Local defect correction with curvilinear grids : algorithm and application to laminar flames

Local defect correction with curvilinear grids: algorithm and application to laminar flames

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 convection-diffusion-reaction problems that are characterised by the presence of high activity regions in a relatively small part of the domain. The improvement of the solution on the 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 curvilinear orthogonal grids and shows how these can be beneficial in reducing the complexity of the LDC method. A convection-diffusion-reaction model problem is used to asses its properties. Also the solution of a laminar flame is computed.

Keywords: local defect correction (LDC), curvilinear orthogonal grid, convection-diffusion-reaction equation, thermo-diffusive model for laminar flames.

1 Introduction

Many boundary value problems (BVPs) have solutions which exhibit very rapid variations in relatively small parts of the spatial domain. Moreover, these so-called *high activity* regions are often of irregular shape. Examples of such BVPs are mathematical models of laminar flames, shock waves or semiconductor devices. Their numerical solution requires a grid that is very fine in the vicinity of the high activity region(s). The finite element method, making use of elements of arbitrary shape and size, offers a solution to that, but it produces unstructured grids and complicated data structures. In order to obtain a simpler data structure, the finite volume or finite difference methods in combination with tensor product grids could be a good alternative, but an irregular shape of the high activity area implies often too many grid points where they are not supposed to be. The Local Defect Correction (LDC) method is a better choice, since it combines a coarse global and a fine local grid(s), providing also a way to exchange information between them. It was first introduced in [1]. Theoretical studies and applications to several fields can be found in [2–8].

Because of the irregular shape of the high activity region(s), the boundaries of the refinement area may not consist of straight lines. Our goal is to show that in this case a local BVP can be efficiently solved using curvilinear fine grids that follow tightly the shape of the solution. This method shows only one drawback: when transforming the local BVP from a cartesian to a curvilinear coordinate system some more terms appear in the PDEs. However, the problem can be partially overcome by using an orthogonal fine grid. This requirement also turns out to have some more advantages: if one set of grid lines is aligned with the solution iso-contours, we manage to get a substantial reduction in the number of fine grid points. It is easy to understand how the fine grid generation assumes, in this context, a crucial role for the feasibility of the method.

This paper is organised as follows. Section 2 gives a brief overview of the LDC method and Section 3 introduces the steps to follow in order to combine that method with curvilinear grids. In Section 4 the procedure to generate a suitable orthogonal grid is described. Section 5 is devoted to solve a convection-diffusion-reaction model problem, useful to assess the properties of the method. Section 6 deals with a real combustion problem where also non-linearities appear. In Section 7 a complexity analysis is presented, in order to evaluate the memory requirements of the LDC algorithm. Conclusions are drawn in the last section.

2 Brief overview of LDC

In this section we briefly recall the LDC method and introduce some notation. Let us consider the BVP:

$$\mathcal{L}[u] = f, \qquad \mathbf{x} \in \Omega, \tag{1a}$$

$$\mathcal{B}[u] = g, \qquad \mathbf{x} \in \partial\Omega, \tag{1b}$$

where $\Omega \subset R^2$ is a simply connected domain, $\mathcal{L}$ a linear elliptic operator, f a source term and g the value of u at the boundary $\partial\Omega$. The boundary conditions, expressed by the operator $\mathcal{B}$, are either of Dirichlet or Neumann type. Let us define a discretisation of (1) on a uniform (coarse) grid of size H covering Ω

$$\mathcal{L}_H[u_H] = f_H, \tag{2}$$

where the right-hand side f_H contains both the source term and the contribution of the boundary conditions. In the iteration procedure below, we denote the solution of (2) by u_H^0.

Suppose now that u changes very rapidly in a small subdomain $\Omega' \subset \Omega$. In this high activity region the grid size H will most likely be too large to capture the behaviour of u, so we formulate a new discrete BVP in Ω' by covering it with a grid Ω'_h. For this latter, h is either the actual grid size, if the fine grid is uniform, or a characteristic grid size, if the fine grid is not uniform. In order to define a discrete BVP on it, let us introduce some notation. We split $\partial\Omega'$ in two parts: the interface Γ between Ω and Ω', with $\Gamma = \partial\Omega' \backslash \partial\Omega$, and $\partial\Omega' \cap \partial\Omega$. Furthermore, we define $\Gamma_h := \Gamma \cap \Omega'_h$ and $\Omega'_H := \Omega_H \cap \Omega'$. We set $\mathcal{B}[u] = g$ on $\partial\Omega' \cap \partial\Omega$ and we interpolate from u_H^0 the boundary conditions on Γ_h by using an interpolation operator $\mathcal{P}^{h,H}$. The discrete problem on Ω' now reads

$$\mathcal{L}_h[u_h] = f_h - \mathcal{G}_\Gamma^h \mathcal{P}^{h,H}[u^H|_\Gamma]. \tag{3}$$

Here the term f_h contains the contribution of both the right-hand side f and the boundary conditions on $\partial\Omega' \cap \partial\Omega$, while $\mathcal{G}_\Gamma^h$ is the part of $\mathcal{L}_h$ operating on Γ_h. By solving (3), we obtain the fine grid solution u_h^0. This will be employed to estimate the local discretisation error of the coarse grid approximation, defined as

$$d_H := \mathcal{L}_H[u] - f_H, \tag{4}$$

i.e. the residual found by substituting the exact solution of (1) into the coarse grid scheme (2). To this end, we introduce the grid function w_H^0

$$w_H^0(\mathbf{x}) := \begin{cases} u_H^0(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega_H \backslash \Omega'_H, \\ \mathcal{R}^{H,h}[u_h^0](\mathbf{x}), & \text{if } \mathbf{x} \in \Omega'_H. \end{cases} \tag{5}$$

Here $\mathcal{R}^{H,h}$ is the restriction operator that maps grid functions on Ω'_h onto grid functions on Ω'_H. Let $S_H(\mathbf{x})$ be the set of coarse grid points belonging to the stencil of a grid point $\mathbf{x} \in \Omega'_H$; then we define the subset $\Omega''_H \subset \Omega'_H$ by the relation

$$\mathbf{x} \in \Omega''_H \iff S_H(\mathbf{x}) \subset \Omega'_H. \tag{6}$$

The approximation of d_H is then given by

$$d_H(\mathbf{x}) \doteq d_H^0(\mathbf{x}) := \chi_{\Omega''_H}\big(\big(\mathcal{L}_H[w_H^0] - f_H\big)(\mathbf{x})\big), \tag{7}$$

with $\chi_{\Omega''_H}$ the characteristic function of Ω''_H, i.e.

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

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

$$\mathcal{L}_H[u_H^1] = f_H + d_H^0. \tag{8}$$

By solving (8), we get a new approximation u_H^1. Again u_H^1 can be used to approximate the interface condition for the fine grid problem, giving rise to the following algorithm.

