Explicitly Solving Vectorial ODEs

B.Tasić, R.M.M.Mattheij
Department of Mathematics and Computing Science,
Eindhoven University of Technology,
PO Box 513, 5600 MB, The Netherlands

Abstract

This paper concerns a new method for finding numerical solutions of multivariate ODE flows, where the flow field is not given explicitly. The method is based on the existing implicit numerical methods, such as Euler Backward and the implicit midpoint rule. The main aspect is that the method as such is explicit, but with the similar accuracy and stability properties as the original (implicit) methods. A higher dimensional analysis is given of both stability and accuracy. A couple of examples illustrates this analysis.

Key words: ODE, flow, Euler Backward, implicit midpoint rule, inverse interpolation, stability, implicit, explicit

1 Introduction

Solving ODEs (Ordinary Differential Equations) numerically is a well-studied problem both in theory and in numerous practical applications. Examples can be found in almost all technical disciplines and many numerical methods are developed for the time discretisation of an autonomous ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^N \tag{1.1}$$

Frequently the method of choice is some implicit method, like for stiff problems (see e.g. [4, 8–10, 18, 23]) or problems where volume preservation is required (see e.g. [11]). The drawback of these methods is that the time discretisation leads to a system of nonlinear equations which needs to be solved at every time-level. This is usually done by some Newton type iterative method, which introduces additional computational costs and numerical errors in finding the solution. Nevertheless this approach gives satisfactory results in many applications and it became a "standard" way to (numerically) implicitly solve certain ODEs. However, in problems where the velocity field $\mathbf{u}$ is not

given *explicitly* on the entire space domain of interest $\Gamma \subset \mathbb{R}^N$, the use of an implicit method is not straightforward. Here one need to approximate $\mathbf{u}$ and its Jacobian matrix $D\mathbf{u}$ on Γ. This introduces new errors into the solution which cannot be neglected in the general case. Examples of such "implicit" problems are numerous. In problems from fluid dynamics the velocity is often numerically computed from a PDE (Partial Differential Equation), i.e it is known at the vertices only. The position then follows from an ODE given by (1.1). Often Γ is bounded, like a deforming material blob (see [11,14–17,24]). In such a case only the boundary may be of interest to describe the evolution of this blob. The velocity can also be the numerical solution of another ODE, like in BVP (Boundary Value Problem), shown in [22], or when it is obtained experimentally. Examples of experimentally obtained $\mathbf{u}$ can be found e.g. in electrical networks with nonlinear elements, such as transistors, diodes, nonlinear resistors, etc. (see [5]).

In a previous paper (see [22]) we introduced a new method (the so-called flow method), developed for solving the autonomous flow problem

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^N, \\ \mathbf{x}(0) \in \mathbf{I}(0), \end{cases} \tag{1.2}$$

where $\mathbf{I}(0) \subseteq \Gamma \subset \mathbb{R}^N$ and the velocity is discretely given, i.e. it is given at some points of Γ only. The method can be viewed as a modification of the Euler Backward method (EB), where the autonomy of the problem is employed and the solution is obtained by applying inverse interpolation. The result is a method which is explicit in a way, but with the accuracy and the stability properties similar to EB. The main goal of this paper is to extend the analysis given in [22], which was given for scalar case only, to multivariate problems. Even though the method principle remains the same, a further analysis is needed since multivariate inverse interpolation is more involved. We will also extend the analysis of our method to the IMR (Implicit Midpoint Rule) method. Since this method is closely related to EB (as we will show later), the implementation of IMR into the flow method is straightforward. Since IMR is a symplectic integrator, this allows us to employ our method in the applications where the volume preservation is essential issue.

The paper is built up as follows. In Section 2, we first give an outline of the mathematical problem and describe the basic idea behind the method. We also briefly discuss the properties of the inverse interpolation, since it is an essential ingredient of the method. The afore mentioned interpolation involves quite a complicated local error, justifying a separate error analysis, which is given in Section 3. In Section 4 it is shown that we obtain a stability behaviour similar to that of EB, despite the fact that our method is *de facto* explicit. We conclude the paper with some practical aspects of the method, given in Section 5, where we illustrate the method by two examples for a given discrete velocity field. The first example represents a problem where the velocity has to be found experimentally and the second problem involves a numerical solution of a discretised PDE.

2 Outline of the Method

Consider an autonomous flow problem (1.2) where $\mathbf{x}$ is a point of the flow and $\mathbf{u}(\mathbf{x})$ is not given explicitly; however, we will assume that $\mathbf{u}(\mathbf{x})$ is Lipschitz continuous. Here the flow, denoted by $\mathbf{I}(t)$, is meant to be the time evolution of a "continuum" of solutions (see [6]). This means that, in order to track $\mathbf{I}(t)$ numerically in time, one needs to properly discretise $\mathbf{I}(t)$ in space. Hence, we denote by $\{\mathbf{x}_j(t)\}_{j=1}^{q} \in \mathbf{I}(t)$ a set of points which gives a numerical spatial representation of the flow. Let us now assume that at a particular time point, $\mathbf{u}(\mathbf{x})$ is obtained numerically (or experimentally) at some spatial points, say $\{\mathbf{x}_k\}_{k=1}^{n} \in \Gamma$, i.e. it is given by the set of values $\{\mathbf{u}_k\}_{k=1}^{n}$. Here Γ represents a convex hull of the set $\{\mathbf{x}_k\}_{k=1}^{n}$. We distinguish $\{\mathbf{x}_k\}_{k=1}^{n}$ from $\{\mathbf{x}_j\}_{j=1}^{q}$ since $\mathbf{u}(\mathbf{x})$ can be obtained not only at the flow points. Of course, $\mathbf{I}(t) \subseteq \Gamma$ should hold for all t to avoid lack of information about the velocity. One very important application is where $\mathbf{u}(\mathbf{x})$ is known only at the flow points ($\{\mathbf{x}_k\}_{k=1}^{n} = \{\mathbf{x}_j\}_{j=1}^{q}$). The method's main idea is inspired by problems from fluid dynamics (see [11]) where the solution $\mathbf{x}$ is in fact the Eulerian position of a deforming material blob. Here one needs to discretise the flow at the particular time point to numerically obtain $\mathbf{u}$, coming from PDE (in particular Stokes equation). After this, information about $\mathbf{u}$ should be used to obtain the flow at the next time level.

