Semi-discrete subdomain iteration for hyperbolic systems

Morten Bjørhus

Abstract

In this paper, a semi-discrete subdomain iteration method for multi-dimensional hyperbolic systems is presented. The semi-discretization is obtained by the method of lines, treating interface and boundary points according to the compatibility conditions. Under certain differentiability conditions, the method is shown to converge uniformly and superlinearly on all finite time intervals. In the one-dimensional case convergence in a finite number of iterations is established. Numerical experiments are included.

1 Introduction

In the context of numerical methods for solving ordinary differential equations (ODEs), one can in general distinguish between two different types of parallelism: parallelism across the method, and parallelism across the system (see [12]). The first type depends on the structure of the numerical scheme being such that different parts of the calculations can be performed simultaneously; this type of parallelism is thus problem-independent. As opposed to this, parallelism across the system is essentially about exploiting certain properties of the time-continuous problem. One popular technique in this respect is the waveform relaxation method (or dynamic iteration method), which allows for a decoupling into independent subsystems at the expense of introducing a functional iteration. For partial differential equations (PDEs), a natural starting point for finding parallel solution techniques is to decompose the domain of the problem. Still working with the continuous problem, the subproblems can then be decoupled using a technique sometimes referred to as the subdomain iteration method. For time-dependent problems, the resulting method is conceptually similar to the waveform relaxation method for ODEs. In the present paper, we show that the two methods are indeed closely related.

Let the generic element of $\mathbb{R}^{n+1}$ be $(x,t) = (x_1, \ldots, x_n, t)$, and let $D_j = \partial/\partial x_j$ and $D_t = \partial/\partial t$. Let $\Omega \subset \mathbb{R}^n$ be open and consider the initial–boundary value problem of finding $u \in C^k(\overline{G})^r$, $k \geq 1$, satisfying

$$\begin{aligned} D_t u + \textstyle\sum_{j=1}^n A_j D_j u + C u = f \quad & \text{in } \ G = \Omega \times (0,T) \ , \\ u = u_0 \quad & \text{on } \ H = \Omega \times \{0\} \ , \\ B^j u = g^j \quad & \text{on } \ \Sigma^j \subset \partial\Omega \times (0,T) \ , \ \ \text{for } \ j \in J \ . \end{aligned} \tag{1}$$

Here, $A_j, C \in C^{k-1}(G)^{r\times r}$, $B^j \in C^0(\mathscr{S}^j)^{r^j \times r}$ for some open neighborhood $\mathscr{S}^j$ of Σ^j, and we use matrix-vector notation. Assume that the system is hyperbolic, in the sense that for all non-zero $\xi \in \mathbb{R}^n$, the characteristic matrix $A(\xi) := \sum_{j=1}^n A_j \xi_j$ has only real eigenvalues and is diagonalizable.

Let $\{G_i\}_{i\in I}$ be a cylindrical decomposition of G, that is, $G_i = \Omega_i \times (0,T)$, and assume that the family $\{\Omega_i\}$ can be constructed by subtracting from Ω the boundaries of a finite number of C^k domains. Let $\Gamma_{ii'}$ be the smooth part of the interface $\partial G_i \cap \partial G_{i'} \cap G$, defined by

$$\Gamma_{ii'} = \partial G_i \cap \partial G_{i'} \cap G \setminus \left(\bigcup_{k \neq i,i'} \partial G_k \right) \ .$$

If ∂G is regular, and the decomposition is non-characteristic with respect to the differential operator $D_t + \sum A_j D_j + C$, then the following *decomposed problem* is equivalent to the above (see [5]): For each $i \in I$, find $v_i \in C^k(\overline{G})^r|_{G_i}$ satisfying

$$\begin{aligned} D_t v_i + \textstyle\sum_{j=1}^n A_j D_j v_i + C v_i = f \quad & \text{in } \ G_i \ , \\ v_i = u_0 \quad & \text{on } \ H_i = H \cap \partial G_i \ , \\ B^j v_i = g^j \quad & \text{on } \ \Sigma_i^j = \Sigma^j \cap \partial G_i, \ \ \text{for } \ j \in J \ , \\ v_i = v_{i'} \quad & \text{on } \ \Gamma_{ii'}, \ \ \text{for } \ i' \neq i \ . \end{aligned} \tag{2}$$

Hence, instead of solving the original problem, we can concentrate on solving the decomposed problem.

In this paper, we present a convergence result for an iterative method which can be considered as a semi-discrete version of the following subdomain iteration method for solving the decomposed problem: For each $\ell = 1, 2, \ldots$, find $v_i^{(\ell)} \in C^1(\overline{G_i})^r$, $i \in I$, satisfying

$$\begin{aligned} D_t v_i^{(\ell)} + \textstyle\sum_{j=1}^n A_j D_j v_i^{(\ell)} + C v_i^{(\ell)} = f \quad & \text{in } \ G_i \ , \\ v_i^{(\ell)} = u_0 \quad & \text{on } \ H_i \ , \\ B^j v_i^{(\ell)} = g^j \quad & \text{on } \ \Sigma_i^j \ , \ \text{for } \ j \in J \ , \\ S_-(\nu_{ii'}) v_i^{(\ell)} = S_-(\nu_{ii'}) v_{i'}^{(\ell-1)} \quad & \text{on } \ \Gamma_{ii'}, \ \ \text{for } \ i' \neq i \ . \end{aligned} \tag{3}$$

Here, $S_-(\nu_{ii'})$ is the rectangular matrix representing the (one-dimensional) characteristic combinations coming into G_i from $G_{i'}$. That is, if $\nu_{ii'}$ is the unit normal vector of $\Gamma_{ii'}$ pointing out of G_i, the rows of $S_-(\nu_{ii'})$ are exactly the left eigenvectors of $A(\nu_{ii'})$ corresponding to negative eigenvalues. This splitting makes sense since our assumptions imply that the eigenvalues never vanish and that the matrix we get by stacking the matrix $S_-(\nu_{ii'})$ on top of $S_-(\nu_{i'i})$ is non-singular, hence if the method converges it produces the solution of (2). On the other hand this particular splitting of the interface condition ensures that the boundary conditions of the decoupled problems in (3) are reasonable.

This type of iterative scheme for solving PDEs is well known in the context of solving elliptic problems, where the corresponding methods can be referred to as non-overlapping Schwarz methods or Neumann-Dirichlet methods (cf. [7, 9]). In [19], Quarteroni discusses the use of subdomain iteration as a method for solving various types of problems, including hyperbolic ones. Convergence results for (3) in the case of one-dimensional hyperbolic systems is given in [5]. Convergence for a 2×2 hyperbolic test problem, discretized by the spectral collocation method in space and backward Euler in time, was established by Quarteroni in [18]. In [2], the author concluded, based on numerical experiments on the one-dimensional Euler equations for adiabatic flow and a two-dimensional linear test problem, that the subdomain iteration method can lead to significant reduction of execution time in a parallel environment.

In Section 2 we semi-discretize the decomposed problem in a hypercube, using the method of lines. The interfaces and boundaries are handled according to the compatibility conditions. After presenting a classical formulation of the semi-discrete problem, we reformulate it as a pure ODE system. In Section 3 we briefly discuss the ODE formulation of more general problems.

In Section 4 we present a general convergence proof of a semi-discretize version of the subdomain iteration (3). Following this, possible strategies for time discretizations are discussed. For one-dimensional problems, it turns out that convergence properties similar to those of the continuous iteration hold also in the semi-discrete and the fully discrete case.

In Section 5 we digress to show a related convergence result for a semi-discrete subdomain iteration scheme for the heat equation. This is to illustrate that convergence results similar to that of Section 4.1 can be given for other problems too, provided the iterative scheme can be formulated as a dynamic iteration method.

To illustrate potential advantages and problems with a particular implementation of the subdomain iteration method, we conclude in Section 6 with some numerical experiments.

2 Semi-discretization of the decomposed problem

2.1 DAE formulation

We will in this section introduce a semi-discretization of the decomposed problem (2). To avoid unnecessary complications, we will assume from the outset that the spatial domain Ω is a (hyper-)cube, that is, $\Omega = (-1,1)^n \subset \mathbb{R}^n$, and that the subdomains Ω_i are smaller cubes of equal size. More precisely, we choose $M \geq 1$ and let the Ω_i be the connected

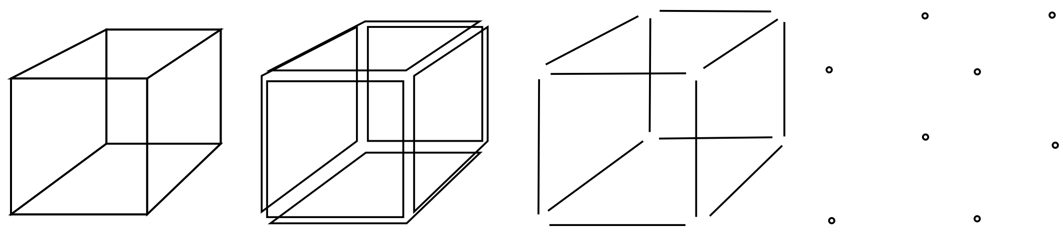

Figure 1: The sets $\partial_k Q$ constituting ∂Q for $n = 3$

components of what is left after subtracting from Ω the $n(M-1)$ hyperplanes

$$H_{j,k} = \{x \mid x_j = 1 - 2k/M\}\ , j = 1, \ldots, n\ ,\ k = 1, \ldots, M-1\ . \tag{4}$$

Hence each Ω_i is a cube with sides of length $2/M$. As before, we let $G = \Omega \times (0, T)$ and $G_i = \Omega_i \times (0, T)$. We will in addition assume that the same degree of discretization is used in each of the subdomains. It will become clear, however, that not all of these simplifying assumptions are necessary for the convergence results to hold (cf. Remark 2.7).

We introduce some notation. The boundary of a cube Q in $\mathbb{R}^n$ is embedded in the union of $2n$ hyperplanes. Let $\partial_0 Q$ be the points that lie in n of these hyperplanes, and for $1 \leq j \leq n-1$, let $\partial_j Q$ be the points that lie in exactly $n - j$ of the hyperplanes, but in none of the $\partial_k Q$, $0 \leq k < j$. That is, for a cube Q in $\mathbb{R}^3$, $\partial_0 Q$ are the vertices, $\partial_1 Q$ are the edges, and $\partial_2 Q$ are the faces; see Figure 1. The components of $\partial_j Q$ are thus j-dimensional manifolds. If Q is a cube in $\mathbb{R}^n$ and $x \in \partial_{n-1} Q$, we denote by $\nu = \nu(x)$ the unit normal vector pointing out of Q. In this notation, $\Gamma_{ii'}$ is given by

$$\Gamma_{ii'} = (\partial_{n-1}\Omega_i \cap \partial_{n-1}\Omega_{i'}) \times (0, T)\ .$$

For a unit vector ξ in $\mathbb{R}^n$, and for $(x, t) \in \partial_{n-1}\Omega \times (0, T)$, let $\Lambda(\xi)$ be the diagonal $r \times r$ matrix with the eigenvalues of $A(\xi)$ on the diagonal, and let $S(\xi)$ be the matrix whose rows are the corresponding left eigenvectors, such that

$$S(\xi)A(\xi) = \Lambda(\xi)S(\xi)\ .$$

Let $r_-(\xi)$ be the number of negative eigenvalues, and let $r_+(\xi) = r - r_-(\xi)$. Let $\Lambda(\xi)$, $S(\xi)$, and $S^{-1}(\xi)$ be partitioned accordingly as

$$\Lambda(\xi) = \begin{pmatrix} \Lambda_+(\xi) & 0 \\ 0 & \Lambda_-(\xi) \end{pmatrix}\ ,\ S(\xi) = \begin{pmatrix} S_+(\xi) \\ S_-(\xi) \end{pmatrix}\ ,\ S^{-1}(\xi) = \begin{pmatrix} S_+^{-1}(\xi) & S_-^{-1}(\xi) \end{pmatrix}\ . \tag{5}$$

We make the following assumptions.

ASSUMPTION 2.1 *Let $\Omega = (-1, 1)^n$ and let $\{\Omega_i\}_{i=1}^{M^n}$ be the connected components of $\Omega \setminus \bigcup H_{j,k}$. Assume that*

1. $A_j, C \in C(\overline{G})^{r\times r}$,

2. *The hyperplanes* $H_{j,k}$ *are non-characteristic for* $\sum A_j D_j$ *for* $0 < t < T$,

3. $J = \{0, 1, \ldots, r\}$, *and* $\Sigma^j = \{(x,t) \in \partial_{n-1}\Omega \times (0,T) \mid r_-(\nu) = j\}$,

4. $B^j = S_-(\nu)$, $0 \le j \le r$ *(characteristic boundary conditions).*

Assumptions *1* and *2* guarantee that the decomposed problem (2) is equivalent to the original problem (1) for C^1 solutions. Assumptions *3* and *4* makes it possible to rewrite the boundary conditions $B^j u = g^j$, $j \in J$, using the shorthand notation

$$S_-(\nu)u = g \quad \text{on} \quad \partial_{n-1}\Omega \times (0,T) \ ,$$

where the number of equations in general will depend on x and t. Thus equations (2) of the decomposed problem can be written as

$$\begin{aligned} D_t v_i + \textstyle\sum_{j=1}^n A_j D_j v_i + C v_i = f \quad & \text{in} \quad \Omega_i \times (0,T) \ , \\ v_i = u_0 \quad & \text{on} \quad \Omega_i \times \{0\} \ , \\ S_-(\nu) v_i = g \quad & \text{on} \quad (\partial_{n-1}\Omega_i \cap \partial_{n-1}\Omega) \times (0,T) \ , \\ v_i = v_{i'} \quad & \text{on} \quad (\partial_{n-1}\Omega_i \cap \partial_{n-1}\Omega_{i'}) \times (0,T), \quad \text{for} \quad i' \neq i \ , \end{aligned} \tag{6}$$

We will now semi-discretize the decomposed problem using the method of lines. To fix the ideas we will in the following assume that a spectral collocation method is used. However, this choice of spatial discretization is never essential to the results given, and everything applies for other choices of the scalar derivative matrix D.