LDC Algorithm

- Initialization
  - Compute u_H^0 from the basic coarse grid problem (2);
  - Build a fine grid enclosing the high activity region;
  - Define the fine grid boundary value problem, interpolating the boundary conditions on Γ_h;
  - Compute u_h^0 from the fine grid BVP (3).
- Iteration $i = 1, 2, \ldots$
  - Compute the grid functions w_H^{i-1} using (5);
  - Estimate the local discretisation error d_H^{i-1} using (7);
  - Solve the new coarse grid problem

    $$\mathcal{L}_H[u_H^i] = f_H + d_H^{i-1};$$

  - Define the new BVP on the fine grid, interpolating boundary conditions on Γ_h;
  - Compute u_h^i from the fine grid BVP (3).

The i-th approximation of the solution of BVP (1) is the composite grid solution

$$u_{H,h}^i(\mathbf{x}) := \begin{cases} u_H^i(\mathbf{x}) & \mathbf{x} \in \Omega_H \backslash \Omega'_H, \\ u_h^i(\mathbf{x}) & \mathbf{x} \in \Omega'_h, \end{cases} \tag{9}$$

i.e. $u_{H,h}^i$ is the fine grid solution inside Ω' and the coarse grid solution outside Ω'.

3 Curvilinear refinement domains

Let us focus on the application of LDC with different grid types. In particular let us consider the combination of a global rectangular and a local curvilinear domain; Ω' will then be described in terms of the local (ξ, η)-coordinates, that are assumed to vary monotonically between their extreme values. Between the physical space (x, y) and the computational space (ξ, η) a one-to-one relation has to be established: this guarantees that only one (ξ, η)-point corresponds to each (x, y)-point, viz.

$$T : \begin{cases} x = x(\xi, \eta), \\ y = y(\xi, \eta). \end{cases} \tag{10}$$

The Jacobian of the transformation, defined as

$$J := \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{vmatrix}, \tag{11}$$

represents the ratio of an infinitesimal area element in the physical and in the computational space. We assume $J \neq 0$ for all (ξ, η), implying that T is invertible. In order to express the differential operators in terms of the new curvilinear coordinates, we introduce the function $v(\xi, \eta)$, defined by

$$u(x, y) = u(x(\xi, \eta), y(\xi, \eta)) =: v(\xi, \eta).$$

Then, the first partial derivatives and the Laplacian of u become [9]

$$\frac{\partial u}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial v}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial v}{\partial \eta}\right), \tag{12a}$$

$$\frac{\partial u}{\partial y} = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}\frac{\partial v}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial v}{\partial \xi}\right), \tag{12b}$$

$$\begin{aligned} \nabla^2 u = {} & \frac{1}{J^2}\left(g_\eta^2 \frac{\partial^2 v}{\partial \xi^2} - 2g_{\xi\eta}\frac{\partial^2 v}{\partial \eta \partial \xi} + g_{\xi\xi}\frac{\partial^2 v}{\partial \eta^2}\right) \\ & - \frac{1}{J^3}\left[\left(g_{\eta\eta}\frac{\partial^2 x}{\partial \xi^2} - 2g_{\xi\eta}\frac{\partial^2 x}{\partial \xi \partial \eta} + g_{\xi\xi}\frac{\partial^2 x}{\partial \eta^2}\right)\left(\frac{\partial y}{\partial \eta}\frac{\partial v}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial v}{\partial \eta}\right)\right. \\ & \left. + \left(g_{\eta\eta}\frac{\partial^2 y}{\partial \xi^2} - 2g_{\xi\eta}\frac{\partial^2 y}{\partial \xi \partial \eta} + g_{\xi\xi}\frac{\partial^2 y}{\partial \eta^2}\right)\left(\frac{\partial x}{\partial \xi}\frac{\partial v}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial v}{\partial \xi}\right)\right], \end{aligned} \tag{13a}$$

with $g_{\eta\eta}$, $g_{\xi\eta}$ and $g_{\xi\xi}$ defined by

$$g_{\eta\eta} := \left|\frac{\partial \mathbf{x}}{\partial \eta}\right|^2, \qquad g_{\xi\eta} := \left(\frac{\partial \mathbf{x}}{\partial \xi}, \frac{\partial \mathbf{x}}{\partial \eta}\right), \qquad g_{\xi\xi} := \left|\frac{\partial \mathbf{x}}{\partial \xi}\right|^2. \tag{13b}$$

The coefficients x_ξ, y_ξ, $x_{\xi\xi}$ etc. can be computed once transformation (10) is known. From (12) and (13a) it appears that the differential operators assume a more complex form in the curvilinear coordinates. This effect can be minimized by using an orthogonal grid, i.e. a grid for which the relation

$$\left(\frac{\partial \mathbf{x}}{\partial \xi}, \frac{\partial \mathbf{x}}{\partial \eta}\right) = g_{\xi\eta} = 0, \tag{14}$$

holds. This implies that the mixed second derivatives $x_{\xi\eta}$, $y_{\xi\eta}$ and $v_{\xi\eta}$ in (13a) vanish. Orthogonal coordinate systems not only reduce the number of the additional terms resulting from translating the BVP from the physical into the computational space, but also prevent the introduction of truncation errors in the difference expression, see [9]. The problem of generating such a grid will be discussed in the next section.

Once the coordinate transformation has been accomplished, the resulting BVP has to be discretized on the local domain. The computational domain is built such that (ξ, η) can be regarded as being a uniform rectangular grid, so that all the discetization techniques suitable for cartesian domains can be used. Furthermore, since the step sizes in (ξ, η) are arbitrary and independent of the actual grid size in the physical space, they can be thought of unitary length. Therefore it is not restrictive to take $\xi_i = i$ and $\eta_j = j$, thus simplifying the discretization of the BVP. Moreover, while switching from a cartesian to a curvilinear coordinate representation, the order of accuracy of the discretized problem is kept; see [9]. It must be pointed out that also the coefficients x_ξ, y_ξ, $x_{\xi\xi}$ etc. need to be discretized in the curvilinear space. As has been shown in [9], the truncation error is reduced if they are evaluated numerically by the same finite difference scheme used for the unknown function.

We close this section by specifying the interpolation operator, $\mathcal{P}^{h,H}$, and the restriction operator, $\mathcal{R}^{H,h}$. In both cases, we choose bilinear interpolation; see also [6]. The former maps, onto a given Γ_h-point, the solution values of its closest four Ω_H-points; the latter maps, onto a given Ω'_H-point, the solution values of its closest four Ω'_h-points.

4 Constructing an orthogonal grid

In the previous section the transformation of the differential operators to a curvilinear coordinate system was introduced. This section is devoted to show how an orthogonal fine grid, able to capture the detailed structure of the unknown function, can be built by exploiting the coarse grid solution. We employ the method that has been presented in [10]. Let us briefly explain the main features of this technique and tailor it to our purposes.

