Local defect correction with slanting grids

Citation for published version (APA):

Document status and date:
Published: 01/01/2002

Publisher 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 author's 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.

Take down policy
If you believe that this document breaches copyright please contact us:
openaccess@tue.nl
providing details. We will immediately remove access to the work pending the investigation of your claim.

Download date: 28. Jan. 2019
Local defect correction with slanting grids

M. Graziadei, R.M.M. Mattheij and J.H.M. ten Thije Boonkkamp
Department of Mathematics and Computer Science, Eindhoven University of Technology,
PO Box 513, 5600MB Eindhoven, The Netherlands

Abstract. The local defect correction (LDC) method is used to solve a convection-diffusion-reaction problem that contains a high activity region in a relatively small part of the domain. The improvement of the solution on a coarse grid is obtained by introducing a correction term computed from a local fine grid solution. This paper studies problems where the high activity region is covered with a rectangular fine grid not aligned with the axes of the global domain. This study shows that the resulting method is less expensive than both a uniform refinement and tensor product grid method.

1 Introduction

Solutions of boundary value problems (BVPs) are often characterised by the presence of regions where gradients are quite large compared to those in the rest of the domain, where the solution can be considered relatively smooth. When solving such problems numerically, this solution behaviour requires a much finer grid in these high activity regions than in regions where the solution is fairly smooth. This is the case, for instance, for the equations that describe laminar flames [1]: most of the activity is concentrated in the 'flame front', a narrow area where the temperature rises steeply and chemical reactions take place. Finite element methods can quite easily cope with this, but for finite difference or finite volume methods the unstructured grids, that are a consequence of this non-uniform behaviour, are much more complicated to handle. For the latter type of methods, one may still obtain a simple data structure by considering tensor product grids. These should be such that they have higher density in the high activity region. However, this most likely leads to too many grid points in other parts of the domain and thus to low efficiency. In particular this is the case when the front is not aligned with either one of the coordinate axes. This paper focuses, in particular, on such type of problems, say where we have a slanting front in a rectangular domain. Similar situations may arise in more general domains. This setting, however, is mainly taken to demonstrate the principal idea: constructing an efficient algorithm based on the use of tensor product grids on rectangular domains. Roughly speaking this goes as follows. First an approximation on a coarse grid in the entire domain is computed. This is then used to define a boundary value problem on a subdomain where a different grid is employed. The thus found solution on the latter subdomain is used to define a defect on the global domain, which, in turn, induces an updated global domain problem. This so-called local defect correction (LDC) method is used in an iterative way. It converges very fast: typically one iteration suffices.

LDC was first introduced in [2]. The convergence behaviour of the LDC method has been studied in [3] and [4] using, as a model problem, the Poisson equation discretised with the finite difference method. In [5] one can find the method combined with the finite volume discretisation. It is shown how this can
be done maintaining the conservation property. In [4] the LDC method is combined with multi-level adaptive grid refinement, with an application to laminar flame simulation. Finally in [6] and [7] it is shown how different kinds of grids, namely a global rectangular and a local cylindrical, can be coupled by LDC and applied to the study of viscous flows.

The problem investigated in this paper consists of a 2D convection-diffusion-reaction equation with a source term which is large in a small slanting strip of the domain and virtually zero outside this strip. This means that the solution contains a high gradient region which runs through the domain and that is not aligned with either the x- or the y-axis. In order to apply LDC just across this area, two choices are possible. One is to use a few overlapping rectangular fine grids, aligned with the coordinate axes and arranged in a staircase shape. The drawback of this method is that redundant grid points are needed. This makes the procedure unnecessarily expensive in terms of memory requirements and CPU time. Moreover, since more than one local fine grid must be used, the convergence of the LDC method might slow down; see [4]. The second possibility is to cover the high activity region with a single fine grid, which is slanted with respect to the x-axis of the global domain. This is the approach followed in this paper. The advantages of the method are that only one single fine grid is needed and the number of fine grid points is considerably reduced. The only, though small, disadvantage is that the fine grid is not aligned with the coarse grid: this fact makes it more difficult to implement.

This paper is set up as follows. In Section 2 we give a brief description of LDC in a general setting. Section 3 contains the elaboration of the LDC method for our specific convection-diffusion-reaction problem. Section 4 presents numerical experiments and results. In Section 5 we compare the efficiency of the LDC and the tensor-product grid methods in a complexity analysis. Section 6 contains conclusions.

2 Formulation of the LDC method

In this section we give a brief description of the LDC method.

Let us consider the BVP for \( u(x) \):

\[
\begin{align*}
\mathcal{L} u &= f, & x \in \Omega, \\
u &= g, & x \in \partial \Omega,
\end{align*}
\]

where \( \Omega \subset \mathbb{R}^2 \) is a simply connected domain, \( \mathcal{L} \) is a linear elliptic operator, \( f \) is a source term and \( g \) is the value of \( u \) on the boundary \( \partial \Omega \).

In order to solve (1), we first choose a coarse grid size \( H \): the corresponding uniform Cartesian grid will be indicated by \( \Omega_H \). The discretised equation reads

\[
\mathcal{L}_H u_H = f_H,
\]

where the right-hand side contains both the source term and the contribution of the boundary conditions. For the moment it is not necessary to specify the operators any further. From (2) we compute \( u^0_H \).