To solve (1.2) we need a proper time discretisation. If the problem is stiff or the volume preservation is required, the method of choice is most likely implicit. We will restrict ourselves to the EB and IMR (with a fixed step size h), although the flow method can be extended to any existing implicit method. For any point of the flow we denote by $\mathbf{x}_j^i$ the approximation of $\mathbf{x}_j(t^i)$, at $t^i = i\,h$. Then, by applying EB, the solution at the next time-level follows from

$$\mathbf{x}_j^{i+1} = \mathbf{x}_j^i + h\,\mathbf{u}(\mathbf{x}_j^{i+1}). \tag{2.1}$$

If we apply IMR then we have

$$\mathbf{x}_j^{i+1} = \mathbf{x}_j^i + h\,\mathbf{u}\left(\frac{\mathbf{x}_j^{i+1} + \mathbf{x}_j^i}{2}\right), \tag{2.2}$$

which can be transformed into

$$\mathbf{x}_j^{i+\frac{1}{2}} = \mathbf{x}_j^i + \frac{h}{2}\,\mathbf{u}(\mathbf{x}_j^{i+\frac{1}{2}}), \tag{2.3}$$

$$\mathbf{x}_j^{i+1} = 2\,\mathbf{x}_j^{i+\frac{1}{2}} - \mathbf{x}_j^i. \tag{2.4}$$

Both (2.1) and (2.3) are systems of nonlinear equations which, in general, cannot be solved directly. One possible way to solve them is to employ some Newton type iterative method, but for that we need $\mathbf{u}(\mathbf{x})$ explicitly. The possible remedy is to find some approximation, say $\tilde{\mathbf{u}}(\mathbf{x})$, from the available information (in particular by interpolation), which can be used for iteration. Also the Newton iterative method requires the inverse of the Jacobian matrix. Hence it must be approximated as well. All this can be expensive computationally, especially if q is large. Also (theoretically more important) one

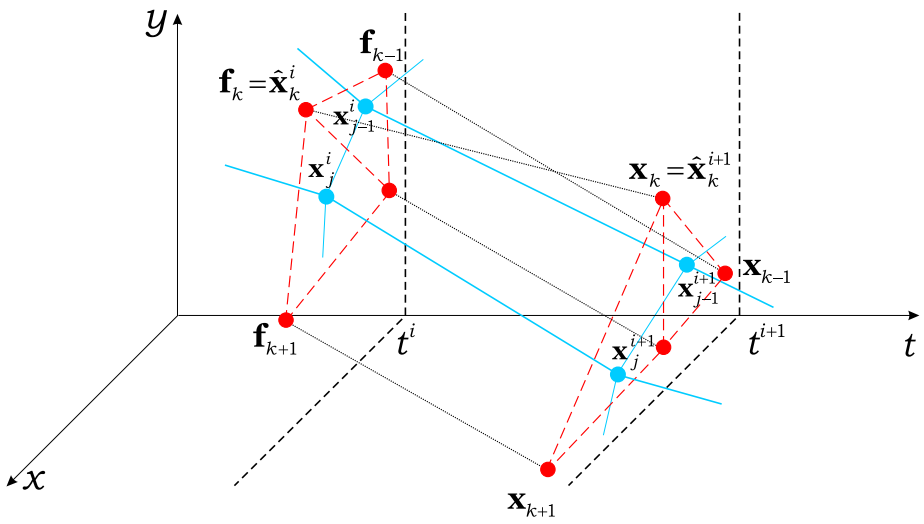


Figure 2.1: Flow method principle for 2-D (original) triangular grid.

would have to answer the obvious question how good these approximations should be to guarantee the convergence of the iterative method, of course, if it converges at all.

We now come to our method. Let us first address the flow method based on EB. At time level t^{i+1} we define

$$\hat{\mathbf{x}}_k^{i+1} := \mathbf{x}_k, \tag{2.5}$$

and since (1.2) is autonomous, we clearly have

$$\mathbf{u}(\hat{\mathbf{x}}_k^{i+1}) = \mathbf{u}(\mathbf{x}_k) = \mathbf{u}_k. \tag{2.6}$$

Taking $\hat{\mathbf{x}}_k^{i+1}$ now as the result of an EB step then this should correspond to a value $\hat{\mathbf{x}}_k^i$ defined by

$$\mathbf{f}_k = \hat{\mathbf{x}}_k^i := \hat{\mathbf{x}}_k^{i+1} - h\,\mathbf{u}(\hat{\mathbf{x}}_k^{i+1}) = \mathbf{x}_k - h\,\mathbf{u}_k. \tag{2.7}$$

In Figure 2.1 a 2-D example is shown where the points $\{\mathbf{x}_k\}_{k=1}^n$ are the vertices of a triangular grid. Clearly there is a functional dependence between points at two consecutive time-levels (which is, of course, analytically unknown). Nevertheless we can employ this fact for finding approximate values for all points in Γ and thus for the solution points. For the general point $\mathbf{x}^i \in \Gamma$ we can rewrite (2.7) as

$$\mathbf{x}^i = \mathbf{x}^{i+1} - h\,\mathbf{u}(\mathbf{x}^{i+1}) =: \mathbf{f}(\mathbf{x}^{i+1}). \tag{2.8}$$

If $\mathbf{f}(\mathbf{x})$ satisfies conditions of the inverse function theorem (see [1]), then there exists

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}^{-1}(\mathbf{x}), \tag{2.9}$$

and we may write

$$\mathbf{x}^{i+1} = \mathbf{g}(\mathbf{x}^i). \tag{2.10}$$

Since $\mathbf{g}$ is unknown in general, we can try to find an approximation, say $\mathbf{p}$, by requiring $\mathbf{p}(\hat{\mathbf{x}}_k^i) = \mathbf{g}(\hat{\mathbf{x}}_k^i)$. An obvious choice is to interpolate points $\{(\mathbf{f}_k, \mathbf{x}_k)\}_{k=1}^n$ in 2N dimensional space. Now the solution at the next time-level follows from

$$\mathbf{x}_j^{i+1} = \mathbf{p}(\mathbf{x}_j^i). \tag{2.11}$$

The choice of the interpolation method involved can be seen as arbitrary, but there are some preferences which can help answering the question which method should be used. Firstly, the interpolation should preferably be local to avoid big costs in computations. Secondly, the additional (interpolation) error should be commensurate with the local discretisation error to preserve the accuracy of EB. Finally, the interpolation method should be applicable for irregular grids, since the new grid is obtained from the original (possibly regular) grid by the nonlinear mapping $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$, defined by (2.8). This mapping of the grid is probably the most crucial part of the method and will be addressed separately.