The starting point is a non-orthogonal coordinate system: one family of non-orthogonal coordinate lines will be kept, the other one will be transformed into a set of lines orthogonal to the first set. As before we will restrict ourselves to 2D domains. Consider the position vector $\mathbf{x} = \mathbf{x}(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta))$ in a non-orthogonal coordinate system. Let us introduce the covariant base vectors $\partial\mathbf{x}/\partial\eta$, $\partial\mathbf{x}/\partial\xi$, tangent to the ξ- and η-lines, respectively, and the contravariant base vectors $\nabla\xi$, $\nabla\eta$ perpendicular to the ξ- and η-lines, respectively; see Figure 1. The relations between the covariant and the contravariant base vectors are given by

$$\nabla\xi = \frac{1}{J^2}\Big[g_{\eta\eta}\frac{\partial \mathbf{x}}{\partial \xi} - g_{\xi\eta}\frac{\partial \mathbf{x}}{\partial \eta}\Big], \tag{15a}$$

$$\nabla\eta = \frac{1}{J^2}\Big[-g_{\xi\eta}\frac{\partial \mathbf{x}}{\partial \xi} + g_{\xi\xi}\frac{\partial \mathbf{x}}{\partial \eta}\Big], \tag{15b}$$

with $g_{\eta\eta}$, $g_{\xi\eta}$, $g_{\xi\xi}$ and J as introduced in (13b) and in (11), respectively. Suppose that we want to keep

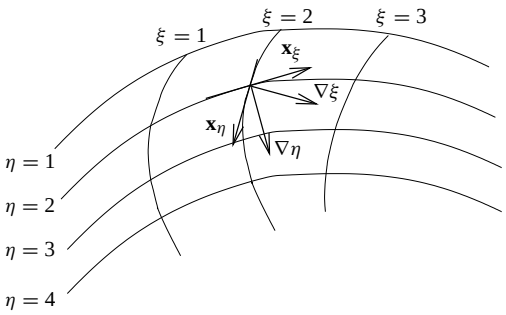


Figure 1: Covariant and contravariant base vectors.

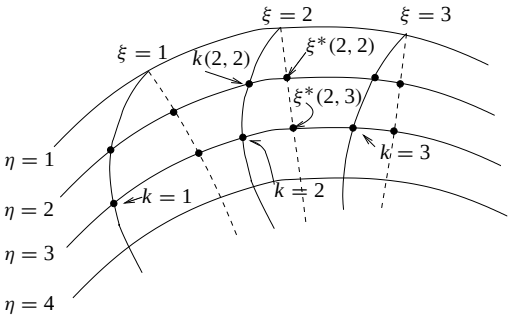


Figure 2: Coordinate systems (ξ, η) (solid lines) and (ζ, η) (dashed lines).

the η-lines and that we want to build a family of coordinate lines, the ζ-lines say, perpendicular to them. To do so, we introduce a function $k(\xi, \eta)$, being constant along the new ζ-lines. Since the covariant base vector $\partial\mathbf{x}/\partial\xi$ is tangent to the η-lines, the orthogonality condition between the ζ-lines and the η-lines can be found to be

$$\nabla k \times \frac{\partial \mathbf{x}}{\partial \xi} = 0, \tag{16}$$

where ∇k is related to the contravariant base vectors by

$$\nabla k = \frac{\partial k}{\partial \xi}\nabla\xi + \frac{\partial k}{\partial \eta}\nabla\eta. \tag{17}$$

Using (15), we can express ∇k in terms of the covariant base vectors. If we insert (17) into (16), we get the hyperbolic equation

$$\frac{\partial k}{\partial \eta} + f(\xi, \eta)\frac{\partial k}{\partial \xi} = 0, \tag{18a}$$

with $f(\xi, \eta)$ given by

$$f(\xi, \eta) = -\frac{\left(\dfrac{\partial \mathbf{x}}{\partial \xi}, \dfrac{\partial \mathbf{x}}{\partial \eta}\right)}{\left|\dfrac{\partial \mathbf{x}}{\partial \xi}\right|^2} = -\frac{g_{\xi\eta}}{g_{\xi\xi}}. \tag{18b}$$

The characteristics of (18a) are

$$\frac{\mathrm{d}\xi}{\mathrm{d}\eta} = f(\xi, \eta), \tag{19a}$$

$$\frac{\mathrm{d}k}{\mathrm{d}\eta} = 0, \tag{19b}$$

i.e. the sought ζ-lines.

As will be explained in the following, the solution of (18a) will enable us to compute the position of the orthogonal trajectories with respect to the global domain. We will use a second order accurate scheme, centered around the point $(\xi, \eta + 1/2)$. This choice is based on the observation that the accuracy by which the position of the fine grid points is computed will affect the value of the coefficients x_ξ, y_ξ, $x_{\xi\xi}$ etc. Since these have to be used to discretize our BVP in curvilinear coordinates, too big a deviation from orthogonality will spoil the local grid solution. If I is the number of non-orthogonal ξ-lines, the derivatives of the function k are approximated as follows

$$\frac{\partial k}{\partial \xi}\big(\xi, \eta + \tfrac{1}{2}\big) \doteq \frac{1}{4}\big(k\big(\xi + 1, \eta + 1\big) - k\big(\xi - 1, \eta + 1\big) + k\big(\xi + 1, \eta\big) - k\big(\xi - 1, \eta\big)\big), \qquad \xi = 2, 3, ..., I - 1, \tag{20a}$$

$$\frac{\partial k}{\partial \eta}\big(\xi, \eta + \tfrac{1}{2}\big) \doteq k\big(\xi, \eta + 1\big) - k\big(\xi, \eta\big). \qquad \xi = 1, 2, 3, ..., I \tag{20b}$$

Note that (20) does not hold for $\xi = 1$ and $\xi = I$. In order to complete the equation set, we have to consider the kind of grid we are going to generate. Since we will work with η-lines that are not closed, i.e. $\mathbf{x}(1, \eta) \neq \mathbf{x}(I, \eta)$, the derivative $\partial k/\partial \xi$ in $\xi = 1$ and $\xi = I$ has to be discretized by a one-sided formula that is at least second order accurate, say

$$\begin{aligned}\frac{\partial k}{\partial \xi}\big(I, \eta + \tfrac{1}{2}\big) \doteq \frac{1}{4}\big(&3k\big(I, \eta + 1\big) - 4k\big(I - 1, \eta + 1\big) + k\big(I - 2, \eta + 1\big) \\ &+ 3k\big(I, \eta\big) - 4k\big(I - 1, \eta\big) + k\big(I - 2, \eta\big)\big).\end{aligned} \tag{21}$$

A similar expression holds for $\xi = 1$. If we use (20) and (21) into (18a), we obtain

$$\begin{aligned}&-\frac{f}{4}k(\xi - 1, \eta + 1) + k(\xi, \eta + 1) + \frac{f}{4}k(\xi + 1, \eta + 1) = \\ &\frac{f}{4}k(\xi - 1, \eta) + k(\xi, \eta) - \frac{f}{4}k(\xi + 1, \eta), \qquad \xi = 2, 3, ..., I - 1,\end{aligned} \tag{22a}$$

$$\begin{aligned}&\left(1 + \frac{3f}{4}\right)k(I, \eta + 1) - fk(I - 1, \eta + 1) + \frac{f}{4}k(I - 2, \eta + 1) = \\ &\left(1 - \frac{3f}{4}\right)k(I, \eta) + fk(I - 1, \eta) - \frac{f}{4}k(I - 2, \eta), \qquad \xi = I,\end{aligned} \tag{22b}$$

$$\begin{aligned}&\left(1 - \frac{3f}{4}\right)k(1, \eta + 1) + fk(2, \eta + 1) - \frac{f}{4}k(3, \eta + 1) = \\ &\left(1 + \frac{3f}{4}\right)k(1, \eta) - fk(2, \eta) + \frac{f}{4}k(3, \eta), \qquad \xi = 1,\end{aligned} \tag{22c}$$

where $f := f\big(\xi, \eta + \tfrac{1}{2}\big)$. The derivatives that appear in (18b) are computed numerically, using the same scheme. Let us briefly describe the procedure to solve (22).