For $t \ge 0$ we look for a polynomial in each (spatial) subdomain Ω_i, with coefficients in $\mathbb{C}^r$ and of degree at most N_j in the jth variable, which is to approximate $v_i(\cdot, t)$; without loss of generality we assume $N_j = N$ for $1 \le j \le n$ (cf. Remark 2.7). These polynomials are uniquely determined by their values in $(N+1)^n$ collocation points $\{x \in \overline{\Omega_i} \mid \tilde{x}_j \in CP,\ 1 \le j \le n\}$, where CP is a set of $N+1$ collocation points in the reference interval $[-1,1]$ and where $x \mapsto \tilde{x}$ is the affine mapping that maps Ω_i into $(-1,1)^n$. It is customary to work only with the values in the collocation points, hence if we denote by $\mathsf{v}_i(t)$ an $r(N+1)^n$ vector containing the $(N+1)^n$ point values, we will need mappings D_j which transforms v_i into the partial derivatives with respect to x_j of the corresponding interpolating polynomial, evaluated in the collocation points. Since we have assumed $N_j = N$, we only have to know the one-dimensional scalar derivative matrix $\mathsf{D} = \mathsf{D}(N)$, as the D_j are easily constructed from this. The $(N+1) \times (N+1)$ matrix D can be computed explicitly in different ways (see e.g., [21]).

We divide $\overline{\Omega} \setminus (\bigcup \Omega_i)$ into the following subsets (cf. Mandel [14]).

DEFINITION 2.2 *Let* $0 \le k \le n-1$. *The set* $\bigcup_i \partial_k \Omega_i$ *is called the* k*-glob of the* Ω_i*, and will be denoted by* g_k. *Let* $g_k^\circ = g_k \cap \Omega$ *and* $\partial g_k = g_k \cap \partial\Omega$ *be the interior and exterior* k*-globs, respectively.*

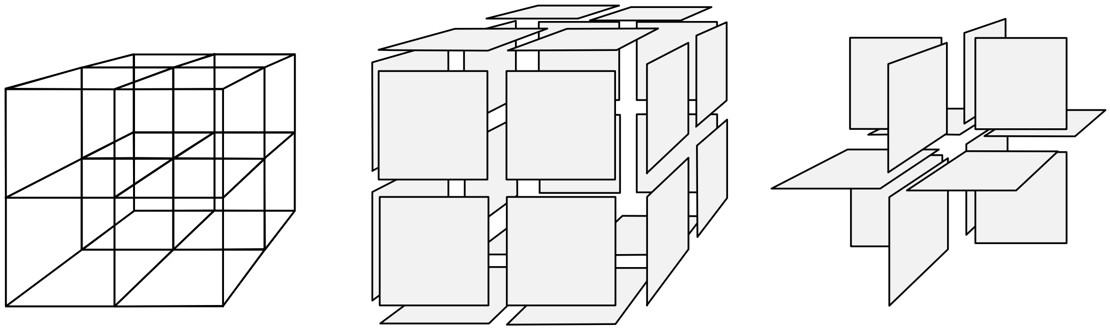

Figure 2: $n = 3$, $M = 2$: the Ω_i, the exterior 2-glob, the interior 2-glob

Let p be among the collocation points. Depending on the location of p, different types of equations will be used to determine the corresponding *point value*, that is, the approximation $\mathsf{v}_{ik}(t)$ to $v_i(p,t)$. If $p \in \Omega_i$, the PDE is collocated directly, and we get the ODE

$$\dot{\mathsf{v}}_{ik} + \sum A_j(\mathsf{D}_j\mathsf{v}_i)_k + C\mathsf{v}_{ik} = f\ . \tag{7}$$

If $p \in g^\circ_{n-1}$, it is on the interface $\Gamma_{ii'}$ between two subdomains Ω_i and $\Omega_{i'}$, hence there are two point values v_{ik} and $\mathsf{v}_{i'k'}$ to be determined in p. This is done the usual way by enforcing the interface condition $\mathsf{v}_{ik} = \mathsf{v}_{i'k'}$ and collocating the PDEs representing outflow from each of the two subdomains. If ξ is a unit normal vector of the interface, pointing into $\Omega_{i'}$, this amounts to imposing the $2r$ equations

$$\begin{aligned} S_+(\xi)\,(\dot{\mathsf{v}}_{ik} + \textstyle\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k + C\mathsf{v}_{ik}) &= S_+(\xi)f \\ S_-(\xi)\mathsf{v}_{ik} &= S_-(\xi)\mathsf{v}_{i'k'} \end{aligned} \tag{8}$$

$$\begin{aligned} S_-(\xi)\,(\dot{\mathsf{v}}_{i'k'} + \textstyle\sum A_j(\mathsf{D}_j\mathsf{v}_{i'})_{k'} + C\mathsf{v}_{i'k'}) &= S_-(\xi)f \\ S_+(\xi)\mathsf{v}_{i'k'} &= S_+(\xi)\mathsf{v}_{ik} \end{aligned} \tag{9}$$

If $p \in \partial g_{n-1}$, the r equations to be satisfied are given by the boundary conditions and by collocating the equations for outgoing information; that is,

$$\begin{aligned} S_+(\nu)\,(\dot{\mathsf{v}}_{ik} + \textstyle\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k + C\mathsf{v}_{ik}) &= S_+(\nu)f \\ S_-(\nu)\mathsf{v}_{ik} &= g\ . \end{aligned} \tag{10}$$

This far in the process of semi-discretizing the decomposed problem, we have used techniques that are well established in the literature (see e.g., [8, 10]). It remains to set up the equations for collocation points in the sets g_k, $0 \le k \le n-2$ (i.e., the corners, edges, etc.). How to do this is not obvious, since there are no interface or boundary conditions imposed in these sets by (6). However, given a continuous solution of the original problem, which we know corresponds to continuous solutions of the decomposed problem, the interface and boundary conditions can be extended continuously to these globs.

Let us first concentrate on the interior globs. If $p \in g_k^\circ$, $0 \le k \le n-2$, then p lies inside Ω and on exactly $n-k$ of the hyperplanes (4); hence there are $2(n-k)$ subdomains meeting at p. In other words, there are $2(n-k)$ point values to be determined in p, and this requires $2r(n-k)$ equations. Divide the $2(n-k)$ subdomains into $n-k$ pairs. For each pair Ω_i, $\Omega_{i'}$ let v_{ik} and $\mathsf{v}_{i'k'}$ denote the corresponding point values, and let ξ be a unit vector in $\mathbb{R}^n$ such that

$$p + \xi \in \overline{\Omega_{i'}}\ , \quad p - \xi \in \overline{\Omega_i}\ . \tag{11}$$

As mentioned above, we can extend the interface conditions to p and require $\mathsf{v}_{ik} = \mathsf{v}_{i'k'}$. We can provide the r remaining equations by collocating the outgoing information along ξ as before. Hence by (11), the $2r$ equations (8), (9) determine the two point values also in this case. Applying this technique to each pair, we get $2r(n-k)$ equations as needed. Kopriva [13] effectively used this technique, by pairing neighboring subdomains.

DEFINITION 2.3 *Let $p \in g_k^\circ$, $0 \le k \le n-1$. If the $2(n-k)$ point values residing at p are divided into pairs, and if each pair v_{ik}, $\mathsf{v}_{i'k'}$ are required to satisfy equations* (8), (9) *for some unit vector ξ satisfying* (11)*, then we say that the values at p are determined by pairwise matching.*

The only remaining points are the ones lying in the exterior globs ∂g_k, $0 \le k \le n-2$. For each such point p, there is either a single subdomain or an even number of subdomains, $2q$ say, meeting at p. Assume the latter situation. Again, by the assumption of continuity of the solution of the PDE, both the interface and the boundary conditions in the neighborhood of p can be extended to p. Hence, one possible way to determine the $2q$ point values in p is to determine $2q'$ of them by pairwise matching, and the remaining $2(q-q')$ by treating each of them as a point in the closure of one of the components of $\partial_{n-1}\Omega_i \cap \partial_{n-1}\Omega$, satisfying the corresponding equations (10). In some situations it may be more natural to determine each of the last $2(q-q')$ point values by prescribing the incoming characteristic variables to the subdomain Ω_i to which it belongs relative to some other unit vector, still denoted by ν, satisfying

$$p - \nu \in \overline{\Omega_i}\ . \tag{12}$$

Collocating the corresponding outgoing equations again yields equations of type (10) (with g now denoting some prescribed value). A necessary condition for this to make sense is clearly that this extra condition is compatible with the other interface and boundary conditions at p. If either of these two techniques are used, we say that the corresponding $2(q-q')$ point values are determined by inflow-outflow conditions. In the case where p belongs to only one subdomain, the single point value is assumed to be treated similarly; in corner points ($n=2$), this procedure is effectively the same as used by Wasberg and Andreassen [22] by determining the point value by an inflow-outflow condition in the direction ν "at 45° with both boundaries".

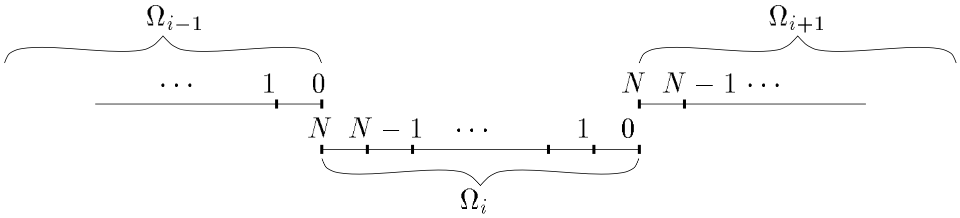

Figure 3: Ordering of subdomains and collocation points for $n = 1$ in Example 2.5

DEFINITION 2.4 *Let $p \in \partial g_k$, $0 \le k \le n-1$. If an even number (possibly zero) of the point values residing at p are divided into pairs and determined by pairwise matching, and if the remaining point values are required to satisfy* (10) *for some unit vectors ν satisfying* (12)*, then we say that the values at p are determined by a combination of pairwise matching and inflow-outflow conditions.*

Summing up, we have assumed that the decomposed problem is semi-discretized in such a way that all point values in the interior of the subdomains are determined by equations of type (7), all point values lying in the interior globs are determined by equations of type (8), (9), and all point values lying in the exterior globs are determined either by equations of type (10) or of type (8), (9). Clearly, from the two equations (8), (9) it follows that $\mathsf{v}_{ik}(t) = \mathsf{v}_{i'k'}(t)$ for all $t \ge 0$; the point value v_{ik} is thus the solution of

$$\dot{\mathsf{v}}_{ik} = S_+^{-1}(\xi) S_+(\xi) \left(-\sum A_j (\mathsf{D}_j \mathsf{v}_i)_k \right) + S_-^{-1}(\xi) S_-(\xi) \left(-\sum A_j (\mathsf{D}_j \mathsf{v}_{i'})_{k'} \right) - C \mathsf{v}_{ik} + f \ .$$

Hence in each collocation point on the interfaces $\Gamma_{ii'}$, the two point values will always coincide, while in the collocation points lying in globs of degree $\le n-2$ there will in general be as many point values as there are matching pairs.

Here and below, the one-dimensional problem in which the principal part of the differential operator has constant coefficients will be used as illustration.

Example 2.5 *Let $n = 1$, and let* ASSUMPTION 2.1 *hold. In this case there is essentially no freedom in the choice of vectors ξ and ν in the globs if the problem is semi-discretized as above. Let the A be constant, and denote the matrix $S(1)$ by S. If the Ω_i are numbered from left to right, and the collocation points within each subdomain are numbered as usual from right to left (see Figure 3), then the equations determining the point values v_{ik}, $k = 0, \ldots, N$ in Ω_i, $1 < i < M$ are*

$$\begin{aligned} S_+ \dot{\mathsf{v}}_{i0} + S_+ A (\mathsf{D}_1 \mathsf{v}_i)_0 + S_+ C \mathsf{v}_{i0} &= S_+ f \ , \\ S_- \mathsf{v}_{i0} &= S_- \mathsf{v}_{i+1,N} \ , \end{aligned} \tag{13}$$

$$\dot{\mathsf{v}}_{ik} + A (\mathsf{D}_1 \mathsf{v}_i)_k + C \mathsf{v}_{ik} = f \ , \quad k = 1, \ldots, N-1 \ , \tag{14}$$

$$\begin{aligned} S_- \dot{\mathsf{v}}_{iN} + S_- A (\mathsf{D}_1 \mathsf{v}_i)_N + S_- C \mathsf{v}_{iN} &= S_- f \ , \\ S_+ \mathsf{v}_{iN} &= S_+ \mathsf{v}_{i-1,0} \ , \end{aligned} \tag{15}$$

If $i = 1$ (resp. $i = M$) the last of the two equations in (15) *(resp.* (13)*) is replaced by the boundary condition.*

Remark 2.6 Clearly, we have $S_{\mp}(\xi) = S_{\pm}(-\xi)$ in equations (8), (9) in the interior $n-1$ globs because of ASSUMPTION 2.1.2; hence the equations are invariant under the change $\xi \to -\xi$. This is not necessarily the case in the remaining globs. When the matrices fail to be equal, that is if $A(\xi)$ has at least one zero eigenvalue, the effect of a choice between ξ and $-\xi$ is that the task of integrating a 'zero mode' is assigned to one of the two subdomains.

