The Large Discretization Step Method for Time-Dependent Partial Differential Equations

Zigo Haras and Shlomo Ta'asan *
Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science
and
Institute for Computer Applications in Science and Engineering

Abstract

A new method for the acceleration of linear and nonlinear time dependent calculations is presented. It is based on the Large Discretization Step (LDS, in short) approximation, defined in this work, which employs an extended system of low accuracy schemes to approximate a high accuracy discrete approximation to a time dependent differential operator. Error bounds on such approximations are derived. These approximations are efficiently implemented in the LDS methods for linear and nonlinear hyperbolic equations, presented here. In these algorithms the high and low accuracy schemes are interpreted as the same discretization of a time dependent operator on fine and coarse grids, respectively. Thus, a system of correction terms and corresponding equations are derived and solved on the coarse grid to yield the fine grid accuracy. These terms are initialized by visiting the fine grid once in many coarse grid time steps. The resulting methods are very general, simple to implement and may be used to accelerate many existing time marching schemes.

The efficiency of the LDS algorithms is defined as the cost of the computing the fine grid solution relative to the cost of obtaining the same accuracy with the LDS methods. The LDS methods typical efficiency is 16 for 2D problems and 28 for 3D problems for both linear and nonlinear equations. For a particularly good discretization of a linear equation an efficiency of 25 in 2D and 66 in 3D was obtained.

i

1   Introduction

In recent years, interest in long-time integration of partial differential equations has increased greatly due to the need to solve diverse problems occurring in various fields of science and engineering such as fluid mechanics, aeroacoustics, electromagnetism and others. In such computations a large system of equations has to be evaluated (explicit schemes) or solved (implicit schemes) for many time steps. These simulations require huge computation time and unless more efficient computational methods are developed, they will be practically intractable in the foreseeable future.

The two possible approaches to employ finite difference approximations to such problems are either to use a highly accurate scheme (i.e., a high order scheme or a scheme for long-time integration [10, 13]) on a grid which resolves all the physical frequencies occurring in the problem, or to use a low order scheme on a significantly finer grid, or a combination of these two. The first approach seems more appealing theoretically. Indeed, high order spatial discretizations [6, 7, 13] and discretizations for long-time integration [10, 13], as well as high order time marching schemes [8] have been in the focus of research lately. Although significant progress has been made, the two main problems investigated in the abovementioned research still lack general solutions. The appropriate treatment of the boundary terms in high order Runge-Kutta schemes that will maintain the interior discretization accuracy still requires further investigation even for linear variable coefficient equations. This problem is mainly of a theoretical interest as it has only a minor effect on most practical computations [8]. The second problem is the lack of a systematic method for constructing numerical boundary conditions of a required accuracy such that the resulting discretization is time-stable. This is a major obstacle to long-time simulations. The Large Discretization Step (LDS) methods, presented here, offer a new and interesting approach to long-time integration. They enable to obtain a fine grid accuracy by time stepping mainly on a coarse grid with rare visits to the fine grid, at a cost substantially smaller than fine grid simulation.

In some cases, the huge computational cost of fine grid simulations may be reduced by using such a grid only at regions where it is required, e.g., to resolve shocks, and employing coarser grids in parts of the computational domain where the solution is relatively smooth [1, 2, 3]. This method of local mesh refinement for systems of conservation laws has been reported to achieve a speedup of up to a factor of 55 for three dimensional problems, relative to performing the computation on a uniform grid with the finest mesh employed [1]. This approach assumes the scheme has the same spatial and temporal accuracy and does not seem applicable to implicit schemes. The programming effort involved in generating and moving the fine grid patches is probably the cause for the limited use of this method.

Multigrid methods have been employed to accelerate time dependent computations in several ways. The naive approach is to use an efficient multigrid solver for implicit time marching schemes. However, in this approach one is still confined to the fine grid time step. A more advanced idea is to use multigrid in time, as well. This approach, the *frozen* τ method, was successfully applied to parabolic equations [4, 5, 9]. There, correction terms are added to the coarse grid equations such that one can time-step on the coarse grid and practically obtain the fine grid solution. This method exploits the smoothness of the change in the solution, typical to parabolic equations, which can be well approximated on coarser grids.

1

The Large Discretization Step (LDS) methods, first introduced by the authors in [11], may be viewed as a generalization of the frozen τ method aimed at accelerating the solution of hyperbolic as well as parabolic equations, for both implicit and explicit time marching schemes. The present work investigates the LDS approach for hyperbolic equations; a class of equations that was not previously amenable to multigrid methods. Although the error bounds derived in this work apply to any time dependent equation, the algorithm for parabolic equations would significantly differ from the hyperbolic solver. Nevertheless, it is expected that introducing the ideas outlined in this work to parabolic solvers will substantially improve their performances, as well.

The present paper extends the preliminary results presented in [11] in several important aspects. The algorithm for high degree LDS was significantly improved; resulting in a more efficient algorithm that requires lower order intergrid transfers. The method was extended to treat nonlinear problems with the same efficiency. Last, problems with non-periodic boundary conditions were solved.

The hyperbolic LDS identifies two grids, a coarse *representation grid* on which all wavelengths occurring in the physical problem can be well resolved, and a finer *computational grid* which is required to obtain the desired accuracy with the given discretization at the prescribed final time. The LDS method employs a grid possibly finer than the representation grid, yet significantly coarser than the computational grid. It introduces one or more correction terms to the coarse grid equations and a system of equations satisfied by these terms is derived, initialized using the fine grid and solved on the coarse grid to yield the fine grid solution. The correction terms are integrated on the coarse grid, hence, their accuracy deteriorates at a rate determined by the coarse grid discretization. However, since their norm is significantly smaller than the solution norm, they can be effectively used for many coarse grid time steps. Thereafter, the fine grid should be revisited to compute new initial data for the correction equations.

The LDS method assumes that a grid which resolves all the physical frequencies occurring in the problem as well as a discretization suitable for a fairly long simulation time are given. However, the requirement to employ the same discretization for substantially longer integration time while maintaining a desired accuracy necessitates the use of significantly finer grids. The algorithm solves on the coarse grid an extended system of equations, using essentially the original time marching subroutines (with at most slight modifications), yielding the fine grid solution. This programming simplicity renders the proposed method easily applicable to any problem similar to these investigated in this work, provided it obeys a few programming conventions. This is an important feature of the proposed method.

The efficiency of the LDS is defined as the cost of computing the solution on the fine grid relative to the cost of obtaining the same solution with the LDS on the coarse grid. The typical efficiency achieved in this work was 16 for 2D problems and 28 for 3D problems. This efficiency was obtained for linear problems with periodic and Dirichlet boundary conditions and for the nonlinear Euler equation in a periodic domain. A particularly good discretization yielded, for a linear problem, an efficiency of 25 in 2D and 66 in 3D.

The organization of this paper is as follows. Section 2 contains a heuristic derivation of the method for both linear and nonlinear time dependent equations. Section 3 presents bounds on the error in the LDS approximation for linear equations. In Section 4 it is shown that the LDS approximation maintains the stability and consistency of the original scheme. The LDS algorithms are described in Section 5. In Section 6 Fourier analysis is employed to analyze various aspects of the algorithm, in particular, to obtain the necessary orders of the intergrid transfers. Section 6

presents numerical results and the conclusions are given in Section 7.

2   Heuristic Derivation

An intuitive and informal derivation of the LDS method will be outlined in this section. A rigorous derivation for linear problems is given in Section 3.

Consider a linear time dependent system of equations with coefficients possibly dependent on x but not on t,

$$
\begin{aligned}
u_t &= L(x, \frac{\partial}{\partial x})u &&\text{for } x \in \Omega \quad t \in [0, T] \\
Mu &= 0 &&\text{for } x \in \partial\Omega \\
u(x, 0) &= u_0(x) &&\text{for } x \in \Omega
\end{aligned}
\tag{2.1}
$$

where $\Omega \subset \mathbb{R}^d$, and $\frac{\partial}{\partial x} = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_d})$.

Let L^h, L^H be the same semi-discretization of equation (2.1) on grids h and H, respectively. Assume that the fine grid is required to obtained the desired accuracy at time T. However, instead of solving on the fine grid

$$
U_t^h = L^h U^h
\tag{2.2}
$$

one would like to modify the coarse grid equation such that it will yield the fine grid solution. The coarse grid solution satisfies,

$$
U_t^H = L^H U^H
\tag{2.3}
$$

A correction term τ to equation (2.3) is sought such that,

$$
u_t^h = L^H u^h + \tau
\tag{2.4}
$$

where $u^h = I_h^H U^h$ denotes a restriction of the fine grid solution to the coarse grid. The relative error, $u^h - U^H$, satisfies

$$
(u^h - U^H)_t = L^H(u^h - U^H) + \tau
\tag{2.5}
$$

Thus, this error satisfies the same equation as u^h. Moreover,

$$
(\partial_t - L^H)\tau = (\partial_t - L^H)^2 u^h = \tau_1
\tag{2.6}
$$

If the following relation holds, which is reasonable to assume when L^H well approximates L^h,

$$
(\partial_t - L^H)^2 u^h \ll (\partial_t - L^H)u^h
\tag{2.7}
$$

i.e., $\tau_1 \ll \tau$, then τ_1 may be neglected; otherwise, the same argument implies that τ_1 satisfies a similar equation to u^h, resulting in a larger system of correcting equations (See Section 3). Thus, when relation (2.7) holds and τ is properly initialized the system of equations

$$
\begin{aligned}
u_t^h &= L^H u^h + \tau \\
\tau_t &= L^H \tau
\end{aligned}
\tag{2.8}
$$

3

yields the fine grid solution on the coarse grid for some integration time. This method of expanding a system of difference equations to obtain a more accurate approximation will be called an *LDS approximation*. In particular, an inflated system of the type (2.8) will be called an LDS approximation in *Correction Scheme* form.

Introducing the new variable $v^h = u^h + \tau$, the system (2.8) can be written as,

$$
\begin{aligned}
u_t^h &= L^H u^h + v^h - u^h \\
v_t^h &= L^H v^h + v^h - u^h
\end{aligned}
\tag{2.9}
$$

This later form might look awkward; however, as will be shown next, this is the form of the LDS for nonlinear problems. This representation of the LDS will be called *Full Approximation Scheme*, following the multigrid naming conventions [4].

In Section 3 rigorous error bounds on such approximations for linear evolution equations are derived.

For nonlinear problems, only a heuristic derivation is given. Consider the nonlinear evolution problem,

$$
\begin{aligned}
u_t &= P(x, u, \frac{\partial}{\partial x}) &&\text{for } x \in \Omega \quad t \in [0, T] \\
M(u) &= 0 &&\text{for } x \in \partial\Omega \\
u(x, 0) &= u_0(x) &&\text{for } x \in \Omega
\end{aligned}
\tag{2.10}
$$

where $\Omega \subset I\!\!R^d$, and $\frac{\partial}{\partial x} = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_d})$.

Let P^h, P^H be the same semi-discretization of equation (2.10) on grids h and H, respectively. A modification of the coarse grid equation is sought that will yield on that grid the fine grid solution, i.e., a forcing term is required which will satisfy

$$
u_t^h = P^H(u^h) + \tau
\tag{2.11}
$$

where $u^h = I_h^H U^h$ denotes a restriction of the fine grid solution to the coarse grid. In this case the relative error satisfies,

$$
\begin{aligned}
(u^h - U^H)_t &= P^H(u^h) - P^H(U^H) + \tau \\
&\approx P_u^H(u^h)(u^h - U^H) + \tau
\end{aligned}
\tag{2.12}
$$

where $P_u^H(u^h)$ is a linearization of P^H around u^h. Assume that the following relation holds, which is reasonable if P^H well approximates P^h,

$$
(\partial_t - P_u^H(u^h))\tau = (\partial_t - P_u^H(u^h))^2(u^h - U^H) \ll (\partial_t - P_u^H(u^h))(u^h - U^H)
\tag{2.13}
$$

then the right hand side of the τ equation may be neglected.

It follows, by the same argument as in the linear case, that the system

$$
\begin{aligned}
u_t^h &= P^H(u^h) + \tau \\
\tau_t &= P_u^H(u^h)\tau
\end{aligned}
\tag{2.14}
$$

yields on the coarse grid the fine grid solution, for some integration time.

This formulation of the LDS is inconvenient to use as it requires explicit linearization. Introducing the new variable $v^h = u^h + \tau$ results in the following system of equations,

$$
\begin{aligned}
u_t^h &= P^H(u^h) + v^h - u^h \\
v_t^h &= P^H(v^h) + v^h - u^h
\end{aligned}
\tag{2.15}
$$

In this setting, the implementation of the nonlinear LDS necessitates only minor modifications to the original time marching program.

3 Approximation Theorems

The LDS approximation introduced in Section 2 may be better appreciated once the initial conditions for the correction equations are determined, and error bounds on these approximations are derived. These two issues are the subject of the present section.

First, an error bound is obtained for a semi-discrete approximation in a restricted setting. This restriction, a commutativity assumption, is introduced merely to maintain a simple and intuitive presentation. Subsequently, error bounds for semi-discrete and fully discrete approximations are derived without this superfluous assumption.

The analysis will be performed for linear equations with coefficients which may depend on x but not on t, of the form,

$$
\begin{aligned}
u_t &= L(x, \frac{\partial}{\partial x}) u + F(x) && \text{for } x \in \Omega \quad t \in [0, T] \\
Mu &= 0 && \text{for } x \in \partial\Omega \\
u(x, 0) &= u_0(x) && \text{for } x \in \Omega
\end{aligned}
\tag{3.1}
$$

where $\Omega \subset \mathbb{R}^d$, and $\frac{\partial}{\partial x} = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_d})$.

3.1 Semi-Discrete Analysis

3.1.1 Motivation

Consider a stable semi-discretization of a linear homogeneous initial boundary value problem of the form (3.1) given by,

$$
\frac{du^h}{dt} - L^h u^h = 0
\tag{3.2}
$$

with initial conditions $u^h(0) = u_0^h$.

Let $\tilde{L}^h$ be an approximation to L^h, e.g., a coarse grid representation of the fine grid operator. Define the system

$$
\frac{d}{dt}
\begin{pmatrix}
v_{0,m}^h \\
\vdots \\
v_{m-1,m}^h \\
v_{m,m}^h
\end{pmatrix}
-
\begin{pmatrix}
\tilde{L}^h & I & 0 & \cdots \\
0 & \tilde{L}^h & I & \cdots \\
& & \ddots & \\
& & & \tilde{L}^h
\end{pmatrix}
\begin{pmatrix}
v_{0,m}^h \\
\vdots \\
v_{m-1,m}^h \\
v_{m,m}^h
\end{pmatrix}
=
\begin{pmatrix}
0 \\
\vdots \\
0 \\
0
\end{pmatrix}
\tag{3.3}
$$

with initial data

$$
v_{j,m}^h(0) = \left(L^h - \tilde{L}^h\right)^j u^h(0)
\tag{3.4}
$$

Henceforth, an approximation of a system of equations by an enlarged system of the form (3.3) will be called *an LDS approximation of degree m*.

Assume that $\tilde{L}^h$ and L^h commute. The solution of this system of linear ordinary differential equations is $e^{\mathcal{A}^h t}(v_{0,m}^h(0), \ldots, v_{m,m}^h(0))^T$, where $\mathcal{A}^h$ denotes the above system. The matrix $\mathcal{A}^h$ has a block Jordan form; hence, an explicit expression for $v_{0,m}^h(t)$ is given by,

$$v_{0,m}^h(t) = e^{\tilde{L}^h t} \sum_{k=0}^m \frac{(L^h - \tilde{L}^h)^k t^k}{k!} u^h(0) = e^{\tilde{L}^h t} \left(e^{(L^h - \tilde{L}^h)t} - \frac{(L^h - \tilde{L}^h)^{m+1} \xi^{m+1}}{(m+1)!} \right) u^h(0)$$

for some $\xi \in [0, t]$.

Assume that $\|e^{\tilde{L}^h t} u^h(0)\| \leq C^h e^{\beta^h t} \|u^h(0)\|$, for constants C^h, β^h; then

$$\|u^h(t) - v_{0,m}^h(t)\| = \frac{\|(L^h - \tilde{L}^h)^{m+1} \xi^{m+1} e^{\tilde{L}^h t} u^h(0)\|}{(m+1)!} \leq \frac{\|L^h - \tilde{L}^h\|^{m+1} t^{m+1}}{(m+1)!} C^h e^{\beta^h t} \|u^h(0)\| \quad (3.5)$$

The bound (3.5) implies that for any fixed final time T, $\lim_{m \to \infty} \|u^h(T) - v_{0,m}^h(T)\| = 0$, with convergence rate depending on the magnitude of the relative truncation error, $\|L^h - \tilde{L}^h\|$.

This bound also suggests that when the relative truncation error is small, then there exists a time T_0 such that the error in the approximation of a fixed degree m is small for $T < T_0$. This observation motivated the LDS algorithm and enables its high efficiency. In this algorithm L^h and $\tilde{L}^h$ stand for the same discretization on the fine and coarse grids, respectively. The algorithm computes initial conditions for the correction equations using the fine grid, then marches with the enlarged system on the coarse grid as long as the LDS error relative to the fine grid solution is of the same magnitude as the error in the later solution. Then, the fine grid is revisited to compute new initial data for these equations. In this manner the fine grid accuracy can be obtained when time marching mainly on the coarse grid.

The identity in (3.5) can be used to obtain the following inequality,

$$\|u^h(t) - v_{0,m}^h(t)\| \leq \frac{\|e^{\tilde{L}^h t}\| \, \|(L^h - \tilde{L}^h)^{m+1} u^h(0)\| \, t^{m+1}}{(m+1)!} \quad (3.6)$$

The stability of the semi-discretizations considered implies that there exists a mesh size h_0 and constants C, β such that for all grids with mesh size $h \leq h_0$,

$$\|e^{L^h t} u^h(0)\| \leq C \, e^{\beta t} \|u^h(0)\| \quad (3.7)$$

Therefore, for $h \leq h_0$ the inequality (3.6) implies that,

$$\|u^h(t) - v_{0,m}^h(t)\| \leq \frac{\|(L^h - \tilde{L}^h)^{m+1} u^h(0)\| t^{m+1}}{(m+1)!} C e^{\beta t} \quad (3.8)$$

Hence, for a fixed degree m and fixed integration time T, the error in the LDS approximation satisfies, $\|u^h(t) - v_{0,m}^h(t)\| = O(h^{(m+1)p})$, provided $(L^h - \tilde{L}^h) = O(h^p)$. Furthermore, if $p > 0$, which is reasonable to assume, this bound suggests that the LDS error decreases as $h^{(m+1)p}$ as mesh is refined.

In the next section, an error bound is proved for non homogeneous equations without the commutativity assumption

6

3.1.2 Error Bound

Let L^h be a discretization of the spatial operator L in (3.1). Let A^h be an approximation to L^h, and denote $B^h = L^h - A^h$. Intuitively, L^h may be viewed as a fine grid discretization and A^h a coarse grid approximation to L^h. However, it should be noted that all the operators L^h, A^h, B^h are defined on the same grid. The semi-discretization of equation (3.1) may be written as

$$u_t^h - A^h u^h = B^h u^h + f^h$$
$$u^h(0) = u_0^h \tag{3.9}$$

Denote the solution of this problem by $U^h(t) \in R^N$,where N is the number of grid points. Assume the following inequalities hold,

$$\|e^{A^h t}\| \leq e^{\beta t} \tag{3.10}$$

$$\|U^h(t)\| \leq C\, e^{\beta t} \left(\|u_0^h\| + \|f^h\| \right) \tag{3.11}$$

for constants C, β which may depend on h. The stability of the semi-discretizations considered implies that for fine enough grids (3.10)–(3.11) may be bounded independently of h.