We introduce and compute a function $\xi^*(\xi, \eta)$ that represents the value of ξ to which the point (ξ, η) must be displaced on an η-line to get a trajectory orthogonal to it. Suppose we know $\xi^*(\xi, \eta)$ and we want to determine $\xi^*(\xi, \eta + 1)$, where the initial conditions are

$$\xi^*(\xi, 1) = \xi, \qquad \xi = 1, 2, ..., I.$$

We set

$$k(\xi, \eta + 1) = \xi, \qquad \xi = 1, 2, ..., I \tag{23}$$

and solve system (22) for $k(\xi, \eta)$ by a backward step. After that we know

- k and $\xi^*(\xi, \eta)$ on the (ξ, η)-points (in fact, $\xi^*(\xi, \eta) = \xi$);
- $\xi^*(\xi, \eta)$ on the (ζ, η)-points from the previous solution step.

It follows that $k(\xi^*, \eta)$ on the (ζ, η)-points can be found by a four point Lagrangian interpolation. Furthermore k is constant on the orthogonal trajectories ($k(\zeta, \eta) = k(\zeta, \eta + 1)$) and, because of (23), $k = \xi$ on the $(\eta + 1)$-line. From this we get

$$\xi^*(\xi, \eta + 1) = k(\zeta, \eta + 1) = k(\zeta, \eta) = k(\xi^*(\xi, \eta), \eta).$$

By exploiting the knowledge of the cartesian coordinates of the $(\xi, \eta + 1)$-points and the values of $k(\zeta, \eta + 1)$ and $k(\xi, \eta + 1)$, it is possible to get the cartesian coordinates of the (ζ, η)-points by inverse interpolation.

The aim of this section was to build an orthogonal grid able to cover the high activity region. In order to achieve this result the previously outlined method will be applied to a non-orthogonal coordinate system, where the η-lines consist of the level curves obtained by solving (2) in the global coarse grid. Since these curves are, in general, not smooth enough to be used for the grid generation, a reasonable approach is to smooth them first by using least squares interpolation polynomials or smoothing splines, see Section 5. There is a fundamental reason to use the iso-curves as the set of coordinate lines to be kept: the possibility to decrease the complexity of the local BVP. This can be achieved in two ways. First, because gradients along level curves are almost equal to zero and secondly, because gradients decrease along lines perpendicular to the level curves when going far from the high activity region. In both cases the solution allows for bigger step sizes.

5 A convection-diffusion model problem

In this section we asses the performance of the LDC method in combination with a local curvilinear grid when applied to the following model problem

$$-\nabla^2 u + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega := (0, l_1) \times (0, l_2) \tag{24a}$$

$$u = g(\mathbf{x}) := 1 - \tanh[\delta s(\mathbf{x})], \qquad \mathbf{x} \in \partial\Omega, \tag{24b}$$

where f is such that the exact solution of (24) is given by

$$u(\mathbf{x}) = 1 - \tanh[\delta s(\mathbf{x})]. \tag{25}$$

We consider the following function s

$$s(x, y) = by + ax^2 - r, \tag{26}$$

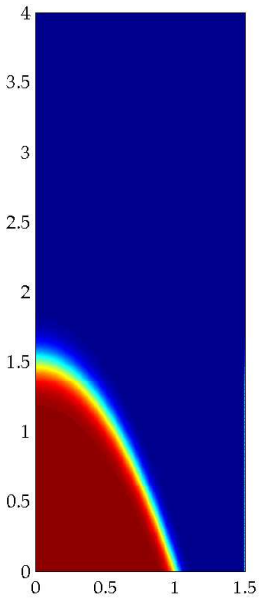


Figure 3: Function $u(\mathbf{x})$: view from above.

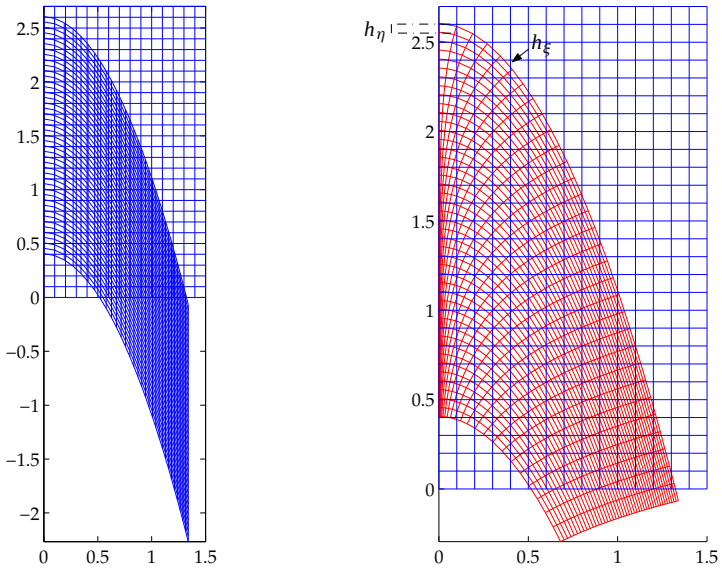
with $a = \frac{1}{2}$, $b = \frac{1}{3}$ and $r = \frac{1}{2}$. Let the global domain be $\Omega = (0, 1.5) \times (0, 4)$ and $\delta = 20$. The function $u(\mathbf{x})$ is approximately equal to 2 in the lower part of the domain, where $s(\mathbf{x}) < 0$, and its value decreases steeply over the center line $s(\mathbf{x}) = 0$, becoming 0 in the upper part of the domain, where $s(\mathbf{x}) > 0$; see Figure 3.