Suppose now that \( u \) changes very rapidly in the subdomain \( \Omega' \). In this high activity subdomain the grid size \( H \) will most likely be too large to capture the behaviour of \( u \), so we need to refine there. Let us
consider the local discretisation error of the coarse grid approximation, defined as
\[ d_H := \mathcal{L}_H u - f_H, \] (3)
i.e. the residual found by substituting the continuous solution of (1) into the discrete scheme (2). If we would know \( d_H \), we could add it to the right-hand side of (2) as a correction term, to obtain the solution on the coarse grid. The continuous solution \( u \) in (3) is not known in general, so we try to get a reasonable estimate of \( d_H \), instead.

To this end, we formulate a new discrete problem in \( \Omega' \) by covering it with a grid \( \Omega'_h \) of size \( h \) say, with \( h < H \). Because boundary conditions on \( \partial \Omega' \) are needed, we split it in two parts: the interface \( \Gamma' \) between \( \Omega \) and \( \Omega' \) and \( \partial \Omega' \cap \partial \Omega \). Note that this last subset can also be empty. Let us now introduce some additional notation. The set of fine grid points on \( \partial \Omega' \) is denoted by \( \partial \Omega'_h \), so \( \partial \Omega'_h = \partial \Omega' \cap \partial \Omega \). On \( \partial \Omega' \cap \partial \Omega \) we already know that \( u = g \). To specify boundary conditions on \( \partial \Omega'_h \) we need an interpolation operator
\[ \mathcal{P}^{h,H} : F(\Omega_H) \to F(\Gamma_h), \]
where \( \mathcal{P}^{h,H} \) maps grid functions on \( \Omega_H \) onto grid functions on \( \Gamma_h \). From this it follows that the boundary conditions for the discrete BVP on the fine grid read
\[ u_h = g_h \quad \text{if } x \in \partial \Omega'_h \cap \partial \Omega, \] (4a)
\[ u_h = \mathcal{P}^{h,H} u_H \quad \text{if } x \in \Gamma_h, \] (4b)
where \( g_h \) is the boundary value function \( g \) evaluated at points of the set \( \partial \Omega'_h \cap \partial \Omega \). With the notation introduced above, we can finally formulate the problem on \( \Omega'_h \) as
\[ \mathcal{L}_h u_h = f_h, \] (5)
and solve it to find \( u_h^0 \). The contribution of all boundary conditions and the source term are included in \( f_h \). The discrete operator \( \mathcal{L}_h \) can be different from \( \mathcal{L}_H \).

Now we introduce \( \Omega''_H := \Omega_H \cap \Omega' \) and the restriction operator
\[ r^{H,h} : F(\Omega'_h) \to F(\Omega''_H), \]
mapping grid functions on \( \Omega'_h \) onto grid functions on \( \Omega''_H \). Furthermore, we define the grid function \( w^0_H \in F(\Omega_H) \) as
\[ w^0_H(x) := \begin{cases} u^0_H(x) & x \in \Omega_H \setminus \Omega'_H, \\ r^{H,h} u^0_H(x) & x \in \Omega'_H. \end{cases} \] (6)
The grid function \( w^0_H(x) \) can be interpreted as an updated approximation of the solution on the global coarse grid \( \Omega_H \).

Let \( S_H(x) \) be the set of coarse grid points belonging to the stencil of \( x \in \Omega_H \); then we define the subset \( \Omega''_H \subset \Omega'_H \) by the relation
\[ x \in \Omega''_H \iff S_H(x) \subset \Omega_H. \] (7)
Exploiting the solution computed in $\Omega'_h$, we give the following approximation of $d_H$

$$d_H = d^0_H := \chi_{\Omega''_H}(L_H w^0_H - f_H).$$

where $\chi_{\Omega''_H}$, the characteristic function of $\Omega''_H$, is defined as

$$\chi_{\Omega''_H}(x) := \begin{cases} 1 & \text{if } x \in \Omega''_H, \\ 0 & \text{if } x \notin \Omega''_H. \end{cases}$$

Note that $d^0_H$ will be computed only at the points belonging to the subset $\Omega''_H$ because their stencil is completely contained in $\Omega'_H$.

Once $d^0_H$ has been calculated, it is possible to add it to the right-hand side of (2), resulting in the equation

$$L_H u^1_H = f_H + d^0_H.$$  

By solving (9), we get a new approximation $u^1_H$. This is expected to be more accurate than $u^0_H$, as has been mentioned before. Again $u^1_H$ will be used to update the interface conditions for the fine grid problem, giving rise to the following algorithm.

**LDC Algorithm**

- **Initialization**
  - Compute $u^0_H$ from the basic coarse grid problem (2);
  - Define the fine grid boundary value problem, interpolating the boundary conditions on $\Gamma_h$;
  - Compute $u^0_h$ from the fine grid BVP (5).