Define the system of equations

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{pmatrix} V_{0,m}^h \\ \vdots \\ V_{m-1,m}^h \\ V_{m,m}^h \end{pmatrix} - \begin{pmatrix} A^h & I & 0 & \cdots \\ 0 & A^h & I & \cdots \\ & & & \ddots \\ & & & A^h \end{pmatrix} \begin{pmatrix} V_{0,m}^h \\ \vdots \\ V_{m-1,m}^h \\ V_{m,m}^h \end{pmatrix} = \begin{pmatrix} f_0^h \\ \vdots \\ f_{m-1}^h \\ f_m^h \end{pmatrix} \tag{3.12}$$

$$\begin{pmatrix} V_{0,m}^h(0) \\ \vdots \\ V_{m-1,m}^h(0) \\ V_{m,m}^h(0) \end{pmatrix} = \begin{pmatrix} B_0^h U_0^h \\ \vdots \\ B_{m-1}^h U_0^h \\ B_m^h U_0^h \end{pmatrix} \tag{3.13}$$

$$f_j^h = B_j^h f^h \qquad 0 \leq j \leq m \tag{3.14}$$

with the B_j^h defined inductively by

$$B_0^h = I$$
$$B_{j+1}^h = [B_j^h, A^h] + B_j^h B^h \qquad j \geq 0 \tag{3.15}$$

It will be shown that the first component of the solution of this system, $V_{0,m}^h(t)$, approximates $U^h(t)$. More precisely, a bound on $\|V_{0,m}^h(t) - U^h(t)\|$ will be presented which, for fixed t, tends to zero as $m \to \infty$.

The vector $(B_0^h U^h(t), \ldots, B_m^h U^h(t))$ satisfies the equation

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{pmatrix} B_0^h U^h \\ \vdots \\ B_{m-1}^h U^h \\ B_m^h U^h \end{pmatrix} - \begin{pmatrix} A^h & I & 0 & \cdots \\ 0 & A^h & I & \cdots \\ & & & \ddots \\ & & & A^h \end{pmatrix} \begin{pmatrix} B_0^h U^h \\ \vdots \\ B_{m-1}^h U^h \\ B_m^h U^h \end{pmatrix} = \begin{pmatrix} f_0^h \\ \vdots \\ f_{m-1}^h \\ f_m^h + B_{m+1}^h U^h \end{pmatrix} \tag{3.16}$$

$$
\begin{pmatrix}
B_0^h U^h(0) \\
\vdots \\
B_{m-1}^h U^h(0) \\
B_m^h U^h(0)
\end{pmatrix}
=
\begin{pmatrix}
B_0^h U_0^h \\
\vdots \\
B_{m-1}^h U_0^h \\
B_m^h U_0^h
\end{pmatrix}
\tag{3.17}
$$

It will be shown that $V_{j,m}^h(t)$ converges to $B_j^h U^h(t)$ as $m \to \infty$. In particular, $V_{0,m}^h(t)$ converges to $U^h(t)$ as $m \to \infty$.

Define $e_{j,m}^h(t) = B_j^h U^h(t) - V_{j,m}^h(t)$. It satisfies the following equation

$$
\frac{\mathrm{d}}{\mathrm{dt}}
\begin{pmatrix}
e_{0,m}^h \\
\vdots \\
e_{m-1,m}^h \\
e_{m,m}^h
\end{pmatrix}
-
\begin{pmatrix}
A^h & I & 0 & \cdots \\
0 & A^h & I & \cdots \\
 & & \ddots & \\
 & & & A^h
\end{pmatrix}
\begin{pmatrix}
e_{0,m}^h \\
\vdots \\
e_{m-1,m}^h \\
e_{m,m}^h
\end{pmatrix}
=
\begin{pmatrix}
0 \\
\vdots \\
0 \\
B_{m+1}^h U^h
\end{pmatrix}
\tag{3.18}
$$

$$
\begin{pmatrix}
e_{0,m}^h(0) \\
\vdots \\
e_{m-1,m}^h(0) \\
e_{m,m}^h(0)
\end{pmatrix}
=
\begin{pmatrix}
0 \\
\vdots \\
0 \\
0
\end{pmatrix}
\tag{3.19}
$$

The solution of this system satisfies

$$
e_{m,m}^h(t) = \int_0^t e^{A^h(t-s)} B_{m+1}^h U^h(s)ds
\tag{3.20}
$$

$$
e_{j,m}^h(t) = \int_0^t e^{A^h(t-s)} e_{j+1,m}^h(s)ds \qquad 0 \le j \le m-1
\tag{3.21}
$$

The norm $\|e_{0,m}^h(t)\|$ is the sought error bound.

$$
\|e_{m,m}^h(t)\| \le \|B_{m+1}^h\| \int_0^t e^{\beta(t-s)} \|U^h(s)\|ds \le \|B_{m+1}^h\| \left(\|u_0^h\| + \|f^h\| \right) C \, te^{\beta t}
\tag{3.22}
$$

By induction one obtains

$$
\|e_{l,m}^h(t)\| \le \|B_{m+1}^h\| \left(\|u_0^h\| + \|f^h\| \right) C \, \frac{t^{m-l+1}}{(m-l+1)!} e^{\beta t}
\tag{3.23}
$$

The following theorem was proved.

Theorem 1 *Let $U^h(t)$ be the solution of (3.9) and $(V_{0,m}^h(t), \ldots, V_{m,m}^h(t))$ be that of (3.12)-(3.13) Then*

$$
\|V_{0,m}^h(t) - U^h(t)\| \le \|B_{m+1}\| \left(\|u_0^h\| + \|f^h\| \right) C \, \frac{t^{m+1}}{(m+1)!} e^{\beta t}
\tag{3.24}
$$

8

A different bound on the error in the LDS approximation will be derived next. Denote

$$\Psi^h(t) = \sup_{0 \le s \le t} e^{\beta s} \tag{3.25}$$

$$\Phi^h_{m+1}(t) = \sup_{0 \le s \le t} \|B^h_{m+1} U^h(s)\| \tag{3.26}$$

The exponent β may depend on the mesh size h. However, for a grid fine enough the stability of the semi-discretization implies that the solution can be bounded independently of h. Therefore, it will be assumed that the grid is fine enough for this property to hold. Then,

$$\|e^h_{m,m}(t)\| \le \Psi^h(t)\Phi^h_{m+1}(t)\,t \tag{3.27}$$

and

$$\|e^h_{j,m}(t)\| \le \left[\Psi^h(t)\right]^{m-j+1} \Phi^h_{m+1}(t)\frac{t^{m-j+1}}{(m-j+1)!} \tag{3.28}$$

Therefore,

$$\|e^h_{0,m}(t)\| \le \left[\Psi^h(t)\right]^{m+1} \Phi^h_{m+1}(t)\frac{t^{m+1}}{(m+1)!} \tag{3.29}$$

This bound is very crude and can be easily improved. However, it implies that for a fixed final time T and degree m, if $B_{m+1} = O(h^{p_{m+1}})$ with $p_{m+1} > 0$, then $\Phi^h_{m+1}(T) = O(h^{p_{m+1}})$ and the error in the LDS approximation decreases like $h^{p_{m+1}}$ as mesh is refined.

The bound (3.29), suggests that looking at B^h_{m+1} may give an insight into the accuracy of the approximation. From the definition of B^h_j, it follows that $B^h_1 = B^h$ and $B^h_2 = [B^h, A^h] + (B^h)^2$.

In the following two examples, A^h is interpreted as a coarse grid approximation to the fine grid spatial discretization $A^h + B^h$.

Example 1 Consider a discretization of the two dimensional wave equation

$$u_{tt} = \Delta u \tag{3.30}$$

when transformed into the system

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = \begin{pmatrix} 0 & I \\ \Delta & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \tag{3.31}$$

This discretization of the wave equation was successfully used in [12] to solve problems in elasticity. Assume that the Laplace operator is approximated by a second order scheme. Then, up to higher order terms,

$$A^h = \begin{pmatrix} 0 & I \\ \Delta + \alpha\,H^2(\partial_{xxxx} + \partial_{yyyy}) & 0 \end{pmatrix}, \qquad B^h = \begin{pmatrix} 0 & 0 \\ \alpha\,(h^2 - H^2)(\partial_{xxxx} + \partial_{yyyy}) & 0 \end{pmatrix} \tag{3.32}$$

where α denotes a generic constant. Since $B^2 = 0$, it follows that

$$B^h_2 = [B^h, A^h] = \begin{pmatrix} -\alpha\,(h^2 - H^2)\,(\partial_{xxxx} + \partial_{yyyy}) & \alpha\,(h^2 - H^2)\,(\partial_{xxxx} + \partial_{yyyy}) \\ 0 & 0 \end{pmatrix} \tag{3.33}$$

It can be seen that B^h_2 consists the same terms as the relative truncation error. Thus, one can not expect that the LDS of degree one will yield an approximation more accurate than the coarse grid discretization. Indeed, applying the LDS algorithm to this discretization shows that the error in the LDS of degree one solution grows like the coarse grid error.

Example 2 Consider the linearized Euler equation

$$
\begin{aligned}
p_t &= a \cdot \nabla + c\,(u_x + v_y) \\
u_t &= a \cdot \nabla + c\,p_x \\
v_t &= a \cdot \nabla + c\,p_y
\end{aligned}
\tag{3.34}
$$

where $a = (a_1, a_2)$ is a two dimensional vector. Assume, that the spatial operator is discretized using a second order scheme. Then, up to higher order terms,

$$
A^h = \begin{pmatrix}
a \cdot (\nabla + \alpha\,H^2\,(\partial_{xxx} + \partial_{yyy})) & c\,(\partial_x + \alpha\,H^2\,\partial_{xxx}) & c\,(\partial_y + \alpha\,H^2\,\partial_{yyy}) \\
c\,(\partial_x + \alpha\,H^2\,\partial_{xxx}) & a \cdot (\nabla + \alpha\,H^2\,(\partial_{xxx} + \partial_{yyy})) & 0 \\
c\,(\partial_y + \alpha\,H^2\,\partial_{yyy}) & 0 & a \cdot (\nabla + \alpha\,H^2\,(\partial_{xxx} + \partial_{yyy}))
\end{pmatrix}
$$

$$
B^h = (h^2 - H^2) \begin{pmatrix}
a \cdot (\alpha\,\partial_{xxx} + \alpha\,\partial_{yyy}) & c\,\alpha\,\partial_{xxx} & c\,\alpha\,\partial_{yyy} \\
c\,\alpha\,\partial_{xxx} & a \cdot (\alpha\,\partial_{xxx} + \alpha\,\partial_{yyy}) & 0 \\
c\,\alpha\,\partial_{yyy} & 0 & a \cdot (\alpha\,\partial_{xxx} + \alpha\,\partial_{yyy})
\end{pmatrix}
\tag{3.35}
$$

where α is a generic constant. It can be easily seen that $(B^h)^2$ consists of sixth order mixed derivatives and fourth order powers of H, h.

$$
[B^h, A^h] = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & \eta \\
0 & -\eta & 0
\end{pmatrix}
\tag{3.36}
$$

with

$$
\eta = c^2\,\alpha^2\,(h^2 - H^2)^2(\partial_{yyyx} - \partial_{xyyy} + \partial_{yyyxxx} - \partial_{xxxyyy})
\tag{3.37}
$$

For smooth solutions,

$$
\eta = c^2\,\alpha^2\,(h^2 - H^2)^2(\partial_{yyyx} - \partial_{xyyy})
\tag{3.38}
$$

Hence, the error bound (3.29) implies that for smooth data the LDS of degree one yields a significantly smaller error than the coarse grid operator. Indeed, our numerical results (see Section 7) show that the LDS algorithm for this equation yields the fine grid solution on the coarse grid.

3.2 Fully Discrete Analysis

Consider a stable finite difference approximation to (3.1) of the form

$$
U^{n+1} = (I + kA)U^n + kBU^n + kf
$$
$$
U^0 = u_0
\tag{3.39}
$$

where $k = \Delta t$.

Assume that

$$
\begin{aligned}
\|U^n\| &< e^{\beta nk}\,(\|u_0\| + \|f\|) & \tag{3.40}\\
\|I + kA\| &\leq e^{\beta k} & \tag{3.41}
\end{aligned}
$$

10

In an analogous way to the semi-discrete case, define $(V_{0,m}^n, \ldots, V_{m,m}^n)$ to be the solution of

$$
\begin{pmatrix} V_{0,m}^{n+1} \\ \vdots \\ V_{m-1,m}^{n+1} \\ V_{m,m}^{n+1} \end{pmatrix} - \begin{pmatrix} I+kA & kI & 0 & \ldots \\ 0 & I+kA & kI & \ldots \\ & & \ddots & \\ & & & I+kA \end{pmatrix} \begin{pmatrix} V_{0,m}^{n} \\ \vdots \\ V_{m-1,m}^{n} \\ V_{m,m}^{n} \end{pmatrix} = \begin{pmatrix} k\,f_0 \\ \vdots \\ k\,f_{m-1} \\ k\,f_m \end{pmatrix}
\tag{3.42}
$$

$$
\begin{pmatrix} V_{0,m}^{0} \\ \vdots \\ V_{m-1,m}^{0} \\ V_{m,m}^{0} \end{pmatrix} = \begin{pmatrix} B_0\,u_0 \\ \vdots \\ B_{m-1}\,u_0 \\ B_m\,u_0 \end{pmatrix}
\tag{3.43}
$$

$$
f_j = B_j f \qquad 0 \le j \le m
\tag{3.44}
$$

with B_j defined recursively by

$$
B_0 = I
\tag{3.45}
$$
$$
B_{j+1} = [B_j, A] + B_j B \qquad j \ge 0
\tag{3.46}
$$

Consider the vector $(B_0 U^n, \ldots, B_{m-1} U^n, B_m U^n)$. It satisfies

$$
\begin{pmatrix} B_0 U^{n+1} \\ \vdots \\ B_{m-1} U^{n+1} \\ B_m U^{n+1} \end{pmatrix} - \begin{pmatrix} I+kA & kI & 0 & \ldots \\ 0 & I+kA & kI & \ldots \\ & & \ddots & \\ & & & I+kA \end{pmatrix} \begin{pmatrix} B_0 U^{n} \\ \vdots \\ B_{m-1} U^{n} \\ B_m U^{n} \end{pmatrix} = \begin{pmatrix} k\,f_0 \\ \vdots \\ k\,f_{m-1} \\ k\,f_m + k\,B_{m+1} U^n) \end{pmatrix}
\tag{3.47}
$$

with the same initial condition as $(V_{0,m}^n, \ldots, V_{m,m}^n)$.

The error $e_{j,m}^n = B_j U^n - V_{j,m}^n$ satisfies

$$
\begin{pmatrix} e_{0,m}^{n+1} \\ \vdots \\ e_{m-1,m}^{n+1} \\ e_{m,m}^{n+1} \end{pmatrix} - \begin{pmatrix} I+kA & kI & 0 & \ldots \\ 0 & I+kA & kI & \ldots \\ & & \ddots & \\ & & & I+kA \end{pmatrix} \begin{pmatrix} e_{0,m}^{n} \\ \vdots \\ e_{m-1,m}^{n} \\ e_{m,m}^{n} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ k\,B_{m+1} U^n \end{pmatrix}
\tag{3.48}
$$

$$
\begin{pmatrix} e_{0,m}^{0} \\ \vdots \\ e_{m-1,m}^{0} \\ e_{m,m}^{0} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}
\tag{3.49}
$$

11

These equations give,

$$1e_{m,m}^n = \sum_{j=0}^{n-1} (I + kA)^{n-1-j} \, k \, B_{m+1} U^j \tag{3.50}$$

$$e_{l,m}^n = \sum_{j=0}^{n-1} (I + kA)^{n-1-j} \, k \, e_{l+1,m}^j \qquad 0 \le l \le m - 1 \tag{3.51}$$

The following theorem can be proved by bounding the solution of this discrete system.

Theorem 2 *Let U^n be the solution of (3.39) and $(V_{0,m}(t), \ldots, V_{m,m}(t))$ be that of (3.42)–(3.43). Then*

$$\|V_{0,m}^n - U^n\| \le \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \binom{n}{m+1} k^{m+1} e^{\beta(n-m-1)\,k} \tag{3.52}$$

Proof : The following, more general, formula will be proved

$$\|e_{l,m}^n\| \le \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \binom{n}{m-l+1} k^{m-l+1} e^{\beta(n-m+l-1)\,k} \qquad \text{for } 0 \le l \le m \tag{3.53}$$

Theorem 2 is the particular case of $l = 0$.

The proof follows by induction on $m - l$. First consider the case $m - l = 0$, i.e., $l = m$. According to formula (3.50)

$$e_{m,m}^n = \sum_{j=0}^{n-1} (I + kA)^{n-1-j} \, k \, B_{m+1} U^j \tag{3.54}$$

Therefore,

$$\|e_{m,m}^n\| \le \sum_{j=0}^{n-1} \|(I + kA)^{n-1-j}\| \, k \, \|B_{m+1}\| \, \|U^j\| \le \sum_{j=0}^{n-1} e^{\beta(n-1-j)k} \, k \, \|B_{m+1}\| \, e^{\beta j k} \left(\|U^0\| + \|f\| \right)$$

$$= \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \, (nk) \, e^{\beta(n-1)k} \tag{3.55}$$

Assume the bound (3.53) is correct for $m - l = m - 1$; that is,

$$\|e_{1,m}^n\| \le \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \binom{n}{m} k^m e^{\beta(n-m)\,k} \tag{3.56}$$

From relation (3.53) it follows that,

$$e_{0,m}^n = \sum_{j=0}^{n-1} (I + kA)^{n-1-j} \, k \, e_{1,m}^j \tag{3.57}$$

Therefore, by the induction hypothesis (3.56),

$$\|e_{0,m}^n\| \le \sum_{j=0}^{n-1} \|(I + kA)^{n-1-j}\| \, k \, \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \binom{j}{m} k^m \, e^{\beta(j-m)\,k} \tag{3.58}$$

$$\le \left(\|U^0\| + \|f\| \right) \|B_{m+1}\| \binom{n}{m+1} k^{m+1} \, e^{\beta(n-m-1)\,k} \tag{3.59}$$

The last inequality follows from the following identity which can be proved by induction on n.

$$\sum_{j=0}^{n-1} \binom{j}{m} = \binom{n}{m+1} \tag{3.60}$$

The relations (3.50)–(3.51) give rise to a somewhat different bound, as well. Denote,

$$\Psi^h(n) = \sup_{0 \le j \le n} \|(I + kA)^j\| \tag{3.61}$$

$$\Phi_{m+1}^h(n) = \sup_{0 \le j \le n} \|B_{m+1}^h U^j\| \tag{3.62}$$

Then,

$$\|e_{m,m}^n\| \le (nk)\Psi^h(n)\Phi_{m+1}^h(n) \tag{3.63}$$

and

$$\|e_{0,m}^n\| \le (nk)^{m+1} \left[\Psi^h(n)\right]^{m+1} \Phi_{m+1}^h(n) \tag{3.64}$$

The stability of the discretization implies that for a fixed time T, if the meshsize is fine enough, i.e., $h \le h_0$, (assuming that k is related to h in a fixed manner), then one can bound

$$\Psi^h(n) \le C(T) \qquad \text{for all } nk \le T, \qquad \text{such that } h \le h_0 \tag{3.65}$$

this bound is independent of the mesh size. Thus,

$$\|e_{0,m}^n\| \le T^{m+1} [C(T)]^{m+1} \Phi_{m+1}^h(n) \tag{3.66}$$

If $B_{m+1} = O(h^{pm+1})$, then as mesh is refined the error in the LDS approximation decreases like h^{pm+1}.

At first glance the discretization (3.39) is a first order in time explicit scheme. However, any single stage explicit or implicit discretization of any order can be brought to a similar form once the source term is appropriately redefined. Therefore, Theorem 2 can be modified and generalized to yield a similar result for a general discretization. Such a generalization would be futile as finding an explicit form for A, B and B_m might require matrix inversion or computing a matrix polynomial. Thus, in those cases the bound (3.52) is hard to compute.

In the fully discrete setting, as in the semi-discrete, an inspection of B_{m+1} might indicate about the applicability of the LDS method for a given discretization.