In order to discretize the global problem, the domain Ω has been covered by a uniform rectangular grid with grid size H. The central difference method has been used both on the global and on the local domain. Suppose that we want to compute the solution with an error order of 10^{-3}. Then we notice that the discretization error on the coarse grid is bounded by

$$\frac{1}{12}\big(C_1 + C_2\big)H^2, \quad \text{with} \quad C_1 = \max\left|\frac{\partial^4 u}{\partial x^4}\right| \quad \text{and} \quad C_2 = \max\left|\frac{\partial^4 u}{\partial y^4}\right|.$$

This means that if we choose the boundaries of Ω' coinciding with the parabolae $by + ax^2 - 0.87 = 0$ and $by + ax^2 - 0.13 = 0$ and set $H = 10^{-1}$, we obtain the solution in $\Omega \backslash \Omega'$ with the desired accuracy. The local grid is found by applying the procedure explained in the previous section. The auxiliary grid consists of two sets of curves: the η-lines, parallel to the level curves, and the ξ-lines, being vertical; see Figure 4(a). The η-lines have been obtained as follows: by using the coarse grid solution the center line of the high activity region, corresponding to $u = 1$, is computed; then this is translated vertically by a certain distance h_η to generate a number of curves sufficient to cover entirely the area to be refined. The orthogonal grid is created starting from the upper curve, where the distance between two consecutive grid points has been set equal to h_ξ. Both h_η and h_ξ can thus be considered as a measure of the high activity region refinement; see Figure 4(b).

The same figure shows that the local fine grid is such that some of its points fall outside the computational domain Ω. This is unavoidable if the level curves are not perpendicular to the boundary and we require the fine grid to be orthogonal and the η-lines to follow the level curves; see also [6]. Suppose then that



(a) Auxiliary grid

(b) Zoom on the new orthogonal grid

Figure 4: Auxiliary and orthogonal grids.

a certain point C is such that one of its neighbours, say S, is outside Ω and that B is the point where the segment connecting C and S crosses the boundary. The function value u_S in S can be expressed via extrapolation in terms of the function boundary value in B and of the function value in C itself. By substituting u_S into the C-equation, we can eliminate the dependence on S. Furthermore, in order to keep the pentadiagonal structure of the matrix, we endow all fine grid points outside the global domain with an equation of the type $u = c$, where c is an arbitrary value. This, in fact, does not influence the behaviour of the solution inside Ω.

In Table 1 we show the infinity norm of the error of the composite grid solution, computed for several values of h_ξ, h_η and H. From these results it can be noticed that the method is asymptotically second order accurate with respect to h_ξ and h_η, as it has been already shown to hold for local cartesian grids; see [6]. Furthermore, by comparing the third and the fifth columns of Table 1, it appears that for constant h_η and h_ξ, errors obtained with $H = 10^{-1}$ are quite close to errors obtained with $H = 20^{-1}$. This behaviour suggests that, for the H-values considered here, the discretization errors on the fine grid are dominant. The only value that does not comply with this remark is the last one in the third column. This is due to the fact that the discretization error on the coarse grid is now of the same order of magnitude as the one on the fine grid. Further refinement of the high activity region will thus not produce any improvement on the composite grid approximation error.

One of the advantages of using the level curves as η-lines is that gradients are equal to zero, or very small, along these lines. As a consequence, the grid size in this direction can be increased without spoiling the accuracy of the solution. Table 2 presents again the composite grid error relevant to model problem (24). Although h_ξ is doubled compared to the values in Table 1, the solution is approximated within almost

h_η	h_ξ	$H = 10^{-1}$	ratio	$H = 20^{-1}$	ratio
$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$4.34 \cdot 10^{-2}$		$3.28 \cdot 10^{-2}$	
$5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$8.01 \cdot 10^{-3}$	5.4	$7.70 \cdot 10^{-3}$	4.3
$2.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.20 \cdot 10^{-3}$	3.6	$1.90 \cdot 10^{-3}$	4.1
$1.25 \cdot 10^{-2}$	$1.25 \cdot 10^{-2}$	$1.30 \cdot 10^{-3}$	1.2	$4.76 \cdot 10^{-4}$	4.0

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

the same accuracy. The last column of Table 2 shows the number of η-lines lines used to cover the high activity region.

h_η	h_ξ	$H = 10^{-1}$	ratio	$H = 20^{-1}$	ratio	$\eta -$ lines
$5 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1.06 \cdot 10^{-2}$		$1.10 \cdot 10^{-2}$		45
$2.5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$2.70 \cdot 10^{-3}$	3.9	$1.70 \cdot 10^{-3}$	6.5	89
$1.25 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$1.30 \cdot 10^{-3}$	1.2	$4.30 \cdot 10^{-4}$	4.0	175

Table 2: Composite grid approximation error $||u^* - u_{H,h}||_\infty$ ($h_\xi = 2h_\eta$).

Another way to reduce the number of the fine grid points can e.g. be to thin out the η-lines while going away from the high activity region. This is done on the basis of the second derivative $\partial^2 u/\partial y^2$, computed along the line $x = 0$. We start from the point ($x_0 = 0, y_0 = 1.5$), placed on the central line of the high activity area. Suppose now that we want to refine the region above this line starting with a certain grid size $h_{\eta\,0}$ that will increase as we move away from $\mathbf{x}_0$. To do so, we evaluate $(\partial^2 u/\partial y^2)(0, y_{i-1} + h_{\eta_{i-1}})$ and update h_η in the following way

- $h_{\eta i} = h_{\eta_{i-1}} \left(\frac{\partial^2 u}{\partial y^2}\big|_{\mathbf{x}_{i-1}} \Big/ \frac{\partial^2 u}{\partial y^2}\big|_{\mathbf{x}_i} \right)$;
- if $h_{\eta_i} < h_{\eta_0}$, set $h_{\eta_i} = h_{\eta_0}$;
- if $h_{\eta i} > \nu h_{\eta_0}$ for some $\nu > 1$, set $h_{\eta_i} = h_{\eta_{i-1}}$.

To understand the meaning of the first *if*-statement, we must realise that $\partial^2 u/\partial y^2$ is an odd function that is equal to zero just across the line from which we start to refine and that reaches its maximum absolute values above and below this line. Thus, without this statement, the first steps of the algorithm would lead to a h_{η_i} smaller than h_{η_0}. In contrast, the second *if*-statement prevents a small second derivative from resulting in a very big h_η. We have used $\nu = 3$. Since we know the analytical form of u on the boundary $x = 0$, we have also its second derivatives at our disposal. In other cases it can be evaluated numerically. The same procedure is used to refine the region below the considered η-line. Results are reported in Table 3, where h_{η_0} is the grid size across the central line of the high activity region. The approximation errors in Table 3 have to be compared with those shown in the third column of Table 2 ($H = 10^{-1}$). It appears that, when h_η varies according to the procedure previously explained, the number of η-lines is

considerably reduced while the accuracy of the approximated solution is almost the same. The last row of Table 3 shows a smaller error even with respect to the corresponding value in Table 2. This is due to the fact that, by using a variable h_η, we cannot impose a priori the fine grid to be exactly as wide as the fine grid built with a constant h_η. Thus, in the former case, the fine grid turns out to be slightly wider and this leads to a more accurate solution. The achievable contour-lines reduction depends, in general, on the shape of the unknown function. Here we have managed to save, on average, 30% of them. Let us now compare Table 1 and Table 2, we see that, for equal values of h_η, the reduction factor achievable by using a bigger h_ξ is equal to 2. On the other hand, by comparing the number of η-lines in Table 2 and Table 3, we notice that the possibility to use h_η variable leads, on average, to a reduction factor equal to 1.5. Then, the total reduction factor that can be obtained by using the level curves as set of coordinate line and without any loss in the accuracy of the solution is roughly equal to 3.

h_{η_0}	h_ξ	$\|\|u^* - u_{H,h}\|\|_\infty$	ratio	η − lines
$5 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$9.80 \cdot 10^{-3}$		33
$2.5 \cdot 10^{-2}$	$5. \cdot 10^{-2}$	$2.70 \cdot 10^{-3}$	3.63	58
$1.25 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$7.42 \cdot 10^{-4}$	3.64	111

Table 3: Composite grid approximation error with $H = 10^{-1}$ and h_η variable.