- **Iteration $i = 1, 2, ...$**
  - Compute the grid functions $w^{i-1}_H$ using (6);
  - Estimate the local discretisation error $d^{i-1}_H$ using (8);
  - Solve the new coarse grid problem
    $$L_H u^i_H = f_H + d^{i-1}_H;$$
  - Define the new BVP on the fine grid, interpolating boundary conditions on $\Gamma_h$;
  - Compute $u^i_h$ from the fine grid BVP (5).

The final numerical solution of BVP (1) will be the composite grid approximation, defined as

$$u^i_{H,h}(x) := \begin{cases} u^i_H(x) & x \in \Omega_H \setminus \Omega'_H, \\ u^i_h(x) & x \in \Omega'_H. \end{cases}$$

i.e. $u^i_{H,h}$ is the fine grid solution inside $\Omega'$ and the coarse grid solution outside $\Omega'$.  


3 LDC applied to slanting grids

In this section we will focus on the application of LDC to a sample problem having a solution characterised by a gradient across a straight line in the domain, which is not aligned with one of the coordinate axes. This setting is quite general. In fact, it is representative of a large class of real physical phenomena having the same behaviour: a high activity region concentrated in a narrow strip along a line. As will become clear later on, it is convenient to track the high activity region with a local fine grid with one of its axes parallel to the gradient of the solution.

Consider the following two-dimensional convection-diffusion-reaction problem (see [1])

\[-\nabla^2 u + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f(x, y) := \frac{2\beta^2(a^2 + b^2) \tanh[\beta(ax + by - c)] - \beta(a + b)}{\cosh^2[\beta(ax + by - c)]}, \quad x \in \Omega := (0, l_1) \times (0, l_2),\]

(11a)

\[u = g(x, y) := 1 - \tanh[\beta(ax + by - c)] \quad x \in \partial \Omega,\]

(11b)

so that the exact solution of (11) is given by

\[u(x) = 1 - \tanh[\beta(ax + by - c)].\]

(12)

The function \(u(x)\) is approximately equal to 2 in the left part of the domain, its value decreases steeply over the line \(ax + by = c\) and becomes 0 in the right part of the domain; see Figure 1. The parameter \(\beta\) determines the steepness of \(u(x)\) in the vicinity of the line \(ax + by = c\). Our purpose is to apply LDC in combination with a slanting local grid enclosing that line.

Figure 1: Function \(u(x)\) in the domain \(\Omega = (0, 1) \times (0, 4)\), with \(\beta = 5, a = 4, b = 2, c = 3\).

To compute the numerical solution of BVP (11) using LDC, we need to specify the following:
- The coarse grid and the discretisation of the problem on it;
- The fine grid and the discretisation of the problem on it;
- An interpolation operator to compute the interface conditions;
- A restriction operator.

Furthermore the use of a slanting grid poses a problem that can easily be understood looking at Figure 6. In fact, it turns out that too small a fine grid is not adequate because it leaves out the corners of \( \Omega \). On the other hand, it is possible to cover the whole steep gradient area with a longer rectangular fine grid; however, we then have to accept that some grid points will fall outside \( \Omega \).

Our first step will be the discretisation of (11a) in \( \Omega \), in order to get system (2). The finite volume method (FVM) on a cell centered grid leads to a second order accurate scheme, that for the general coarse grid point \( C \), reads

\[
- \frac{u_E - 2u_C + u_W}{H^2} - \frac{u_N - 2u_C + u_S}{H^2} + \frac{u_E - u_W}{2H} + \frac{u_N - u_S}{2H} = f_C.
\]

The fluxes are approximated by central differences. The subscripts indicate the locations where terms are evaluated and \( N, S, W \) and \( E \) are the northern, southern, western and eastern neighbouring points of the central point \( C \), respectively.

Let us now investigate how to impose the Dirichlet boundary conditions (11b). The choice of a cell centered grid is such that it is not possible to assign a value on the boundary of the domain, because it contains no grid points. This problem can be solved introducing virtual grid points, as in Figure 2. The Dirichlet boundary condition in this example is approximated by

\[
\frac{1}{2}(u_W + u_C) = g(x_w),
\]

from which \( u_W \) can be expressed as a function of \( u_C \) and \( g(x_w) \). Using (13) in the internal grid points of \( \Omega_H \), we obtain a pentadiagonal algebraic system that can easily be solved.

As the next step we will define the local problem (5). Obviously, a tensor product fine grid would cover a substantial part of the entire domain, resulting in almost a similar number of grid points as would be obtained from application of this type of grid on the entire domain. Instead, we solve the local problem (5)
using a slanting fine grid and thus trying to reduce the number of grid points needed to give a good representation of the unknown function. The slanting domain $\Omega'$ is defined such it has to contain the line $ax + by = c$ completely; furthermore some redundant points are needed to make the data structure simpler (see Section 4).