4 Stability, consistency and convergence

In the previous section a method for obtaining highly accurate approximations using an expanded system of lower order approximations was analyzed. It should be proved that if the original scheme was consistent and stable then so is the resulting LDS scheme.

For simplicity of presentation the discussion is limited to LDS of degree one. The generalization to LDS of general degree m is straightforward. It is further assumed that the source term $F = 0$. This assumption does not effect stability analysis [14] and its effect on consistency will be shortly discussed.

In an LDS of degree one instead of using the stable and consistent scheme,

$$U^{n+1} = (I + kA)U^n \qquad (4.1)$$
$$U^0 = u_0 \qquad (4.2)$$

the following LDS system is employed,

$$V_{0,1}^{n+1} = (I + kA)V_{0,1}^n + kV_{1,1}^n \qquad (4.3)$$
$$V_{1,1}^{n+1} = (I + kA)V_{1,1}^n \qquad (4.4)$$
$$V_{0,1}^0 = u_0 \qquad (4.5)$$
$$V_{1,1}^0 = B_1 u_0 \qquad (4.6)$$

where $B_1 = B$ is defined in (3.39).

The stability of the scheme (4.3)-(4.4) follows from the structure of this system. It consists of a principal part which is coupled through lower order terms. The principal part is diagonal; thus, its discretization is stable for the same time step as the original scheme. The lower order term does not affect stability.

The consistency of the single equation discretization implies that the LDS is a consistent approximation of the system

$$u_t = Lu + \tau \qquad (4.7)$$
$$\tau_t = L\tau$$

for initial data

$$u(x,0) = u_0(x) \qquad (4.8)$$
$$\tau(x,0) = \tau_0(x)$$

In the LDS approximation $V_{1,1}^0 = B_1 u_0$. Thus, if the following relation holds

$$B_1 = O(h^{p_1} + k^{q_1}) \qquad \text{with} \quad p_1, q_1 \geq 1 \qquad (4.9)$$

then the LDS solution is a consistent approximation of (4.7)-(4.8) with initial data

$$u(x,0) = u_0(x) \qquad (4.10)$$
$$\tau(x,0) = 0$$

Thus, it consistently approximates the equation

$$u_t = Lu \qquad (4.11)$$
$$u(x,0) = u_0(x)$$

The source term for the $V_{1,1}$ equation is $B_1 F$. Thus, by the above reasoning, if (4.9) holds then the τ equation is a consistent approximation of an equation with $F = 0$. Therefore, this term does not effect consistency, either.

It follows, by Lax equivalence theorem [14], that the consistency and stability of the original equation together with condition (4.9) ensure the LDS solution convergence to the analytic solution.

14

The LDS approximation maintains the stability of the original scheme, however, it does not necessarily preserve the time-stability of the discretization, i.e., the LDS solution might exhibit a non-physical growth in time although the underlying discretization did not allow such a growth. This phenomena is demonstrated in Lemma 1. Recall, if A^h is a discrete operator acting on grid functions on a periodic or infinite domain, then the *symbol of* A^h, $\hat{A}^h(\theta)$, is defined by the identity,

$$\hat{A}^h(\theta)e^{i\theta \cdot x} = A^h e^{i\theta \cdot x} \tag{4.12}$$

Lemma 1: *Let A^h be a space discretization of a scalar time-dependent equation with periodic boundary conditions, such that its symbol satisfies $\hat{A}^h(\theta_0) = 0$, for $\theta_0 \neq 0$ and $\hat{B}^h(\theta_0) \neq 0$. Then the error in the LDS system based on A^h grows polynomially in time for this Fourier component. The order of the polynomial equals the LDS degree.*

Proof: Consider a semi-discrete LDS approximation of degree one based on A^h. According to Section 3.1.1 it has the form,

$$\begin{array}{rcl} u_t^h & = & A^h\, u^h + \tau^h \\ \tau_t^h & = & A^h\, \tau^h \end{array} \tag{4.13}$$

with initial data

$$\begin{array}{rcl} u^h(x,0) & = & u_0^h(x) \\ \tau^h(x,0) & = & B^h u_0^h(x) \end{array} \tag{4.14}$$

The solution of this system for the θ_0 component is,

$$\hat{u}^h(\theta_0, t) = e^{\hat{A}^h(\theta_0)t}\left(\hat{u}_0^h(\theta_0) + t\hat{B}^h(\theta_0)\hat{u}^h(\theta_0)\right) = \left(1 + t\,\hat{B}^h(\theta_0)\right)\,\hat{u}^h(\theta_0) \tag{4.15}$$

The proof for higher degree LDS is similar.

Note, that the property $\hat{A}^h(\theta_0) = 0$ is common to central discretizations of the first derivative on non-staggered grids, for $\theta_0 = \pi$ (see also Sec. 6.3).

It should be emphasized that although the LDS transformation preserves stability, the resulting algorithm might not be stable. Consider, for instance, a discretization satisfying the condition of Lemma 1; then the LDS solution for the θ_0 component will grow exponentially with the number of visits to the fine grid. Other possible sources for such a growth are large errors introduced by the intergrid transfer employed by the algorithm which are not damped during the cycle (see Sec. 5.1.4) or improper use of Richardson extrapolation (see Sec. 5.2). These last remarks can be understood once the LDS algorithm is presented, in the next section.

5 Large Discretization Step (LDS) Methods

The LDS approximation, introduced in the previous sections, approximates a high accuracy scheme by an enlarged system of equations of a lower accuracy discretization. In the present work the two schemes are the same discretization of a differential operator on two different grids. An LDS algorithm computes the fine grid solution on the coarse grid by solving there an extended system of equations which are initialized using the fine grid. The LDS system is integrated on the coarse grid; hence, the accuracy of the correction terms deteriorates at a rate determined by

that grid discretization. However, since the norm of these terms is significantly smaller than the solution norm, they can be effectively used for many coarse grid time steps. Then, the fine grid has to be revisited to compute new initial data for the correction terms. These terms initialization is computationally costly and the LDS approximation consists of more equations than the original problem; however, the large number of steps performed on the coarse grid before revisiting renders the resulting algorithm very efficient.

This section details the algorithmic implementation of these ideas. The algorithm is presented both in its Correction Scheme and Full Approximation Scheme forms and efficient initialization procedures for these schemes are described and analyzed. Richardson extrapolation is introduced to the LDS method which, for smooth solutions, yields a higher order approximation. The efficiency of the algorithm is discussed and evaluated.

Given a system of hyperbolic differential equations with coefficients which may depend on x but not on t, of the form

$$
\begin{aligned}
\frac{\partial U(x,t)}{\partial t} - A(x, \frac{\partial}{\partial x}) U(x,t) &= F(x) && \text{for } x \in \Omega, \quad t \in [0,T] \\
M u(x,t) &= 0 && \text{for } x \in \partial\Omega \\
U(x,0) &= U_0(x) && \text{for } x \in \Omega
\end{aligned}
\tag{5.1}
$$

where $\Omega \subset {\rm I\!R}^d$, and $\frac{\partial}{\partial x} = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_d})$.

Consider a discretization of the form

$$
U^{n+1} = E(x,k,h) U^n + S(x,k,h) F^n
\tag{5.2}
$$

where h, k denotes Δx and Δt, respectively, and $U^n = U^n_j$ approximates $U(jh, nk)$. In this work $E(x,k,h)$ is an explicit or implicit two level time marching operator. However, the method may be used with multilevel integration schemes, as well. In the sequel, the notation $E^{k,h}$ will be used, omitting the possible dependence on x. In an LDS application two grids (in space-time) are given, a fine one with spacing (h,k) and a coarse with (H,K), where $H = \alpha h, K = \alpha k$. Given $U^H(x,0)$ on the (H,K) grid, one needs to calculate the solution up to a prescribed final time T and obtain the fine grid accuracy.

5.1 The LDS Method of General Degree

The *degree* of an LDS approximation is defined as the number of its correction terms. The error bounds obtained in Section 3 suggest that in many cases (e.g., for smooth solutions) the higher the degree the better the LDS approximates the fine grid solution. In practice, efficiency considerations limit the degree to at most two (see Section 5.3.2). In the sequel, algorithms for LDS of general degree are described.

The initialization procedures necessitate the transfer of the solution from the coarse grid to the fine and back. In this section, it is assumed that appropriate intergrid transfers are given. The order and properties of these transfers will be discussed in Section 6.1. For presentation simplicity the algorithm is described for the case $F(x) = 0$; the treatment of a source term is straightforward.

16

5.1.1 Correction Scheme LDS

The fundamental idea of the LDS method is to look for correction terms to the coarse grid equations, derive and solve the equations satisfied by these terms to obtain the fine grid accuracy on the coarse grid. In Section 3, it was shown that for linear problems these terms approximately satisfy the same equation as the solution. The resulting system of equations, which is valid only for linear problems, was named Correction Scheme LDS.

Correction Scheme LDS of Degree One when $H = 2h$. For clarity of presentation, the algorithm is first presented its simplest form, i.e., Correction Scheme LDS of degree one with $\frac{H}{h} = \frac{K}{k} = 2$.

The algorithm consists two stages : initialization of the correction terms using the fine grid, and time marching on the coarse grid for a predetermined number of steps. The results in Section 3 show that for linear problems the correction terms satisfy approximately the same equation as the solution. However, they do no indicate how to effectively and efficiently compute initial values for these terms. The requirement that the LDS solution should yield the fine grid solution suggests that on the first time steps these solutions should be identical. This observation leads to the following initialization routine,

```
Initialize(V_{0,1}^N, V_{1,1}^N)
```

$$
\begin{aligned}
\texttt{Set} \quad & U^N && = && I_{H,K}^{h,k}\, V_{0,1}^N \\
\texttt{Solve} \quad & U^{N+\frac{m}{2}} && = && E^{h,k}\; U^{N+\frac{m-1}{2}}, && \texttt{m=1,2} \\
& \widetilde{V}_{0,1}^{N+1} && = && E^{H,K}\; V_{0,1}^N \\
\texttt{Set} \quad & V_{1,1}^{N+1} && = && I_{h,k}^{H,K} U^{N+1} \; - \; \widetilde{V}_{0,1}^{N+1} \\
& V_{0,1}^{N+1} && = && \widetilde{V}_{0,1}^{N+1} \; + \; V_{1,1}^{N+1} \\
\texttt{Set} \quad & N && = && N \; + \; 1
\end{aligned}
$$

The initialization consists of interpolating the LDS solution to the fine grid and stepping for the same time on the fine and coarse grids. Then, the correction term is set to the discrepancy between the two solutions and the LDS solution is updated to the fine grid. Here, $\widetilde{V}_{0,1}^{N+1}$ stands for an intermediate value assigned to this variable.

The time advance of the LDS has the following simple form,

```
     LDS Method of Degree One, Correction Scheme
```

$$
\begin{aligned}
&\texttt{Initialize} \quad V_{0,1}^0 \\
&N = 0 \\
&\texttt{While } N \; \leq \; \lceil \tfrac{T}{K} \rceil \quad \texttt{Do} \\
&\quad \texttt{Call Initialize}(V_{0,1}^N, V_{1,1}^N) \\
&\quad \texttt{For i =1,\ldots,Revisit Do} \\
&\qquad \texttt{Solve } V_{1,1}^{N+1} \; = \; E^{H,K}\; V_{1,1}^N
\end{aligned}
$$

$$V_{0,1}^{N+1} = E^{H,K} V_{0,1}^N + V_{1,1}^{N+1}$$

```
    Set       N  =  N  +  1
        End
    End
```

The relation between the theory derived in Section 3 and the initialization procedure is analyzed in Section 5.1.2.

Correction Scheme LDS of General Degree when $H = 2h$. The correction term in the LDS of degree one is initialized to equalize the fine and LDS solutions. If one could solve the exact equation satisfied by this term, which is approximately the fine grid equation, then the fine grid and the LDS solutions would be identical. Clearly, this would be as difficult as obtaining the fine grid solution on the coarse grid. Instead, one can add a new term to correct the first correction term equation. Thus, in an LDS of general degree, the ith correction term may be viewed as correcting the $(i-1)^{th}$ term.

The general degree algorithm consists of two stages : initialization of the correction terms and time marching on the coarse grid for a predetermined number of steps. The time marching procedure has the following simple form, where term i corrects term $i-1$ for $1 \leq i \leq d$.

```
    LDS Method of General Degree - d , Correction Scheme

    Initialize     V⁰_{0,d}
    N = 0
    While N  ≤  ⌈T/K⌉  Do
        Call Initialize(Vᴺ_{0,d},...,Vᴺ_{d,d},d )
        For i = 1,...,Revisit Do
          Solve Vᴺ⁺¹_{d,d}  =  Eᴴ'ᴷ Vᴺ_{d,d}
          For l = d-1,...,0, Step = -1, Do
            Solve Vᴺ⁺¹_{l,d}  =  Eᴴ'ᴷ Vᴺ_{l,d}  +  Vᴺ⁺¹_{l+1,d}
          End
          Set       N  =  N  +  1
        End
    End
```

In an LDS of degree d, the ith correction term $(1 \leq i \leq d)$ corrects the $(i-1)^{th}$ term and both satisfy approximately the same equation as the coarse grid solution. Therefore, the initialization of the ith term to correct the $(i-1)^{th}$ term is identical to the initialization of the first term to correct the solution in the LDS of degree one. Once $V_{i,d}$ is initialized and $V_{i-1,d}$ is updated using this value; the lower index variables can be time advanced using these new values. This procedure is repeated for all $1 \leq i \leq d$. Section 5.1.2 outlines the connection between the error bound derived in Section 3.1.1 and these intuitive arguments which are implemented in the procedure listed below.

```
    Initialize(Vᴺ_{0,d},...,Vᴺ_{d,d},d )
```

```
For i = 1,...,d Do
    Set    U^N     =  I_{H,K}^{h,k} V_{i-1,d}^N
    Solve  U^{N+\frac{m}{2}} =  E^{h,k}  U^{N+\frac{m-1}{2}},              m=1,2
           \widetilde{V}_{i-1,d}^{N+1}  =  E^{H,K}  V_{i-1,d}^N
    Set    V_{i,d}^{N+1}  =  I_{h,k}^{H,K} U^{N+1}  -  \widetilde{V}_{i-1,d}^{N+1}
           V_{i-1,d}^{N+1}  =  \widetilde{V}_{i-1,d}^{N+1}  +  V_{i,d}^{N+1}
    For l = i-2,...,0, Step = -1, Do
        Solve  V_{l,d}^{N+1}  =  E^{H,K}  V_{l,d}^N  +  V_{l+1,d}^{N+1}
    End
    Set     N     =  N  +  1
End
```

The initialization of an LDS of degree two is graphically illustrated in Figure 1.
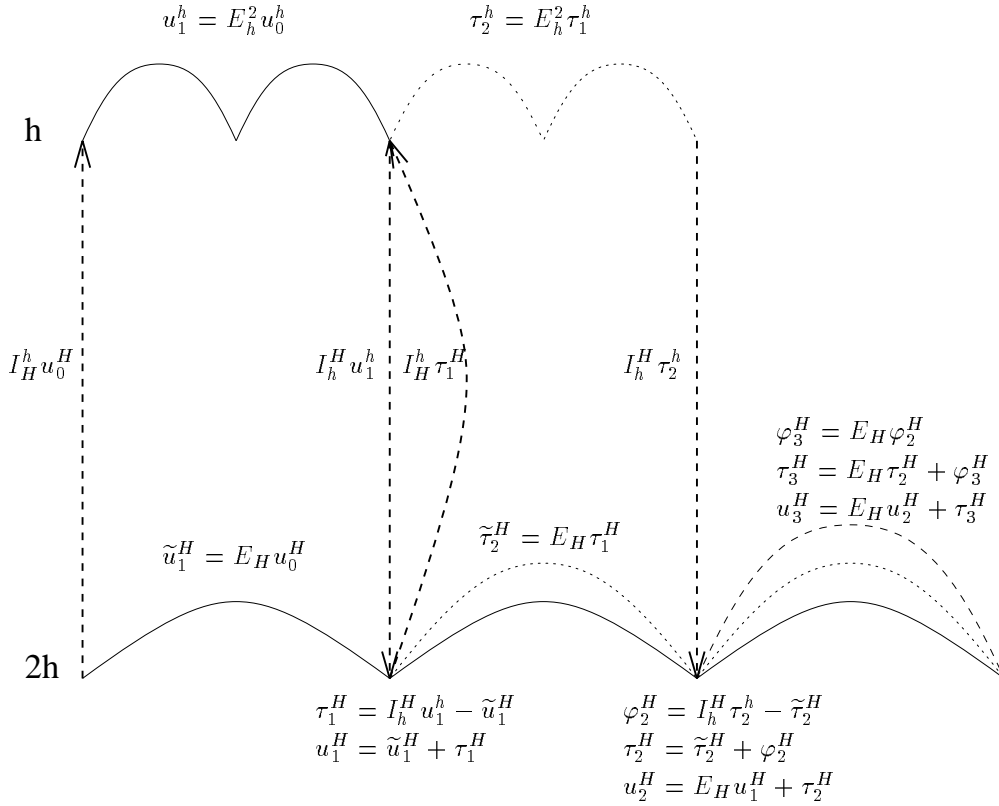
Figure 1: Initialization of an LDS of degree two. With the notation : $u^H = V_{0,2}$, $\tau^H = V_{1,2}$, $\varphi^H = V_{2,2}$

Correction Scheme LDS of General Degree when $H = \alpha h$. The above described procedure can be easily adapted for the general case when $\frac{H}{h} = \alpha$. The simplest approach is to perform α time steps on the fine grid for each coarse grid time step and initialize the correction terms correspondingly, see Figure 2. This procedure of *direct initialization* is very costly and greatly
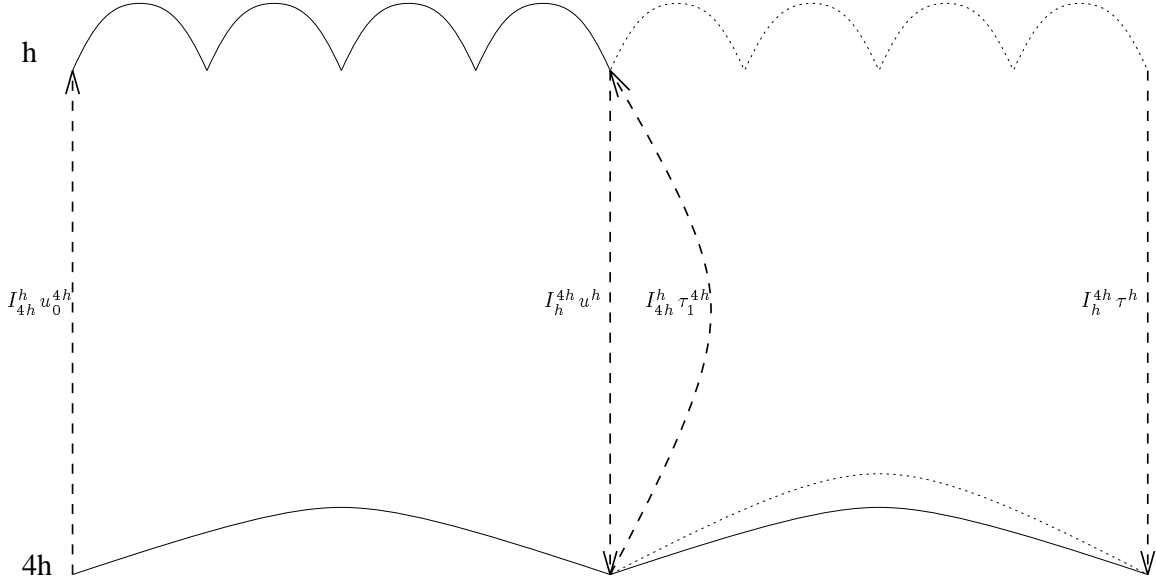

Figure 2: Direct Initialization

reduces the LDS efficiency.