The results in Table 1, 2 and 3 are obtained with one complete LDC iteration, after which convergence is achieved.

6 The thermo-diffusive model for laminar flames

In this section the LDC technique in combination with curvilinear grids will be applied to a laminar premixed flame. Let us first give a brief outline of the two-dimensional thermo-diffusive combustion model.

6.1 The thermo-diffusive model

The classical combustion equations (mass conservation, momentum equations, conservation of energy and chemical species conservation) constitute a coupled non-linear system that is quite hard to solve. However, it can be simplified by applying the *isobaric* and *constant density approximations*, see [11], and then normalized, as explained in [12]. This procedure eventually leads to the following system

$$\frac{\partial \Theta}{\partial t} = \nabla^2 \Theta + \omega(Z, \Theta) - V \cos\left(\frac{\pi y}{2L}\right) \frac{\partial \Theta}{\partial x}, \tag{27a}$$

$$\frac{\partial Z}{\partial t} = \frac{1}{Le} \nabla^2 Z - \omega(Z, \Theta) - V \cos\left(\frac{\pi y}{2L}\right) \frac{\partial Z}{\partial x}, \tag{27b}$$

where $V \cos(\pi y/2L)$ is a given non-uniform velocity field inside the tube, Z is the normalized mass fraction of the reactant, Θ the non-dimensional temperature and Le the *Lewis number*. The source term

is given by

$$\omega = \frac{\beta^2}{2Le} Z \exp\left(- \frac{\beta(1-\Theta)}{1-\alpha(1-\Theta)} \right), \tag{28}$$

with β and α non-dimensional parameters. In the following we set $Le = 1$.

Our purpose is to study the propagation of a flame in an infinite tube of height L, whose upper and lower walls are adiabatic and non-catalytic. The boundary conditions for system (27) read

$$\Theta(-\infty, y, t) = 0, \qquad \Theta(\infty, y, t) = 1, \qquad \Theta_y(x, 0, t) = \Theta_y(x, L, t) = 0, \tag{29a}$$

$$Z(-\infty, y, t) = 1, \qquad Z(\infty, y, t) = 0, \qquad Z_y(x, 0, t) = Z_y(x, L, t) = 0. \tag{29b}$$

Obviously, the computational domain will be restricted to a portion $\Omega = [-a, a] \times [0, L]$ of the infinite tube

It has been shown in [13] that this problem allows for a traveling wave solution of the form $\Theta(x + V_0 t,\ y)$, $Z(x + V_0 t,\ y)$. In this context, V_0 represents the velocity of the traveling wave (the flame front); it is directed towards the unburnt gases, i.e. the left part of the computational domain. Furthermore, it can easily be seen that the relation $\Theta + Z = 1$ always holds. This implies that (27) reduces to the problem

$$-\nabla^2\Theta + \left(V_0 + V \cos\left(\frac{\pi y}{2L}\right)\right)\frac{\partial \Theta}{\partial x} = \omega(1-\Theta, \Theta), \tag{30}$$

with the following boundary conditions

$$\Theta(-\infty, y) = 0, \qquad \Theta(\infty, y) = 1, \qquad \Theta_y(x, 0) = \Theta_y(x, L) = 0. \tag{31}$$

Eq. (30) implies that the flow is stationary in a frame of reference attached to the flame front. Moreover, the velocity of the flow with respect to the aforementioned frame of reference is equal to $V_0 + V \cos\left(\frac{\pi y}{2L}\right)$. An expression for the speed V_0 can be obtained by integrating (30) on the whole computational domain Ω. It gives

$$V_0 = \frac{1}{L} \iint_{\Omega} \omega \, dS - \frac{2V}{\pi}. \tag{32}$$

This only holds if the ratio $2a/L$ is big enough that in the vicinity of the boundaries $\Theta_x(\pm L, y)$ is small. Depending on the value of V, the velocity V_0 can be negative. If this is the case, it may happen that in some parts of the tube also the term $V_0 + V \cos\left(\pi y/2L\right)$ becomes negative, producing an inversion of the flow, which is thus directed from the burnt towards the unburnt gases.

6.2 Modification of the LDC Algorithm

This subsection is devoted to describe how the solution of the problem (30)-(32) in the unknowns Θ and V_0 has been obtained and the way the LDC Algorithm of Section 2 has been modified.

Both the global and the local problem have been discretized with central differences. The integral of the source term ω extended to the domain Ω is computed by the Cavalieri-Simpson formula for the

2-dimensional rectangular element, i.e.

$$I \cong \frac{(x_{i+1}-x_i)(y_{j+1}-y_j)}{36}\Big[\omega(i,j)+\omega(i+1,j)+\omega(i,j+1)+\omega(i+1,j+1) + 4\Big(\omega(i,j+\tfrac{1}{2})+\omega(i+1,j+\tfrac{1}{2})+\omega(i+\tfrac{1}{2},j)+\omega(i+\tfrac{1}{2},j+1)\Big) + 16\omega(i+\tfrac{1}{2},j+\tfrac{1}{2})\Big]. \tag{33}$$

The problems that have to be solved in the global and in the local domain are different. In fact, the first one consists of both the temperature equation (30) and the velocity equation (32): V_0 represents a part of the convection coefficient and it must be updated every time the non-linear solver gives a converged iteration for the temperature field. While solving the local problem, V_0 cannot be computed after each converged iteration, since it is essentially a global variable, so it is only updated after the restriction step. Furthermore, the correction step will likely induce some change in the level curves. This calls for a re-gridding step before solving the local problem again. These considerations lead to the following new algorithm

LDC Regridding Algorithm

- Initialization
  - Compute Θ_H^0 from the basic coarse grid problem (30) and V_0 from (32);
  - Build a fine grid enclosing the high activity region by using the coarse grid solution;
  - Define the fine grid BVP, interpolating the boundary conditions on Γ_h;
  - Compute Θ_h^0 from the fine grid BVP.
- Iteration $i = 1, 2, ...$
  - Compute the grid functions w_H^{i-1} using (5);
  - Estimate the local discretisation error d_H^{i-1} using (7);
  - Solve the new coarse grid problem

    $$\mathcal{L}_H[\Theta_H^i] = f_H + d_H^{i-1}, \tag{34}$$

    and update V_0 from (32);
  - Build a new fine grid enclosing the high activity region by using the coarse grid solution;
  - Define the new BVP on the fine grid, interpolating boundary conditions on Γ_h;
  - Compute Θ_h^i from the fine grid BVP.

