75=300

Table 2: Results of run #0. N_g and N_l indicate the sum of the dimensions of all the linear systems that have to be solved per time step Δt on the global and local grid respectively.

The results of a first numerical experiment, run #0, are presented in Table 2. In this run the width of the local region is 0.6. In the table the subscript next to LDC indicates the number of LDC iterations that are performed at each time step (e.g. LDC₃ means three LDC iterations for every t_n). In this experiment we want to test the robustness of the LDC algorithm. For that reason we choose a very coarse global grid ($H = 1/25$) and a rather big time step ($\Delta t = 1/5$). Locally we refine in space and time and we set $h = 1/125$, $\delta t = 1/25$; in this way $\sigma = \tau = 5$. LUGR is not able to provide a good boundary condition for the local problem and it fails dramatically. LDC, on the other hand, can lead to results of order 10^{-2} accurate if enough LDC iterations (3 in our example) are performed at each time step. Thanks to the process of defect correction, LDC proves to be a more robust technique than LUGR. Of course, Table 2 also shows that increasing the number of LDC iterations, we proportionally increase the complexity that has to be computed per time step Δt : each defect correction (see Algorithm 1) means in fact reperforming the entire time step Δt again and computing new (more accurate) coarse and fine grid approximations. In particular, we have that

$$\frac{N_{\text{LDC}_1}}{N_{\text{LUGR}}} = 2, \quad (23)$$

where N is the total number of unknowns (global grid + local grid) per time step. With total number of unknowns we mean the sum of the dimensions of all the linear systems that have to be solved per time step.

In Table 3 we present the results of other two numerical simulations: run #1 and run #2. In both of them the width of the local region is 0.2, the factors of grid and time refinement are both equal to 5 and we only consider the comparison between

	H	Δt	$\epsilon_{\infty, \text{LUGR}}$	$\epsilon_{\infty, \text{LDC}_1}$	$\frac{\epsilon_{\infty, \text{LUGR}}}{\epsilon_{\infty, \text{LDC}_1}}$
run #1	1/50	1/20	$3.9 \cdot 10^0$	$3.6 \cdot 10^{-2}$	108
run #2	1/100	1/80	$3.1 \cdot 10^{-2}$	$5.5 \cdot 10^{-3}$	5.6

Table 3: Coarse grid size, time step Δt and results of run #1 and run #2. In both runs $\sigma = \tau = 5$.

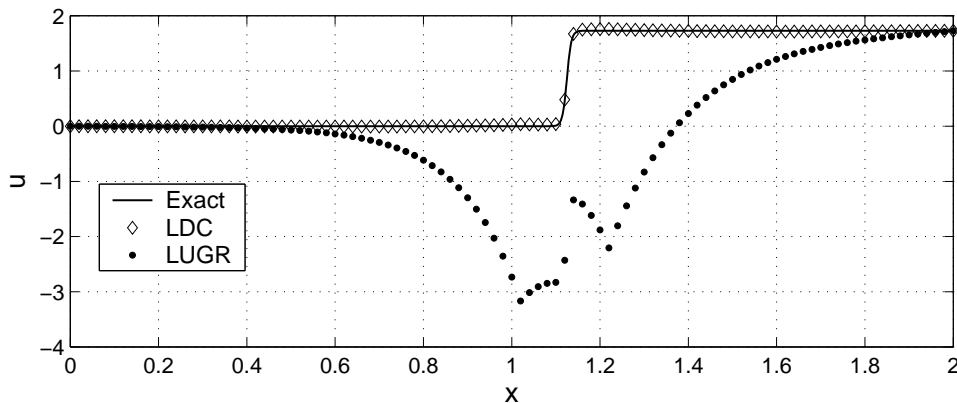


Figure 7: Exact solution (21) at $t_{\text{end}} = 1$ and numerical approximations computed using LDC and LUGR in run #1.

LUGR and LDC_1 . In run #1 we have a situation similar to run #0: LUGR fails, while the approximation computed using LDC has an accuracy of order 10^{-2} (see Figure 7). Keeping in mind that we are using a second order method in space and a first order method in time, in run #2 we take H twice as small and Δt four times as small with respect to run #1. LUGR finally gives meaningful results; yet LDC is a factor 5.6 more accurate than LUGR, costing only twice as much in terms of total number of unknowns per time step, see (23).