If we use IMR, the method goes essentially similar. Indeed, it can be seen, from (2.3) and (2.4), that IMR is just an EB step on a half-interval followed by an (explicit) algebraic evaluation. This means that we can apply the flow method with a step-size $\frac{h}{2}$ and obtain the solution at the half-interval by

$$\mathbf{x}_j^{i+\frac{1}{2}} = \mathbf{p}(\mathbf{x}_j^i). \tag{2.12}$$

Now by using (2.12) in (2.4) we obtain the desired solution at the next time level.

The nonlinear system (2.8) is equivalent to a well-studied problem

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}. \tag{2.13}$$

The literature about this problem is rich (see e.g. [13]) and in the last decade a number of papers addressed the particular case arising from the use of implicit numerical methods for solving ODEs. Conditions for existence and uniqueness of the solution of (2.13) as well as a time step constraints for which these conditions are guaranteed are given in [27] and [25]. Hence we will assume that the solution exists and that it is unique. Our main interest here is to analyse the influence of the nonlinear mapping $\mathbf{f}$ (under the influence of $\mathbf{u}$) on our interpolation technique. Also we will assume that $\mathbf{f}$ is a diffeomorphism (see [1]).

Let us assume that Γ (i.e $\mathbf{I}(t^i)$ where all interpolation points are the flow points) is discretised in space, i.e. the grid which covers Γ is defined by points $\{\mathbf{x}_k\}_{k=1}^n$ and adequate elements: For example, one can think of a triangular grid in 2-D. For our algorithm, the grid needs to be mapped into a new grid defined by points $\{\mathbf{f}_k\}_{k=1}^n$ and elements of the same type. The diffeomorphism $\mathbf{f}$ can cause a change in the orientation of the grid elements, causing an overlapping of elements in the new grid; this makes interpolation difficult to handle or even highly ill-posed, which one should avoid of course. It is known (see [3]) that the orientation of certain manifold mapped by $\mathbf{f}$:

$\mathbb{R}^N \to \mathbb{R}^N$ is preserved if the Jacobian, say $|D\mathbf{f}|$, of $\mathbf{f}$ is positive on that manifold. This means that if

$$|D\mathbf{f}| > 0, \tag{2.14}$$

holds on a particular element in the original grid, the element in the new grid will have the same orientation. Of course, this should hold for all elements, i.e. on the entire Γ. Typically in stiff systems we have multiple time scales which are related to time-varying eigenvalues of the Jacobian matrix of $\mathbf{u}$, say $\lambda_l(t),\ l = 1, \ldots, N$ that are widely spread (but with a significant gap between them) in the left half of the complex plane. In other words all eigenvalues have negative real part. From (2.8) we see that the Jacobian matrix of $\mathbf{f}$ reads

$$D\mathbf{f} = \mathbf{I}_N - h\, D\mathbf{u}, \tag{2.15}$$

where $\mathbf{I}_N$ is the identity matrix of order N. If we now express the Jacobian $|D\mathbf{f}| := det(D\mathbf{f})$ via the aforementioned eigenvalues of $D\mathbf{u}$ we have

$$|D\mathbf{f}| = \prod_{l=1}^{N} [1 - h\,\lambda_l(t)]. \tag{2.16}$$

From the characteristics of the stiff problems mentioned above, i.e. $Re\{\lambda_l(t)\} < 0$ and (2.16), it is clear that (2.14) holds on the entire $\mathbb{R}^N$. The condition (2.14) we therefore define as a well-posedness of the method.

Throughout this section we have been concentrating on autonomous problems. We remark that the method principle also holds for the class of the non-autonomous problems given by

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}) + \mathbf{w}(t), \tag{2.17}$$

where $\mathbf{w}$ is an explicitly given time dependent function. The interpolation is then done only on the autonomous part of the velocity field and the solution at the next time-level reads

$$\mathbf{x}_j^{i+1} = \mathbf{p}(\mathbf{x}_j^i + h\,\mathbf{w}(t^{i+1})). \tag{2.18}$$

It can easily be shown (cf. [22]) that, for a case where the autonomous part is linear in $\mathbf{x}$ and by applying (2.18) with $\mathbf{p}$ as the linear interpolation function, the result is *identical* to one obtained by EB. For the general (non-autonomous) case the system

$$\dot{\mathbf{x}} = \mathbf{u}(t, \mathbf{x}), \tag{2.19}$$

can be transformed into an autonomous one, which will increase the system dimension by one, i.e. by adding an equation

$$\dot{t} = 1, \tag{2.20}$$

we have the autonomous problem.

3 Error Analysis

In [22] it was shown that for the univariate case the local error of the flow method consists of two components: the local discretisation error of EB and the interpolation error. Of course, the same also holds for the multivariate case. Assuming that $\mathbf{u}$ is Lipschitz continuous and smooth enough, the local error of a particular point in the flow can be expressed as

$$\delta(\mathbf{x}_j(t^{i+1}), h) = d(\mathbf{x}_j(t^{i+1}), h) + r(\mathbf{x}_j(t^i), h) \tag{3.1}$$

where d is the local discretisation error and r is the interpolation error. It is well known (see e.g. [12]) that EB has consistency order 1 and IMR order 2, i.e. the error d is $O(h)$ and $O(h^2)$ respectively. The goal is thus to have the interpolation error commensurate with the discretisation error. In the previous section we pointed out that the interpolation method can be of arbitrary type if it satisfies all of three preferences noted there. However, we will restrict ourselves to (piecewise) linear interpolation, i.e. the interpolation by linear polynomials to function values at $(N+1)$ points in $\mathbb{R}^N$, just for the ease of argument. Also, this interpolation is local and it can be applied on irregular grids, which means that the only requirement left is that of sufficient accuracy. The accuracy of the linear interpolation is a well-studied problem in theory (for the error bounds for various special cases see [2, 7, 20, 21]). We will use the result obtained in [26], where a sharp pointwise L_∞-error bound is obtained. But before applying this theory to our problem, we remark that our interpolation domain, say $\hat{\Gamma}$, is actually a range of the flow domain Γ. Therefore we denote all symbols related to $\hat{\Gamma}$ by providing them with a hat symbol above. Assume that $\hat{\Gamma}$ is covered by a set of (nondegenerate) simplices, say $\hat{\Gamma}_m$, $m = 1, \ldots, M$. The simplex $\hat{\Gamma}_m$ is defined by a set, say $\hat{S}_m$, of affinely independent $N+1$ points in $\mathbb{R}^N$. These points, which we denote by $\hat{\mathbf{x}}_{m1}, \hat{\mathbf{x}}_{m2}, \ldots, \hat{\mathbf{x}}_{m,N+1}$, are actually the vertices of the new (mapped) grid simplex. Now, $\hat{\Gamma}_m$ can be seen as a convex hull of $\hat{S}_m$, i.e.