For this, we have to carry out a coordinate transformation. Suppose that the $x'$-axis of the frame of reference $(x', y')$ is rotated over an angle $\alpha$ with respect to the $x$-axis of the frame of reference $(x, y)$ and that its origin coincides with the point having coordinates $(n_1, n_2)$. The transformation that relates $(x, y)$ and $(x', y')$ is clearly given by

$$
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix}
= R^T \left[ \begin{pmatrix}
  x \\
  y
\end{pmatrix} - \begin{pmatrix}
  n_1 \\
  n_2
\end{pmatrix} \right],
$$

(14)

with $R$ the orthogonal matrix

$$
R = \begin{pmatrix}
  \cos \alpha & -\sin \alpha \\
  \sin \alpha & \cos \alpha
\end{pmatrix}.
$$

From now on we will use the single quotation mark to indicate a vector in $(x', y')$. Let us introduce the chain rule for scalar and for vectorial quantities

$$
\nabla u = R \nabla' u, 
$$

(15a)

$$
\nabla \cdot \mathbf{v} = (R \nabla') \cdot \mathbf{v}.
$$

(15b)

Applying relations (15) to the Laplacian on the right-hand side of (11a), we get

$$
\nabla^2 u = \nabla \cdot (\nabla u) = (R \nabla') \cdot (R \nabla'u) = \nabla' \cdot (R^T R \nabla'u) = \nabla' \cdot (\nabla' u) = (\nabla')^2 u,
$$

(16)

where $(\nabla')^2 u = \frac{\partial^2 u}{\partial x'^2} + \frac{\partial^2 u}{\partial y'^2}$. The convection terms in (11a) can be written as

$$
\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = (\mathbf{a} \cdot \nabla)u, \quad \text{with } \mathbf{a} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.
$$

(17)

Expressing (17) in the new frame of reference, we get

$$
(\mathbf{a} \cdot \nabla')u = (\mathbf{a} \cdot R \nabla')u = (R^T \mathbf{a} \cdot \nabla')u = (\mathbf{b} \cdot \nabla')u, \quad \text{with } \mathbf{b} = \begin{pmatrix} \cos \alpha + \sin \alpha \\ \cos \alpha - \sin \alpha \end{pmatrix}.
$$

(18)

Substitution of (16) and (18) into (11a) yields

$$
-(\nabla')^2 u + (\mathbf{b} \cdot \nabla')u = f.
$$

(19)

The method used to discretise the global problem on the coarse grid will be applied to the discretisation of (19) on the fine grid as well. We will choose a step size equal to $h$ along the $x'$-axis and a step size equal to $mh$ along the $y'$-axis, with $m \geq 1$. Indeed, since $\partial u/\partial y' = 0$, it is possible to use a coarser grid size along the $y'$-axis without spoiling the accuracy of the solution. The discretisation of (19) yields

$$
\begin{align*}
-\frac{u_E - 2u_C + u_W}{h^2} - \frac{u_N - 2u_C + u_S}{(mh)^2} + (\cos \alpha + \sin \alpha)\frac{u_E - u_W}{2h} + (\cos \alpha - \sin \alpha)\frac{u_N - u_S}{2mh} &= f_C,
\end{align*}
$$

(20)
where C is a general fine grid point and W, E, N and S are its neighbours. In order to solve (20), we must specify the angle \( \alpha \). We note that the high activity region, see (12), is a straight line, inclined at an angle equal to \( \arctan\left(-\frac{a}{b}\right) \). Therefore the most natural choice is to put \( \alpha = \arctan\left(\frac{b}{a}\right) \): in this way the \( x' \)-axis will be parallel to the gradient of \( u \). Since we want the fine grid not to be aligned with the coarse grid, we consider \( a \neq 0 \) and \( b \neq 0 \). Furthermore the point \((n_1, n_2)\) will correspond to the intersection of the line \( ax + by = c \) and the \( x \)-axis. In Section 4 we will carry out numerical experiments using two different configurations: one with \( \alpha = \frac{\pi}{4} \), and another one with \( \alpha = \arctan 2 \).

One of the most important steps of the LDC Algorithm is the interpolation of the interface conditions on \( \Gamma_h \): this will be done using a bi-linear interpolation operator. This is a reasonable choice since it has the same order of accuracy as the discretisation operator used in the global and the local problems. This is confirmed by the numerical experiments of next section. Suppose that the interface condition is needed at a point \( V' \) on \( \Gamma \). Then \( \phi^{h,H} \) determines \( u_{V'} \) by interpolation from the four nearest neighbours on the coarse grid; see Figure 3. For a more general discussion on this topic see [3].

![Figure 3](image-url)

Figure 3: In the boundary point \( V' \) the interface condition is interpolated from the coarse grid points A, B, C and D. The slanting line represents the interface between \( \Omega' \) and \( \Omega \).

Making use of the slanting fine grid, as depicted in Figure 6, presents an additional difficulty in that there are some points belonging to \( \Omega'_h \) that do not belong to the computational domain, as noted before. As a consequence, \( \Omega'_h \nsubseteq \Omega \). To overcome this problem we note again that the solution at a point \( C \) is influenced by the four neighbouring points \( N, S, W \) and \( E \), only. This means that we can divide the fine grid points that fall outside \( \Omega \) in two groups: the points which affect the solution of the fine grid points inside \( \Omega \) and the points that do not. To the second group (points marked by ‘⊙’ in Figure 5), an arbitrary value of the grid function can be assigned because they will have no relation with the grid function in the interior of the computational domain. The points belonging to the first group (marked by ‘∗’ in Figure 5), are linked to the points inside the computational domain (marked by ‘×’), using a linear interpolation of the boundary value function \( g \). In fact, we use linear extrapolation to determine the \( u \)-values in ‘∗’-points from the \( u \)-values in the ‘×’-points having the same \( x' \)-coordinates and from
the (known) boundary values on $\partial \Omega$. Thus, we get an algebraic equation connecting the values of the unknown function in such couple of consecutive grid points. The matrix that we obtain has a nice banded pentadiagonal structure; see Figure 4.