Remark 2.7 Note that the method presented relies essentially only on the fact that (i) a collocation point lying in one of the $\partial\Omega_i$ is present in all of the $\partial\Omega_j$ meeting at that point, and that (ii) the (one-dimensional) characteristic variables are integrated in the domain for which they represents outflow. Hence it is possible to use the same type of semi-discretization for any domain which is written as the union of cubes, say, as long as the collocation points match at the interfaces. Moreover, there is no problem from a computational point of view in allowing some of the hyperplanes $H_{j,k}$ to be characteristic for some values of t and x. The interpretation of this is as in the previous remark.

2.2 ODE formulation

In this subsection we will see how the semi-discrete problem can be written equivalently as a pure ODE system (cf. [4]). Besides of facilitating the combination with standard ODE solvers when it comes to numerical time integration, this will be the key to proving the convergence theorem of the subdomain iteration method presented in Section 4.1.

Assume that the decomposed problem (6) is semi-discretized like above, using pairwise matching in the interior globs, and a combination of pairwise matching and inflow-outflow conditions in exterior globs. Then, as mentioned above, in each collocation point the point values are determined either by equations of type (7), of type (8) and (9), or of type (10). The first type is already an ODE, while the others are DAEs, in the sense that they involve both ODEs and algebraic constraints. We will now show how to rewrite the latter as pure ODEs.

First consider the $2r$ equations (8), (9) for the 2 point values v_{ik}, $\mathsf{v}_{i'k'}$ in a collocation point p. Using equations (8) we have

$$\begin{aligned} \dot{\mathsf{v}}_{ik} &= S^{-1}(\xi)S(\xi)\dot{\mathsf{v}}_{ik} \\ &= S_+^{-1}(\xi)S_+(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right) + S_-^{-1}(\xi)S_-(\xi)\dot{\mathsf{v}}_{ik}\ , \end{aligned} \tag{16}$$

and

$$\begin{aligned} S_-(\xi)\dot{\mathsf{v}}_{ik} &= \tfrac{d}{dt}(S_-(\xi)\mathsf{v}_{ik}) - \dot{S}_-(\xi)\mathsf{v}_{ik} \\ &= \tfrac{d}{dt}(S_-(\xi)\mathsf{v}_{i'k'}) - \dot{S}_-(\xi)\mathsf{v}_{ik} \\ &= S_-(\xi)\dot{\mathsf{v}}_{i'k'} + \dot{S}_-(\xi)(\mathsf{v}_{i'k'} - \mathsf{v}_{ik})\ ; \end{aligned}$$

hence v_{ik} satisfies the ODE

$$\begin{aligned}\dot{\mathsf{v}}_{ik} = \;& S_+^{-1}(\xi)S_+(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right)\\ &+ S_-^{-1}(\xi)S_-(\xi)\dot{\mathsf{v}}_{i'k'} + S_-^{-1}(\xi)\dot{S}_-(\xi)(\mathsf{v}_{i'k'} - \mathsf{v}_{ik})\ .\end{aligned} \tag{17}$$

Similarly, $\mathsf{v}_{i'k'}$ satisfies the ODE

$$\begin{aligned}\dot{\mathsf{v}}_{i'k'} = \;& S_-^{-1}(\xi)S_-(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_{i'})_{k'} - C\mathsf{v}_{i'k'} + f\right)\\ &+ S_+^{-1}(\xi)S_+(\xi)\dot{\mathsf{v}}_{ik} + S_+^{-1}(\xi)\dot{S}_+(\xi)(\mathsf{v}_{ik} - \mathsf{v}_{i'k'})\ .\end{aligned} \tag{18}$$

It is not difficult to eliminate the derivatives from the right hand sides of (17), (18) to get

$$\begin{aligned}\dot{\mathsf{v}}_{ik} = \;& S_+^{-1}(\xi)S_+(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right)\\ &+ S_-^{-1}(\xi)S_-(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_{i'})_{k'} - C\mathsf{v}_{i'k'} + f\right) + S_-^{-1}(\xi)\dot{S}_-(\xi)(\mathsf{v}_{i'k'} - \mathsf{v}_{ik})\end{aligned} \tag{19}$$

and

$$\begin{aligned}\dot{\mathsf{v}}_{i'k'} = \;& S_+^{-1}(\xi)S_+(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right)\\ &+ S_-^{-1}(\xi)S_-(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_{i'})_{k'} - C\mathsf{v}_{i'k'} + f\right) + S_+^{-1}(\xi)\dot{S}_+(\xi)(\mathsf{v}_{ik} - \mathsf{v}_{i'k'})\ .\end{aligned} \tag{20}$$

This is the explicit ODE formulation of the pair of equations (8), (9).

The equations of type (10) are treated completely analogously. By equation (16) and the fact that

$$S_-(\nu)\dot{\mathsf{v}}_{ik} = \frac{d}{dt}(S_-(\nu)\mathsf{v}_{ik}) - \dot{S}_-(\nu)\mathsf{v}_{ik} = \dot{g} - \dot{S}_-(\nu)\mathsf{v}_{ik}\ , \tag{21}$$

it follows immediately that

$$\dot{\mathsf{v}}_{ik} = S_+^{-1}(\nu)S_+(\nu)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right) + S_-^{-1}(\nu)\dot{g} - S_-^{-1}(\nu)\dot{S}_-(\nu)\mathsf{v}_{ik} \tag{22}$$

is the ODE formulation of (10).

Since our semi-discretization of (6) is thus equivalent to an ODE initial value problem, it makes sense to introduce the following assumptions.

ASSUMPTION 2.8 *For all equations of type* (8), (9), *let*

$$S_-^{-1}(\xi)S_-(\xi)\ ,\ S_-^{-1}(\xi)\dot{S}_-(\xi)\ ,\ S_+^{-1}(\xi)S_+(\xi)\ ,\ S_+^{-1}(\xi)\dot{S}_+(\xi)\ ,$$

be continuous (matrix-valued) functions of t. Similarly, for all equations of type (10), *let*

$$S_+^{-1}(\nu)S_+(\nu)\ ,\ S_-^{-1}(\nu)\dot{S}_-(\nu)\ ,\ S_-^{-1}(\nu)\dot{g}$$

be continuous.

Clearly, in conjunction with ASSUMPTION 2.1, this implies that the right hand side of the ODEs (7), (19), (20), (22) are continuous with respect to t, and linear, hence Lipschitz continuous, with respect to the point values (for all f admissible for $C^1(\overline{G})$ solutions of (1)). By a classical result in ODE theory this is sufficient to guarantee the existence and uniqueness of a C^1 solution of our semi-discretization of (6) (see e.g., Yosida [23]).

Example 2.9 *Consider the system in* Example 2.5. *Since A is constant,* ASSUMPTION 2.8 *holds for all g admissible for $C^1(\overline{G})$ solutions of* (1). *The ODE formulation of equations* (13), (14), (15) *is*

$$\dot{\mathsf{v}}_{i0} = S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_i)_0 - C\mathsf{v}_{i0} + f\right) + S_-^{-1}S_-\dot{\mathsf{v}}_{i+1,N} \ , \tag{23}$$

$$\dot{\mathsf{v}}_{ik} = -A(\mathsf{D}_1\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f \ , \quad k = 1, \ldots, N-1 \ ,$$

$$\dot{\mathsf{v}}_{iN} = S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_i)_N - C\mathsf{v}_{iN} + f\right) + S_+^{-1}S_+\dot{\mathsf{v}}_{i-1,0} \ . \tag{24}$$

The equivalent formulation yielding explicit ODEs is

$$\begin{aligned}\dot{\mathsf{v}}_{i0} = \ & S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_i)_0 - C\mathsf{v}_{i0} + f\right) \\ & + S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_{i+1})_N - C\mathsf{v}_{i+1,N} + f\right) \ ,\end{aligned} \tag{25}$$

$$\dot{\mathsf{v}}_{ik} = -A(\mathsf{D}_1\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f \ , \quad k = 1, \ldots, N-1 \ ,$$

$$\begin{aligned}\dot{\mathsf{v}}_{iN} = \ & S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_i)_N - C\mathsf{v}_{iN} + f\right) \\ & + S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_{i-1})_0 - C\mathsf{v}_{i-1,0} + f\right) \ .\end{aligned} \tag{26}$$

If $i = 1$ (resp. $i = M$) the last term of equations (24), (26) *(resp.* (23), (25)*) is replaced by $S_+^{-1}\dot{g}$ (resp. $S_-^{-1}\dot{g}$).*

3 ODE formulation in more general cases

The main part of this paper is within the framework of hyperbolic systems with variable coefficients and characteristic boundary conditions. For completeness, however, we will in this section show how one can obtain the ODE formulation in cases that are more general than those discussed in Section 2.2; namely the case of a more general boundary condition, and the case of a quasilinear hyperbolic system. To avoid an unnecessary complicated notation, we will in this section suppress the explicit dependence on ξ in the matrices (5).

First let us stay in the variable coefficient case, and show how to handle the boundary condition

$$Mu = g \ , \tag{27}$$

where M is an $r_- \times r$ matrix-valued function and $\det(MS_-^{-1}) \neq 0$ ("the Kreiss condition"). We have

$$Mu = MS^{-1}Su = (MS_-^{-1})(S_-u) + (MS_+^{-1})(S_+u) = g \ ,$$

so (27) is equivalent to the (reflection) condition

$$S_-u = (MS_-^{-1})^{-1}\left[g - (MS_+^{-1})S_+u\right]$$

in which the incoming (one-dimensional) characteristic variable S_-u is expressed as a combination of g and the outgoing characteristic variable S_+u. To use (16) we have to express this condition as a condition on $S_-\dot{\mathsf{v}}_{ik}$, and by (21) we get

$$S_-\dot{\mathsf{v}}_{ik} = \frac{d}{dt}\left\{(MS_-^{-1})^{-1}\left[g - (MS_+^{-1})S_+\mathsf{v}_{ik}\right]\right\} - \dot{S}_-\mathsf{v}_{ik}\ .$$

Applying the chain rule and substituting into (16) yields

$$\begin{aligned}\left[I + S_-^{-1}(MS_-^{-1})^{-1}(MS_+^{-1})S_+\right]\dot{\mathsf{v}}_{ik} = \quad & S_+^{-1}S_+\left(\textstyle\sum A_j(\mathsf{D}_j\mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right)\\ & + S_-^{-1}\tfrac{d}{dt}\left[(MS_-^{-1})^{-1}g\right]\\ & - S_-^{-1}\tfrac{d}{dt}\left[(MS_-^{-1})^{-1}(MS_+^{-1})S_+ + S_-\right]\mathsf{v}_{ik}\ .\end{aligned} \tag{28}$$

Hence we can solve to get an explicit ODE for v_{ik} provided

$$\det\left[I + S_-^{-1}(MS_-^{-1})^{-1}(MS_+^{-1})S_+\right] \neq 0\ . \tag{29}$$

Of course, (28) reduces to (22) in the particular case $M = S_-$.

Example 3.1 *Consider the system $D_tu + AD_xu = f$ in $(-1,1)\times(0,T)$, where*

$$A = \begin{pmatrix} a & 1\\ 1 & a\end{pmatrix}\ ,\ a\in\mathbb{R}\ ,\ |a|<1\ ,$$

subject to the boundary conditions $u_1(-1,t) = u_1(1,t) = 0$ for $0<t<T$. The boundary conditions are of the form (27) with $M = \begin{pmatrix} 1 & 0\end{pmatrix}$. For this problem we can take

$$S(\xi) = S^{-1}(\xi) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1\\ 1 & -1\end{pmatrix}\ ,\ \Lambda(\xi) = \begin{pmatrix}\xi(a+1) & 0\\ 0 & \xi(a-1)\end{pmatrix}\ ,$$

and it is not difficult to see that the inverse of the matrix $[I + S_-^{-1}(MS_-^{-1})^{-1}(MS_+^{-1})S_+]$ is given by

$$\frac{1}{2}\begin{pmatrix} 1 & 1\\ -1 & 3\end{pmatrix}\quad \textit{and}\quad \frac{1}{2}\begin{pmatrix} 1 & -1\\ 1 & 3\end{pmatrix}\ ,$$

on the left and right boundaries, respectively. Hence the semi-discrete problem can be expressed as a pure ODE system.