$$\hat{\Gamma}_m := \operatorname{conv} \hat{S}_m, \tag{3.2}$$

with diameter

$$\hat{a} := \operatorname{diam} \hat{S}_m = \max_{\hat{\mathbf{x}}_{mr}, \hat{\mathbf{x}}_{ms} \in \hat{S}_m} \|\hat{\mathbf{x}}_{mr} - \hat{\mathbf{x}}_{ms}\| . \tag{3.3}$$

The error bound is given for a scalar function $g : \mathbb{R}^N \to \mathbb{R}$, while our problem concerns a vectorial mapping $\mathbf{g} : \mathbb{R}^N \to \mathbb{R}^N$. However, all elements of the vector $\mathbf{g}$, say g_l, $l = 1, \ldots, N$, are defined on the same simplex $\hat{\Gamma}_m$, which allows us to do an element-wise analysis first. For any g_l we can define a linear interpolation map p_l, $l = 1, \ldots, N$, defined on $\hat{\Gamma}_m$. One can show that (see [26])

$$|g_l(\hat{\mathbf{x}}) - p_l(\hat{\mathbf{x}})| \leq \frac{1}{2} \left(\hat{R}^2 - \|\hat{\mathbf{x}} - \hat{\mathbf{c}}\|^2 \right) \left\| D^2 g_l \right\|_{\infty, \hat{\Gamma}_m} , \tag{3.4}$$

holds, where $\hat{R}$ and $\hat{\mathbf{c}}$ are the radius and the center of the (unique) sphere containing $\hat{S}$. The norm $\|\cdot\|$ is the Euclidian norm and $\left\| D^2 g_l \right\|_{\infty, \hat{\Gamma}_m}$ represents the $\infty-$norm of the

second derivative $D^2 g_l$ on $\hat{\Gamma}_m$, defined as

$$\left\|D^2 g_l\right\|_{\infty,\hat{\Gamma}_m} := \sup_{\hat{x}\in\hat{\Gamma}_m} \sup_{\substack{\hat{y}\in\mathbb{R}^N \\ \|\hat{y}\|=1}} \left|D^2_{\hat{y}} g_l(\hat{x})\right|. \tag{3.5}$$

Here $D_{\hat{y}} g_l$ represents the derivative of g_l in the direction $\hat{y}$.

Since all g_l (and p_l) are defined on the same simplex we have, by taking the $\infty-$norm over $|g_l(\hat{x}) - p_l(\hat{x})|$ and $\left\|D^2 g_l\right\|_{\infty,\hat{\Gamma}_m}$, $l = 1,\ldots,N$

$$\begin{aligned} \|\mathbf{g}(\mathbf{x}) - \mathbf{p}(\mathbf{x})\|_{\infty,\hat{\Gamma}_m} := \max_l |g_l(\hat{x}) - p_l(\hat{x})| &\leq \frac{1}{2}\,(\hat{R}^2 - \|\hat{x} - \hat{c}\|^2)\, \max_l \left\|D^2 g_l\right\|_{\infty,\hat{\Gamma}_m} \\ &= \frac{1}{2}\,(\hat{R}^2 - \|\hat{x} - \hat{c}\|^2)\, \left\|D^2 \mathbf{g}\right\|_{\infty,\hat{\Gamma}_m} \end{aligned} \tag{3.6}$$

The expression $\hat{R}^2 - \|\hat{x} - \hat{c}\|^2$ in $\hat{\Gamma}_m$ has a maximum for the point $\hat{x}^* \in \hat{\Gamma}_m$, closest to $\hat{c}$. Defining

$$\hat{d} := \operatorname{dist}(\hat{c}, \hat{\Gamma}_m) = \min_{\hat{x}\in\hat{\Gamma}_m} \|\hat{x} - \hat{c}\| = \|\hat{x}^* - \hat{c}\|, \tag{3.7}$$

we then find

$$r := \|\mathbf{g} - \mathbf{p}\|_{L_\infty(\hat{\Gamma}_m)} \leq \frac{1}{2}\,(\hat{R}^2 - \hat{d}^2)\, \left\|D^2 \mathbf{g}\right\|_{\infty,\hat{\Gamma}_m}. \tag{3.8}$$

Now, we apply (3.8) to our analysis. This error bound is given, of course, for direct interpolation. In our case it does not give explicit information since it still requires knowledge of a second derivative of the unknown inverse function $\mathbf{g}$ and depends on the geometry of the new grid. However, from (2.9), we find

$$\begin{aligned} D\mathbf{g} &= [D\mathbf{f}]^{-1}, \\ D^2\mathbf{g} &= -[D\mathbf{f}]^{-1}\, D^2\mathbf{f}\, [D\mathbf{f}]^{-2}, \end{aligned} \tag{3.9}$$

and, since $D\mathbf{f} = \mathbf{I}_N - h\, D\mathbf{u}$ and $D^2\mathbf{f} = -h\, D^2\mathbf{u}$, we have

$$D^2\mathbf{g} = h\,[\mathbf{I}_N - h\, D\mathbf{u}]^{-1}\, D^2\mathbf{u}\, [\mathbf{I}_N - h\, D\mathbf{u}]^{-2}. \tag{3.10}$$

Substituting (3.10) into (3.8), we can eliminate $\mathbf{g}$, i.e the interpolation error bound reads

$$r(\mathbf{x}^i_j, h) \leq \frac{h}{2}(\hat{R}^2 - \hat{d}^2)\, \left\|[\mathbf{I}_N - h\, D\mathbf{u}]^{-1}\, D^2\mathbf{u}\, [\mathbf{I}_N - h\, D\mathbf{u}]^{-2}\right\|_{\infty,\Gamma_m}, \tag{3.11}$$

where Γ_m is a simplex of the original grid, i.e the original of $\hat{\Gamma}_m$. Clearly, if $\|D\mathbf{u}\|_{\infty,\Gamma_m}$ is large, which typically occurs in stiff problems (our problems of interest), then $\left\|D^2\mathbf{g}\right\|_{\infty,\hat{\Gamma}_m}$ is not necessarily large due to the inversion of the matrix $[\mathbf{I}_N - h\, D\mathbf{u}]$.