![Figure 4: Sparsity pattern (lower corner) of the matrix used to solve the local BVP.](image)

Once the system (20) has been solved, the grid functions (6) can be calculated. To do so, the restriction operator $r^{H,h}$ has to be specified: also in this case we will choose bi-linear interpolation because it is not computationally expensive and, at the same time, it guarantees enough accuracy of the numerical results. Therefore, the operator $r^{H,h}$ computes the function value $u(x)$ with $x \in \Omega'_H$ from the function values of the four nearest fine grid points. The subsequent steps are straightforward: once we know $w_H$ we can obtain, $d_H$ to perform the last three steps of the LDC Algorithm.

In Section 2 the domain $\Omega'_H$ was introduced: in Figure 6 its points are marked by '•', while the grid points belonging to $\Omega'_H \setminus \Omega''_H$ are marked by '∗'. They are defined in (7) noticing that the central difference discretisation employs a five-point stencil.

The following features are interesting to note: under certain conditions we can adapt the width of the local domain $\Omega'$ to the steepness of the solution and thus we can make the number of fine grid points along the $x'$-axis independent of $\beta$. In the frame of reference $(x',y')$, we have $u(x',y') = 1 - \tanh(\beta s x')$, with $s = \sqrt{a^2 + b^2}$. We start noticing that the longitudinal boundaries of the high activity area can be determined by requiring the function $u$ to differ from its asymptotic value by more than a certain tolerance, say $\epsilon$. Because this value can be approximated by 0, when $ax + by > c$, and by 2, when $ax + by < c$, then $(x', y') \in \Omega'$ if $\epsilon \leq u(x', y') \leq 2 - \epsilon$. Since $\text{Artanh}(z) = \frac{1}{2} \ln \left( \frac{1 + z}{1 - z} \right)$, this translates into

$$-\frac{1}{2s\beta} \ln \left( \frac{2 - \epsilon}{\epsilon} \right) \leq x' \leq \frac{1}{2s\beta} \ln \left( \frac{2 - \epsilon}{\epsilon} \right).$$

The domain $\Omega'$ is symmetric with respect to $x'$ and its width goes to zero when $\beta$ increases.
Figure 5: The points marked by a circle can assume whatever value; the points outside Ω marked by a star are interpolated from the points internal to Ω marked by x and from the boundary values.

Figure 6: Slanting fine grid.

If we consider the scaled coordinate

\[ x'' := \beta x', \quad (22a) \]
\[ y'' := \beta y', \quad (22b) \]

the inequality (21) transforms into

\[ -\frac{1}{2s} \ln \left( \frac{2 - \epsilon}{\epsilon} \right) \leq x'' \leq \frac{1}{2s} \ln \left( \frac{2 - \epsilon}{\epsilon} \right). \quad (23) \]
Scaling equation (19) by using the coordinate transformation (22), we get

\[-(\nabla'')^2 u + \frac{1}{\beta} (b \cdot \nabla'') u = f_1(x'', y'') + \frac{1}{\beta} f_2(x'', y''), \tag{24}\]

where the double quotation mark on the differential operators means derivative with respect to the variables \(x''\) and \(y''\). Furthermore \(u(x'', y'')\), unlike \(u(x, y)\) in (19), does not depend explicitly on \(\beta\) anymore. The error that one makes discretising (24) is bounded by the sum of two terms:

\[\left(\frac{1}{12} \left( |\frac{\partial^4 u}{\partial x''^4}| + |\frac{\partial^4 u}{\partial y''^4}| \right) h''^2 \right.\]

\[
\left. + \frac{1}{6\beta} \left( |\frac{\partial^3 u}{\partial x''^3}| + |\frac{\partial^3 u}{\partial y''^3}| \right) h''^2 \right). \]

Since the second term is negligible with respect to the first one, the error bound turns out to be independent of \(\beta\) and proportional to \(h''^2\).

Therefore, if \(\beta\) varies, the domain \(\Omega'\) can be stretched, according to (22a), while the \((x'', y'')\)-domain is kept constant, as well as the number of grid points along the \(x'\)-axis.

The elements that have been used to show this are the fact that \(u\) is invertible, i.e. it is monotonously increasing or decreasing with respect to \(x'\), and the fact that \(u = u(\beta x')\). Furthermore, we have supposed that the error bound of the convective term in (24) is small, as \(\beta\) increases, with respect to the error bound of the diffusion term. We could thus likely extend these results to other functions that satisfy the same requirements.

4 Numerical results

In this section we consider two examples.