Now let us look at the quasilinear case. Assume that the boundary conditions are characteristic, but the coefficients of (1) depend on u. In this case the time derivative of

S_- in (21) must be interpreted as a total derivative, since v_{ik} is a function of t, and S_- now in general depends on v_{ik}. That is,

$$\frac{d}{dt}(S_-)u = \frac{\partial}{\partial t}(S_-)u + \sum_{j=1}^{r} \frac{\partial}{\partial u_j}(S_-)\frac{\partial u_j}{\partial t}u = \frac{\partial}{\partial t}(S_-)u + \mathcal{R}(S_-, u)\frac{\partial u}{\partial t} ,$$

where the matrix $\mathcal{R}(E, u)$ is defined by

$$\mathcal{R}(E, u) = \begin{pmatrix} \frac{\partial E}{\partial u_1}u & \frac{\partial E}{\partial u_2}u & \cdots & \frac{\partial E}{\partial u_r}u \end{pmatrix} .$$

Substituting in (16), we get

$$\begin{aligned} \left[I + S_-^{-1}\mathcal{R}(S_-, \mathsf{v}_{ik})\right] \dot{\mathsf{v}}_{ik} = \quad & S_+^{-1}S_+ \left(\sum A_j(\mathsf{D}_j \mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right) \\ & + S_-^{-1}\dot{g} - S_-^{-1}\tfrac{\partial}{\partial t}(S_-)\mathsf{v}_{ik} , \end{aligned}$$

which is uniquely solvable for $\dot{\mathsf{v}}_{ik}$ provided

$$\det\left[I + S_-^{-1}\mathcal{R}(S_-, \mathsf{v}_{ik})\right] \neq 0$$

for the values of t, v_{ik} of interest. Similarly, the quasi-linear versions of (19), (20) are

$$\begin{aligned} \left[I + S_-^{-1}\mathcal{R}(S_-, \mathsf{v}_{ik} - \mathsf{v}_{i'k'})\right] \dot{\mathsf{v}}_{ik} = \quad & S_+^{-1}S_+ \left(-\sum A_j(\mathsf{D}_j \mathsf{v}_i)_k - C\mathsf{v}_{ik} + f\right) \\ & + S_-^{-1}S_-\dot{\mathsf{v}}_{i'k'} + S_-^{-1}\dot{S}_-(\mathsf{v}_{i'k'} - \mathsf{v}_{ik}) , \end{aligned} \tag{30}$$

$$\begin{aligned} \left[I + S_+^{-1}\mathcal{R}(S_+, \mathsf{v}_{i'k'} - \mathsf{v}_{ik})\right] \dot{\mathsf{v}}_{i'k'} = \quad & S_-^{-1}S_- \left(-\sum A_j(\mathsf{D}_j \mathsf{v}_{i'})_{k'} - C\mathsf{v}_{i'k'} + f\right) \\ & + S_+^{-1}S_+\dot{\mathsf{v}}_{ik} + S_+^{-1}\dot{S}_+(\mathsf{v}_{ik} - \mathsf{v}_{i'k'}) . \end{aligned} \tag{31}$$

Unless in special cases, for instance when $\|S_\pm^{-1}\mathcal{R}(S_\pm, \mathsf{v}_{ik} - \mathsf{v}_{i'k'})\| < 1$, it is not immediate that these equations reduce to explicit ODEs for v_{ik}, $\mathsf{v}_{i'k'}$ as before, but provided

$$\det\left[I + S_-^{-1}\mathcal{R}(S_-, \mathsf{v}_{ik} - \mathsf{v}_{i'k'})\right] \neq 0 ,$$

$$\det\left[I + S_+^{-1}\mathcal{R}(S_+, \mathsf{v}_{i'k'} - \mathsf{v}_{ik})\right] \neq 0$$

for the interesting values of t, v_{ik}, $\mathsf{v}_{i'k'}$, they can still be useful in the context of the subdomain iteration scheme introduced below.

Remark 3.2 One should be careful to distinguish between, for instance, the matrix S_- appearing in (30) and the matrix S_- appearing in (31). These are notational shorthands for $S_-(t, \mathsf{v}_{ik}; \xi)$ and $S_-(t, \mathsf{v}_{i'k'}; \xi)$, respectively, and will in general not be the same when the requirement $\mathsf{v}_{ik} = \mathsf{v}_{i'k'}$ is relaxed through the iterative scheme introduced in the next section.

Example 3.3 (2D Euler equations) *For the Euler equations for two-dimensional compressible adiabatic flow (see e.g., Wasberg and Andreassen [22]), the assumption of subsonic flow velocity ensures that for any unit vector ξ, there will be only two different expressions for the matrices $S_-(\xi)$ involved, corresponding to flow in the direction of ξ or $-\xi$, respectively. Therefore there are only two expressions for the matrix $[I + S_-^{-1}\mathcal{R}(S_-, \mathsf{v}_{ik})]$ that have to be investigated. It turns out that the determinant of the matrix appearing in the outflow case is*

$$\frac{1}{4}\left(3 - 2\frac{\xi_1 u - \xi_2 v}{c} + \frac{1}{\gamma}\right) ,$$

while the determinant in the inflow case is

$$\frac{3}{4}\frac{\gamma - 1}{\gamma} .$$

Since $\gamma > 1$ and since $\xi_1 u - \xi_2 v < c$ where c is the speed of sound, these are never zero. Hence we can always invert the matrices to express the evolution in time of the point values at the boundary by pure ODEs.

Of course, we can combine the two approaches above to handle a quasilinear system with the general boundary condition (27).

4 A convergent subdomain iteration method

4.1 The semi-discrete case

We will now introduce a subdomain iteration method for the semi-discrete problem. The system of equations is decoupled into independent subsystems, one per subdomain, by solving in iteration $\ell = 1, 2, \ldots$ for unknown vectors $\mathsf{v}_i^{(\ell)}$, replacing the equations of type (8), (9) by

$$\begin{aligned} S_+(\xi)\left(\dot{\mathsf{v}}_{ik}^{(\ell)} + \sum A_j(\mathsf{D}_j \mathsf{v}_i^{(\ell)})_k + C\mathsf{v}_{ik}^{(\ell)}\right) &= S_+(\xi) f \\ S_-(\xi)\mathsf{v}_{ik}^{(\ell)} &= S_-(\xi)\mathsf{v}_{i'k'}^{(\ell-1)} \end{aligned} \tag{32}$$

$$\begin{aligned} S_-(\xi)\left(\dot{\mathsf{v}}_{i'k'}^{(\ell)} + \sum A_j(\mathsf{D}_j \mathsf{v}_{i'}^{(\ell)})_{k'} + C\mathsf{v}_{i'k'}^{(\ell)}\right) &= S_-(\xi) f \\ S_+(\xi)\mathsf{v}_{i'k'}^{(\ell)} &= S_+(\xi)\mathsf{v}_{ik}^{(\ell-1)} , \end{aligned} \tag{33}$$

where the values at iteration level zero are given. We see that we have obtained a semi-discrete version of the continuous subdomain iteration (3).

Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_{M^n})^T$, let $\mathsf{v}^{(\ell)} = (\mathsf{v}_1^{(\ell)}, \ldots, \mathsf{v}_{M^n}^{(\ell)})^T$, and let v_* (resp. $\mathsf{v}_*^{(\ell)}$) denote the subvector of v (resp. $\mathsf{v}^{(\ell)}$) corresponding to those point values determined by pairwise matching. That is, v_* consists of all point values of the semi-discrete problem belonging to the interior globs, plus possibly some belonging to the ∂g_k, $1 \le k \le n-2$. Let $\|\mathsf{v}\|$ and $\|\mathsf{v}_*\|_*$ denote the corresponding Euclidean vector norms. We have the following result.

THEOREM 4.1 *Let* ASSUMPTION 2.1 *hold, and let the decomposed problem* (6) *be semi-discretized by the method of lines, using pairwise matching in the interior globs, and a combination of pairwise matching and inflow-outflow conditions in exterior globs. If* ASSUMPTION 2.8 *holds, then the* $\{\mathsf{v}^{(\ell)}\}$ *given by the subdomain iteration based on* (32), (33) *converges uniformly to* v *on all finite time intervals given any continuously differentiable* $\mathsf{v}_*^{(0)}$ *satisfying* $\mathsf{v}_*^{(0)}(0) = \mathsf{v}_*(0)$*; that is, for each* $\ell \geq 2$ *and* $T > 0$ *we have an estimate of the type*

$$\sup_{0 \leq t \leq T} \|\mathsf{v}(t) - \mathsf{v}^{(\ell)}(t)\| \leq C(\ell, T) \sup_{0 \leq t \leq T} \|\mathsf{v}_*(t) - \mathsf{v}_*^{(1)}(t)\|_* \,, \tag{34}$$

where $C(\ell, T) \to 0$ *as* $\ell \to \infty$. *Moreover, the rate of convergence is superlinear, that is,* $[C(\ell, T)]^{1/\ell} \to 0$ *as* $\ell \to \infty$.

Proof. In the proof we make use of the results presented in the Appendix. By the results of Section 2.2 we know that the semi-discrete problem is equivalent to a system of ODEs in which all equations are either of type (7), of type (19), (20), or of type (22). By applying the same techniques to the pair of equations (32), (33), we get the equivalent formulation

$$\begin{aligned} \dot{\mathsf{v}}_{ik}^{(\ell)} = \;& S_+^{-1}(\xi) S_+(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_i^{(\ell)})_k - C\mathsf{v}_{ik}^{(\ell)} + f\right) \\ & + S_-^{-1}(\xi) S_-(\xi) \dot{\mathsf{v}}_{i'k'}^{(\ell-1)} + S_-^{-1}(\xi) \dot{S}_-(\xi) (\mathsf{v}_{i'k'}^{(\ell-1)} - \mathsf{v}_{ik}^{(\ell)}) \,, \end{aligned} \tag{35}$$

$$\begin{aligned} \dot{\mathsf{v}}_{i'k'}^{(\ell)} = \;& S_-^{-1}(\xi) S_-(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_{i'}^{(\ell)})_{k'} - C\mathsf{v}_{i'k'}^{(\ell)} + f\right) \\ & + S_+^{-1}(\xi) S_+(\xi) \dot{\mathsf{v}}_{ik}^{(\ell-1)} + S_+^{-1}(\xi) \dot{S}_+(\xi) (\mathsf{v}_{ik}^{(\ell-1)} - \mathsf{v}_{i'k'}^{(\ell)}) \,. \end{aligned} \tag{36}$$

In particular, the assumptions of the theorem imply that the $\mathsf{v}^{(\ell)}$ exists and are continuously differentiable for each $\ell = 1, 2, \ldots$.

For $\ell \geq 2$, (35), (36) can be rewritten as the pair of explicit ODEs

$$\begin{aligned} \dot{\mathsf{v}}_{ik}^{(\ell)} = \;& S_+^{-1}(\xi) S_+(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_i^{(\ell)})_k - C\mathsf{v}_{ik}^{(\ell)} + f\right) \\ & + S_-^{-1}(\xi) S_-(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_{i'}^{(\ell-1)})_{k'} - C\mathsf{v}_{i'k'}^{(\ell-1)} + f\right) \\ & + S_-^{-1}(\xi) \dot{S}_-(\xi) (\mathsf{v}_{i'k'}^{(\ell-1)} - \mathsf{v}_{ik}^{(\ell)}) \end{aligned} \tag{37}$$

and

$$\begin{aligned} \dot{\mathsf{v}}_{i'k'}^{(\ell)} = \;& S_+^{-1}(\xi) S_+(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_i^{(\ell)})_k - C\mathsf{v}_{ik}^{(\ell)} + f\right) \\ & + S_-^{-1}(\xi) S_-(\xi) \left(-\textstyle\sum A_j (\mathsf{D}_j \mathsf{v}_{i'}^{(\ell-1)})_{k'} - C\mathsf{v}_{i'k'}^{(\ell-1)} + f\right) \\ & + S_+^{-1}(\xi) \dot{S}_+(\xi) (\mathsf{v}_{ik}^{(\ell-1)} - \mathsf{v}_{i'k'}^{(\ell)}) \,. \end{aligned} \tag{38}$$

We note that the right hand sides of these ODEs are splittings of the right hand sides of the corresponding ODEs (19), (20). Moreover, they are continuous with respect to t and linear, hence Lipschitz continuous, with respect to each of the other two arguments $\mathsf{v}_{ik}^{(\ell)}$, $\mathsf{v}_{i'k'}^{(\ell)}$. Hence, for $\ell \geq 2$ the subdomain iteration corresponds to a dynamic iteration scheme

(49) with initial guess $\mathsf{v}^{(1)}$ (corresponding to $y^{(0)}$ in (49)), and an error estimate is provided by (50) or (51) with k replaced by $\ell-1$. Since it is evident from (35), (36) that $\mathsf{v}^{(\ell)}$ depends only on $\mathsf{v}_*^{(\ell-1)}$, it follows that the estimate can be written in terms of the $\|\cdot\|_*$-norm. ■

COROLLARY 4.2 *If* $\mathsf{v}^{(0)}$ *satisfies the (compatibility) conditions*

$$\dot{\mathsf{v}}_{ik}^{(0)} = S_+^{-1}(\xi)S_+(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_i^{(0)})_k - C\mathsf{v}_{ik}^{(0)} + f\right) + S_-^{-1}(\xi)h_{ik}(t,\mathsf{v}^{(0)})$$

$$\dot{\mathsf{v}}_{i'k'}^{(0)} = S_-^{-1}(\xi)S_-(\xi)\left(-\sum A_j(\mathsf{D}_j\mathsf{v}_{i'}^{(0)})_{k'} - C\mathsf{v}_{i'k'}^{(0)} + f\right) + S_+^{-1}(\xi)h_{i'k'}(t,\mathsf{v}^{(0)})$$

for all pairs of equations (32), (33), *for some* h_{ik}, $h_{i'k'}$, *then the following estimate holds:*

$$\sup_{0\le t\le T}\|\mathsf{v}(t)-\mathsf{v}^{(\ell)}(t)\| \le C(\ell+1,T)\sup_{0\le t\le T}\|\mathsf{v}_*(t)-\mathsf{v}_*^{(0)}(t)\|_* \,. \tag{39}$$

Proof. The result follows like above, except now the explicit ODE formulation (37), (38) is valid also for $\ell = 1$. Hence the entire subdomain iteration can is a dynamic iteration scheme, and the error is estimated using the initial iterate, namely $\mathsf{v}^{(0)}$. ■

Remark 4.3 It is clear that the same results are valid also for problems involving the more general boundary condition (27), provided (29) is satisfied and the inverse of this matrix is continuous with respect to t. In fact, the proof is exactly the same as above, since we never use the explicit form of the corresponding ODEs.