The expression $\hat{R}^2 - \hat{d}^2$, present in the error bound, is strongly depending on the diameter of the simplex $\hat{\Gamma}_m$ and the geometry of the new grid. Hence, we will first relate $\hat{a}$ to the diameter of the original grid, say a. By applying (2.7) in (3.3) we have

$$\hat{a} = \|\hat{\mathbf{x}}_{mq} - \hat{\mathbf{x}}_{ms}\| = \|\mathbf{x}_{mq} - \mathbf{x}_{ms} - h\,(\mathbf{u}_{mq} - \mathbf{u}_{ms})\|\,. \tag{3.12}$$

From the mean value theorem (see [1]), we have

$$\mathbf{u}(\mathbf{x}_{mq}) - \mathbf{u}(\mathbf{x}_{ms}) = D\mathbf{u}(\tilde{\mathbf{x}})(\mathbf{x}_{mq} - \mathbf{x}_{ms}), \quad \tilde{\mathbf{x}} \in \Gamma_m, \tag{3.13}$$

Substituting (3.13) into (3.12) we obtain an estimate of $\hat{a}$, i.e.

$$\hat{a} = \|(\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}}))\,(\mathbf{x}_{mq} - \mathbf{x}_{ms})\| \le \|\mathbf{x}_{mq} - \mathbf{x}_{ms}\|\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\| \le a\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\|\,. \tag{3.14}$$

Note that $\|\mathbf{x}_{mq} - \mathbf{x}_{ms}\| \le a$, since the diameter of the original grid is not necessarily attached to the corresponding points with the same indexes of the new grid.

One cannot give a simple relation between $\hat{R} - \hat{d}$ and $\hat{a}$ (i.e. a) in $\mathbb{R}^N$ in general. Hence, we will access some important special cases by the following examples.

Example 1. For $\hat{\mathbf{c}} \in \hat{\Gamma}_m$ it can be shown (cf. [26]) that

$$\sup\left\{\frac{\hat{R}^2}{\hat{a}^2} : \hat{\mathbf{c}} \in \hat{\Gamma}_m\right\} = \frac{N}{2(N+1)}, \tag{3.15}$$

which allows us to estimate

$$\hat{R}^2 \le \frac{N}{2(N+1)}\,a^2\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\|^2. \tag{3.16}$$

By using (3.16) and the fact that $\hat{d} = 0$, i.e. $\hat{R}^2 - \hat{d}^2 = \hat{R}^2$, the interpolation error bound for a particular point in the flow reads

$$r\left(\mathbf{x}_j^i, h\right) \le \frac{h}{4}\,\frac{N}{N+1}\,a^2\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\|^2\,\left\|[\mathbf{I}_N - h\,D\mathbf{u}]^{-1}\,D^2\mathbf{u}\,[\mathbf{I}_N - h\,D\mathbf{u}]^{-2}\right\|_{\infty,\Gamma_m}\,.\ \square \tag{3.17}$$

Example 2. For the bivariate case ($N = 2$) we can also give the error bound if $\hat{\mathbf{c}} \notin \hat{\Gamma}_m$. Here we have that $\hat{\mathbf{x}}^*$is exactly in the middle of the facet of length $\hat{a}$, see Figure 3.1. Since the line segment from $\hat{\mathbf{c}}$ to $\hat{\mathbf{x}}^*$ is orthogonal to the facet, Pythagoras' theorem gives

$$\hat{R}^2 - \hat{d}^2 = \frac{1}{4}\,\hat{a}^2. \tag{3.18}$$

This leads to the error bound

$$r\left(\mathbf{x}_j^i, h\right) \le \frac{h}{8}\,a^2\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\|^2\,\left\|[\mathbf{I}_N - h\,D\mathbf{u}]^{-1}\,D^2\mathbf{u}\,[\mathbf{I}_N - h\,D\mathbf{u}]^{-2}\right\|_{\infty,\Gamma_m}\,.\ \square \tag{3.19}$$

For $\hat{\mathbf{c}} \notin \hat{\Gamma}_m$ and $N > 2$ the relation (3.18) does not hold in general. Here we have an interval of possible values for $\hat{R} - \hat{d}$ depending on the geometry of $\hat{S}$. We analyse this for the case of interest $N = 3$ by the following example.

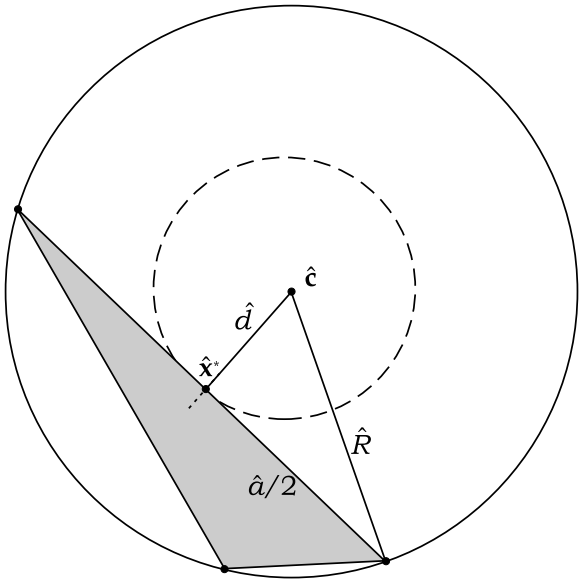


Figure 3.1: The circumscribed sphere of an obtuse triangle

Example 3. For $N = 3$ and $\hat{\mathbf{c}} \notin \hat{\Gamma}_m$, the point $\hat{\mathbf{x}}^* \in \hat{\Gamma}_m$ lies on a facet closest to $\hat{\mathbf{c}}$. Since the facet is a triangle, it can be acute or obtuse angled. This means that $\hat{R} - \hat{d}$ may vary between $\frac{1}{3}\,\hat{a}^2$ (in case of equilateral triangle) and $\frac{1}{4}\,\hat{a}^2$ (obtuse angled triangle). Hence, by taking

$$\hat{R}^2 - \hat{d}^2 \leq \frac{1}{3}\,\hat{a}^2, \tag{3.20}$$

we have the error bound

$$r\left(\mathbf{x}_j^i, h\right) \leq \frac{h}{6}\,a^2\,\|\mathbf{I}_N - h\,D\mathbf{u}(\tilde{\mathbf{x}})\|^2 \left\|[\mathbf{I}_N - h\,D\mathbf{u}]^{-1}\,D^2\mathbf{u}\,[\mathbf{I}_N - h\,D\mathbf{u}]^{-2}\right\|_{\infty,\Gamma_m}. \quad \square \tag{3.21}$$

From (3.17), (3.19) and (3.21) it follows that the interpolation error is $O(a^2)$, meaning that the local error of the flow method is order $O(h^s) + O(a^2)$, where $s = 1, 2$ for EB and IMR respectively. Clearly this means that the original grid simplex size (defined by a) should be as such that both local error components are of the same order. If a is larger, then the interpolation error can become a dominant source of error, i.e. the accuracy of the original implicit method may be lost. On the other hand by doing the interpolation more accurately than needed, the error will not decrease below the error of the original implicit method.