In case α is a composite number, e.g., $\alpha = 2^l$, a more efficient approach is available, exploiting the LDS high accuracy by employing intermediate grids. In this approach, the *simultaneous initialization*, the fine grid is used to initialize an LDS system of degree one on the grid $H_1 = 2h$; since this approximation is very accurate it can be used to initialize an LDS of degree one on grid $H_2 = 2H_1$. This procedure is repeated until the correction term on grid $H_l = H$ is initialized. This process is repeatedly employed for all correction terms. In this method, a correction term on a coarse grid is initialized as soon as enough time marching on finer grids was performed, see Figure 3.

A few important points should be emphasized regarding the initialization procedures. First, in this work the computationally efficient method was favored. There is, however, a trade off between the computational cost and the storage requirements of these two methods (see Sec. 5.3). Therefore, whenever storage is limited, direct initialization might be preferred. Second, it might have been expected that direct initialization will yield more accurate solutions than the simultaneous approach. However, our numerical results show that these methods are indistinguishable for integrations times of interest, i.e., as long as the error in the LDS solution is small. Last, the direct initialization is of practical interest, being used with Richardson extrapolation (see Sec. 5.2).

A simple way to predict the LDS performance is to look at the relative magnitude of the correction terms immediately after initialization. According to the result presented in Section 2,
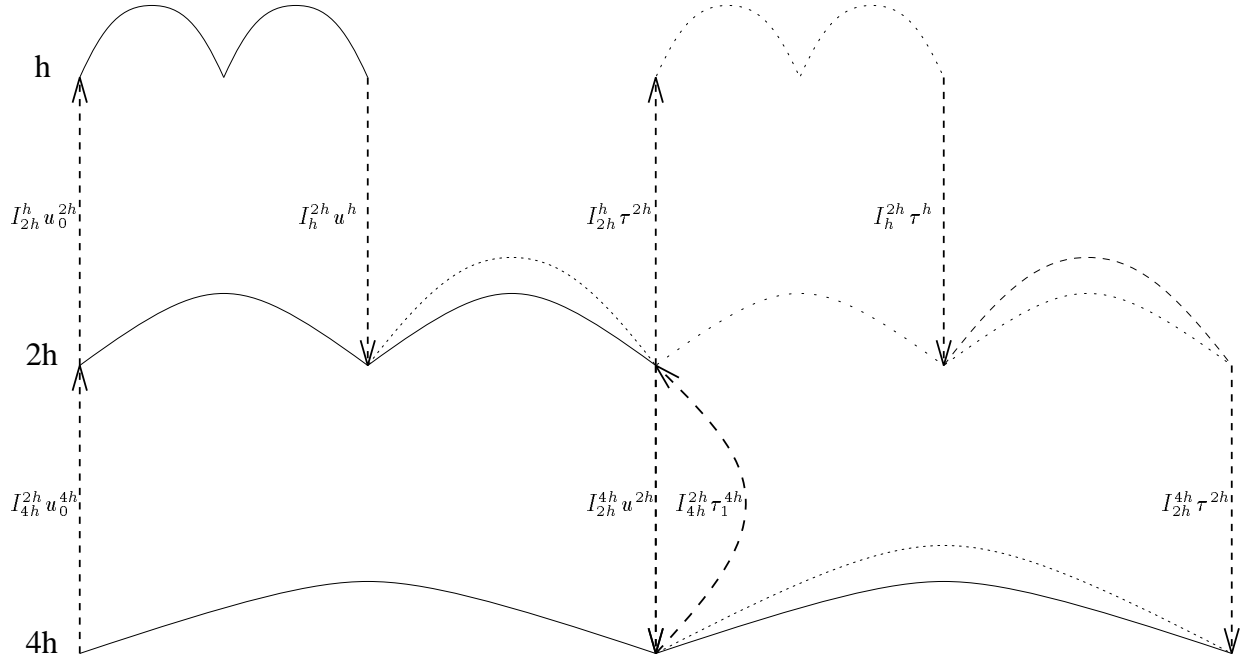
Figure 3: Simultaneous Initialization

the ratio $\frac{\|\tau_j(x,0)\|}{\|\tau_{j-1}(x,0)\|}$ should be roughly constant. Thus, a large variation in this quantity suggests a large error in the initialization of τ_j, causing the LDS failure.

The error bounds derived in Section 3 apply to approximations of general degree m. In practice however, due to efficiency considerations two is the highest degree used (see Sec. 5.3.2).

5.1.2 Initialization Analysis

The procedure for the correction terms initialization was justified by the intuitive arguments that when properly initialized, the LDS solution should agree with the fine grid solution for the first time steps; and that in an approximation of degree d the ith term $V_{i,d}$ corrects the $(i-1)^{th}$ term $V_{i-1,d}$ (for $1 \leq i \leq d$). Thus, each term is initialized similarly to the manner the first term is set to correct the solution. This section outlines the relation between the error bound derived in Section 3.1.1 for the semi-discrete case with the commutativity assumption and the way initialization is implemented. Specifically, it will be shown that the solution at the termination of the initialization procedure described in Section 5.1.1 and the solution of the LDS approximation introduced in Section 3.1.1 are equal, up to higher order terms and multiplicative constants which are used for computational efficiency.

For simplicity, the analysis is performed for the case $\frac{H}{h} = 2$. It is assumed that the fine and coarse grid spatial discretizations commute, i.e., $[L^H, L^h] = 0$; and that intergrid transfers introduce no error. By an abuse of notation these transfers are omitted from the analysis in this section. Nevertheless, whenever the fine and coarse grid solutions appear in the same formula, it should be understood that an appropriate restriction of the fine grid solution to the coarse grid

21

is implied. Similarly, whenever the fine and coarse grid operators appear in the same formula it stands for a restriction of the fine grid operator to the coarse grid. The properties of these transfers necessary to guarantee the algorithm performance are discussed in Section 6.1.

In Section 3.1.1 an error bound was derived for a semi-discrete LDS approximation of general degree d. This approximation which is given by equations (3.3)–(3.4) can be succinctly written as,

$$\frac{dv_{i,d}}{dt} = L^H v_{i,d} + v_{i+1,d} \qquad \text{for } i \leq 0 < d \tag{5.3}$$

$$\frac{dv_{d,d}}{dt} = L^H v_{d,d}$$

with initial conditions

$$v_{i,d}(x,0) = \left(L^h - L^H\right)^i u_0(x) \tag{5.4}$$

where $u_0(x)$ is evaluated at grid points. The solution of this system is given by,

$$v_{i,d}(x,t) = \left[\sum_{k=0}^{d-i} \left(L^h - L^H\right)^{k+i} \frac{t^k}{k!}\right] e^{L^H t} u_0(x) \qquad \text{for } 0 \leq i \leq d \tag{5.5}$$

For an LDS algorithm of degree one denote by u_{LDS}, τ the variables approximating $v_{0,1}, v_{1,1}$, respectively. At time Δt the solution of the LDS approximation is given by,

$$v_{0,1}(x,\Delta t) = \left[1 + \Delta t \left(L^h - L^H\right)\right] e^{L^H \Delta t} u_0(x) \tag{5.6}$$

$$v_{1,1}(x,\Delta t) = e^{L^H \Delta t} \left(L^h - L^H\right) u_0(x)$$

For linear problems, the time marching operator E^H approximates $e^{L^H \Delta t}$. Therefore, in the following semi-discrete analysis $e^{L^H \Delta t}$ will denote the time-stepping on grid H for time Δt. Assuming $u(x,\Delta t) = O(1)$, then, at the end of the initialization phase the variables u_{LDS}, τ satisfy

$$u_{LDS}(x,\Delta t) = e^{L^h \Delta t} u_0(x) = \left[1 + \Delta t \left(L^h - L^H\right) + \frac{(\Delta t)^2}{2}\left(L^h - L^H\right)^2\right] e^{L^H \Delta t} u_0(x) + \text{h.o.t}$$

$$= v_{0,1}(x,\Delta t) + O\left((\Delta t)^2 \left(L^h - L^H\right)^2\right) \tag{5.7}$$

$$\tau(x,\Delta t) = \left(e^{L^h \Delta t} - e^{L^H \Delta t}\right) u_0(x)$$

$$= \Delta t \left\{\left(L^h - L^H\right) + \frac{\Delta t}{2}\left(L^h - L^H\right)^2\right\} e^{L^H \Delta t} u_0(x) + \text{h.o.t}$$

$$= \Delta t \left\{v_{1,1}(x,\Delta t) + O\left(\Delta t \left(L^h - L^H\right)^2\right)\right\} \tag{5.8}$$

The factor Δt is maintained to reduce the number of multiplications during the time-stepping stage. It can be seen that, up to higher order expressions, the algorithm correctly initializes the first correction term.

For an LDS algorithm of degree two denote by u_{LDS}, τ, φ the variables approximating $v_{0,2}, v_{1,2}$ and $v_{2,2}$, respectively. At time $2\Delta t$ the solution of the LDS approximation is given by,

$$
\begin{aligned}
v_{0,2}(x, 2\Delta t) &= \left[1 + 2\Delta t\left(L^h - L^H\right) + 2(\Delta t)^2\left(L^h - L^H\right)^2\right] e^{L^H 2\Delta t} u_0(x) \\
v_{1,2}(x, 2\Delta t) &= \left[1 + 2\Delta t\left(L^h - L^H\right)\right] e^{L^H 2\Delta t}\left(L^h - L^H\right) u_0(x) \qquad (5.9) \\
v_{2,2}(x, 2\Delta t) &= e^{L^H 2\Delta t}\left(L^h - L^H\right)^2 u_0(x)
\end{aligned}
$$

It can be easily seen that at the end of the initialization procedure the variables u_{LDS}, τ, φ satisfy,

$$
\begin{aligned}
u_{LDS}(x, 2\Delta t) &= e^{L^h 2\Delta t} u_0(x) = v_{0,2}(x, 2\Delta t) + O\left((\Delta t)^3\left(L^h - L^H\right)^3\right) \qquad (5.10) \\
\tau(x, 2\Delta t) &= e^{L^h \Delta t}\left(e^{L^h \Delta t} - e^{L^H \Delta t}\right) u_0(x) = \left[e^{(L^h - L^H)2\Delta t} - e^{(L^h - L^H)\Delta t}\right] e^{L^H 2\Delta t} u_0(x) \\
&= \Delta t\left\{\left(L^h - L^H\right) + \frac{3\Delta t}{2}\left(L^h - L^H\right)^2\right\} e^{L^H 2\Delta t} u_0(x) + \text{h.o.t} \\
&= \Delta t\left\{v_{1,2}(x, 2\Delta t) + O\left(\Delta t\left(L^h - L^H\right)^2\right)\right\} \qquad (5.11) \\
\varphi(x, 2\Delta t) &= \left(e^{L^h \Delta t} - e^{L^H \Delta t}\right)^2 u_0(x) \\
&= (\Delta t)^2\left\{\left(L^h - L^H\right)^2 + \Delta t\left(L^h - L^H\right)^2\left(L^h + L^H\right)\right\} e^{L^H 2\Delta t} u_0(x) + \text{h.o.t.} \\
&= (\Delta t)^2\left\{v_{2,2}(x, 2\Delta t) + O\left(\Delta t\left(L^h - L^H\right)^3\right)\right\} \qquad (5.12)
\end{aligned}
$$

It can be seen the algorithm correctly initializes the correction terms in the LDS of degree two, up to higher order terms and multiplicative constants.

5.1.3 Full Approximation Scheme LDS

The FAS form of the LDS is appropriate for both linear and nonlinear problems. However, since the Correction Scheme has a simpler form and necessitates less modifications to the code, it is more conveniently used for linear problems.

In the present work only LDS in FAS form of degree one was implemented , and it will be described for a homogeneous system of equations.

Recall, the Full Approximation Scheme of degree one is given by,

$$
\begin{aligned}
u_t^h &= P^H(u^h) + v^h - u^h \qquad (5.13) \\
v_t^h &= P^H(v^h) + v^h - u^h
\end{aligned}
$$

where P^H may be either a linear or nonlinear operator and $v^h = u^h + \tau$, with τ corresponding to the first correction term in the Correction Scheme form.

The algorithmic implementation is slightly more involved than the Correction Scheme as it requires some modification of the time marching procedure. Denote by $\widetilde{E}_1^{H,K}$ the integration scheme obtained by modifying the coarse grid operator $E^{H,K}$ to time advance the LDS system of degree one (5.13).

The algorithm has the following form

```
        LDS Method of Degree one , Full Approximation Scheme
```

$$\text{Initialize} \quad V_{0,1}^0$$
$$N = 0$$
$$\text{While } N \leq \lceil \tfrac{T}{K} \rceil \text{ Do}$$
$$\quad \text{Call Initialize}(V_{0,1}^N, V_{1,1}^N)$$
$$\quad \text{For i =1,\dots,Revisit Do}$$
$$\quad\quad \text{Solve } V_1^{N+1} = \widetilde{E}_1^{H,K} V_1^N,$$
$$\quad\quad \text{Set} \quad N = N + 1$$
$$\quad \text{End}$$
$$\text{End}$$

Here, V_1 denotes the vector $(V_{0,1}, V_{1,1})$. For presentation simplicity, the initialization procedure is described for the case $\frac{H}{h} = 2$. Generalizations to the case $\frac{H}{h} = 2^l$ are identical to those for the Correction Scheme. In Section 5.1.2 it was shown (see Eq. (5.8)–(5.7)), that after the Correction Scheme initialization $V_{0,1}^1$ contains the fine grid solution $u^h(x, \Delta t)$ and $V_{1,1}^1$ contains $\Delta t \, \tau(x, \Delta t)$. Thus, the FAS initialization consists of the Correction Scheme initialization supplemented with the additional computation of $(u^h + \tau)(x, \Delta t)$ at the end of the procedure. For completeness, the whole initialization procedure for the Full Approximation Scheme is listed below

$$\text{Initialize}(V_{0,1}^N, V_{1,1}^N)$$

$$\text{Set } U^N = I_{H,K}^{h,k} V_{0,1}^N$$
$$\text{Solve } U^{N+\frac{m}{2}} = E^{h,k} \, U^{N+\frac{m-1}{2}}, \qquad\qquad \text{m=1,2}$$
$$\qquad\quad \widetilde{V}_{0,1}^{N+1} = E^{H,K} \, V_{0,1}^N$$
$$\text{Set} \quad \widetilde{V}_{1,1}^{N+1} = I_{h,k}^{H,K} U^{N+1} - \widetilde{V}_{0,1}^{N+1}$$
$$\qquad\quad V_{0,1}^{N+1} = \widetilde{V}_{0,1}^{N+1} + \widetilde{V}_{1,1}^{N+1},$$
$$\qquad\quad V_{1,1}^{N+1} = V_{0,1}^{N+1} + \tfrac{1}{\Delta t} \widetilde{V}_{1,1}^{N+1},$$
$$\text{Set} \quad N = N + 1$$

The generalization to higher degree LDS is straightforward, instead of the original variables $(V_{0,d}, V_{1,d}, \dots, V_{d,d})$ a new set of variables $(V_{0,d}, V_{0,d} + V_{1,d}, \dots, V_{0,d} + \cdots + V_{d,d})$ is introduced. The equations satisfied by these new variables can be determined from the equivalence between the Full Approximation Scheme and the Correction Scheme for linear problems.

5.1.4 Treatment of Boundary conditions

The LDS treatment of the boundary conditions will be discussed under a restrictive commutativity assumption, which at this stage we do not know how to dispose.

Assume u^h satisfies the boundary condition

$$M u^h + N u^h = g \tag{5.14}$$

If $[M, N] = 0$, then $\tau = N u^h$ satisfies,

$$M \tau = M N u^h = N M u^h = N g - N^2 u \tag{5.15}$$

If $M\tau \ll \tau$ the right hand side term may be neglected and τ satisfies the boundary condition

$$M \tau = 0 \tag{5.16}$$

Otherwise, when higher degree LDS is employed, the τ_j satisfy the boundary conditions

$$M \tau_1 + \tau_2 \ = \ g_1 \tag{5.17}$$
$$\vdots$$
$$M \tau_k \ = \ 0 \tag{5.18}$$

where

$$\tau_j \ = \ N^j \tau \tag{5.19}$$
$$g_j \ = \ N^j g \tag{5.20}$$

Thus, a large error at the boundary discretization may require adding correction terms and corresponding equations in the whole domain.

The assignment of the appropriate boundary conditions to the correction terms when the commutativity assumption does not hold should be further investigated.

The presence of non-periodic boundary conditions may pose problems even when the commutativity assumption holds. This is due to errors introduced by the one-sided high order interpolation near the boundaries. These large and localized errors excited during the initialization phase might not be damped before the next visit to the fine grid; resulting in an error which grows exponentially in the number of visits to the fine grid. At this stage of research, it seems that one should use the differential equation to design appropriate near boundary interpolation with reduced errors (see Sec. 7 for an example).

5.2 Richardson Extrapolation

The simultaneous time stepping on two grids during the initialization phase can be used to estimate the local truncation error. This estimate can be used in various ways. In [1, 2, 3] it was used to implement adaptive mesh refinement for hyperbolic equations. In the multigrid method this estimate is used for the τ *extrapolation* technique which applies a weighted transfer of the correction term to obtain higher accuracy using a lower order scheme [4] .

For simplicity, let Q_h be a two-level explicit difference operator. If the solution is smooth enough, the local truncation error is

$$u(x, t + k) - Q_h u(x, t) \ = \ k \left[k^{q_1} a(x, t) + h^{q_2} b(x, t) \right] + k O(k^{q_1 + 1} + h^{q_2 + 1}) \tag{5.21}$$
$$= \ \tau + k O(k^{q_1 + 1} + h^{q_2 + 1}) \tag{5.22}$$

where the leading term is denoted by τ. If u is smooth enough, then if one takes two time steps with the method Q_h, the leading error is 2τ. That is,

$$u(x, t + 2k) - Q_h^2 u(x, t) = 2\tau + kO(k^{q_1+1} + h^{q_2+1}) \tag{5.23}$$

Let Q_{2h} be the same difference scheme as Q_h but based on mesh width of $2h$ and $2k$. Also, assume that the order of accuracy in time and space are equal, $q_1 = q_2 = q$. Then

$$u(x, t + 2k) - Q_{2h} u(x, t) = (2k) \left[(2k)^q a(x, t) + (2h)^q b(x, t) \right] + O(h^{q+2}) \tag{5.24}$$
$$= 2^{q+1} \tau + O(h^{q+2}) \tag{5.25}$$

Since $u(x, t + 2k) - Q^2 u(x, t) \approx 2\tau$, forming the difference

$$\frac{Q_h^2 u(x, t) - Q_{2h} u(x, t)}{2^{q+1} - 2} = \tau + O(h^{q+2}) \tag{5.26}$$

gives an estimate of the local truncation error at time t. In other words, the difference between the solution on grid $(2h, 2k)$ and (h, k) uses to estimate of the local truncation error.

This procedure has several advantages. First, it is not necessary to know the exact form of the truncation error to apply it. The error estimation procedure is independent of the difference method. The restriction of this method, that the accuracy in time and space should be the same, is not a severe one. Many popular finite difference methods share this property, for example, second order methods like Lax Wendroff or MacCormack's method and Leap Frog, and first order method such as upstream differencing. For methods where the accuracy in space and time is not the same, a more expensive variant of this procedure is possible. For example, one could estimate the spatial and temporal error separately: first keep k constant and take a step based on $2h$ differences, then keep h constant and take a step with time step $2k$. Other variations are possible. In the present work the LDS with Richardson extrapolation was employed only for schemes with the same spatial and temporal accuracy. The usefulness of this approach applied to discretizations without this property should be further investigated.

The initialization step of the LDS method computes the term $Q_h^2 u - Q_{2h} u$ and uses it as the initial value of the correction equation on the next coarser grid. Taking

$$\frac{2^q}{2^q - 1} \left(Q_h^2 u - Q_{2h} u \right) \tag{5.27}$$

as the initial value will yield an $O(h^{q+1})$ approximation on the coarse grid.

The initialization of the extrapolated LDS of degree one is performed directly from the finest grid. The extra cost associated with this initialization is compensated by the added accuracy.