4.1 Example 1

Consider a domain \(\Omega = (0, 1) \times (0, 1)\) and a value \(\beta = 20\). The choice \(a = b = c = 1\) implies \(\alpha = \frac{\pi}{4}\); furthermore the vector \(b\) becomes

\[b = \left( \sqrt{2} \right).\]

We choose \(m = 1\), see (20). Since \(\alpha = \frac{\pi}{4}\), it is possible to arrange the fine grid in such a way that each point of \(\Omega'H\) coincides with a point of \(\Omega'h\). This happens when the line \(ax + by = c\) coincides with the center line of the grid and \(h = \frac{H}{k\sqrt{2}}\), with \(k\) an integer. In this case the restriction operator simplifies to

\[r^{H,h} u_h(x, y) := u_h(x, y), \quad \text{for } x \in \Omega'H. \tag{26}\]

Table 1 shows the errors \(||u^* - u_{H,h}||_\infty\) of the composite grid solution, and the ratio between the errors, for several values of \(H\), \(u^*\) being the exact solution of (11). It turns out that convergence is achieved already after one iteration of the LDC algorithm. The width of the fine grid is kept almost constant and equal to 0.39. Even if interface conditions have been interpolated using a bi-linear operator, the solution approximation is asymptotically second order accurate with respect to \(h\), provided that \(H\) is small.
enough. Note also that, for \( h \) fixed and \( H \) decreasing, the error is approximately constant. In fact, in this situation, the effect of the discretisation errors on the coarse grid are negligible compared to the local discretisation errors on the fine grid.

\[
H = 10^{-1} = 0.1, \quad H = 10^{-2} = 0.01, \quad H = 10^{-3} = 0.001
\]

Table 1: Composite grid approximation error \( ||u^* - u_{H,h}||_\infty \).

<table>
<thead>
<tr>
<th>( h )</th>
<th>( H = 10^{-1} ) ratio</th>
<th>( H = 20^{-1} ) ratio</th>
<th>( H = 40^{-1} ) ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H/2\sqrt{2} )</td>
<td>4.68 ( \cdot ) 10(^{-2} )</td>
<td>1.72 ( \cdot ) 10(^{-2} )</td>
<td>4.10 ( \cdot ) 10(^{-3} )</td>
</tr>
<tr>
<td>( H/4\sqrt{2} )</td>
<td>1.99 ( \cdot ) 10(^{-2} )</td>
<td>4.20 ( \cdot ) 10(^{-3} )</td>
<td>1.10 ( \cdot ) 10(^{-3} )</td>
</tr>
<tr>
<td>( H/8\sqrt{2} )</td>
<td>6.90 ( \cdot ) 10(^{-3} )</td>
<td>4.10 ( \cdot ) 10(^{-3} )</td>
<td>2.600 ( \cdot ) 10(^{-4} )</td>
</tr>
</tbody>
</table>

The results reported in Table 2 show that, when \( \beta \) varies, the number of fine grid points along the \( x' \)-axis can be kept constant without affecting the error of the fine grid solution; see Section 3. The width of the local domain is computed by using (21), with \( \epsilon \) approximately equal to 10\(^{-5} \). It turns out that the error is almost constant, for a fixed \( H \), when different values of \( \beta \) are considered and when the local domain is correspondingly stretched. This approach fails for large values of \( H \) and \( \beta \), because, in this case, the domain \( \Omega'_{H} \) does not contain enough points to apply the correction.

Table 2: Error \( ||u^* - u_h||_\infty \) for different values of \( H \) and \( \beta \); \( h = \frac{H}{2\sqrt{2}} \) for \( \beta = 10 \), \( h = \frac{H}{4\sqrt{2}} \) for \( \beta = 20 \), \( h = \frac{H}{8\sqrt{2}} \) for \( \beta = 40 \).

<table>
<thead>
<tr>
<th>( H )</th>
<th>( \beta = 10 )</th>
<th>( \beta = 20 )</th>
<th>( \beta = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10(^{-1} )</td>
<td>1.81 ( \cdot ) 10(^{-2} )</td>
<td>1.99 ( \cdot ) 10(^{-2} )</td>
<td>7.10 ( \cdot ) 10(^{-1} )</td>
</tr>
<tr>
<td>20(^{-1} )</td>
<td>4.30 ( \cdot ) 10(^{-3} )</td>
<td>4.20 ( \cdot ) 10(^{-3} )</td>
<td>6.70 ( \cdot ) 10(^{-3} )</td>
</tr>
<tr>
<td>40(^{-1} )</td>
<td>1.10 ( \cdot ) 10(^{-3} )</td>
<td>1.10 ( \cdot ) 10(^{-3} )</td>
<td>1.10 ( \cdot ) 10(^{-3} )</td>
</tr>
</tbody>
</table>

4.2 Example 2

Consider problem (11) in the domain \( \Omega = (0, 1) \times (0, 4) \), where \( a = 4, b = 2, c = 3 \) and \( \beta = 5 \). Then the exact solution is

\[
u(x, y) = 1 - \tanh[\beta(4x + 2y - 3)].
\]

The high activity region lies across the straight line \( 4x + 2y = 3 \). In Figure 7 we depict the grids that have been used to solve (11), when \( H = 10^{-1} \) and \( h = 20^{-1} \).