4 Stability

Numerical stability properties of the flow method should be similar to those of the implicit method involved. In particular we will analyse EB for stiff problems, by studying

first variations. Let $\mathbf{z}_j^i|_{j\ fixed}$ denote a small perturbation of the solution $\mathbf{x}_j^i|_{j\ fixed}$ of (2.11), such that $\mathbf{x}_j^i + \mathbf{z}_j^i|_{j\ fixed}$ also satisfies (2.11) to first order. Now we have

$$\mathbf{x}_j^{i+1} + \mathbf{z}_j^{i+1} = \mathbf{p}(\mathbf{x}_j^i + \mathbf{z}_j^i) \doteq \mathbf{p}(\mathbf{x}_j^i) + D\mathbf{p}(\mathbf{x}_j^i)\,\mathbf{z}_j^i. \tag{4.1}$$

By neglecting higher order terms we have

$$\mathbf{z}_j^{i+1} = D\mathbf{p}(\mathbf{x}_j^i)\,\mathbf{z}_j^i. \tag{4.2}$$

For stability in a nonlinear situation it is sufficient to prove that the contractivity condition of the discrete equation (4.2) is satisfied (see e.g. [12]), i.e.

$$\|D\mathbf{p}(\mathbf{x}_j^i)\| < 1. \tag{4.3}$$

Before continuing we would like to relate (4.3) to the equivalent condition for EB, which reads

$$\|[D\mathbf{f}]^{-1}\| = \|[\mathbf{I}_N - h\,D\mathbf{u}]^{-1}\| < 1. \tag{4.4}$$

Since $D\mathbf{p} \doteq D\mathbf{g} = [D\mathbf{f}]^{-1}$ one should expect that (4.3) and (4.4) are equivalent in a way. Indeed, this can be shown as follows. Again for the ease of argument, we will stick to the case where the interpolation (vectorial) polynomial is the piecewise linear function w.r.t. $\mathbf{x}$, i.e. $\mathbf{p}(\mathbf{x})$ on a certain simplex reads

$$\mathbf{p}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}. \tag{4.5}$$

Clearly $\mathbf{A} = D\mathbf{p} \doteq D\mathbf{g}$, which means that the matrix $\mathbf{A}^{-1}$ should be an approximation of $D\mathbf{f}$. To find $\mathbf{A}$ let us choose the simplex Γ_m from the original grid with vertices $\mathbf{x}_k = [x_{1k}, x_{2k}, \ldots, x_{N,k}]^T,\ k = 1, \ldots, N+1$, and corresponding $\hat{\Gamma}_m$ from the new grid with vertices $\mathbf{f}_k = [f_{1k}, f_{2k}, \ldots, f_{N,k}]^T,\ k = 1, \ldots, N+1$, defined by (2.7). By applying this to (4.5), we have

$$\mathbf{A} = \mathbf{B}\,\mathbf{C}^{-1}, \tag{4.6}$$

where

$$\mathbf{B} = \begin{bmatrix} x_{12} - x_{11} & x_{13} - x_{11} & \ldots & x_{1,N+1} - x_{11} \\ x_{22} - x_{21} & x_{23} - x_{21} & & x_{2,N+1} - x_{21} \\ \vdots & & & \\ x_{N,2} - x_{N,1} & x_{N,3} - x_{N,1} & & x_{N,N+1} - x_{N,1} \end{bmatrix}, \tag{4.7}$$

$$\mathbf{C} = \begin{bmatrix} f_{12} - f_{11} & f_{13} - f_{11} & \ldots & f_{1,N+1} - f_{11} \\ f_{22} - f_{21} & f_{23} - f_{21} & & f_{2,N+1} - f_{21} \\ \vdots & & & \\ f_{N,2} - f_{N,1} & f_{N,3} - f_{N,1} & & f_{N,N+1} - f_{N,1} \end{bmatrix}. \tag{4.8}$$

For ease of argument we will choose vertices of Γ_m with coordinates

$$x_{lk} = x_{l1} + \delta_{l+1,k}\Delta x_l, \quad l = 1, \ldots, N,\ k = 1, \ldots, N+1, \tag{4.9}$$

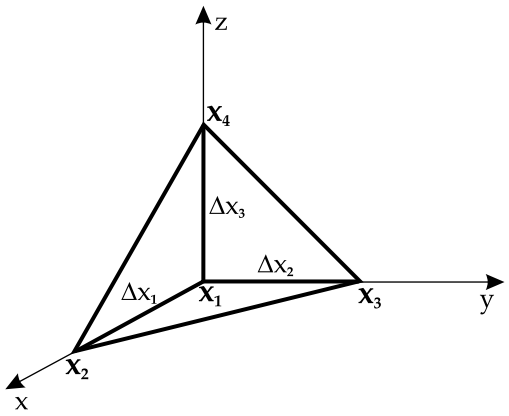


Figure 4.1: 3-D example of the original grid simplex.

where $\delta_{l+1,k}$ is the Kronecker delta. This gives the (original) simplex where all vertices are at the axes of the orthogonal coordinate system with the origin at $\mathbf{x}_1$ and Δx_l, $l = 1, \ldots, N$ are the lengths of edges connecting vertices $\mathbf{x}_2, \ldots, \mathbf{x}_{N+1}$ with the origin $\mathbf{x}_1$. As a 3-D illustration one should think of a tetrahedron which has three edges parallel to x, y and z axes respectively, see Figure 4.1. By substituting (4.9) into (4.7) we have

$$\mathbf{B} = \mathrm{diag}[\Delta x_1, \Delta x_2, \ldots, \Delta x_N]. \tag{4.10}$$

On the other hand, for elements in $\mathbf{C}$, we have

$$f_{l,k+1} - f_{l1} = \Delta x_k \left(\delta_{l,k} - h \frac{\partial u_l(x_{11}, x_{21}, \ldots, c_k, \ldots, x_{N,1})}{\partial x_k} \right), \; l, k = 1, \ldots, N, \tag{4.11}$$

where $x_{k1} \leq c_k \leq x_{k1} + \Delta x_k$, i.e. $c_k \in \Gamma_m$.