Richardson extrapolation is based on Taylor expansion of the error and is valid only for smooth data. For non-smooth solutions, this procedure is incorrect and might lead to an error that grows exponentially with the number of visits to the fine grid. Therefore, a great care should be taken when considering this method, to ensure that the discretization is dissipative enough to prohibit any undesired growth. Nevertheless, since the dissipative schemes employed by the LDS damp the oscillatory components (see Sec. 6.2), this technique might give excellent results when properly used (see Figure 12).

5.3   Work Considerations

The amount of computational work and the memory requirement in a cycle of an LDS of degree m will be evaluated and compared with the corresponding requisites on the finest grid in the cycle.

The simultaneous time-stepping on the finer grids during initialization constitutes a large fraction of the algorithm computational cost and dominates its storage requirements. There is a trade off between the storage requirements and the efficiency in the two initialization procedures described in Section 5.1. In the present work the computationally efficient simultaneous initialization was employed since only moderate storage was needed. However, when storage is limited, direct initialization might be favored. In this section, only the efficiency of the simultaneous scheme is analyzed, but the storage requirements are compared for both methods.

In order to simplify analysis, it will be assumed that the finest and coarsest grid meshsizes satisfy $H = 2^l h$; and that a grid is refined by halving its meshsize.

The problem is solved in a d-dimensional space, for $d = 2, 3$. Typically, real world problems occur in 3-dimensional space.

5.3.1   Storage Requirements

The LDS method of degree m employs on the coarsest level $m + 1$ times as many equations as on the finest grid; while on intermediate grids, twice the number of the fine grid equations are solved. The number of points in a spatial grid on any level is 2^d larger than in the next coarser one. In the *simultaneous initialization*, see Figure 3, if all grids are allocated simultaneously the storage requirement is,

$$\left(1 + \frac{m + 1}{2^{dl}} + \sum_{j=1}^{l-1} \frac{1}{2^{dj-1}} \right) S \tag{5.28}$$

where S is the storage required for the finest grid. Noting that at all times merely two grids are time advanced simultaneously, then a careful management of memory may reduce this requirement to

$$\left(1 + \frac{1}{2^{d-1}} + \frac{m + 1}{2^{dl}} \right) S \tag{5.29}$$

These two requisites are equivalent for $l \leq 2$ (i.e., for $\frac{H}{h} = 2$, or 4) as is the case in the present research. Therefore, this possible small reduction of memory usage will not be further elaborated.

If memory is at premium, storage may be traded for efficiency by using *direct initialization*, see Figure 2. This procedure employs only the finest and coarsest grids with memory requirement of

$$(1 + \frac{m + 1}{2^{dl}}) S \tag{5.30}$$

It should be noted that typically $l = 2$, $d \geq 2$, and due to efficiency considerations the degree satisfies $m \leq 3$ (see Section 5.3.2); thus, the storage overhead associated with the LDS algorithm is fairly small.

5.3.2   Efficiency

The computational cost of fine grid time-stepping relative to the cost of obtaining the same solution using the LDS algorithm will be evaluated. In this estimate, the cost of the intergrid

transfers is neglected, as in many cases it is small relative a fine grid time step. First, the work associated with the initialization of an LDS of degree m is computed. Define

$$I_{l,m}(d) = \left\{ \begin{array}{c} \text{The cost to initialize an LDS of degree } m \text{ using the simultaneous} \\ \text{initialization when the spatial dimension is } d \text{ and } H = 2^l h \end{array} \right\} \qquad (5.31)$$

An inspection of Figure 3 leads to the following formula for $I_{l,m}(d)$,

$$I_{l,m}(d) = 2\,m + \frac{m(m+1)}{2^{dl+1}} + 3\,m \sum_{k=1}^{l-1} \frac{1}{2^{dk}} \qquad (5.32)$$

During initialization a time equal to $m\,2^l$ fine grid time steps is marched. Denote by N the number of coarse grid time steps performed before revisiting the fine grid. The efficiency of an LDS cycle is defined as the computational cost of time-stepping on the fine grid relative to the work required to obtain the same solution on the coarse grid with an LDS cycle. It is given by the formula,

$$\frac{(N+m)2^l}{I_{l,m}(d) + N\,(m+1)\,2^{-dl}} \qquad (5.33)$$

The following table and figure list the LDS efficiency for $H = 4h$ and $d = 2,3$, for various values of N.

Revisit	*Efficiency for 2D problems*			*Efficiency for 3D problems*		
	$m = 1$	$m = 2$	$m = 3$	$m = 1$	$m = 2$	$m = 3$
5	6.98	4.23	3.24	9.42	5.56	4.25
10	10.83	6.35	4.67	16.28	9.11	6.63
15	13.15	8.00	5.82	22.38	12.36	8.83
20	15.81	9.32	6.75	27.85	15.35	10.86
25	17.52	10.41	7.53	32.80	18.09	12.75

Typically to multilevel methods, this algorithm efficiency increases with the problem dimensionality.

6   Fourier Analysis

Fourier analysis is a major tool for the analysis of numerical approximations to hyperbolic equations [17], as well as for analyzing multigrid algorithms [4]. In this section it is employed to obtain necessary conditions for algorithm convergence.

6.1   Properties of the intergrid transfers

The transfer of the solution between the various grids plays a central role in the LDS algorithm. In the initialization stage the solution is first interpolated to the finer grids and after several time steps on these grids is restricted back to the coarsest grid. For an LDS of higher degree, this procedure is repeated for the correction terms. Inevitably, this process introduces errors. Therefore, an appropriate choice of these operators is essential to guarantee the algorithm performances and the desired accuracy of the solution.
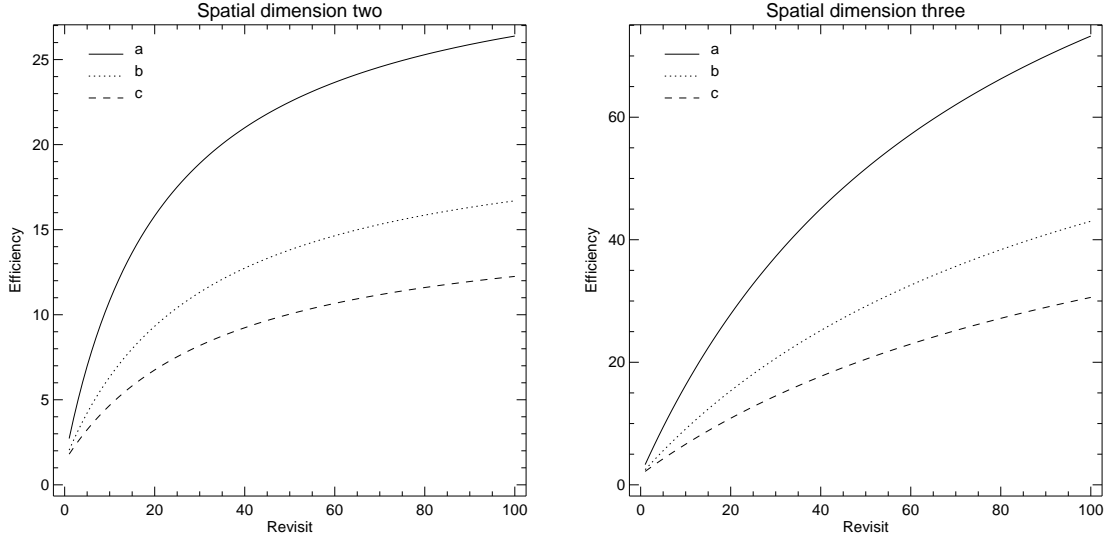
Figure 4: LDS efficiency for 2D and 3D problems when $H = 4h$. (a) LDS of degree one (b) LDS of degree two (c) LDS of degree three.

The analysis will be first performed for first order operators. This is no limitation since every problem may transformed to a first order system by introducing additional variables. Assume that $\lambda = \frac{\Delta t}{\Delta x}$ is fixed on all grids; thus, the temporal error can be expressed in terms of $h = \Delta x$. For simplicity, the analysis will be performed for one dimensional problems with $H = 2h$; generalizations to higher dimensions is straightforward. Assume that the spatial discretization is of order p. Furthermore, assume that the order of the spatial discretization coincides with the order of the full discretization. Thus, for this analysis, the effect of the temporal error on the accuracy may be ignored by investigating a semi-discrete system.

Let the I_H^h, I_h^H be the interpolation and restriction operators, respectively; and let $\hat{I}_H^h(\theta)$, $\hat{I}_h^H(\theta)$ be their corresponding symbols. Assume that for smooth components the intergrid operators satisfy,

$$
\begin{aligned}
\hat{I}_H^h(\theta) &= 1 + c\,\theta^{p_1} + O(\theta^{p_1+1}) \\
\hat{I}_h^H(\theta) &= 1 + c\,\theta^{p_2} + O(\theta^{p_2+1})
\end{aligned}
\tag{6.1}
$$

where throughout this section c is a generic constant. For the harmonic oscillatory component $\theta' = \theta + \pi$ holds,

$$
\begin{aligned}
\hat{I}_H^h(\theta') &= c\,\theta^{q_1} + O(\theta^{q_1+1}) \\
\hat{I}_h^H(\theta') &= c\,\theta^{q_2} + O(\theta^{q_2+1})
\end{aligned}
\tag{6.2}
$$

Denote by $\hat{I}(\theta) = \hat{I}_h^H(\theta)\hat{I}_H^h(\theta)$. Then under the previous assumptions, for smooth components

$$
\hat{I}(\theta) \approx (1 + c\,\theta^{p_1})(1 + c\,\theta^{p_2}) = 1 + c\left(\theta^{p_1} + \theta^{p_2}\right) + \text{h.o.t}
\tag{6.3}
$$

29

For the harmonic oscillatory components holds,

$$\hat{I}(\theta') = c\,\theta^{q_1+q_2} + \text{h.o.t} \tag{6.4}$$

Let $\hat{L}^h(\theta)$ denote the symbol of L^h. For LDS of first degree one wants to compute,

$$\hat{\tau}(\theta, \Delta t) = \left[e^{\hat{L}^h(\theta)\Delta t} - e^{\hat{L}^H(2\theta)\Delta t} \right] \hat{u}_0(\theta) \tag{6.5}$$

Instead one computes,

$$
\left[\hat{I}(\theta)e^{\hat{L}^h(\theta)\Delta t} + \hat{I}(\theta')e^{\hat{L}^h(\theta')\Delta t} - e^{\hat{L}^H(2\theta)\Delta t} \right] \hat{u}_0(\theta) = \hat{\tau}(\theta, \Delta t) + \left[c\,(\theta^{p_1} + \theta^{p_2})e^{\hat{L}^h(\theta)\Delta t} \right.
$$
$$
\left. + \quad c\,\theta^{q_1+q_2}e^{\hat{L}^h(\theta')\Delta t} \right] \hat{u}_0(\theta) + \text{h.o.t} \tag{6.6}
$$

For the low frequencies, if L is the first derivative,

$$\hat{L}^h(\theta) = i\frac{\theta}{h} + ch^p \left(\frac{\theta}{h}\right)^{p+1} + \text{h.o.t} \tag{6.7}$$

For the high frequencies,

$$\left| \hat{L}^h(\theta) \right| = O\left(\frac{1}{h}\right) \tag{6.8}$$

Thus, for this operator,

$$\hat{\tau}(\theta, \Delta t) = c\,\theta^{p+1}\frac{\Delta t}{h}\hat{u}_0(\theta) + \text{h.o.t} \tag{6.9}$$

In order to obtain the desired accuracy, the following inequalities should hold

$$\theta^{p_1} \quad < \quad \theta^{p+1}\frac{\Delta t}{h} \tag{6.10}$$

$$\theta^{p_2} \quad < \quad \theta^{p+1}\frac{\Delta t}{h} \tag{6.11}$$

$$\theta^{q_1+q_2} \quad < \quad \theta^{p+1}\frac{\Delta t}{h} \tag{6.12}$$

In general, for first order hyperbolic equations, $\frac{\Delta t}{h} = O(1)$, thus this term may be neglected and one obtains conditions on the order of the intergrid transfers required to guarantee the algorithm performances

$$p + 2 \quad \leq \quad p_1, p_2 \tag{6.13}$$
$$p + 2 \quad \leq \quad q_1 + q_2$$

It follows from the above bounds that if $\frac{\Delta t}{h} \to 0$, increasingly higher order interpolations will be necessary. Hence, decreasing the time step on a fixed spatial grid will have undesired consequences. In practice, one tries to use a time step as large as possible, thus, this observation poses no real restrictions.

If L is an operator of order m, the previous argument implies that,

$$\hat{\tau}(\theta, \Delta t) = c\,\theta^{p+m}\frac{\Delta t}{h^m}\hat{u}_0(\theta) + \text{h.o.t} \tag{6.14}$$

Therefore, in order to obtain the desired accuracy, the following inequality should hold

$$\theta^{p_1} \quad < \quad \theta^{p+m}\frac{\Delta t}{h^m} \tag{6.15}$$

and similar inequalities should hold for $p_2, (q_1 + q_2)$. This implies the following conditions,

$$p + m + 1 \quad \leq \quad p_1, p_2 \tag{6.16}$$
$$p + m + 1 \quad \leq \quad q_1 + q_2$$

provided $\frac{\Delta t}{h^m} = O(1)$, i.e., it is bounded away from zero.

For an LDS of degree two, φ is initialized to correct τ in the same way that τ is initialized to correct u_{LDS}. Hence, these conditions are sufficient for LDS of degree two, as well. It follows that these conditions are sufficient for an LDS of a general degree m.

These orders of the intergrid transfer are necessary to maintain the scheme accuracy. In practice, one might prefer to employ transfers of order higher than the minimum necessary. Consider, for example, a dissipative scheme which strongly damps the oscillatory components. Thus, each visit to the fine grid somewhat damps the smoother components of the solution through the interpolation error which transfers some of their energy to the fine grid oscillatory components. High order interpolations transfer less energy to these components and therefore might be preferable in such circumstances.

Another undesired property of the intergrid transfer is captured in the following lemma.

Lemma 2: *For an LDS algorithm, if $\hat{I}(\theta) + \hat{I}(\theta') = \epsilon$, with $\epsilon \neq 1$, then initialization procedure introduces an $O(1)$ error in the Fourier component θ.*

Proof : By the assumption, instead of $\hat{\tau}(\theta, \Delta t)$ in Eq (6.9) one computes

$$\left[\hat{I}(\theta)e^{\hat{L}^h(\theta)\Delta t} + \left(\epsilon - \hat{I}(\theta)\right)e^{\hat{L}^h(\theta')\Delta t} \quad - \quad e^{\hat{L}^H(2\theta)\Delta t}\right]\hat{u}_0(\theta) = \hat{\tau}(\theta, \Delta t) + \left[c\left(\theta^{p_1} + \theta^{p_2}\right)e^{\hat{L}^h(\theta)\Delta t}\right.$$
$$+ \quad (\epsilon - 1 - c\,\theta^{p_1} - c\,\theta^{p_2})e^{\hat{L}^h(\theta')\Delta t}\Big]\,\hat{u}_0(\theta) + \text{h.o.t} \tag{6.17}$$

The staggered grid transfers, often, satisfy the condition of Lemma 2 for the oscillatory components. However, since the LDS discretizations are, usually, dissipative (see Sec. 6.2), this result has little practical consequences.

6.2 The Symbol of the LDS cycle

The LDS method employs several operators which should be simultaneously analyzed in order to ensure the proper performance of the algorithm. In the sequel, Fourier analysis will be used to analyze the cycle of an LDS of degree one for constant coefficient equations in one dimensional space with periodic boundary conditions when $H = 2\,h$.

Let E_h, E_H, be the time marching operators on the fine and coarse grids, respectively. Denote the intergrid transfers by I_h^H, I_H^h. The correction term τ is initialized by,

$$\tau^1 = \left(I_h^H\,E_h^2\,I_H^h - E_H\right)u_H^0 \tag{6.18}$$

The solution of the LDS of degree one algorithm, for $n \geq 1$, is given by,

$$u_{LDS}^n \quad = \quad E_H^n u_H^0 + (n-1)E_H^{n-1}\tau^1 \tag{6.19}$$
$$= \quad \left(E_H + (n-1)(I_h^H E_h^2 I_H^h - E_H)\right)u_H^{n-1} \tag{6.20}$$

Let $\widehat{E}_h, \widehat{E}_H, \hat{I}_h^H, \hat{I}_H^h$ denote the symbols of the respective operators. Let $G(\theta, \nu)$ denote the amplification factor of the Fourier component θ in a cycle consisting of ν consecutive steps on the coarse grid before revisiting the fine grid. It is given by

$$G(\theta, \nu) = \| \left\{ \widehat{E}_H(2\theta) + \nu \left[\hat{I}_h^H(\theta)\widehat{E}_h^2(\theta)\hat{I}_H^h(\theta) + \hat{I}_h^H(\theta')\widehat{E}_h^2(\theta')\hat{I}_H^h(\theta') - \widehat{E}_H(2\theta) \right] \right\} \widehat{E}_H^\nu(2\theta)\|$$

The amplification factor of any Fourier component should approximate as closely as possible the analytic growth rate of that component. In particular, when the analytic solution does not grow in time, one would like to guarantee that no Fourier component is amplified by the LDS cycle, i.e., that the amplification factor of the cycle $A(\nu)$ satisfies,

$$A(\nu) = \max_{\theta \in [-\pi, \pi]} G(\theta, \nu) \leq 1 \tag{6.21}$$

Moreover, it would be desirable if the LDS cycle was time-stable for any underlying discretization with this property.

For scalar equations, it can be easily seen that, if $|\widehat{E}_H(2\theta_0)| = 1$, and

$$\hat{\tau}^1(2\theta_0) = \left[\hat{I}_h^H(\theta_0)\,\widehat{E}_h^2(\theta_0)\,\hat{I}_H^h(\theta_0) + \hat{I}_h^H(\theta_0')\,\widehat{E}_h^2(\theta_0')\,\hat{I}_H^h(\theta_0') - \widehat{E}_H(2\theta_0) \right] \hat{u}_0(2\theta_0) \neq 0$$

then

$$\lim_{\nu \to \infty} G(\theta_0, \nu) = \infty \tag{6.22}$$

This is in accordance with the polynomial error bounds derived in Section 3. It should be noted that although the bounds were polynomial, whenever condition (6.21) does not hold, the algorithm exhibits a growth exponential in the number of visits to the fine grid. Thus, the time-stability of the underlying discretization does not necessarily imply time-stability of the LDS scheme. Moreover, if this condition is violated for all $\nu < \nu_0$, (e.g., when the discretization is not dissipative enough), then for revisiting index $\nu < \nu_0$ the more frequently the fine grid is visited the faster the solution blows up.

Dissipation induces an exponential decay of the solution. Therefore, a dissipative scheme will eventually suppress any polynomial growth of the solution, i.e., for such discretizations $A(\nu) < 1$ for large enough ν. However, since the LDS algorithm should maintain the fine grid accuracy, that grid has to be visited sufficiently often. Therefore, the discretization should have enough dissipation to guarantee that condition (6.21) holds for a schedule ν prescribed by the accuracy requirement. Quite often, artificial dissipation should be added to the coarse grid discretization to ensure the cycle is time-stable. This dissipation may be added either during the time marching on the coarse grid, or during the initialization stage as well, in which case it should also be introduced to the fine grid scheme.

The additional dissipation typically results in a dissipative cycle. This might limit the method applicability for truly long integration times. Hence, one should add just enough dissipation to ensure that no Fourier component is amplified by the cycle. This, of course, does not imply that no wavenumber may grow during the cycle; to the contrary, imposing such a restrictive requirement leads to a dissipative cycle of limited practical interest.

In general, the length of the cycle ν and the amount of artificial dissipation should be determined simultaneously to achieve the fine grid accuracy.

6.3 Staggered vs Non-staggered grids

Time dependent problems are often discretized and solved on staggered grids. The dispersion error of discretizations on such grids is, typically, significantly smaller than the error in schemes of the same order on a non-staggered grid. Hence, one can accurately solve on staggered grids for substantially longer integration times. In the sequel the benefits and disadvantages of using such grids when employing the LDS method will be briefly discussed. The exposition is rather general and is demonstrated through a particular example of the linearized Euler equation, which has been investigated in the present research.