Suppose that we want to compute the function \( u \) within an accuracy of 10\(^{-3} \). Taking into account that the discretisation error is approximately bounded by \( \frac{1}{12} \left( \left| \frac{\partial^2 u}{\partial x^2} \right| + \left| \frac{\partial^2 u}{\partial y^2} \right| \right) H^2 \), we can try to give an estimate of this quantity in order to choose a suitable grid size. For \( (x, y) \) \( \in \Omega \) such that \( 4x + 2y < 1.6 \) and
Figure 7: Coarse and slanting fine grid.

4x + 2y > 4.4, we find that $|\frac{\partial^2 u}{\partial x^2}| + |\frac{\partial^2 u}{\partial y^2}| < 4.5$. Hence, we may define $\Omega'$ as the region bounded by the lines $4x + 2y = 1.6$, $4x + 2y = 4.4$ and by the two lines $x - 2y = -4.4$ and $x - 2y = 1$. If we choose $H = 10^{-1}$, the discretisation error in $\Omega \setminus \Omega'$ will be of the order of $10^{-3}$. Actually the dimensions of the refined area are slightly different from the one defined above, in order to make it a multiple of the chosen fine grid step size. The result of the computations performed by LDC are presented in Table 3. This gives the error of the composite grid solution when $H = 10^{-1}$. Table 4 shows the results obtained from solving the same problem with a uniformly refined grid. By comparing these two tables it is clear that the LDC algorithm is much more efficient with respect to the memory usage, because it gives a comparable accuracy with substantial less grid points.

Table 5 gives the error of the composite grid solution, for several values of $H$ and $h$. It appears that, once the asymptotic behaviour has been reached, the solution is second order accurate with respect to $h$. The error for $h$ fixed and $H$ decreasing is again approximately constant; see Example 1.

Until now we have uniformly refined the high activity region: this assumption can be dropped if one realizes that the gradient of the continuous solution $u$ is parallel to the $x'$-axis. This implies that in (20) a value $m > 1$ can be chosen. As a consequence, the number of fine grid points will be strongly reduced, while the accuracy of the approximated solution will remain comparable. This is shown in Table 6, where the global errors of the approximation and the total number of grid points are given. The comparison of Table 6 and Table 3 shows that the solution on the non-uniform fine grid needs much less grid points, while having roughly the same accuracy.
Table 3: Error of the composite grid solution. The last column shows the number of grid points. The global domain is discretised using $400$ points ($H^{-1} = 10$).

| $h$  | $||u^* - u_{H,h}||_\infty$ | Grid points |
|------|----------------------------|-------------|
| $20^{-1}$ | $6.72 \cdot 10^{-2}$ | $400 + 517$ |
| $40^{-1}$ | $2.85 \cdot 10^{-2}$ | $400 + 2162$ |
| $80^{-1}$ | $7.7 \cdot 10^{-3}$ | $400 + 8883$ |

Table 4: Global error obtained with uniform refinement. The last column shows the number of the grid points.

| $H$  | $||u^* - u_H||_\infty$ | Grid points |
|------|--------------------------|-------------|
| $10^{-1}$ | $4.839 \cdot 10^{-1}$ | $400$ |
| $20^{-1}$ | $8.06 \cdot 10^{-2}$ | $1600$ |
| $40^{-1}$ | $2.22 \cdot 10^{-2}$ | $6400$ |
| $80^{-1}$ | $5.70 \cdot 10^{-3}$ | $25600$ |

Table 5: Composite grid approximation error $||u^* - u_{H,h}||_\infty$ for different values of $H$ and $h$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$H = 10^{-1}$</th>
<th>ratio</th>
<th>$H = 20^{-1}$</th>
<th>ratio</th>
<th>$H = 40^{-1}$</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H/2$</td>
<td>$6.72 \cdot 10^{-2}$</td>
<td></td>
<td>$2.95 \cdot 10^{-2}$</td>
<td></td>
<td>$7.00 \cdot 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$H/4$</td>
<td>$2.85 \cdot 10^{-2}$</td>
<td>2.36</td>
<td>7.0 $\cdot 10^{-3}$</td>
<td>4.2</td>
<td>1.70 $\cdot 10^{-3}$</td>
<td>4.1</td>
</tr>
<tr>
<td>$H/8$</td>
<td>7.7 $\cdot 10^{-3}$</td>
<td>3.6</td>
<td>1.70 $\cdot 10^{-3}$</td>
<td>4.1</td>
<td>4.32 $\cdot 10^{-4}$</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Table 6: Error of the composite grid solution and number of grid points as a function of $m$. The coarse mesh size $H$ is constant and equal to $10^{-1}$.

| $h$ | $m$ | $||u^* - u_{H,h}||_\infty$ | Grid points |
|-----|-----|--------------------------|-------------|
| $20^{-1}$ | 2 | $6.71 \cdot 10^{-2}$ | $400 + 231$ |
| $40^{-1}$ | 4 | $2.73 \cdot 10^{-2}$ | $400 + 483$ |
| $80^{-1}$ | 4 | $8.20 \cdot 10^{-3}$ | $400 + 1845$ |
5 Complexity of the method

A way to assess the efficiency of a numerical method is estimating its complexity. To this end, we will compare LDC to the tensor-product grid refinement. Suppose we have a rectangular domain $\Omega$, as in Figure 8, covered with a uniform grid. Let $N$ and $AN$, with $A$ positive, be the number of coarse grid points along the $x$-axis and the $y$-axis, respectively.