We can now use (4.10) and (4.11) to obtain $\mathbf{A}^{-1}$. From (4.6) we have

$$\mathbf{A}^{-1} = \mathbf{C}\,\mathbf{B}^{-1} = \left\{ \delta_{l,k} - h \frac{\partial u_l(x_{11}, x_{21}, \ldots, c_k, \ldots, x_{N,1})}{\partial x_l} \right\}_{k,j=1}^{N}. \tag{4.12}$$

Clearly we have that $\mathbf{A}^{-1}$ is close to $D\mathbf{f}(\mathbf{x}_j^{i+1}) = \mathbf{I}_N - D\mathbf{u}(\mathbf{x}_j^{i+1})$ since $\mathbf{x}_j^{i+1} \in \Gamma_m$. This means that the contractivity condition $\|\mathbf{A}\| < 1$ of the flow method is equivalent to one of EB, given by (4.4). Of course, this holds only if the well-posedness condition (2.14) is satisfied; otherwise the interpolation can become ill-posed. In particular obtaining $\mathbf{A}$ can represent a problem if $\mathbf{C}$ is ill-conditioned. However, as mentioned before, that is typically not the case for stiff problems of interest.

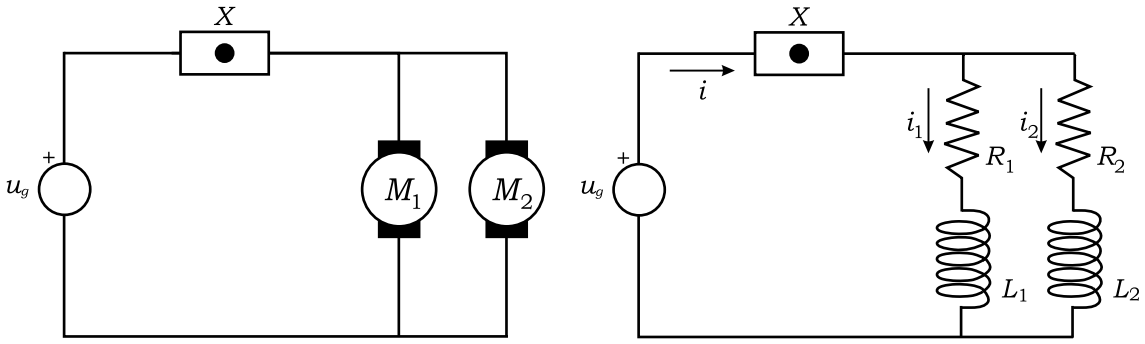
5 Practical Aspects

In the previous sections we introduced our method and gave an error and stability analysis. In this section we will apply the flow method to two examples, where typically the

velocity field is not known explicitly. The first one concerns the situation where the velocity is obtained experimentally. In particular we will solve a stiff problem coming from electrical networks with nonlinear elements. Hence we apply the flow method based on EB, showing that it has the desirable accuracy and stability properties. The second example concerns a problem where the velocity is a numerical solution of the divergence free velocity field, coming from boundary problem, that is part of a PDE. The resulting ODE can be related to a Hamiltonian form, which means that the volume, defined by the flow, should be preserved during the time integration. A typical symplectic numerical method, which has this property, is IMR (see [19]). Hence we will apply the flow method based on IMR and show that a sufficient accuracy can be achieved even for relatively long time integration intervals.

5.1 An Electrical Network

Consider an electrical network with two DC motors, power-supplied by the same source and current protected by a nonlinear resistor, see Figure 5.1a. The most severe situation is when both motor shafts (i.e. rotors) are blocked. Then currents are largest and the equivalent electrical circuit is shown in Figure 5.1b. Here both motors are modeled as serial connections of a resistor (R_k, $k = 1, 2$) and an inductor (L_k) and since both R_k and L_k are relatively small all currents tend to increase under the influence of the relatively large input voltage.



(a) Principle scheme

(b) Electrical scheme with rotors blocked

Figure 5.1: DC motors supplied from the same power source

The task of the nonlinear resistor is to prevent currents not to increase (motors current protection) under the high voltage at the input, here $u_g = u_g(t)$. According to this the transfer function (U-I characteristic) of the nonlinear resistor X is very steep, making the problem stiff. Moreover, this characteristic is not known in closed form and usually given by some tabular values. The mathematical model of the system follows from the

second Kirchhoff's law, i.e

$$\begin{aligned} L_1 \frac{di_1}{dt} &= -u(i_1 + i_2) - R_1 i_1 + u_g(t), \\ L_2 \frac{di_2}{dt} &= -u(i_1 + i_2) - R_2 i_2 + u_g(t), \end{aligned} \tag{5.1}$$

where $i_k = i_k(t)$ are the currents in the motors loops and $u(i)$ is the voltage on the nonlinear resistor. The system (5.1) can be rewritten as

$$\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x}) + \mathbf{w}(t), \tag{5.2}$$

where

$$\begin{aligned} \mathbf{x} = [\ i_1,\ \ i_2\ ]^T, \quad \mathbf{w}(t) = u_g(t)[\ \tfrac{1}{L_1},\ \ \tfrac{1}{L_2}\ ]^T, \\ \mathbf{v}(\mathbf{x}) = [\ -\tfrac{1}{L_1}\left(u(x_1 + x_2) + R_1 x_1\right),\ \ -\tfrac{1}{L_2}\left(u(x_1 + x_2) + R_2 x_2\right)\ ]^T. \end{aligned} \tag{5.3}$$

Here we take $u_g(t) = V \cos \omega t$ and as typical parameters values $R_1 = 2\Omega$, $R_2 = 1\Omega$, $L_1 = 10^{-2}$H, $L_2 = 10^{-4}$H, $V = 220$V and $\omega = 1\,rad/s$. The U-I transfer function $u(x)$ is obtained experimentally (at n points $\{y_k\}_{k=1}^n$), see Table 5.1 and Figure 5.2.