A non-staggered Cartesian grid consists of a set of discrete variables defined at its vertices and the system of equations is discretized at these points. In contrast, on a *staggered grid* the various variables are located at different positions in the computational cell and the various equations are evaluated at distinct points of the cell. Figure 5 depicts the variables distribution and the corresponding computational cells for the linearized Euler equation,

$$
\begin{aligned}
u_t &= a \cdot \nabla u + c\, p_x \\
v_t &= a \cdot \nabla v + c\, p_y \\
p_t &= a \cdot \nabla p + c\,(u_x + v_y)
\end{aligned}
\tag{6.23}
$$

where $a = (a_1, a_2)$ is a two dimensional vector. The superiority of staggered grid discretizations
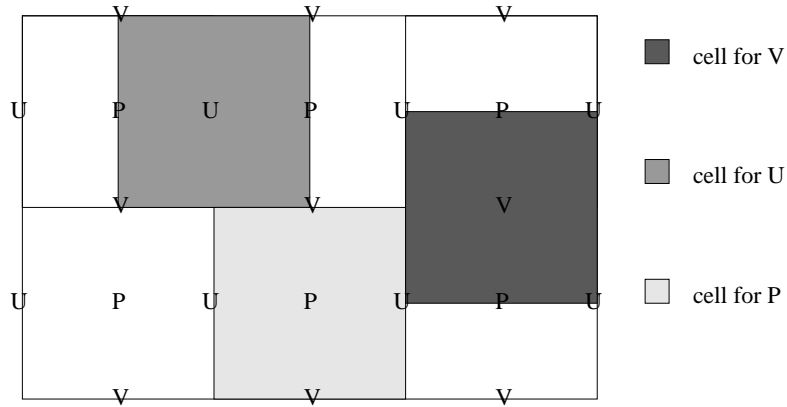
Figure 5: Staggered grid discretization of the linearized Euler equation

to those on non-staggered grid stems from the fact that the symbol of a general order mid-cell discretization of the first derivative has the form $\sum_{k=0}^{m} 2\, i\, \alpha_k \frac{\sin(k+\frac{1}{2})\theta}{h}$ (where α_k depends on the scheme coefficients), while the symbol of a central discretization of this derivative on a regular grid is $\sum_{k=1}^{n} i\beta_k \frac{\sin k\theta}{h}$ (β_k are scheme dependent). The later symbol vanishes at π; hence, poorly approximates the continuous symbol $i\pi$. The phase error of a non-staggered central discretization is larger than that of a mid-cell discretization of the same order over a major fraction of the spectrum of wavelength representable on the grid. Moreover, in non-dissipative full discretizations based on central spatial discretizations the Fourier components in the upper half of the spectrum have negative group velocity and move in a direction opposite to the physical waves. These components can be spuriously excited by discrete boundary conditions or mesh refinements [15, 17]. Therefore, they might deteriorate the computation accuracy even when
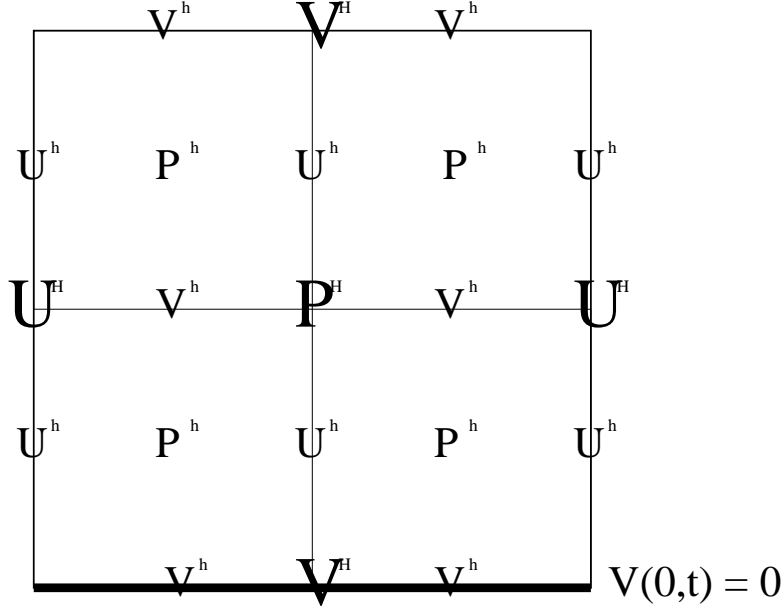
33

Figure 6: Refinement of a staggered grid discretization of the linearized Euler equation with zero Dirichlet boundary condition in the y direction.

they do not occur in the physical problem. In contrast, in mid-cell discretizations all Fourier components have positive group velocity. A considerable drawback of the staggered grid stems from the variables placement at different positions in the computational cell which does not allow implementation of characteristic boundary conditions.

The small phase error of staggered grid discretizations and consequently the significantly less artificial dissipation required to suppress the polynomial growth of the LDS error give rise to highly efficient algorithms applicable for very long integration time. The efficiency obtained by the LDS algorithm for the linear acoustics equation solved on staggered grid was 25 in 2D and 66 in 3D, (see Figures 15 and 16).

The staggered grid has two drawbacks associated with the LDS implementation. Both of them occur in the model problem investigated here and are associated with the intergrid transfers. The first, is that the intergrid transfers satisfy the condition of Lemma 2. Therefore, for some high frequencies there is a $O(1)$ error and enough dissipation should exist to eliminate this error. The second problem is that the solution interpolation to the fine grid requires extrapolation in the cells nearest to the boundary (see Figure 6). This extrapolation strongly amplifies the oscillatory components in the solution and a way had to be found to circumvent this effect for our model problem. Despite these deficiencies, for the problems investigated in this work the staggered grid supports substantially better results than the regular grid.

7   Numerical Results

The LDS method was introduced in the previous sections and its properties were analyzed. The numerical examples presented in this section aim at demonstrating the method strength, as well

as supporting the claims and analysis performed in previous sections.

All the examples in this work have spatial dimension two. However, generalization to higher dimensional problems is straightforward and was avoided due to the heavy computational cost of such simulations.

In all examples $\frac{H}{h} = 4$ with $H = \frac{1}{32}$ and $h = \frac{1}{128}$; in Example 6 finer grids were used as well.

The notation $\text{LDS}(d = a, \gamma = b)$ used in this section denotes an LDS algorithm of degree a which revisits the fine grid after performing b time steps on the coarse grid with the LDS system.

In all relevant plots, the error is normalized with respect to the solution norm, e.g, the coarse grid error is normalized by $\frac{\|I^H U - u^H\|}{\|I^h U\|}$, where $I^H U$ denotes a restriction of the exact solution to the coarse grid. Similarly, the relative error which measures the error in an approximation to the fine grid solution is normalized by the latter solution norm; e.g., the coarse grid relative error is normalized by $\frac{\|u^h - u^H\|}{\|u^h\|}$, where u^h stands for a restriction of the fine grid solution to the coarse grid.

7.1 The Advection Equation

The first set of examples consists of solving the advection equation on the domain $[0,1] \times [0,1]$ with periodic boundary conditions. The equation is given by,

$$u_t - a(x,y)\, u_x - b(x,y)\, u_y = 0 \tag{7.1}$$

In case $a(x,y), b(x,y)$ are constant, an explicit solution of this equation with initial data

$$u(x,y,0) = u_0(x,y) \tag{7.2}$$

is given by

$$u(x,y,t) = u_0(x + at, y + bt) \tag{7.3}$$

Two instances of this equation are solved, the constant coefficient equation with,

$$a(x,y) = 1 \tag{7.4}$$
$$b(x,y) = 0.3 \tag{7.5}$$

and the variable coefficient equation with,

$$a(x,y) = 1 + 0.3\,\sin 2\pi x \tag{7.6}$$
$$b(x,y) = 0.3\,(1 + 0.4\,\sin 2\pi y) \tag{7.7}$$

In this set of examples, except Example 6, the spatial discretization was second order upwind and integration was performed by third order Runge-Kutta. In all these examples $\frac{k}{h} = \frac{K}{H} = 0.3$.

In this set of examples the solution was restricted to the coarse grids by injection, and unless otherwise specified, quintic interpolation was employed.

Example 1. The LDS yields the fine grid solution.   Throughout this work, it was claimed that the LDS method yields the fine grid solution on the coarse grid. Clearly, these solutions may not be identical; rather, one would like to ensure that the error norm in the LDS solution relative

to the fine grid solution is similar to the error norm in the fine grid solution relative to the exact solution, that is,

$$\frac{\|I_h^H u^h - u_{LDS}\|}{\|I_h^H u^h\|} \approx \frac{\|I^h U - u^h\|}{\|I^h U\|} \tag{7.8}$$

where $I^h U$ denotes a restriction of the analytic solution to the fine grid. Figure 7 demonstrates that the LDS yields the fine grid solution in the sense defined by (7.8).

Example 2. The LDS effective integration time. Figure 8 shows two examples of the LDS applied to the variable coefficient advection equation, for smooth and oscillatory data. It can be readily seen that the effective integration time the discretization can be used (i.e., the integration time when the error is small) drastically decreases for oscillatory data. This figure should be born in mind as a reference, since in some of the next examples the same equation with these initial data are used for integration times significantly longer than the LDS effectiveness time, in order to emphasize difference between solutions for various parameters.

Example 3. Direct and simultaneous initialization are practically indistinguishable. In Section 5.1.1 two initialization procedures for the correction terms were presented. These methods vary in their computational cost and memory requirements. Figure 9 compares the direct and simultaneous initialization procedures. It can be seen that at a time longer than the algorithm effectiveness time, the LDS solutions are hardly distinguishable and the observable difference between these solutions is significantly smaller than the error in the LDS approximation relative to the fine grid solution. Thus, it may be concluded that these procedures have essentially the same accuracy.

Example 4. The necessary and desirable orders of the intergrid transfers. In Section 6.1 the orders of the intergrid transfers required to ensure the algorithm performances were analyzed. This analysis implies that for a second order scheme the interpolation should be at least cubic for an LDS of general degree d. The injection introduces no error to the high frequencies, thus is of infinite order. Figure 10 displays an LDS of degree one and two (on the left and right, respectively) with linear, cubic and quintic interpolations. It can be seen, for both approximations, that linear interpolation results in an approximation worse than the coarse grid solution. Cubic interpolation is sufficient to guarantee the LDS performances, however, quintic interpolation yields significantly better results. This phenomena might be due to the interpolation that transfers some of the smooth components energy to the fine grid high frequencies which are strongly damped on the fine grid. This interpolation error is larger for the cubic interpolation.

Example 5. The efficiency of the LDS of various degrees. An LDS approximation of higher degree provides a better approximation to the fine grid solution than a low degree one. Hence, it may be used for longer integration time before the fine grid should be revisited. On the other hand, for such a scheme the initialization procedure as well as the coarse grid time-stepping are more costly.

Figure 11 addresses the question which approximation is more cost effective, a low or a high degree one. In this figure LDS algorithms of various degrees with the same efficiencies are compared for both smooth and nonsmooth data and for diverse costs. The efficiency of LDS($d = 1, \gamma = 8$),

LDS($d = 2, \gamma = 20$) and LDS($d = 3, \gamma = 41$) is 9.4; while that of LDS($d = 1, \gamma = 20$) and LDS($d = 2, \gamma = 79$) is 15.8. In order for an LDS of degree three to achieve efficiency of 15 the fine grid should be visited once in 470 coarse grid steps, while an efficiency of 15.8 can not be achieved even if the fine grid is visited once in 1000 time steps. This suggests that high efficiency can not be achieved with high degree LDS. The efficiency of LDS($d = 3, \gamma = 350$) is only 14.6. For short integration times this efficiency can not be achieved with LDS of degree three since not enough time steps are marched during the simulation, e.g., see Figure 11 left.

Figure 11 plots the error in various solutions relative to the fine grid solution. Inspection of Figure 11 reveals that for small relative error, e.g. ~ 0.02 (i.e., $T \leq 1$ for the smooth data example and $T \leq 0.3$ for the oscillatory data), then for smooth data the LDS of degree one provides a better approximation for both the low and high efficiency scheduling used. For more oscillatory data, the degree two LDS yields better results for the same efficiency. This might be explained by the fact that LDS of degree one provides a fairly good approximation for the smoother components, but not for the more oscillatory ones. The LDS of degree three does not seem an appropriate alternative to the lower degree approximations.

It should be noted that in 3D problems, the performance of LDS of degree two significantly improves. Thus, the efficiency of the abovementioned scheduling is : for LDS($d = 1, \gamma = 8$) it is 13.63 , for LDS($d = 2, \gamma = 20$) it is 15.34 and for LDS($d = 3, \gamma = 41$) it is 18. Moreover, for LDS($d = 1, \gamma = 20$) it is 27.9, and for LDS($d = 2, \gamma = 79$) it is 38.11. Thus, although in 2D problems there seems to be little advantage to use second degree LDS rather than first; in 3D, this changes drastically. If these schedules yield similar results for a similar equation in 3D, then the LDS of degree two is more cost effective for these problems than the first degree algorithm.

It should be born in mind, that the performances of LDS of various degrees might change depending on the equations or on the data. Therefore, it is hard to give conclusive recommendations which method to prefer; and each problem should be investigated separately.

Example 6. Richardson Extrapolation. In Section 5.2 the LDS method with Richardson extrapolation was introduced. Although this technique is limited in scope and, hence, should be used with great care; it might be highly beneficial when it is applicable. Figure 12, provides an example when this idea works. It consists of solutions of the constant coefficient advection equation discretized with first order upwind forward Euler method, which is first order in time and space. The equation is solved on fine grids of 128 and 256 points and corresponding coarse grids of 32 and 64 points on which the LDS is solved. In this example the normalized errors are computed with respect to the exact solution. For smooth data, mesh refinement of the finest grid by a factor of two yields a decrease in error by a factor of 1.875 for the first order scheme versus a factor of 5.81 for the LDS with Richardson extrapolation. For nonsmooth data the fine grid error is reduced by a factor of 1.71 while for the extrapolated LDS the factor is 4.03. Thus, the error reduction for the extrapolated LDS is second order both for smooth and oscillatory data.

7.2 The Linearized Euler Equation

The next set of examples involves solution of the linearized Euler equation, given by

$$
\begin{aligned}
p_t &= a(x, y)\, p_x + b(x, y)\, p_y + c(x, y)\,(u_x + v_y) \\
u_t &= a(x, y)\, u_x + b(x, y)\, u_y + c(x, y)\, p_x
\end{aligned}
\tag{7.9}
$$

$$v_t \;\;=\;\; a(x,y)\,v_x + b(x,y)\,v_y + c(x,y)\,p_y$$

This system was solved for various settings. In all of them interpolation and restriction were of sixth order.

Example 7. Staggered vs. non-staggered grid. Figure 13 demonstrates the superiority of the staggered grid discretization over the regular one for the single grid solution and consequently for the LDS approximation. It shows solutions of the variable coefficient linearized Euler equation on $[0,1] \times [0,1]$ domain with periodic boundary conditions. The discretization is second order upwind for the advection terms and second order central for the terms involving $c(x,y)$; the integration was performed with third order Runge-Kutta with $\frac{\Delta t}{\Delta x} = 0.3$. In the fine non-staggered grid solution one can see an oscillatory component which dominates the solution. This component is visible due to a large phase error of the nonsmooth wavelengths on this grid. The error in these components is even more visible in the coarse grid and LDS solutions. Clearly, the LDS is ineffective for this integration time. However, since the error in the non-staggered fine grid solution is already very large, this is not a real drawback. The large dispersive error is also the cause for the little resemblance between the staggered and non-staggered solutions for the same data and integration time. In contrast, all the staggered grid solutions do not have those oscillations, and the LDS provides an excellent approximation to the fine grid solution.

Example 8. The effect of diminishing CFL. In Section 6.1, it was pointed out that when $\frac{\Delta t}{\Delta x} \to 0$, the order of interpolation should increase. Figure 14 demonstrates this phenomenon for the constant coefficient linearized Euler equation on $[0,1] \times [0,1]$ with periodic boundary conditions. Discretization on a staggered grid is second order upwind for the advection term and second order central for the terms involving the c factor. Integration was done by low storage third order Runge-Kutta. It can be seen that for very short integration time an oscillatory error prevails when $\frac{\Delta t}{\Delta x} = 0.01$. Increasing by a factor of 10 the CFL as well as the integration time, eliminates these oscillations, yielding an excellent approximation to the fine grid solution.

Example 9. High efficiency LDS on periodic staggered grid. The example in Figure 15, is a solution of the constant coefficients acoustics equation,

$$
\begin{aligned}
p_t &= u_x + v_y \\
u_t &= p_x \\
v_t &= p_y
\end{aligned}
\tag{7.10}
$$

on the domain $[0,1] \times [0,1]$ with periodic boundary conditions discretized on a staggered grid with the same discretization as in the previous example, with $\frac{\Delta t}{\Delta x} = 0.3$.

   This discretization has only little dissipation through the Runge-Kutta scheme. However, since the mid-cell discretization provides an very good approximation to the differential operator, the coarse grid operator well approximates the fine grid operator. Hence, only a little artificial dissipation should be added. Sixth order artificial dissipation was added to all equations of the form $\epsilon h \Delta^3$, with $\epsilon = 0.005$. The small dispersive error of the mid-cell discretization enabled both very long integration time, as well as very high LDS efficiency of 26 in 2D and 66 in 3D. Note that at this stage the LDS error relative to the fine grid is <u>forty</u> times smaller than the relative error of the coarse grid.

Example 10. High efficiency LDS on non-periodic staggered grid. Figure 16 plots the solution of the acoustics equation on $[0, 1] \times [0, 1]$ with periodic boundary conditions in the x direction for all variables and zero Dirichlet boundary conditions in the y direction for v,

$$V(x, 0, t) = V(x, 1, t) = 0 \tag{7.11}$$

except for this difference, all the other parameters are identical to those in the previous example.

Recall that high order one sided interpolation near the boundary strongly amplifies the oscillatory components. This problem had to be circumvented in this equation for the interpolation in the y direction. The observation that led to a resolution of this difficulty is that for this equation and these boundary conditions p, u are symmetric in the y direction, and v is asymmetric in this direction. These properties were exploited in the design of the intergrid transfers as well as in the introduction of the high order artificial dissipation. The p, u variables were symmetricly extended around the boundary in the y direction and the interpolation and dissipation were calculated for the extended solution. An assymetric extension for the v variable was similarly defined and used.

The assymetry v is follows from the *dual initial data* argument, which asserts that the boundary condition (7.11) may be viewed as requiring that a *dual* solution with the same magnitude but opposite sign will constantly hit the boundary from the exterior of the domain (e.g., see [16]). The symmetry of p in the y direction follows from

$$p_{yt}(x, 0, t) = v_{tt}(x, 0, t) = 0 \tag{7.12}$$

Thus, if initially

$$p_y(x, 0, 0) = 0 \tag{7.13}$$

this symmetry is maintained for later time. A similar argument holds for the boundary condition at $y = 1$. For the u variable,

$$u_{yt}(x, 0, t) = u_{ty}(x, 0, t) = p_{xy}(x, 0, t) = p_{yx}(x, 0, t) = v_{tx}(x, 0, t) = 0 \tag{7.14}$$

Again, if the initial solution was symmetric in this direction, symmetry is preserved. These assumptions hold for the initial data taken in our examples.

It can be seen that the efficiency and accuracy of the LDS algorithm were not affected by the imposition of non-periodic boundary conditions.

Example 11. The effect of low order artificial dissipation. The parameters taken in the example in Figure 17 are identical to those in Figure 15 except for the use of lower (fourth) order of dissipation of the form $\epsilon h \Delta^2$, with $\epsilon = 0.02$. This choice of ϵ yields the same damping of the oscillatory components as the sixth order dissipation used in Example 9. The resulting LDS solution does not provide a satisfactory approximation to the fine grid solution even for integration time significantly shorter than the one used in Example 9 and more frequent visits to the fine grid. It can be concluded that higher order dissipation is indeed essential for the algorithm performances, as lower order dissipation damps too strongly the smooth components.