Example 4.4 *Apply the subdomain iteration to the system of* Example 2.5 *and* Example 2.9. *According to* (35), (36), *the point values in* Ω_i *satisfy the equations*

$$\dot{\mathsf{v}}_{i0}^{(\ell)} = S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_0 - C\mathsf{v}_{i0}^{(\ell)} + f\right) + S_-^{-1}S_-\dot{\mathsf{v}}_{i+1,N}^{(\ell-1)} \,, \tag{40}$$

$$\dot{\mathsf{v}}_{ik}^{(\ell)} = -A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_k - C\mathsf{v}_{ik}^{(\ell)} + f \,, \quad k = 1,\ldots,N-1 \,,$$

$$\dot{\mathsf{v}}_{iN}^{(\ell)} = S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_N - C\mathsf{v}_{iN}^{(\ell)} + f\right) + S_+^{-1}S_+\dot{\mathsf{v}}_{i-1,0}^{(\ell-1)} \,, \tag{41}$$

Hence, for $\ell \ge 2$, *this set of equations is equivalent to*

$$\begin{aligned}\dot{\mathsf{v}}_{i0}^{(\ell)} = {} & S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_0 - C\mathsf{v}_{i0}^{(\ell)} + f\right) \\ & + S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_{i+1}^{(\ell-1)})_N - C\mathsf{v}_{i+1,N}^{(\ell-1)} + f\right) \,,\end{aligned}$$

$$\dot{\mathsf{v}}_{ik}^{(\ell)} = -A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_k - C\mathsf{v}_{ik}^{(\ell)} + f \,, \quad k = 1,\ldots,N-1 \,,$$

$$\begin{aligned}\dot{\mathsf{v}}_{iN}^{(\ell)} = {} & S_-^{-1}S_-\left(-A(\mathsf{D}_1\mathsf{v}_i^{(\ell)})_N - C\mathsf{v}_{iN}^{(\ell)} + f\right) \\ & + S_+^{-1}S_+\left(-A(\mathsf{D}_1\mathsf{v}_{i-1}^{(\ell-1)})_0 - C\mathsf{v}_{i-1,0}^{(\ell-1)} + f\right) \,,\end{aligned}$$

which clearly represents a dynamic iteration method for the solution of the ODEs (25)–(26).

The general convergence theory presented in the Appendix is used to establish superlinear convergence on all finite time intervals as $\ell \to \infty$. Note that for T greater than some problem dependent number, the general error estimates (50), (51) opens for a blow-up of the error in the first iterations, before the superlinear convergence comes into action. Such a blow-up in the estimates (34), (39) seems unavoidable when we use e.g. the Chebyshev collocation method, since even for the simplest hyperbolic equation $D_t u + D_x u = 0$ the one-sided Lipschitz constant corresponding to the ODE formulated iteration grows like $\mathcal{O}(N^2)$. This is connected to the fact that this particular semi-discretization is merely algebraically stable (see e.g., [10]); the error in the approximation of the initial data has in this case a component growing like $\exp(N^2)$ at $t = 0^+$. Although this "hump" is theoretically countered by the rate of which this component vanishes, it can obviously lead to numerical problems. For smooth initial data, however, this "hump" does not, in our experience, lead to any noticeable blow-up in the first iterations. This is reflected in our numerical experiments using this type of discretization, in which we find that the iteration is generally well-behaved. For spatial discretizations in which the derivative matrix is normal, the situation is different; the estimate is now reflecting the true global growth property of the solution.

The iteration can also be performed using a "windowing" strategy, in which one divides the time interval $(0, T)$ into smaller subintervals, or windows, (T_j, T_{j+1}) and solve the problem in each window by the iterative method. Since the error in a window of length $T^\star > 0$ is bounded by $C(K_1)(K_2 T^\star)^{\ell-1}/(\ell-1)!$ (see the Appendix), it is clear that we can get a nice control over the error by choosing $T^\star < K_2^{-1}$. Since the error estimate of the iteration is often too conservative, the iteration will in many cases behave nicely also in longer windows. Convergence of the semi-discrete subdomain iteration method using the windowing strategy follows by a slight modification of the results given in [6].

4.2 The fully discrete case

To be practical, the subdomain iteration method of Section 4.1 must in general be discretized in time. There are at least two approaches to this issue, one corresponding to the formulation (32), (33) (or the equivalent semi-discrete formulation (35), (36)), and one corresponding to the formulation (37), (38).

In the first case, assume that for each $i = 1, \ldots, M^n$, we have at hand a sequence in which the jth term represents an approximation to $\mathsf{v}_i^{(\ell-1)}(t_{i,j})$, with $t_{i,0} = 0$ and $t_{i,j} < t_{i,j+1}$. We then supply the adjacent subdomains $\Omega_{i'}$ with approximations to the factors $\mathsf{v}_i^{(\ell-1)}$ (resp. $\mathsf{v}_i^{(\ell-1)}$ and $\dot{\mathsf{v}}_i^{(\ell-1)}$) which are needed during the time integration of (33) (resp. (36)). Unless in the case where the same time step is used in all subdomains and in all iterations, this must in general be accomplished by some sort of interpolation of the sequence. This suggests that numerical methods such as continuous Runge-Kutta methods might be interesting for this approach (see e.g., Owren and Zennaro [17]); these are methods which provide high order interpolants as part of the numerical approximation. An important practical aspect of this approach is that such approximants can in general be constructed only given the point values on the interfaces, hence we avoid storing the point values of the previous iterates at all collocation points. At present we have no convergence results for this type of fully discrete subdomain iteration method.

The other approach, using the form (37), (38) of the iteration, is easier to analyze but perhaps less interesting for actual implementation purposes. It is easier to analyze since it represents a dynamic iteration method, and there exists some convergence results for time-discretized versions of such. In the case where the Euler method or the backward Euler method with constant step size is used for the time integration, the results in [3] guarantee convergence of the method, and provide error estimates in the same spirit as above. For general linear multistep methods, the results of [16] can be used. It is also not difficult to verify that explicit methods yield convergence in a finite number of iterations. Bellen and Zennaro [1] has investigated the effect of using continuous Runge-Kutta methods in connection with the dynamic iteration, as this facilitates the implementation of a variable step size implementation, much like in the approach discussed above. They establish convergence also in this case. However, using the form (37), (38) in an actual implementation would in general mean that we have to store more point values from the previous iteration; the exact amount depending on the bandwidth in the rows of the derivative matrix corresponding to the derivatives at the interfaces. For the spectral collocation method this means that all point values must be stored in each iteration, thus making the first approach more attractive from a practical point of view. However, for finite differences this approach might be of interest.

As mentioned above, it might be interesting to do the iteration using a windowing strategy. In a parallel implementation it is nice to have as long windows as possible in order to (i) minimize the ratio between communication and computation, and (ii) to allow for individual time-stepping in each subdomain. On the other hand, the error estimate for the general case indicates that the convergence rate improves when the window length is reduced. Hence there seems to be need for some sort of trade-off in the determination of the optimal window length. Intuitively, it is reasonable to relate the window length to the "sound speed" of the problem. However, in the next section, it is shown that in special one-dimensional cases the window length has no influence on the rate of convergence. This corresponds to the behaviour seen in the continuous case.

Remark 4.5 The fully discrete subdomain iteration method analyzed by Quarteroni in [18] for a 2×2 problem is effectively applying the windowing strategy, using the shortest possible choice of window length: the size of a time step.

4.3 Special case: One-dimensional constant coefficient problems

In one space dimension ($n = 1$), we know that the continuous subdomain iteration scheme (3) converges in $2M - 1$ iterations when the coefficients are constant and the matrix C has the property that either

$$S_-(1) C S_+^{-1}(1) = 0 \tag{42}$$

or

$$S_+(1) C S_-^{-1}(1) = 0 \ . \tag{43}$$

If both (42) and (43) are satisfied, the iteration converges in M iterations (see [5]). In these situations $S(1)$ and the differential operators D_t, D_x commute, enabling us to introduce new variables $S(1)u$ for which the corresponding system of ODEs decouples in two independent subsystems. One of these represents flow to the left, the other represents flow to the right.

Let us consider the semi-discrete case. Let us use the notation of the examples above, in which we denote $S(1)$ by S, etc. Obviously S and $\frac{d}{dt}$ commute. Since the discrete differential operator is given by

$$(\mathsf{D}_1 \mathsf{v}_i)_k = \sum_{j=0}^{N} d_{kj} \mathsf{v}_{ij} \ ,$$

where $d_{kj} \in \mathbb{C}$ are the elements of the scalar derivative matrix $\mathsf{D} = (d_{jk})$, it is clear that S also commutes with this operator. Hence we can reformulate the semi-discrete subdomain iteration in terms of the transformed point values

$$S\mathsf{v}_{ik}^{(\ell)} = \begin{pmatrix} S_+ \mathsf{v}_{ik}^{(\ell)} \\ S_- \mathsf{v}_{ik}^{(\ell)} \end{pmatrix} \ .$$

Let $\mathsf{w}_{ik}^{(\ell)} = S\mathsf{v}_{ik}^{(\ell)}$, and let $\mathsf{w}_{ik\pm}^{(\ell)} = S_\pm \mathsf{v}_{ik}^{(\ell)}$. Similarly, let $\mathsf{w}_i^{(\ell)} = \operatorname{diag}(S, \ldots, S)\mathsf{v}_i^{(\ell)}$, and let $\mathsf{w}_{i\pm}^{(\ell)} = \operatorname{diag}(S_\pm, \ldots, S_\pm)\mathsf{v}_i^{(\ell)}$. The equations (40)–(41) for the point values in Ω_i are transformed to

$$\begin{aligned}
\dot{\mathsf{w}}_{i0} = & -\begin{pmatrix} \Lambda_+ & 0 \\ 0 & 0 \end{pmatrix} (\mathsf{D}_1 \mathsf{w}_i^{(\ell)})_0 - \begin{pmatrix} S_+ C S_+^{-1} & S_+ C S_-^{-1} \\ 0 & 0 \end{pmatrix} \mathsf{w}_{i0}^{(\ell)} \\
& + \begin{pmatrix} S_+ \\ 0 \end{pmatrix} f + \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \dot{\mathsf{w}}_{i+1,N}^{(\ell-1)} \ , \\
\dot{\mathsf{w}}_{ik} = & -\begin{pmatrix} \Lambda_+ & 0 \\ 0 & \Lambda_- \end{pmatrix} (\mathsf{D}_1 \mathsf{w}_i^{(\ell)})_k - \begin{pmatrix} S_+ C S_+^{-1} & S_+ C S_-^{-1} \\ S_- C S_+^{-1} & S_- C S_-^{-1} \end{pmatrix} \mathsf{w}_{ik}^{(\ell)} \\
& + Sf \ , \quad k = 1, \ldots, N-1 \ , \\
\dot{\mathsf{w}}_{iN} = & -\begin{pmatrix} 0 & 0 \\ 0 & \Lambda_- \end{pmatrix} (\mathsf{D}_1 \mathsf{w}_i^{(\ell)})_N - \begin{pmatrix} 0 & 0 \\ S_- C S_+^{-1} & S_- C S_-^{-1} \end{pmatrix} \mathsf{w}_{iN}^{(\ell)} \\
& + \begin{pmatrix} 0 \\ S_- \end{pmatrix} f + \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \dot{\mathsf{w}}_{i-1,0}^{(\ell-1)} \ ;
\end{aligned}$$

with the obvious modifications for $i = 1$ and $i = M$. Hence, if (42), say, is satisfied, the ODE for $\mathsf{w}_{i-}^{(\ell)}$ is independent of $\mathsf{w}_{i+}^{(\ell)}$, and depends only on $\mathsf{w}_{i+1-}^{(\ell-1)}$. So all the $\mathsf{w}_{i-}^{(\ell)}$ have converged after M iterations, which means that the $\mathsf{w}_{i+}^{(\ell)}$ have converged after $M-1$ additional iterations. If both (42) and (43) are satisfied, the whole system has converged after M iterations. It is clear that the assumption that C is constant is not essential here, hence we have established the following result.

THEOREM 4.6 *Let $n = 1$, let A be constant, and let* ASSUMPTION 2.1 *and* ASSUMPTION 2.8 *hold. Let the decomposed problem* (6) *be semi-discretized by the method of lines, using pairwise matching in the interior globs, and inflow-outflow conditions in exterior globs. Then, if either* (42) *or* (43) *is satisfied for all (collocation) points, the semi-discrete subdomain iteration based on* (32), (33) *converges in $2M - 1$ iterations. If both* (42) *and* (43) *are satisfied, it converges in M iterations.*

For any matrices Q and W, with W non-singular, the sequence $\{W^{-1}u_n\}$, where u_n is an approximation to $u(t_n)$ of the initial value problem

$$\dot{u} = WQW^{-1}u + Wf \quad , \quad u(0) = Wy_0$$

computed by a Runge-Kutta (RK) method, coincides with the sequence $\{y_n\}$, where y_n is the approximation to $y(t_n)$ of the problem

$$\dot{y} = Qy + f \quad , \quad y(0) = y_0$$

computed by the same RK method. That is, the RK method commutes with a simple transformation of the unknown. It follows immediately that if we use an RK method to approximate each of the $\mathsf{w}^{(\ell)} = (\mathsf{w}_1^{(\ell)}, \ldots, \mathsf{w}_M^{(\ell)})^T$ by any of the two approaches discussed in the previous section, and the sequence of step sizes used in each subdomain are independent of ℓ, it will have the same convergence properties as the semi-discrete iteration. Since w is given by a simple transformation of v, it follows by the above that the same is true also for the non-transformed system. Hence we have the following corollary.