6.3 Numerical results

To solve the thermo-diffusive model, the following setting is considered: $\Omega = (-6.1, 6.1) \times (0, 4)$, $\beta = 10$, $\alpha = 0.84$; furthermore $V = 3$ in the imposed flow field.

The global domain is discretized uniformly, by using a grid size $H = 2.0 \cdot 10^{-1}$. We have built the local fine grid by the following procedure. Considering that the normalized temperature ranges between

0 and 1, three level curves are drawn at $\Theta_a = \epsilon_a$, Θ_b and $\Theta_c = 1 - \epsilon_c$, with $\epsilon_a = 2.0 \cdot 10^{-1}$ and $\epsilon_c = 0.01 \cdot 10^{-1}$. We choose the level curve $\Theta = \Theta_b$ at the location where ω reaches its maximum value; it can be computed analytically. Across the Θ_b iso-line, $h_{\eta_0} = H/4$ is set. Then the step sizes on the left and on the right of Θ_b are computed according to $h_i = r_{i-1} h_{i-1}$, where

$$r_{i-1} = \min\left(\frac{\omega_{i-1}}{\omega_i}, 1.1\right).$$

Furthermore, $h_\xi = H$. The curves in between Θ_a, Θ_b and Θ_b, Θ_c are not real iso-lines, but are determined by interpolation from the aforementioned three lines.

After two LDC iteration the speed V_0 converges to a value of -0.4237. Figures 6(a) and 6(b) show the normalized temperature and reaction rate coarse solutions, respectively. Figure 7 depicts the fine grid solution. The two-norm of the fine grid solution residual is found to be of the order of 10^{-5}.

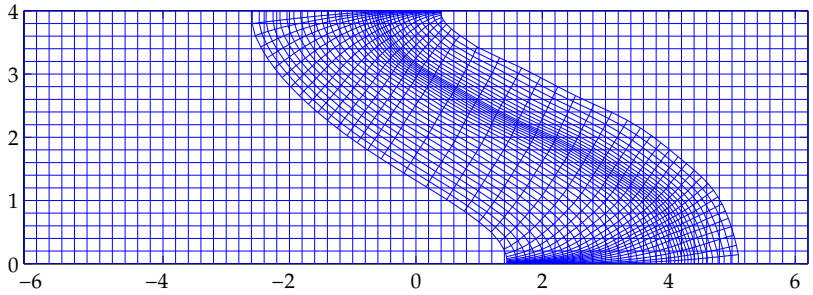
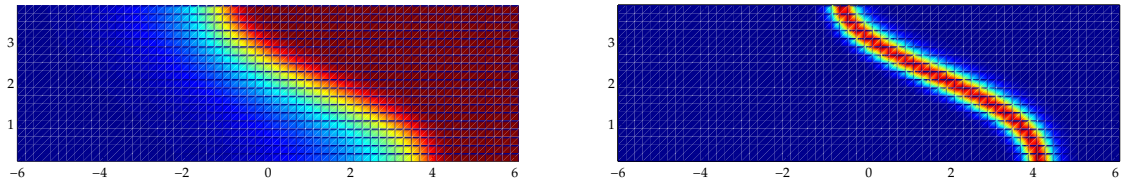


Figure 5: Coarse and fine grids.



(a) Normalized temperature

(b) Normalized reaction rate

Figure 6: Global solution.

7 Complexity analysis

The aim of this section is to assess the complexity of the algorithm that combines the LDC method with curvilinear orthogonal grids. We assume the total number of grid points, i.e., the memory usage, to be a

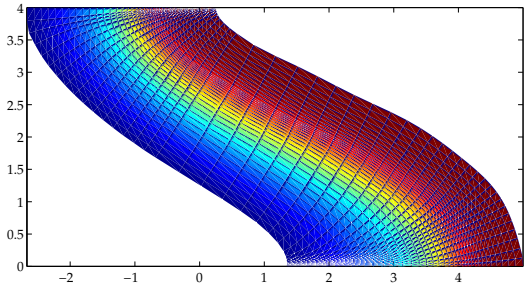


Figure 7: Local solution.

measure of it.

The setting is defined as follows (see Figure 8). The number of coarse grid points along the x- and the y-axis are N and AN, respectively. We introduce a characteristic length l of the fine grid: it can be, for instance, the length of the contour-line η_0 where the activity reaches its maximum value. Moreover, we name b the average width of the fine grid. The step sizes along l and b are h_ξ and h_η, respectively. Let N_{LDC} be the number of points used by the LDC algorithm. If M_ξ and M_η are the number of the fine grid

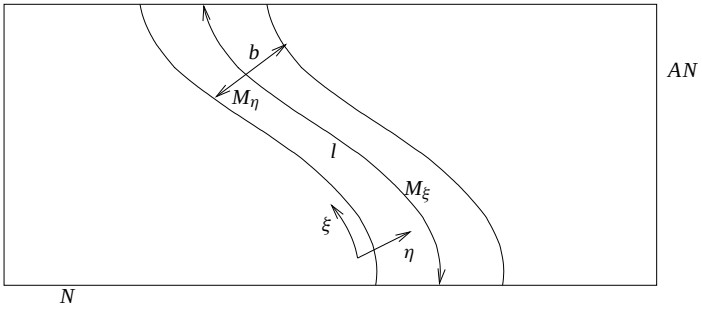


Figure 8: Characteristics of the global and the local domain.

points along the ξ- and the η-axis, then

$$N_{\mathrm{LDC}} = AN^2 + M_\xi M_\eta.$$

Suppose now that in the worst case, i.e., when the high activity region runs from the bottom to the top of the global domain, l is comparable with the height of the global domain Ω. Then, since the η-lines are level curves, the solution is slowly varying along them and we can set $h_\xi = H$. From that follows $M_\xi = BN = \mathcal{O}(N)$ and the complexity of the LDC method becomes

$$N_{\mathrm{LDC}} = AN^2 + BN \cdot M_\eta.$$

The order of magnitude of M_η is usually comparable with N. This implies that

$$N_{\mathrm{LDC}} = \mathcal{O}(N^2), \tag{35}$$

i.e., a highly detailed solution can be obtained by solving two systems having the same complexity of the coarse grid problem. However, we can try to reduce some more the complexity of the fine grid BVP by adapting the number of the η-lines to the behaviour of the solution. In fact, we can vary the step size h_η according to

$$h_{\eta_i} w_i = h_{\eta_{i-1}} w_{i-1} = \text{const}, \quad \text{or} \quad h_{\eta_i} = r_{i-1} h_{\eta_{i-1}}. \tag{36}$$