![Figure 8: Domain $\Omega$ with the high activity region.](image)

The high activity region, characterised by an angle $\delta = \frac{\pi}{2} - \alpha$ (with respect to the $x$-axis), is covered by $\frac{M \times BM}{m}$ fine grid points, with $B > 1$. The number of points used by LDC will then be

$$N_{LDC} := AN^2 + \frac{BM^2}{m},$$

where $m$ is the ratio between the step size along the $y'$-axis and the step size along the $x'$-axis; see Section 3.

For the tensor-product grid refinement, we cover the part of $\Omega$ containing the high activity area with a finer grid. We need approximately $BM \cos \delta$ grid points on the $x$-axis and $BM \sin \delta$ grid points on the $y$-axis, in order to get a step-size comparable with the step-size used in LDC. Thus, we need roughly $N_{tensor}$ grid points, with

$$N_{tensor} := (BM \cos \delta + N)(BM \sin \delta + AN).$$

Note that (29) is valid when the ordinate of the upper right corner of the fine grid is smaller than $l_2$ (height of the global domain).

Let us define the gain $G$ as the ratio between $N_{tensor}$ and $N_{LDC}$. Assuming $N = \gamma M$, $G$ is found to be

$$G = \frac{(B \cos \delta + \gamma)(B \sin \delta + \gamma A)}{Ay^2 + \frac{B}{m}}.$$
We now assess $G$ by letting $\gamma$ vary. First suppose that $\gamma \ll 1$: in this case the number of coarse grid points is far less than the number of fine grid points. Such a situation can occur if one reduces the fine grid size $h$, while keeping constant all geometrical parameters and the width of the local grid. Then the gain $G$ becomes

$$G = \frac{mB \sin(2\delta)}{2},$$

which is apparently independent of $A$.

Suppose now that $\gamma \gg 1$. This means that the number of fine grid points is much less than the number of coarse grid points; for instance when the high activity area takes up just a small corner of the global domain. Then we obtain that $G$ does not depend on $\delta$

$$G = \frac{A\gamma^2}{A\gamma^2} = 1.$$

(32)

In the latter situation it is not worthwhile to use LDC because a comparable complexity can be achieved with the tensor product grid refinement.

If $\gamma = 1$, we find

$$G = \frac{(B \cos \delta + 1)(B \sin \delta + A)}{A + \frac{B}{m}}.$$

(33)

In Figure 9 the gain $G$ has been plotted against $\gamma$. The geometrical parameters correspond to the second example presented in Section 4: only the dimensions of the global domain are varied to parameterise the curves. The coarse grid size is equal to $10^{-1}$ and the fine grid size is $h = l\gamma$, where $l$ is the width of the local domain. Of course we restrict the value of $\gamma$ such that $mh \leq H$. In the previous section we have shown that, if a bigger grid size along the $y'$-axis is chosen, the total number of grid points reduces considerably. Figure 9 gives an idea of the increase of the gain $G$ when a factor $m > 1$ is considered. The first set of curves (dotted lines) are computed with $m = 1$ and the second set (solid lines) with $m = 4$. The asymptotic behaviour of the gain when $\gamma$ increases is clearly shown: once $\delta$ has been fixed, the curves tend to 1 when $\gamma$ is much larger than one, see (32).

6 Conclusions

In this paper an application of the LDC method with a slanting fine grid has been presented. A convection-diffusion-reaction equation has been discretised using the finite volume method, both in the global and in the local domain. Fluxes have been approximated by finite differences, producing a second order accurate scheme. The interface conditions between the fine and the coarse grids have been computed using a bi-linear interpolation operator. The results show that the second order accuracy is preserved. The fact that the axes of the global and the local domain are not aligned, implies that some points belonging to the fine grid fall outside the global domain: when this happens a linear interpolation is used to extend the function and to compute the solution outside the boundaries. Furthermore the memory use of the LDC
Figure 9: Dotted lines have been drawn with $m = 1$, solid lines with $m = 4$. Both sets are parameterised with respect to the dimensions of the domain $\Omega$, that have been assumed equal to $(0, 1) \times (0, 4)$, $(0, 2) \times (0, 8)$, $(0, 4) \times (0, 16)$ and $(0, 8) \times (0, 32)$, respectively. The bigger $\Omega$ is, the higher is the gain-curve. On the first dotted line the values of the gain relevant to the second example are reported by circles. They represent the ratio between the number of points that would have been used with the tensor product grid refinement and the number of points reported in Table 3.

method and of the tensor product grid refinement have been compared via a complexity analysis: the gain function $G$ is used to highlight the asymptotic behaviour of LDC when the geometrical parameters of the problem are varied.
References