COROLLARY 4.7 *Assume that we use an RK method to integrate the ODE formulated semi-discrete subdomain iteration method of* THEOREM 4.6. *Assume further that the sequence of step sizes used in each subdomain is independent of ℓ, and that any approximation of incoming quantities from an adjacent subdomain is a function only of the sequences of step sizes and the corresponding outgoing quantities in the neighboring subdomain. Then the same convergence property as presented in* THEOREM 4.6 *is valid also for the fully discrete case.*

Remark 4.8 Of course, in the case in which the boundary conditions are not characteristic, that is, there is reflection at the boundaries, the situation is not so simple (cf. Quarteroni [18]). However, if there is reflection at only one of the boundaries, and this is the boundary for which the 'outgoing characteristic combinations' have converged after M iterations, the same argument would guarantee convergence in $2M - 1$ iterations.

5 A related result for the heat equation

In this section we use the results from the Appendix to present a convergence result for a subdomain iteration method for a semi-discretization of the one-dimensional heat equation. This digression from the main topic of this paper is included to emphasize that the convergence theorem of the previous section is nothing but a convergence theorem in ODE theory, and that the same technique can be applied to establish convergence of subdomain iteration also for other problems.

Consider the heat equation $D_t u = D_x^2 u$ for $-1 < x < 1$ and $0 < t < T$, with initial condition u_0 and zero boundary conditions. We know that for solutions in $C^2([-1,1] \times [0,T])$ this problem is equivalent to the two coupled problems

$$\begin{aligned} D_t v = D_x^2 v & \quad \text{in } (-1,0) \times (0,T) \\ v = 0 & \quad \text{on } \{-1\} \times (0,T) \\ v = u_0 & \quad \text{on } (-1,0) \times \{0\} \,, \end{aligned}$$

$$\begin{aligned} D_t w = D_x^2 w & \quad \text{in } (0,1) \times (0,T) \\ w = 0 & \quad \text{on } \{1\} \times (0,T) \\ w = u_0 & \quad \text{on } (0,1) \times \{0\} \,, \end{aligned}$$

$$\begin{aligned} v = w & \quad \text{on } \{0\} \times (0,T) \\ D_x v = D_x w & \quad \text{on } \{0\} \times (0,T) \,. \end{aligned}$$

Now, let us discretize in space by finite differences, using central differences in the interior of each subdomain, and one-sided differences for the derivative at the interface. Using a left-to-right ordering in each of the two domains, we get the differential-algebraic system for the $2N$ unknown functions v_i, $i = 1, \ldots, N$ and w_i, $i = 0, \ldots, N-1$:

$$\begin{aligned} \dot{\mathsf{v}}_1 = & \quad (\Delta x)^{-2} (-2\mathsf{v}_1 + \mathsf{v}_2) \\ \dot{\mathsf{v}}_k = & \quad (\Delta x)^{-2} (\mathsf{v}_{k+1} - 2\mathsf{v}_k + \mathsf{v}_{k-1}) \,, \; k = 2, \ldots, N-1 \,, \end{aligned} \tag{44}$$

$$\begin{aligned} \dot{\mathsf{w}}_k = & \quad (\Delta x)^{-2} (\mathsf{w}_{k-1} - 2\mathsf{w}_k + \mathsf{w}_{k+1}) \,, \; k = 1, \ldots, N-2 \\ \dot{\mathsf{w}}_{N-1} = & \quad (\Delta x)^{-2} (\mathsf{w}_{N-2} - 2\mathsf{w}_{N-1}) \,, \end{aligned} \tag{45}$$

$$\begin{aligned} \mathsf{v}_N &= \mathsf{w}_0 \\ \mathsf{v}_N - \mathsf{v}_{N-1} &= \mathsf{w}_1 - \mathsf{w}_0 \,, \end{aligned} \tag{46}$$

for all $t > 0$. The initial conditions are given by $\mathsf{v}_k(0) = u_0(-1 + k\Delta x)$ for $k = 1, \ldots, N-1$, $\mathsf{w}_k(0) = u_0(k\Delta x)$ for $k = 1, \ldots, N-1$, and $\mathsf{v}_N(0) = \mathsf{w}_0(0) = (u_0(-\Delta x) + u_0(\Delta x))/2$.

The system (44)–(46) can be equivalently formulated as a pure ODE system by taking the time derivative of the algebraic interface conditions (46). The ODEs for v_N and w_0 are then

$$\dot{\mathsf{v}}_N = \dot{\mathsf{w}}_0 = \frac{1}{2} (\dot{\mathsf{v}}_{N-1} + \dot{\mathsf{w}}_1) = (\Delta x)^{-2} \left(\frac{1}{2}\mathsf{v}_{N-2} - \mathsf{v}_{N-1} + \frac{1}{2}\mathsf{v}_N + \frac{1}{2}\mathsf{w}_0 - \mathsf{w}_1 + \frac{1}{2}\mathsf{w}_2 \right) \,. \tag{47}$$

By the results of the Appendix, it follows that a subdomain iteration method for (44), (45), (46) always converges if it is a dynamic iteration scheme for the equivalent ODE system (44), (45), (47). In particular, this is the case for the following subdomain iteration method: For each $\ell = 1, 2, \ldots$, find $\mathsf{v}^{(\ell)}, \mathsf{w}^{(\ell)}$ satisfying

$$
\begin{aligned}
\dot{\mathsf{v}}_1^{(\ell)} &= (\Delta x)^{-2}\left(-2\mathsf{v}_1^{(\ell)} + \mathsf{v}_2^{(\ell)}\right) \\
\dot{\mathsf{v}}_k^{(\ell)} &= (\Delta x)^{-2}\left(\mathsf{v}_{k+1}^{(\ell)} - 2\mathsf{v}_k^{(\ell)} + \mathsf{v}_{k-1}^{(\ell)}\right) \ , \ k = 2, \ldots, N-1 \ ,
\end{aligned}
$$

$$
\begin{aligned}
\dot{\mathsf{w}}_k^{(\ell)} &= (\Delta x)^{-2}\left(\mathsf{w}_{k-1}^{(\ell)} - 2\mathsf{w}_k^{(\ell)} + \mathsf{w}_{k+1}^{(\ell)}\right) \ , \ k = 1, \ldots, N-2 \\
\dot{\mathsf{w}}_{N-1}^{(\ell)} &= (\Delta x)^{-2}\left(\mathsf{w}_{N-2}^{(\ell)} - 2\mathsf{w}_{N-1}^{(\ell)}\right) \ ,
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{v}_N^{(\ell)} &= \mathsf{w}_0^{(\ell)} \\
\mathsf{v}_N^{(\ell)} - \mathsf{v}_{N-1}^{(\ell-1)} &= \mathsf{w}_1^{(\ell-1)} - \mathsf{w}_0^{(\ell)} \ ,
\end{aligned} \tag{48}
$$

for all $t > 0$. This follows directly by the fact that (48) is equivalent to

$$
\dot{\mathsf{v}}_N^{(\ell)} = \dot{\mathsf{w}}_0^{(\ell)} = (\Delta x)^{-2}\left(\frac{1}{2}\mathsf{v}_{N-2}^{(\ell-1)} - \mathsf{v}_{N-1}^{(\ell-1)} + 2\mathsf{v}_N^{(\ell-1)} - \mathsf{w}_1^{(\ell-1)} + \frac{1}{2}\mathsf{w}_2^{(\ell-1)}\right) \ .
$$

which is a splitting of (47). Note that this method consists of two independent problems of size N to be solved in each iteration. The method is nothing but a Picard iteration for the ODE system. A block-Jacobi splitting can also be applied; in this case the functions $\mathsf{v}_N^{(\ell)}$ and $\mathsf{w}_0^{(\ell)}$ will in general not coincide for $t > 0$.

Remark 5.1 We note that this technique can not be used directly to analyze the convergence properties of a Dirichlet-Neumann subdomain iteration, given by the interface conditions

$$
\begin{aligned}
\mathsf{v}_N^{(\ell)} &= \mathsf{w}_0^{(\ell-1)} \\
\mathsf{v}_N^{(\ell-1)} - \mathsf{v}_{N-1}^{(\ell-1)} &= \mathsf{w}_1^{(\ell)} - \mathsf{w}_0^{(\ell)} \ ,
\end{aligned}
$$

since this system does not correspond to a dynamic iteration scheme for ODE problem (44), (45), (47). This might be a motivation for studying generalized dynamic iteration schemes, in which more than two iteration levels are involved in each iteration.

6 Numerical experiments

In this section we report some results of numerical experiments made with the subdomain iteration method. The purpose is twofold; firstly we would like to point out that the method is fairly simple to implement given a one-domain solver, and that considerable savings in execution time is obtainable in a parallel environment. Secondly, the experiments indicate that there are several unsolved problems in connection with the implementation of the method. In particular a more careful treatment of the approximation of the incoming information than just simple linear interpolation seems to be required to obtain sufficiently accurate approximations. Moreover, the two-dimensional example show, not unexpectedly, that it is the corner points that are the trouble-makers also in our implementation of the subdomain iteration method. To distinguish between the unknown vector u and its components, we will in this section denote the vector by $\vec{u}$.

We have applied the technique to two constant coefficient problems. The first is the one-dimensional test problem with $\vec{u} = (u_1, u_2)^T$:

$$
\begin{aligned}
D_t\vec{u} + \begin{pmatrix} 1/2 & 1 \\ 1 & 1/2 \end{pmatrix} D_x\vec{u} = \vec{0} \quad & \text{for} \quad -1 < x < 1\,,\; 0 < t < T\,, \\
\vec{u} = \begin{pmatrix} 0 \\ \sin \pi x \end{pmatrix} \quad & \text{for} \quad -1 < x < 1\,,\; t = 0\,, \\
S_-(-1)\vec{u} = \sin \tfrac{3\pi t}{2} \quad & \text{for} \quad x = -1\,,\; t > 0\,, \\
S_-(+1)\vec{u} = -\sin \tfrac{\pi t}{2} \quad & \text{for} \quad x = 1\,,\; 0 < t < T\,.
\end{aligned}
$$

The other is the two-dimensional linearized shallow water equations with a Coriolis term but without bottom friction; here $\vec{u} = (u, v, h)^T$:

$$
D_t\vec{u} + \begin{pmatrix} u_0 & 0 & 1 \\ 0 & u_0 & 0 \\ 1 & 0 & u_0 \end{pmatrix} D_x\vec{u} + \begin{pmatrix} v_0 & 0 & 0 \\ 0 & v_0 & 1 \\ 0 & 1 & v_0 \end{pmatrix} D_y\vec{u} + \begin{pmatrix} 0 & -K & 0 \\ K & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{u} = \vec{0}
$$

$$
\begin{aligned}
& \text{for} \quad -1 < x, y < 1\,,\; 0 < t < T\,, \\
\vec{u} = \vec{u}_0 := \begin{pmatrix} u_0 \\ v_0 \\ h_0 + h_p \end{pmatrix} \quad & \text{for} \quad -1 < x, y < 1\,,\; t = 0\,, \\
S_-(\nu)\vec{u} = S_-(\nu)\vec{u}_0 \quad & \text{for} \quad x \vee y = \pm 1\,,\; 0 < t < T\,.
\end{aligned}
$$

Here $K = 0.005$, and we have chosen $u_0 = 0.1$, $v_0 = 0.05$, and $h_0 = K(v_0 x - u_0 y)$. The initial function $\vec{u}_0$ thus represents a perturbation in the h variable from this steady state solution by the amount h_p, which is given by

$$
h_p = 0.05\, e^{-50(x^2+y^2)}\,.
$$

Since the subproblems are of the same type as the original problem, the iteration can be implemented using an existing one-domain solver. Hence, having at hand a one domain solver based on the ODE formulation of systems semi-discretized by the Chebyshev spectral collocation method, we used the ODE formulation (35), (36) of the subdomain iteration method to obtain the numerical approximation. This was combined with a second order

Runge-Kutta method with variable step size based on a first order error estimate. The associated Butcher tableau is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \\ \hline & 1 & 0 \end{array} ,$$

(see e.g., [11]). In our experiments, the absolute and the relative error tolerance, based on the maximum norm, was set to 10^{-4}. In addition to the computational routines, we had to add administrative tasks such as performing the iteration loop, storing the iterates, and communicating data between the subdomains. The point values and their time derivatives which are part of the right hand sides of (35), (36) were obtained simply by linear interpolation in the linked lists holding the point values at the interfaces.

The first problem was integrated to $T = 5.0$, using a total of 81 and 161 collocation points in space, respectively. As the initial guess we chose the constant values given by the initial condition. In each of the two cases we computed the solution letting M, the number of equally sized subdomains, take the values $1, 2, 4$, and 8. Even though the sequence of step sizes used in each subdomain was not independent of ℓ, hence violating one of the assumptions in COROLLARY 4.7, it seemed reasonable to stop the iteration after M iterations. The overall time consumption of the program when run on an Intel Paragon parallel machine is reported in Figure 4. Below the solid line which represents the time used, there are two other lines representing the maximum and minimum of the time used at each processor for computations. It is thus seen that when the number of subdomains increase in this moderate way, the load imbalance increases. The exact solution of this problem is known to be

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\sin\pi(x+\frac{t}{2}) + \frac{1}{2}\sin\pi(x-\frac{3t}{2}) \\ \frac{1}{2}\sin\pi(x+\frac{t}{2}) + \frac{1}{2}\sin\pi(x-\frac{3t}{2}) \end{pmatrix} ,$$

hence we can compare the quality of the computed approximations. Since the error tolerance per step in the ODE solver is 10^{-4}, we should not expect getting anything closer to the exact solution than this. This is reflected in Figure 5 in which we plot the maximum error over all collocation points and over 100 equidistant time levels, and where the error in the one-domain solution is roughly 10^{-4}. In Figures 6 and 7 we plot the maximum error over all time levels in each collocation point. We see that the approximation clearly deteriorates as M increases. One plausible explanation for this behaviour is that it is caused by errors introduced by the simple approximation used for the time derivatives in the right hand side of the ODE formulated problem. However, this clearly needs to be investigated in more detail.