Here w is a given weight function that reflects, in some way, the shape of the solution: it can be, for instance, the source term. Furthermore $r_{i-1} = w_{i-1}/w_i$. If we choose h_η to vary according to (36) then, since r is variable, it becomes very difficult to make predictions about the number of contour-lines to be used. So, we take r constant and equal to the geometric average of the variable r_i. This way, we expect not to make a big error in the evaluation of N_η. We can split the area to be refined in two parts: one on the left (decreasing η) and the other on the right (increasing η) of η_0. Let us consider what happen to the area on the left of η_0. We introduce the following quantities: b_l, its average width, N_{η_l}, the number of contour-lines with which it will be covered, r_l, the value that r assumes there. We then get

$$b_l = h_{\eta_0}\left(1 + r_l + ... + r_l^{N_{\eta_l}-1}\right) = h_{\eta_0}\frac{1 - r_l^{N_{\eta_l}}}{1 - r_l}. \tag{37}$$

If $M_{\eta_l} + 1$ is the number of η-lines that is to be used if b_l is uniformly refined, (37) can be written as

$$M_{\eta_l}(1 - r_l) = 1 - r_l^{N_{\eta_l}}, \tag{38}$$

from which follows

$$N_{\eta_l} = \frac{\log\left(1 + M_{\eta_l}(r_l - 1)\right)}{\log r_l}. \tag{39}$$

A relation similar to (39) holds for the high activity area placed on the right of η_0. We know that, by definition, $r_l > 1$. Furthermore, since the step sizes are not allowed to change too rapidly where gradients are high, see [9], its value, as well as the values of the single r_i, must be limited. Suppose then to set $1 < r_l < c_l$, with c_l small and define $q = r_l - 1$. By expanding the denominator of (39) by a Taylor series, we get

$$N_{\eta_l} = \frac{\log\left(1 + M_{\eta_l} a_l\right)}{a_l - \frac{a_l^2}{2} + \frac{a_l^3}{3} + \ldots} \approx \frac{\log\left(1 + M_{\eta_l} a_l\right)}{a_l}. \tag{40}$$

We see that the value of N_{η_l} in (40) tends, as expected, to M_{η_l} when a_l goes to zero. Eventually, the complexity of the LDC method can be expressed by the sum of three terms

$$N_{\text{LDC}} \approx AN^2 + BN \cdot \frac{\log\left(1 + M_{\eta_l} a_l\right)}{a_l} + BN \cdot \frac{\log\left(1 + M_{\eta_r} a_r\right)}{a_r}, \tag{41}$$

where M_{η_r} and a_r are the analogous of M_{η_l} and a_l on the right of η_0. The leading term of the sum in (41) still remains the coarse grid contribution. Nevertheless, the second and third term can be considerably smaller than M_{η_l} and M_{η_r}, depending on the value of a_l and a_r, respectively. In Figure 9 the behaviour of $N_\eta = \log(1 + M_\eta a)/a$ versus M_η and for several values of a is plotted. We see that the choice of a variable h_η leads to a considerable reduction of the computational efforts required to solve the local BVP.

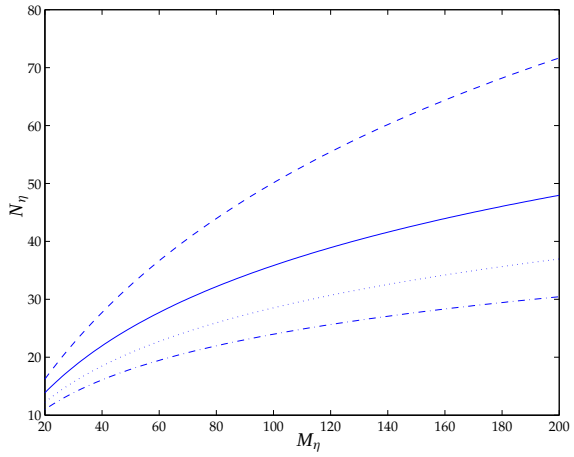


Figure 9: The function $N_\eta = \log(1 + M_\eta a)/a$ is plotted versus M_η. It is parameterised with respect to a, assumed equal to 0.025 (dashed line), 0.05 (solid line), 0.075 (dotted line) and 0.1 (dashdot line).

8 Conclusions

In this paper the LDC method in combination with curvilinear orthogonal grids has been presented. An algorithm that allows to exploit the coarse grid solution of a BVP problem to defi ne a new local problem to be solved on a curvilinear orthogonal grid has been developed. Its properties have been tested with an application to a convection-diffusion-reaction problem. Both the global and the local operators have been discretised by using central differences. The method has been shown to preserve the second order accuracy. A real, though simplifi ed, non-linear combustion problem has then been solved. Finally, the memory requirements of the method have been studied. It has been shown how its complexity can be strongly reduced by using an orthogonal grid that follows closely the path of the high activity region(s). In this framework, the possibility to build such a grid has been demostrted to be crucial.

References

[1] W. Hackbusch, *Local Defect Correction Method and Domain Decomposition Techniques*, Suppl. 5 Computing, 89-113 (1984).

[2] M.J.H.Anthonissen, B.van 't Hof, A.A.Reusken, A finite volume scheme for solving elliptic boundary value problems on composite grids, Computing 61, 285-305 (1998).

[3] M.J.H.Anthonissen, Convergence Results for the Local Defect Correction Method as an Iterative Process, Numerische Mathematik, to be published.

[4] M.J.H.Anthonissen, Local Defect Correction Techniques: analysis and application to combustion, Ph.D. Thesis, Eindhoven University of Technology, Eindhoven (2001).

[5] P.J.J.Ferket and A.A.Reusken: Further Analysis of the Local Defect Correction Method, Computing 56, 117-139 (1996).

[6] M. Graziadei, R.M.M. Mattheij, J.H.M. ten Thije Boonkkamp, *Local Defect Correction with Slanting Grids*, Numerical Methods for Partial Differential Equations (to be published).

[7] V.Nefedov, R.M.M.Mattheij, Local defect correction with different grid types, RANA 01-11, Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven (2001).

[8] V.Nefedov, Numerical Analysis of Viscous Flow Using Composite Grids with Application to Glass Furnaces, Ph.D. Thesis, Eindhoven University of Technology, Eindhoven (2001).

[9] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, *Numerical grid generation (foundations and applications)*, North Holland (1985).

[10] C.W. Davies, *An initial value Approach to the production of Discrete Orthogonal Coordinates*, J. Comp. Phys., vol 59, no. 1, 164-178 (1981).

[11] B. Larrouturou in T.J. Chung, *Numerical Modeling in Combustion*, Taylor and Francis (1993).

[12] B. Larrouturou, *The Equations of One-Dimensional Unsteady Flame Propagation: Existance and Uniqueness*, SIAM J. Math. Anal., 19, 32-59 (1988).

[13] H. Berestycki, B. Larrouturou, P.L. Lions, *Multi-Dimensional Travelling-Wave Solutions of a Flame Propagation Model*, Arch. Rat. Mech. Anal. 111, 33-49 (1990).