7.3 The nonlinear Euler equation

Example 12. The LDS method for the nonlinear Euler equation In Figure 18, the nonlinear Euler equation is solved on $[0, 1] \times [0, 1]$ domain with periodic boundary conditions.

This system is given by,

$$
\begin{aligned}
\rho_t \;\; + \;\; & \rho \left(u_x + v_y \right) + u\,\rho_x + v\,\rho_y = 0 \\
u_t \;\; + \;\; & \left(u\,u_x + v\,u_y \right) + \frac{c^2}{\rho}\rho_x = 0 \\
v_t \;\; + \;\; & \left(u\,v_x + v\,v_y \right) + \frac{c^2}{\rho}\rho_y = 0
\end{aligned}
\tag{7.15}
$$

where $p = \rho^\gamma$, $c^2 = \gamma\rho^{\gamma-1}$ and $\gamma = 1.4$. Second order central discretization was used for all terms, with third order low storage Runge-Kutta ($\frac{\Delta t}{\Delta x} = 0.3$). Artificial sixth order dissipation was added with $\epsilon = 0.8$. The interpolation and restriction were of sixth order accuracy.

The LDS efficiency in this example is 17.5 for 2D problems and 32.8 for 3D problems.

8 Conclusions

The Large Discretization Step methods for time dependent problems were presented. First, the LDS approximation was defined. It consists of a system of lower accuracy discretizations approximating a more accurate time dependent discrete operator. Error bounds on this type of approximations to linear problems were obtained for the semi-discrete and fully discrete cases. These estimates hold for both hyperbolic and parabolic equations. The research reported herein aimed at deriving efficient algorithmic implementation of the LDS approximation for hyperbolic equations, a type of equations which were not previously amenable to multigrid methods. A heuristic argument motivated the design of the LDS algorithm for nonlinear problems, as well.

The LDS methods enables to obtain the fine grid accuracy on a coarse grid by adding correction terms to the coarse grid equations, initializing them using the fine grid and solving a system of equations for these terms. The accuracy of the correction terms deteriorates at a rate determined by the coarse grid discretization. However, since their norm is significantly smaller than the solution norm, they may be effectively used for many coarse grid time steps. Thereafter, the fine grid should be revisited to compute new initial data for them. Fourier analysis was employed to analyzed different aspects of the algorithm; in particular, to obtain conditions on the necessary orders of the intergrid transfers.

The resulting algorithm has a typical efficiency of 16 for 2D problems and 28 for 3D equations. This efficiency was achieved for linear problems with periodic and Dirichlet boundary conditions and the for the nonlinear Euler equation with periodic boundary conditions. A particularly good discretization of a linear equation yielded efficiency of 25 in 2D and 66 in 3D problem.

The results presented in this work are very promising as for the potential of the proposed approach to tackle even more complex problems. Still, a lot of research should be carried out to better understand the method abilities and limitations. Several aspects of the LDS algorithm should be further investigated. The frequency the fine grid is visited is important both to the algorithm efficiency and to ensure the accuracy of the resulting solution. It would be highly beneficial if there was a systematic, preferably adaptive, way to determine when the fine grid should be visited. The dissipativity of the LDS cycle is essential to damp the polynomially growing error. For dissipative discretizations, this requirement typically does not pose any problems. Quite often, though, some artificial dissipation should be added to guarantee the algorithm performances.

A general way is required to determine the right amount of artificial dissipation which will suppress the error growth without affecting the algorithm accuracy. Boundary conditions treatment within the LDS does not seem to pose particular problems. Nevertheless, a large variety of such conditions should be implemented and tested. In particular, a general approach should be found to reduce the high interpolation error near the boundaries.

The novel character of these methods opens many research directions. The application of the LDS to systems of conservations laws and shocks calculations should be investigated. Another interesting direction is to investigate possible generalizations of this method. In the present research, one solves for the correction terms of a linear problem the same equation as the solution. This approximation does not always yield the desired results, e.g., consider the discretization of the wave equation discussed in Section 3. In a more general setting, there might be several correction terms each satisfying a different equation. Such a generalization might significantly reduce the simplicity of the present approach; however, it could be applicable to a broader class of equations.

It is expected that the incorporation of the LDS ideas into parabolic solvers would significantly improve their performances, as well. This is another promising research direction.

References

[1] J. Bell, M. Berger, J. Saltzman and M. Welcome, *Three-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws*, SIAM J. Sci. Comput., 15 (1993), pp. 127-138.

[2] M. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comp. Phys., 82 (1989), pp. 64-84.

[3] M. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comp. Phys., 53 (1984), pp. 484-512.

[4] A. Brandt, *Multigrid Techniques: Guide with Applications to Fluid Dynamics*, GMD-Studien Nr 85, Bonn 1984.

[5] A. Brandt and J. Greenwald, *Parabolic Multigrid Revisited*, Int. Series of Num. Math., Birkhaüser, Verlag, Basel (1991).

[6] M. Carpenter, D. Gottlieb and S. Abarbanel, *The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes*, J. Comp. Phys., 108 (1993), pp. 272-295.

[7] M. Carpenter, D. Gottlieb and S. Abarbanel, *Time-Stable Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems : Methodology and Application to High-Order Compact Schemes*, ICASE Report 93-9.

[8] M. Carpenter, D. Gottlieb, S. Abarbanel and W.S Don, *The Theoretical Accuracy of Runge-Kutta Time Discretizations for the Initial Boundary Value Probelm : A Careful Study of the Boundary Error*, ICASE Reprot 93-83.

[9] J. Greenwald, *Multigrid Methods for Sequences of Problems*, Ph.D. Thesis, The Weizmann Institute of Science, 1992.

[10] Z. Haras and S. Ta'asan, *Finite Difference Schemes for Long-Time Integration*, J. Comp. Phys, 114 (1994), pp. 265-279; and ICASE Report 93-25.

[11] Z. Haras and S. Ta'asan, *Large Discretization Step methods for Evolution Equations*, In P.W. Hemker and P. Wesseling, editors, *Proceedings of the fourth European Multigrid Conference, Multigrid Methods IV* , Int. Series on Num. Math., 116, Birkhaüser Verlag, Basel (1993) pp. 233-245.

[12] D. Kishoni and S. Ta'asan, *Improved Finite Difference Method for Long Distance Propagation of Waves*, Review of Progress in NDE, August 1992, La Jolla California.

[13] S. K. Lele, *Compact Finite Difference Schemes with Spectral-like Resolution*, J. Comp. Phys., 103 (1992), pp. 16-42.

[14] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems*, 2nd ed ,(John Wiley & Sons, 1967).

[15] L.N. Trefeten, *Group Velocity in Finite Difference Schemes*, SIAM Review, 24 (1982), pp. 113-136.

[16] L.N. Trefeten, *Instability of Difference Models for Hyperbolic Initial Boundary Value Problems*, Comm. Pure and Applied Math, 37 (1984), pp. 329-367.

[17] R. Vichnevestsky and J. B. Bowels, *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*, (SIAM, 1982), 140pp.
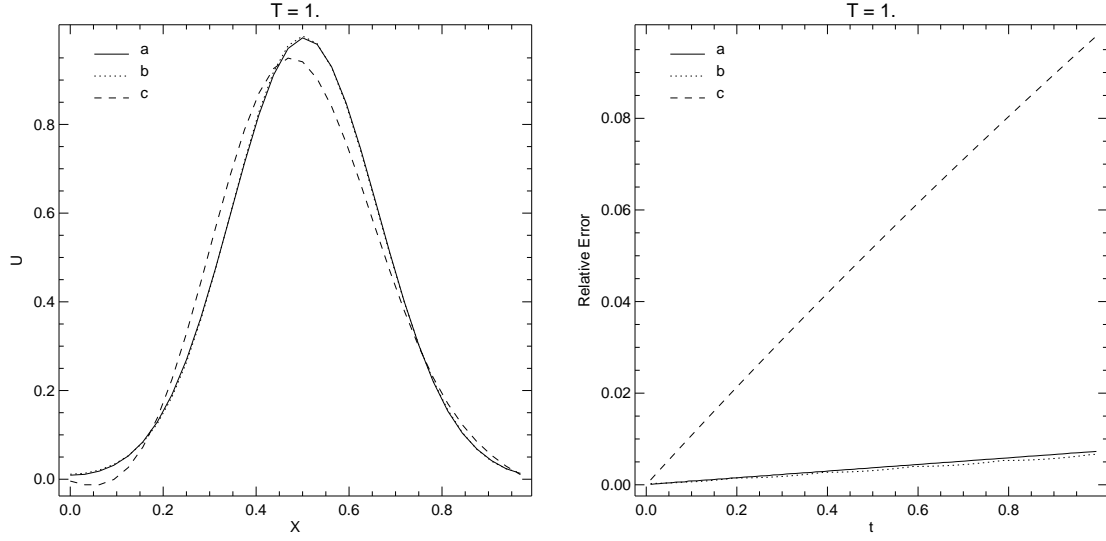
Figure 7: The advection equation, $u_t + u_x + 0.3\,u_y = 0$, with initial data $u_0 = e^{-20(x^2+y^2)}$. Left : (a) fine grid solution (b) the 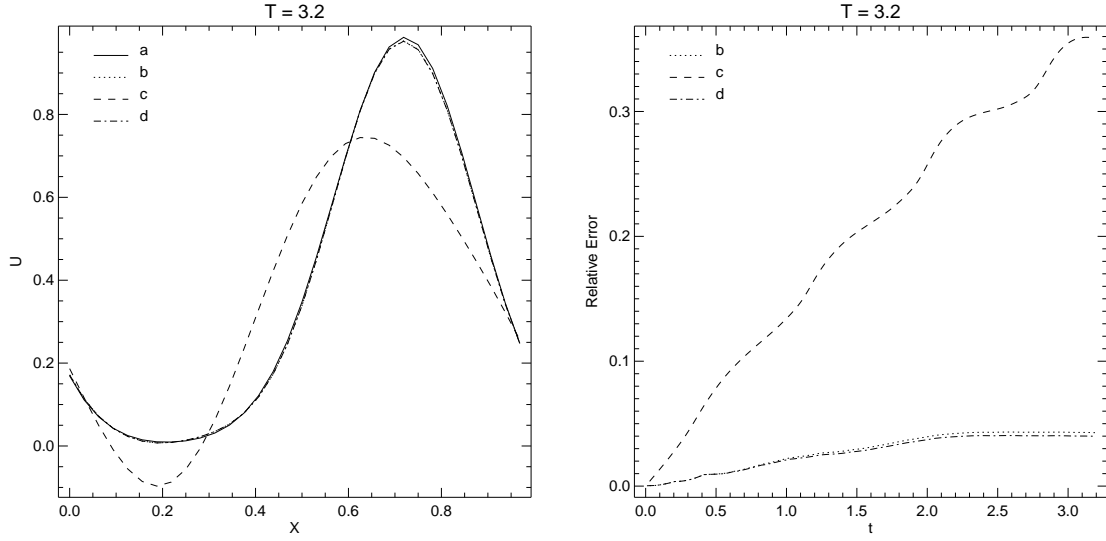LDS($d = 1, \gamma = 20$) solution (c) coarse grid solution. Right : Relative errors. (a) $\|I^h u - u^h\|/\|I^h u\|$, (b) $\|u^h - u^H_{LDS}\|/\|u^h\|$, (c) $\|u^h - u^H\|/\|u^h\|$.
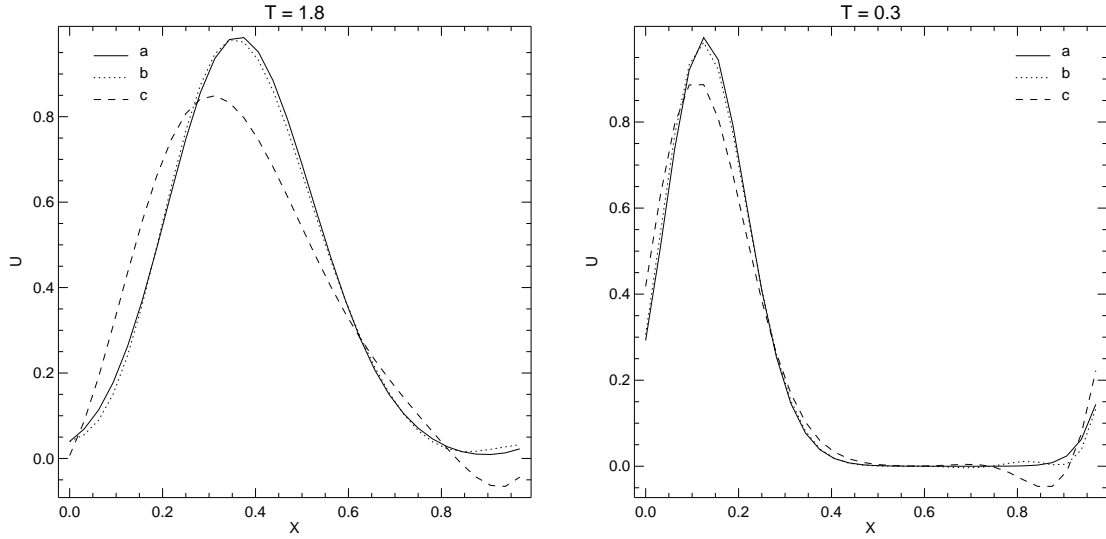
Figure 8: Solutions of $u_t + (1 + 0.3\,\sin(2\pi x))u_x + 0.3\,(1 + 0.4\,\cos(2\pi y))u_y = 0$. Left : Initial data is $u_0 = e^{-20(x^2+y^2)}$. Right : Initial data is $u_0 = e^{-50(x^2+y^2)}$. In both figures : (a) the fine grid solution (b) the LDS($d = 1, \gamma = 20$) solution (c) the coarse grid solution.

Figure 9: The advection equation, $u_t + (1 + 0.3 \sin(2\pi x))u_x + 0.3\,(1 + 0.4\cos(2\pi y))u_y = 0$, with $u_0 = e^{-20(x^2+y^2)}$. Left : (a) the fine grid solution (b) the LDS($d = 2, \gamma = 20$) solution when employing direct initialization. (c) the coarse grid solution (d) the LDS($d = 2, \gamma = 20$) solution when employing simultaneous initialization. Right : Relative errors. (b) relative error of the LDS($d = 2, \gamma = 20$) solution when employing direct initialization. (c) relative error of the coarse grid solution (d) relative error of the LDS($d = 2, \gamma = 20$) solution when employing simultaneous initialization.
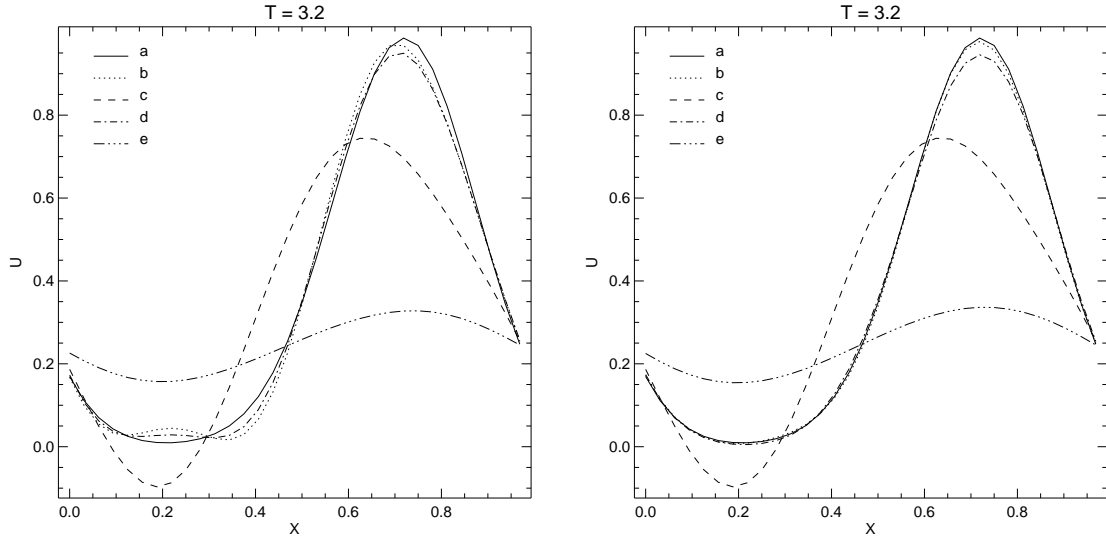
Figure 10: Solution of $u_t + (1 + 0.3 \sin(2\pi x))u_x + 0.3 (1 + 0.4 \cos(2\pi y))u_y = 0$, with $u_0 = e^{-20(x^2+y^2)}$. Left : (a) the fine grid solution (b) the LDS($d = 1, \gamma = 20$) solution when using quintic interpolation (c) coarse grid solution (d) the LDS($d = 1, \gamma = 20$) solution when using cubic interpolation (e) the LDS($d = 1, \gamma = 20$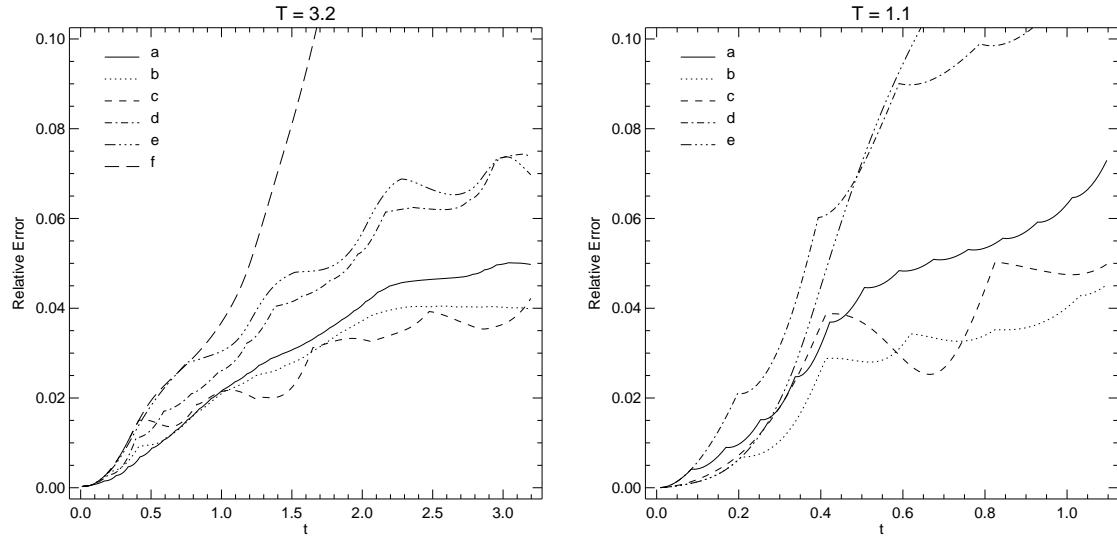) solution when using linear interpolation. Right : (a) the fine grid solution (b) the LDS($d = 2, \gamma = 20$) solution when using quintic interpolation (c) coarse grid solution (d) the LDS($d = 2, \gamma = 20$) solution when using cubic interpolation (e) the LDS($d = 2, \gamma = 20$) solution when using linear interpolation.