The second problem was discretized by using pairwise matching in the left-right direction, implying that no subdomain would have to communicate with more than four other subdomains. On the boundary, the four corner points were treated by imposing the condition $S_-(\xi)\vec{u} = S_-(\xi)\vec{u}_0$ for ξ in the direction 45 degrees on the corner, pointing out of the domain. The remaining exterior 0-globs were treated as points on the boundary.

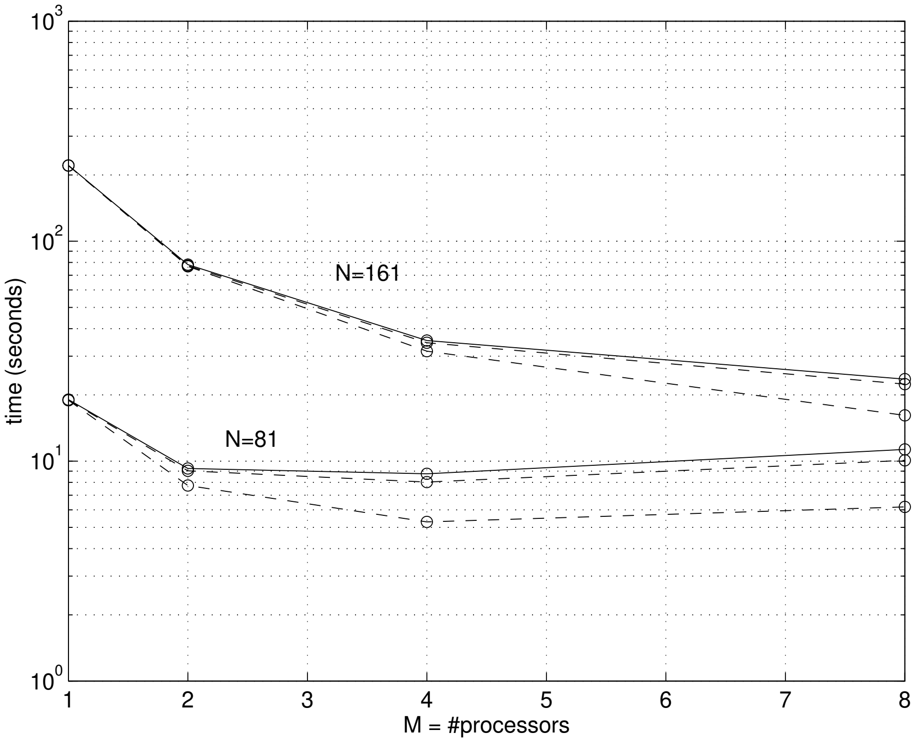

Figure 4: 1-dim. test problem: Time used on the Paragon

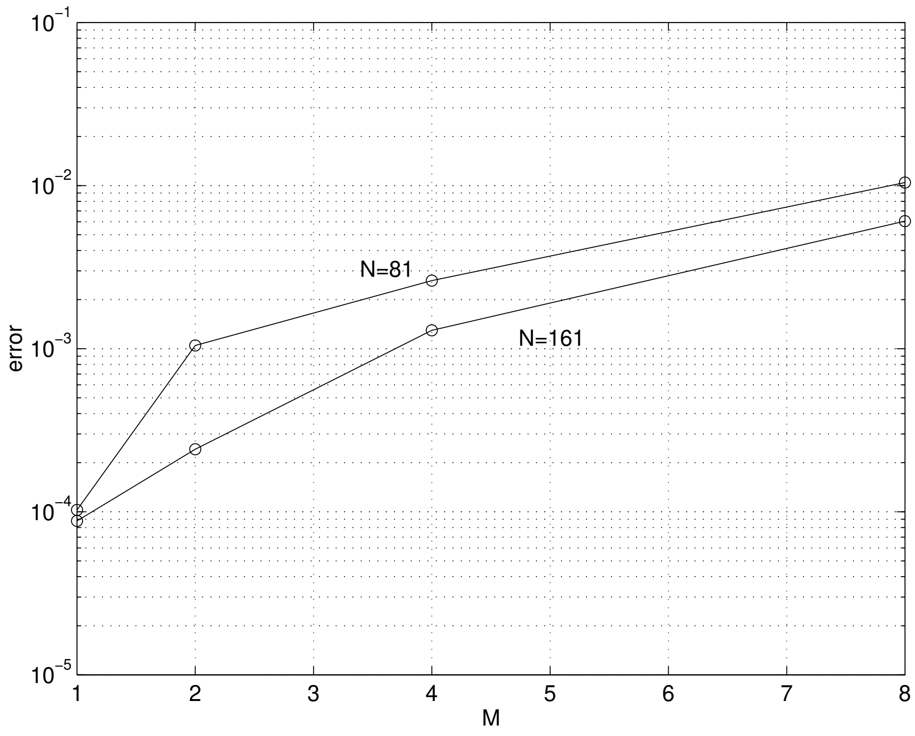

Figure 5: 1-dim. test problem: Error w.r.t. the exact solution

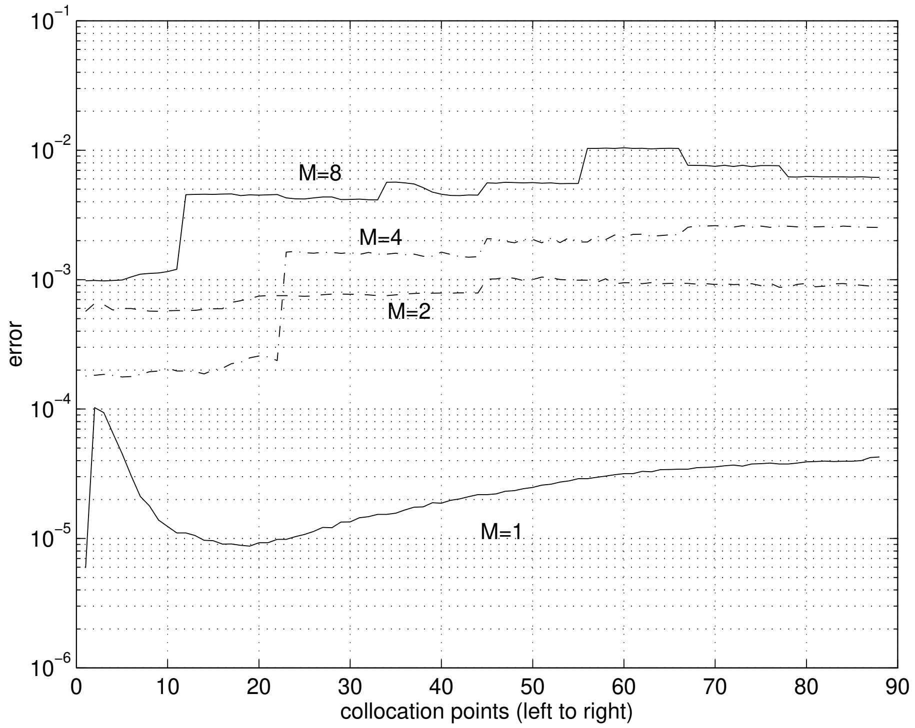

Figure 6: 1-dim. test problem (N=81): Error in the collocation points

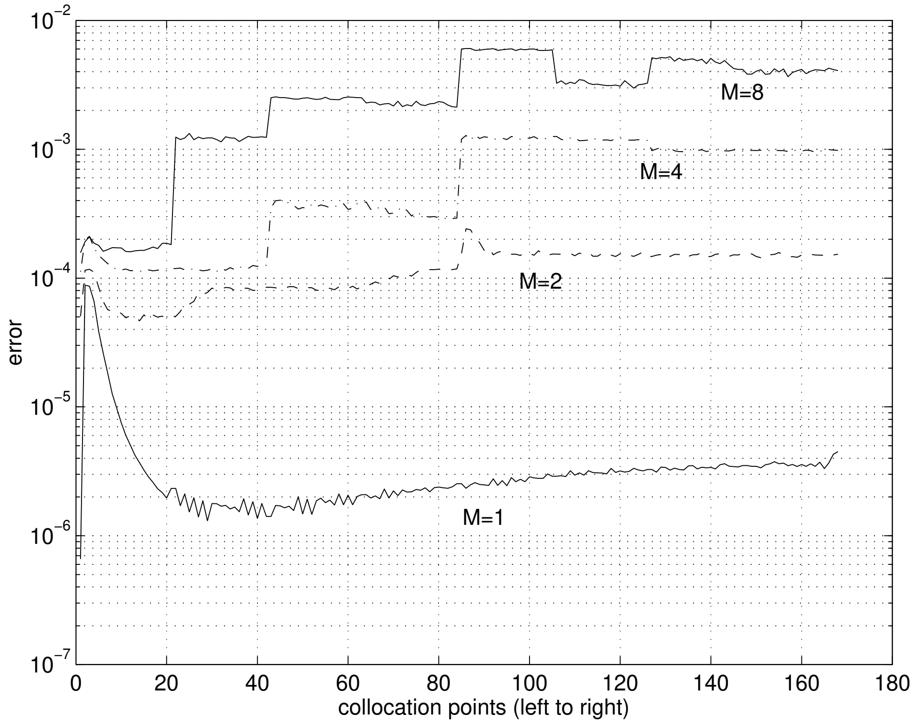

Figure 7: 1-dim. test problem (N=161): Error in the collocation points

However, to get a stable treatment of the corner points in our computations of the subproblems, all the matching and inflow-outflow directions were rotated so that they too was 45 degrees with respect to the corner. It is not clear to us why the Chebyshev spectral collocation method is so sensitive to this change in directions. Again, the initial point values at the interfaces were chosen constant. In this case we have no (useful) results about how many iterates to take. However, according to Nevanlinna [16], it is reasonable to use the difference between two successive iterates as a measure of convergence in a dynamic iteration scheme. Using this measure in lack of something better we found, for the test with a total of 81×81 collocation points, that for the cases $M = 4, 5, 6$ a choice of $2M$ iterations was enough to give a difference between two successive iterates at the same level as the error allowed by the ODE solver. For $M = 2, 3$ this number of iterations (4 and 6 respectively) yielded a difference that was about one magnitude larger. In the experiment we integrated the equations to $T = 0.1$ using $2M$ sweeps (iterations) on M^2 processors for $M = 1, 2, 3, 4, 5$, and 6. In the first test series the total number of collocation points was 81×81, in the second it was 101×101. Timing results for the Intel Paragon is reported in Figures 8 and 9. The meaning of the two lines below the solid line is as for the one-dimensional example. The reason for the large load imbalance is in this case clear, since the short time interval and the nature of the initial condition implies that very little is going on in the subdomains close to the boundary compared to the subdomains close to the middle of the domain. For the 81×81 case, we have plotted in Figure 10 the maximum difference between the interpolated solutions in the single domain case and in the case $M = 4$, over 81×81 equidistant points in space and 4 equidistant points in time. The difference is relatively large, and it turns out that its main contributor is a corner point in one of the four domains in the middle of the square. This is yet another illustration of the fact that corner points in spectral collocation approximations need special attention. Figure 11 shows contour plots of the solution at $t = 0.1$ in the $M = 4$ case after 10 sweeps.