4 Conclusions

In this paper we have presented a Local Defect Correction method for time-dependent problems. LDC is an iterative process suitable for efficiently solving problems characterized at each time by a region of high activity that covers a small part of the physical domain. At each time step the problem is first solved on a global coarse grid; the computed coarse grid approximation defines an artificial boundary condition for a local problem, where a more accurate solution can be computed by means of a smaller time step and a smaller grid size. The local solution is then used to improve the first coarse grid approximation through the defect correction. The new coarse grid solution can in turn define new artificial boundary condition for a new local problem. The process can be repeated till convergence, which is in general very fast.

Results of numerical experiments show that LDC can achieve the same accuracy as a uniform grid solver whose grid size and time step coincide with those used in the local problem of the LDC algorithm. At the same time LDC guarantees a lower computational cost than the uniform grid solver.

The LDC method shares with the Local Uniform Grid Refinement technique the advantage of working with uniform grids only at each time step. LUGR differs from LDC

though in that it does not have the defect correction step: LUGR totally relies on the fact that the first coarse grid approximation provides an accurate artificial boundary condition for the local problem. This cannot be guaranteed in general. Results of numerical experiments show that LDC is a more robust technique than LUGR: when the latter fails, LDC can still lead to accurate results.

References

- [1] M. J. H. Anthonissen. *Local defect correction techniques: analysis and application to combustion*. PhD thesis, Eindhoven University of Technology, Eindhoven (Netherlands), 2001.
- [2] M. J. H. Anthonissen, R. M. M. Mattheij, and J. H. M. ten Thije Boonkkamp. Convergence analysis of the local defect correction method for diffusion equations. *Numerische Mathematik*, 95(3):401–425, 2003.
- [3] M. J. H. Anthonissen, B. van ’t Hof, and A. A. Reusken. A finite volume scheme for solving elliptic boundary value problems on composite grids. *Computing*, 61:285–305, 1998.
- [4] B. A. V. Bennett. *Solution-adaptive gridding methods with application to combustion problems*. PhD thesis, Yale University, New Haven, CT, 1997.
- [5] B. A. V. Bennett and M. D. Smooke. Local rectangular refinement with application to axisymmetric laminar flames. *Combustion Theory and Modelling*, 2:221–258, 1998.
- [6] B. A. V. Bennett and M. D. Smooke. Local rectangular refinement with application to nonreacting and reacting fluid flow problems. *Journal of Computational Physics*, 151:684–727, 1999.
- [7] P. J. J. Ferket and A. A. Reusken. Further analysis of the local defect correction method. *Computing*, 56:117–139, 1996.
- [8] P. J. J. Ferket and A. A. Reusken. A finite difference discretization method on composite grids. *Computing*, 56:343–369, 1996.
- [9] J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis. *Adaptive Methods for Partial Differential Equations*. SIAM Publications, Philadelphia, 1989.
- [10] M. Graziadei, R. M. M. Mattheij, and J. H. M. ten Thije Boonkkamp. Local defect correction with slanting grids. *Numerical Methods for Partial Differential Equations*, 20(1):1–17, 2003.
- [11] W. Hackbusch. Local defect correction and domain decomposition techniques. In K. Böhrer and H. J. Stetter, editors, *Defect Correction Methods. Theory and Applications*, *Computing, Suppl. 5*, pages 89–113, Wien, New York, 1984. Springer.
- [12] D. F. Hawken, J. J. Gottlieb, and J. S. Hansen. Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *Journal of Computational Physics*, 95:254–302, 1991.
- [13] K. Miller. Moving finite elements 2. *Journal of Numerical Analysis*, 18:1033–1057, 1981.
- [14] K. Miller and R.N. Miller. Moving finite elements 1. *Journal of Numerical Analysis*, 18:1019–1032, 1981.
- [15] V. Nefedov and R. M. M. Mattheij. Local defect correction with different grid types. *Numerical Methods for Partial Differential Equations*, 18(4):454–468, 2002.
- [16] R. A. Trompert. *Local Uniform Grid refinement for Time-Dependent Partial Differential Equations*. PhD thesis, University of Amsterdam, Amsterdam (Netherlands), 1994.
- [17] J. U. Wappler. *Die lokale Defektkorrekturmethode zur adaptiven Diskretisierung elliptischer Differentialgleichungen mit finiten Elementen*. PhD thesis, Christian-Albrechts-Universität, Kiel (Germany), 1999. In German.
- [18] P. A. Zegelting. *Moving-Grid Methods for Time-Dependent Partial Differential Equations*. PhD thesis, University of Amsterdam, Amsterdam (Netherlands), 1992.