Figure 11: Relative error in the solution of $u_t + (1 + 0.3\,\sin(2\pi x))u_x + 0.3\,(1 + 0.4\,\cos(2\pi y))u_y = 0$, with Left: Initial data is $u_0 = e^{-20(x^2+y^2)}$. Right : Initial data is $u_0 = e^{-50(x^2+y^2)}$. In both figures : (a) the LDS($d = 1, \gamma = 8$) solution (b) the LDS($d = 2, \gamma = 20$) solution (c) the LDS($d = 3, \gamm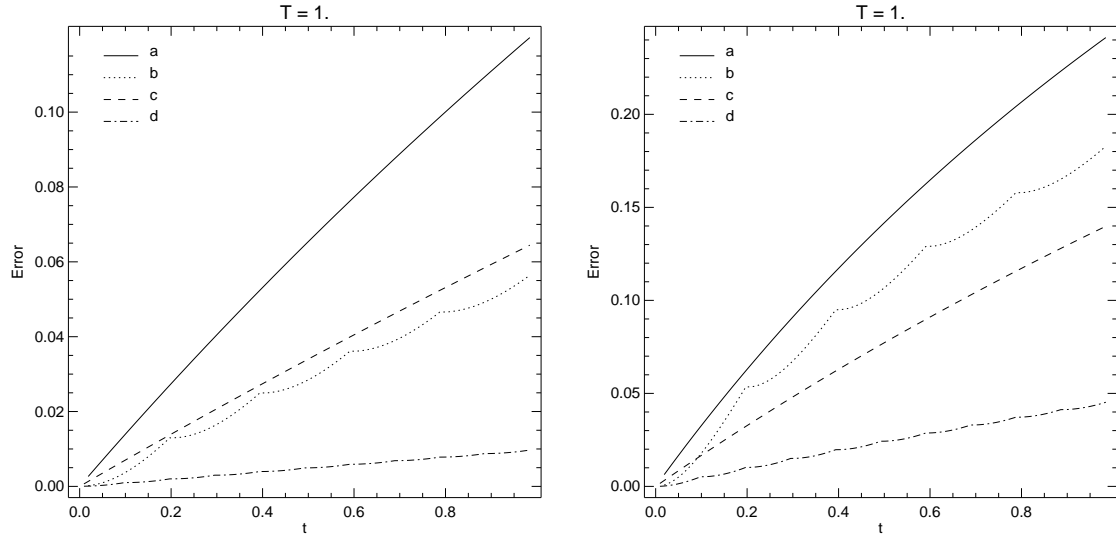a = 41$) solution (d) the LDS($d = 1, \gamma = 20$) solution (e) the LDS($d = 2, \gamma = 79$) solution (f) the LDS($d = 3, \gamma = 350$) solution.

Figure 12: Errors in solutions of $u_t + u_x + 0.3\,u_y = 0$. Left : Initial data is $u_0 = e^{-25(x^2+y^2)}$. Right : Initial data is $u_0 = e^{-60(x^2+y^2)}$. In both figures : (a) error in solution on a 128×128 points grid. (b) error in LDS($d = 1, \gamma = 20$) solution on a 32×32 points grid with Richardson extrapolation. (c) error in solution on a 256×256 points grid. (d) error in LDS($d = 1, \gamma = 20$) solution on a 64×64 points grid with Richardson extrapolation.
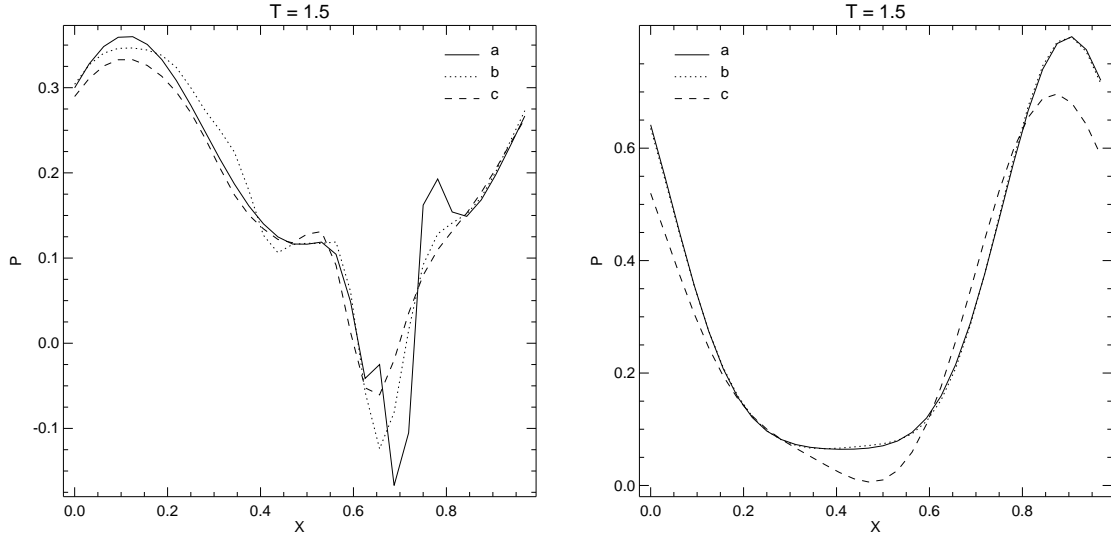
Figure 13: Solution of the variable coefficient linearized Euler equation, with coefficients : $a(x, y) = 0.6 (1 + 0.2 \cos 2\pi x)$, $b(x, y) = 0.4 (1 + 0.4 \cos 2\pi y)$, $c(x, y) = 1 + 0.3 \sin 2\pi x$. Initial data is $p_0 = e^{-30(x^2+y^2)}$, $u_0 = 0$, $v_0 = 0$. Left : non-staggered grid. Right : staggered grid. In both figures : (a) the fine grid solution (b) the LDS($d = 1, \gamma = 20$) solution (c) the coarse grid solution.
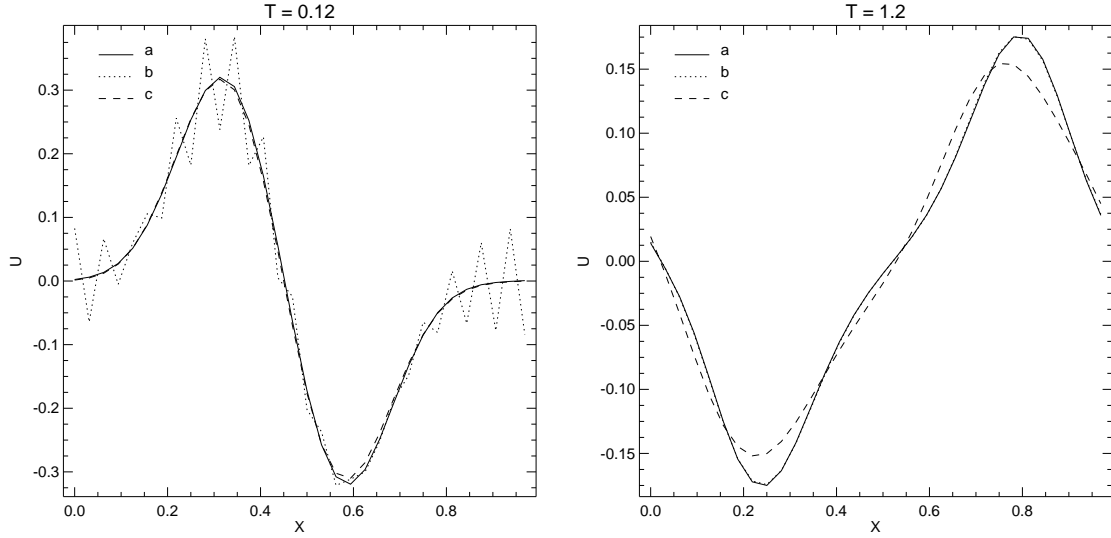

Figure 14: Solution of the constant coefficients linearized Euler equation, with coefficients : $a = 0.4$, b = 0.5, $c = 1$. Initial data is $p_0 = e^{-45(x^2+y^2)}$, $u_0 = 0$, $v_0 = 0$. Left : $\frac{\Delta t}{\Delta x} = 0.01$. Right : $\frac{\Delta t}{\Delta x} = 0.1$. In both figures : (a) the fine grid solution (b) the LDS($d = 1, \gamma = 20$) solution (c) the coarse grid solution.
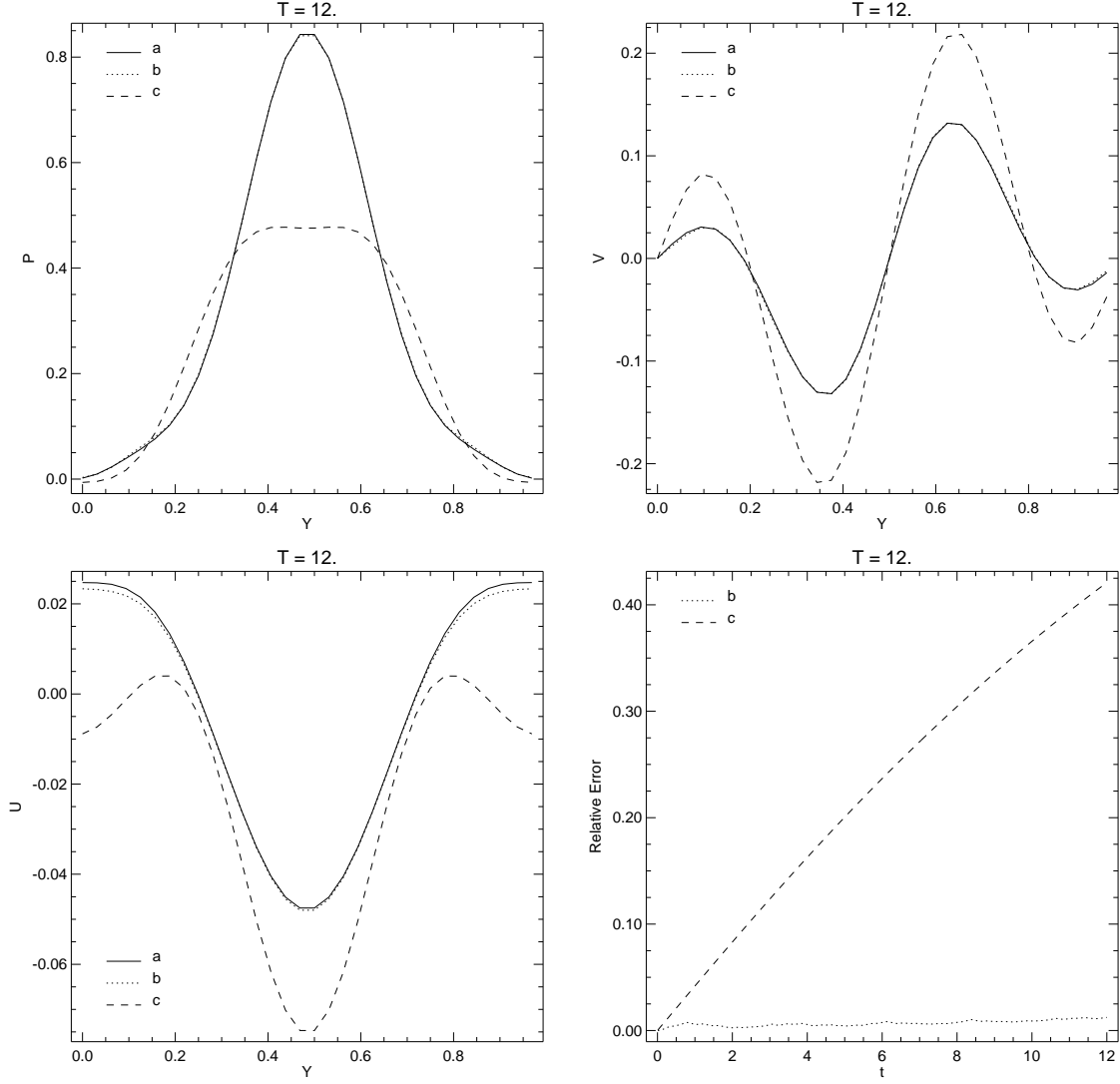
48

Figure 15: The acoustics equation on periodic domain, with initial data $p_0 = e^{-35(x^2+y^2)}$, $u_0 = 0$, $v_0 = 0$. In the plots of the solutions : (a) the fine grid solution (b) the LDS($d = 1, \gamma = 80$) solution when adding sixth order dissipation (c) the coarse grid solution. In the plot of the relative error : (b) the relative error in the LDS($d = 1, \gamma = 80$) solution (c) the relative error in the coarse grid solution.
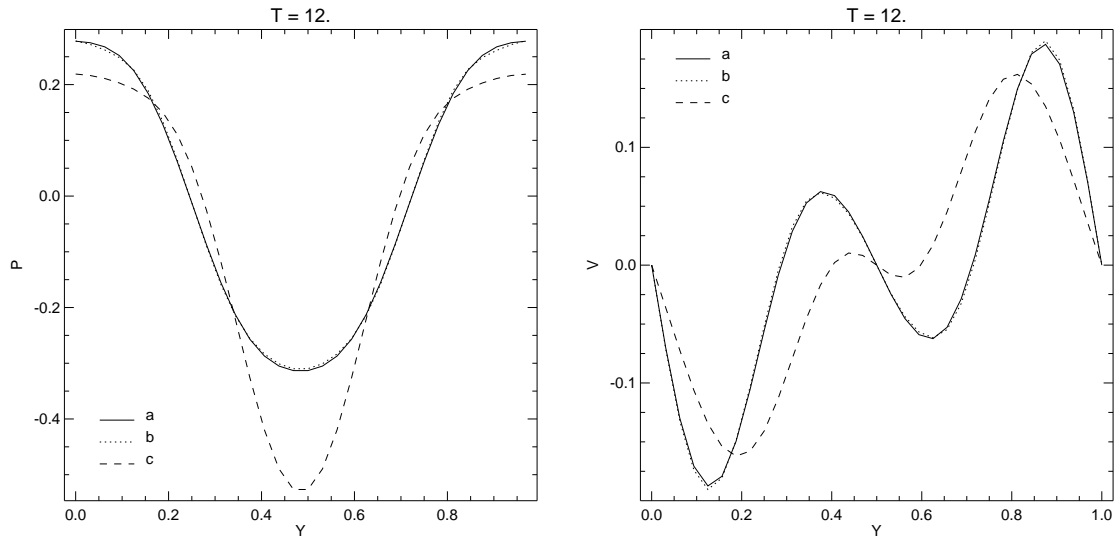
Figure 16: Solution of the acoustics equation with $p_0 = e^{-35(x^2+y^2)}$, $u_0 = 0$, $v_0 = 0$ on non-periodic domain. (a) the fine grid solution (b) the LDS($d = 1, \gamma = 80$) solution when adding sixth order dissipation (c) the coarse grid solution.
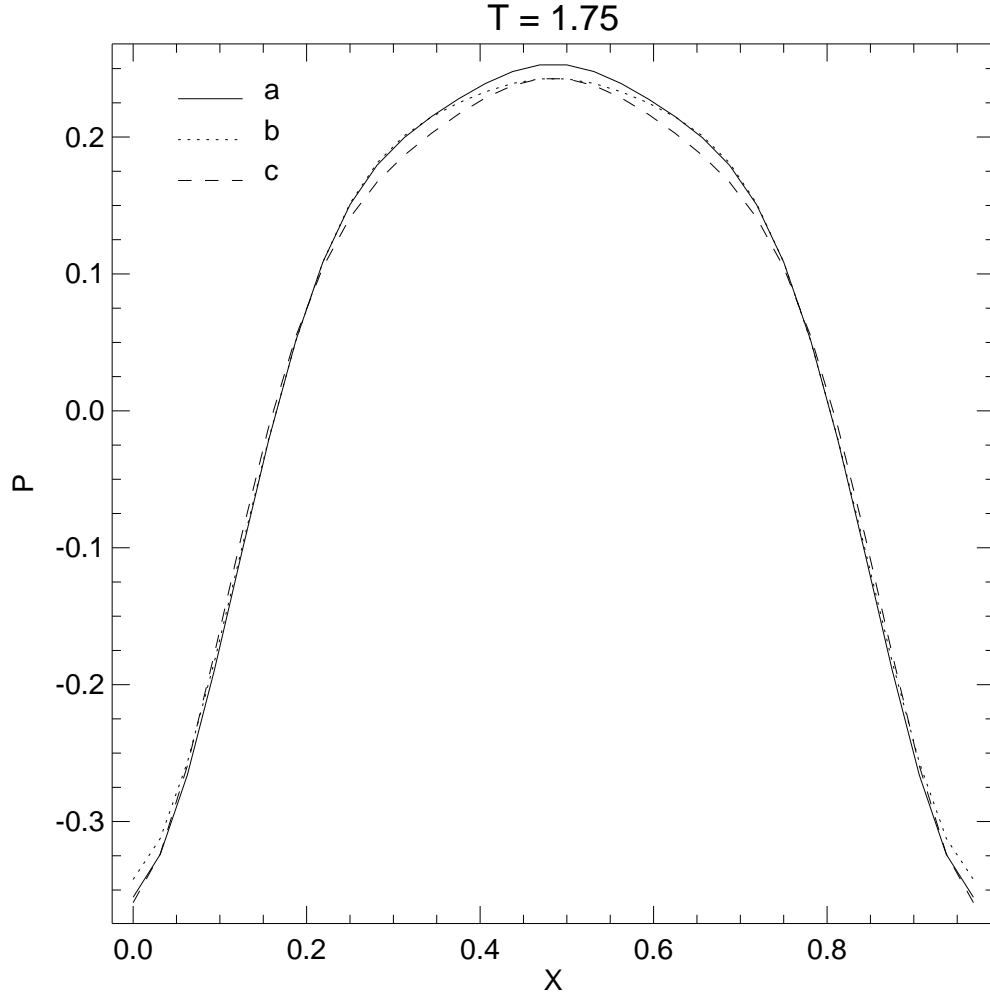
Figure 17: Solution of the acoustics equation on periodic domain with initial data $p_0 = e^{-35(x^2+y^2)}$, $u_0 = 0$, $v_0 = 0$. (a) the fine grid solution (b) the LDS($d = 1, \gamma = 20$) solution when adding fourth order dissipation (c) the coarse grid solution.
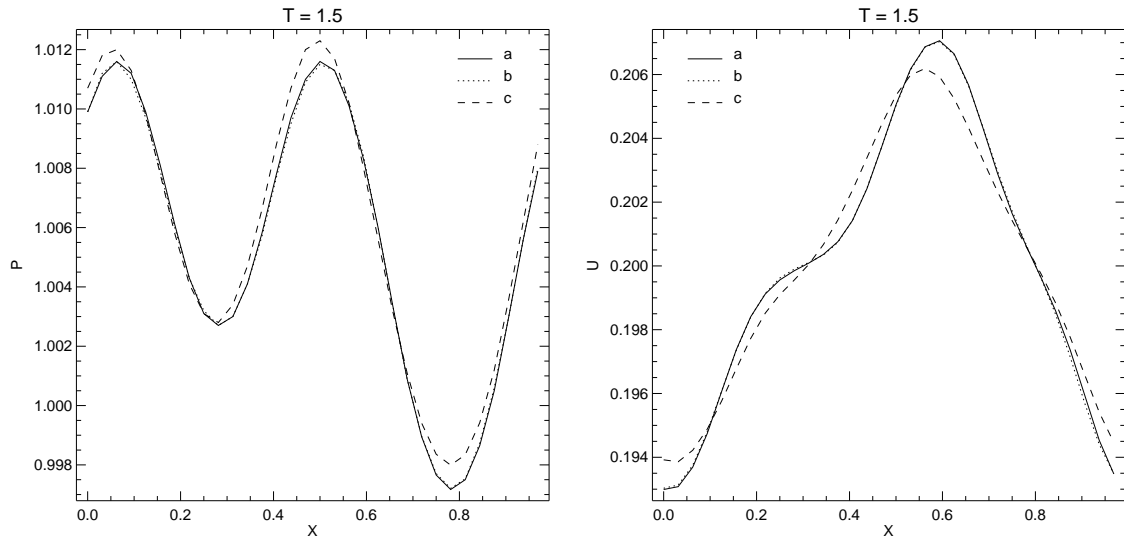
Figure 18: Solution of the nonlinear Euler equation on a periodic domain, with initial data $p_0 = 1 + 0.05e^{-50(x^2+y^2)}$, $u_0 = 0.2$, $v_0 = 0$. (a) the fine grid solution (b) the LDS($d = 1, \gamma = 25$) solution when adding sixth order dissipation (c) the coarse grid solution.