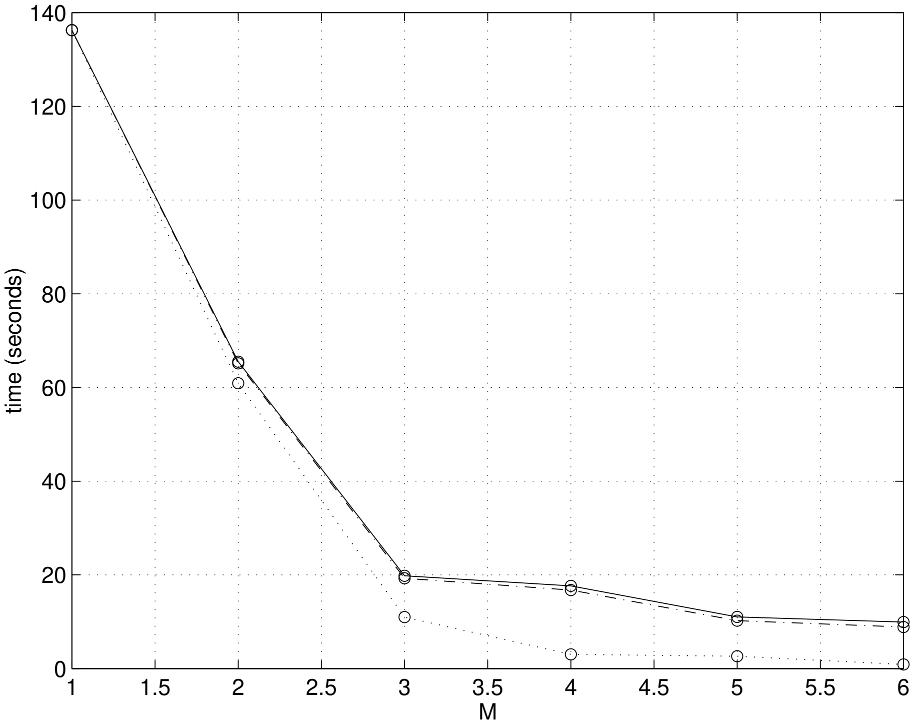

Figure 8: Shallow water equations (N=81): Time used on the Paragon

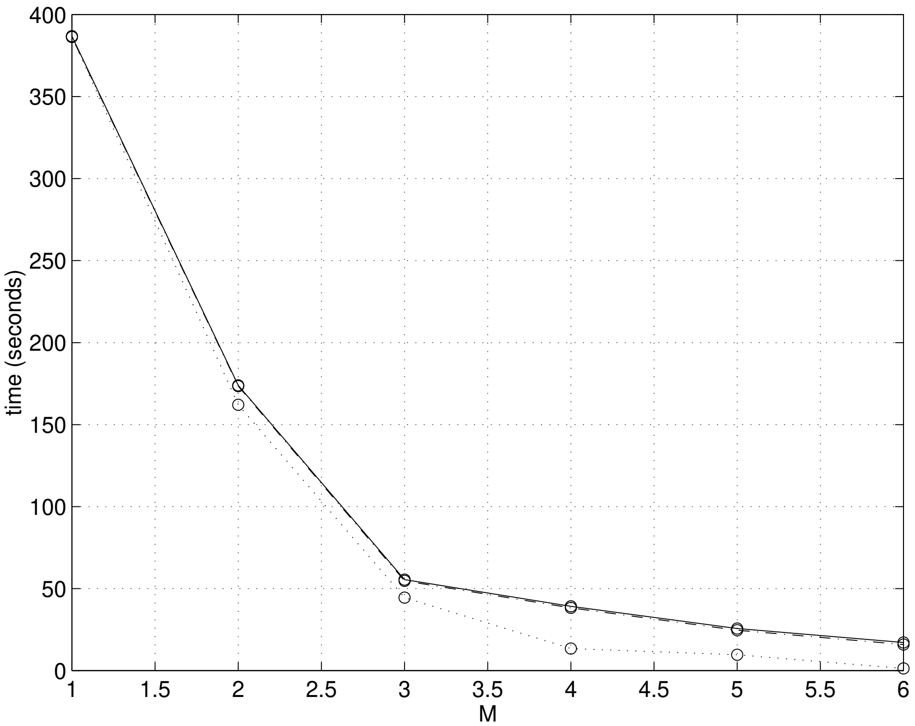

Figure 9: Shallow water equations (N=101): Time used on the Paragon

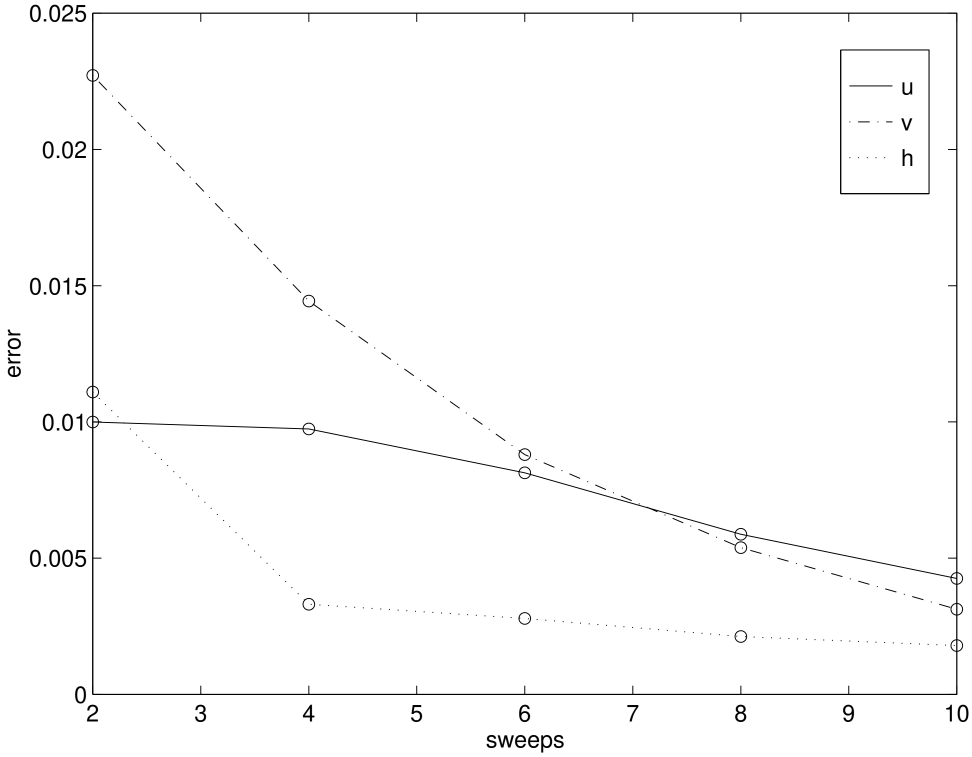

Figure 10: Shallow water equations (N=81, M=4): 'error' w.r.t. single domain solution

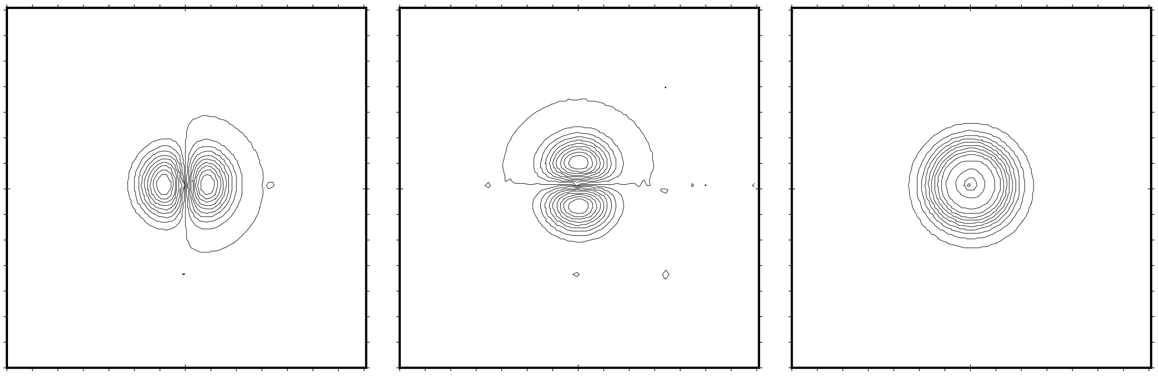

Figure 11: Shallow water equations (N=81, M=4): u, v, and h at $t = 0.1$

A Appendix: Dynamic Iteration for ODEs

We list some well-known results from ODE theory, concerning the so-called *dynamic iteration* or *waveform relaxation* methods, which we use above. For these and related results we refer the reader to references [1, 3, 15, 16, 20]. All norms refer to the euclidean norm.

Let $f : \mathbb{R} \times \mathbb{C}^m \to \mathbb{C}^m$. We say that $\tilde{f} : \mathbb{R} \times \mathbb{C}^m \times \mathbb{C}^m \to \mathbb{C}^m$ is a splitting of f if

$$\tilde{f}(t, y, y) = f(t, y)$$

for all $t \in \mathbb{R}$ and $y \in \mathbb{C}^m$.

The following result establishes the convergence of the dynamic iteration method.

LEMMA A.1 *Let $f : \mathbb{R} \times \mathbb{C}^m \to \mathbb{C}^m$ be continuous with respect to its first argument and Lipschitz continuous with respect to its second argument, and let $y \in C^1([0,\infty))^m$ be the solution of*

$$\begin{aligned} \dot{y} &= f(t, y)\ ,\ t > 0 \\ y(0) &= y_0\ . \end{aligned}$$

Let $\tilde{f}$ be a splitting of f. If $\tilde{f}$ is continuous with respect to its first argument and Lipschitz continuous with respect to its second and third arguments, and if $y^{(0)}$ is an arbitrary continuous function satisfying $y^{(0)}(0) = y_0$, then the sequence of functions $y^{(k)} \in C^1([0,\infty))^m$, $k = 1, 2, \ldots$ satisfying

$$\begin{aligned} \dot{y}^{(k)} &= \tilde{f}(t, y^{(k)}, y^{(k-1)})\ ,\ t > 0 \\ y^{(k)}(0) &= y_0\ , \end{aligned} \tag{49}$$

converges uniformly to y on every finite interval $(0, T)$. If the Lipschitz constants of $\tilde{f}$ with respect to the second and third arguments are K_1 and K_2, respectively, an error estimate is given by

$$\sup_{0 \le t \le T} \|y(t) - y^{(k)}(t)\| \le \frac{(K_2 T)^k}{k!} e^{K_1 T} \sup_{0 \le t \le T} \|y(t) - y^{(0)}(t)\|\ ; \tag{50}$$

from which it follows that the rate of convergence is superlinear as $k \to \infty$.

If instead K_1 is a one-sided Lipschitz constant (or logarithmic norm in the linear case), it is possible to get a finer estimate.

COROLLARY A.2 *If $K_1 \neq 0$ denotes a one-sided Lipschitz constant of $\tilde{f}$ with respect to its second argument, then an error estimate is given by*

$$\sup_{0 \le t \le T} \|y(t) - y^{(k)}(t)\| \le \left(\frac{K_2}{-K_1}\right)^k \left[e^{K_1 T} \sum_{j=k}^{\infty} \frac{(-K_1 T)^j}{j!} \right] \sup_{0 \le t \le T} \|y(t) - y^{(0)}(t)\|\ . \tag{51}$$

Remark A.3 Note that if $K_1 > 0$ the term in square brackets is signed as $(-1)^k$, and it is bounded in absolute value by $e^{K_1 T}$. On the other hand, if $K_1 < 0$ the same term takes values in $(0, 1)$ and it decreases as k increases. In any case, the bound suggests that splittings with small values of the ratio $K_2/|K_1|$ are to be preferred.

References

[1] A. Bellen and M. Zennaro, *The use of Runge-Kutta formulae in waveform relaxation methods*, Appl. Numer. Math., 11 (1993), pp. 95–114.

[2] M. Bjørhus, *Dynamic iteration for time-dependent partial differential equations: A first approach*, Tech. Report Numerics no. 1, Departments of Mathematics, The University of Trondheim, Norway, 1992.

[3] ——, *A note on the convergence of discretized dynamic iteration*, BIT, 35 (1995), pp. 291–296.

[4] ——, *The ODE formulation of hyperbolic PDEs discretized by the spectral collocation method*, SIAM J. Sci. Comput., 16 (1995), pp. 542–557.

[5] ——, *On domain decomposition and subdomain iteration*, Tech. Report Numerics no. 3, Departments of Mathematics, The University of Trondheim, Norway, 1995.

[6] M. Bjørhus and A. Stuart, *Waveform relaxation as a dynamical system.* To appear in Math. Comp.

[7] P. E. Bjørstad and O. B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., 23 (1986), pp. 1097–1120.

[8] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics, Springer-Verlag, Berlin Heidelberg, 1988.

[9] D. Funaro, A. Quarteroni, and P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal., 25 (1988), pp. 1213–1236.

[10] D. Gottlieb and S. A. Orszag, *Numerical analysis of spectral methods: Theory and Applications*, no. 26 in Regional conference series in applied mathematics, SIAM, Philadelphia, 1977.

[11] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, no. 8 in Springer Series in Computational Mathematics, Springer-Verlag, Berlin Heidelberg, 1987.

[12] ——, *Solving Ordinary Differential Equations I*, no. 8 in Springer Series in Computational Mathematics, Springer-Verlag, Berlin Heidelberg, second revised ed., 1993.

[13] D. A. Kopriva, *Computation of hyperbolic equations on complicated domains with patched and overset chebyshev grids*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 120–132.

[14] J. Mandel, *Hybrid domain decomposition with unstructured subdomains*, in Domain decomposition methods in science and engineering, A. Quarteroni et al., eds., vol. 157 of Contemporary Mathematics, AMS, 1994, pp. 103–112.

[15] U. Miekkala and O. Nevanlinna, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 459–482.

[16] O. Nevanlinna, *Remarks on Picard-Lindelöf iteration*, BIT, 29 (1989), pp. 328–346, 535–562.

[17] B. Owren and M. Zennaro, *Order barriers for continuous explicit Runge-Kutta methods*, Mathematics of Computation, 56 (1991), pp. 645–661.

[18] A. Quarteroni, *Domain decomposition methods for systems of conservation laws: Spectral collocation approximations*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 1029–1052.

[19] ——, *Domain decomposition and parallel processing for the numerical solution of partial differential equations*, Surveys on Mathematics for Industry, 1 (1991), pp. 75–118.

[20] R. Skålin, *Numerisk løsning av ordinære differensialligninger ved parallellisering på systemnivå. et studium av Runge-Kutta metoder*, master's thesis, Department of Mathematical Sciences, The Norwegian Institute of Technology, Trondheim, Norway, 1988.

[21] A. Solomonoff and E. Turkel, *Global properties of pseudospectral methods*, J. Comp. Phys., 81 (1989), pp. 239–276.

[22] C. E. Wasberg and Ø. Andreassen, *Pseudospectral methods with open boundary conditions for the study of atmospheric wave phenomena*, Comp. Meth. Appl. Mech. Eng., 80 (1990), pp. 459–465.

[23] K. Yosida, *Lectures on differential and integral equations*, Dover, New York, 1991. Reprint of edition published by Interscience, 1960.