y	-10.0	-9.75	...	-0.25	0.0	0.25	...	10.0
$u(y)$	-1.0e+7	-8.376e+6	...	-6.1e-5	0.0	6.1e-5	...	1.0e+7

Table 5.1: Discrete U-I transfer function of the nonlinear transistor

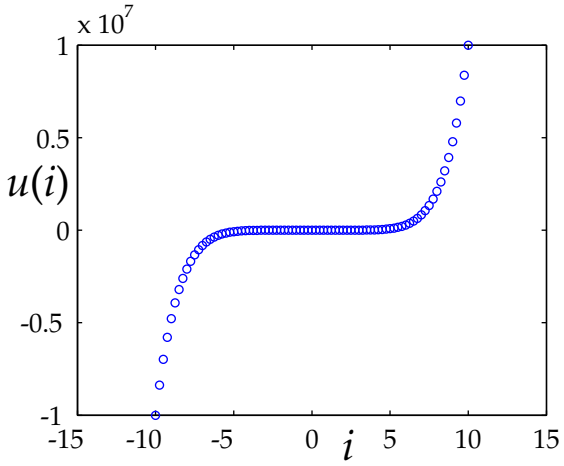


Figure 5.2: Discrete U-I transfer function of the nonlinear transistor

Of course, this set can be used both partially and totally. Hence we will assume that $n \leq 81$. Since we know u only for the given set $\{y_k\}_{k=1}^n$ we can create a 2-D triangular

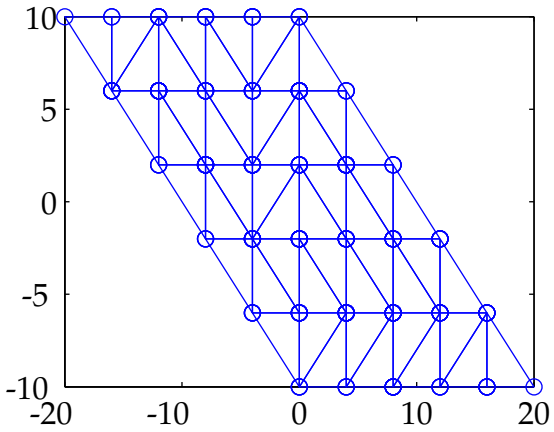


Figure 5.3: Triangular 2-D grid

grid with points $\mathbf{x}_{k,l} = (x_{1,k,l}, x_{2,k,l})$, $l = 1, \ldots, M$, where the coordinates of all grid points satisfy $y_k = x_{1,k,l} + x_{2,k,l}$. Of course, the number of points M is arbitrary. But since we would like to keep our original grid as good as possible, we keep the symmetry of the grid, meaning that the distances between neighbouring points in both directions, say Δx_1 and Δx_2, are equal to Δx, which is determined by the experimentally obtained data. Hence we take $M = n$. Now we have a set of n^2 points which can be triangularised, obtaining 2-D grid (see Figure 5.3 where $n = 6$), for which $\mathbf{v}(\mathbf{x})$ is known in all vertices.

To show the influence of the interpolation error we do the following numerical experiment. For fixed $h = 0.01$ and the flow points initial values $\mathbf{I}(0) = \left\{ [1.0, 0.0]^T, [0.5, 0.5]^T \right\}$, we compute the solutions up to the final time of computation $T_f = 3.5$. The computation is performed by using different numbers of the given tabular points, i.e. $n = 11, 21, 41, 81$, which correspond to the spatial step sizes $\Delta x = 2.0, 1.0, 0.5, 0.25$ respectively. Of course for a triangularisation as shown in Figure 5.3, the diameter of all simplices reads $a = \sqrt{2}\Delta x$. To assess the accuracy of the flow method we compare results with the EB solutions with a much smaller time step $h^* = 0.0001$. By taking the $\infty-$norm of the global error at T_f over all flow points (and their both coordinates) we obtain the dependance between the error and the diameter of the original grid simplex. According to (3.19) this dependance should be quadratic (up to the constant), which can be seen in Figure 5.4. Here the solid line represents the error of the flow method and the dashed line a quadratic function $q(a) = Ca^2$. Of course, such a behaviour of the global error is due to the fact that the time step is much smaller than the simplex diameter, making the interpolation error the dominant error source.

As the final remark we note that the flow method is numerically stable, i.e. there are no time step constraints even for the highly stiff problems as this one.

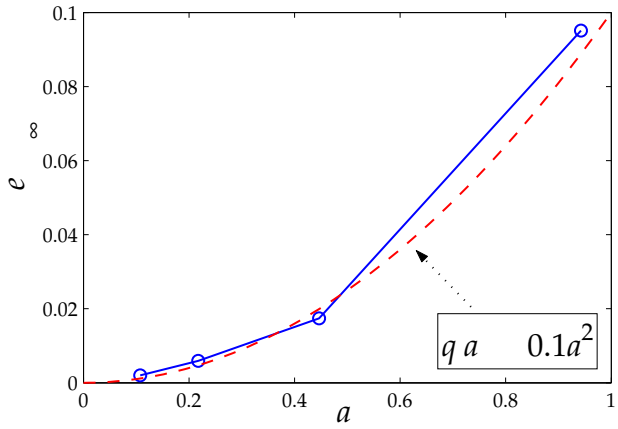


Figure 5.4: Global error of the flow method as a function of the original grid diameter

5.2 A Boundary Problem

Consider the following problem

$$\nabla^2 \cdot \mathbf{u} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{5.4}$$

defined on some domain Γ with the boundary $\partial\Gamma$. Here $\mathbf{x}$ represents the point in Γ with the velocity $\mathbf{u} = \mathbf{u}(\mathbf{x})$. By describing Dirichlet boundary conditions (BC) for $\mathbf{u}$, we have a boundary value problem. To obtain the exact solution of such problem is in general impossible. However, there is a variety of different numerical schemes, such as e.g. finite elements or finite differences which give an approximation of the solution (up to a certain accuracy) on a discrete (finite dimensional) subdomain, say $\Gamma^* \subset \Gamma$. Actually Γ^* represents the set of nodes of the grid which covers Γ. To solve the flow problem (1.2) defined by such velocity and the initial condition $\mathbf{I}(0) \subseteq \Gamma$, one needs to solve an autonomous ODE with discretely given $\mathbf{u}$.

Let us consider the problem (5.4), where $\mathbf{x} = [x, y]^T$ is the point in Cartesian coordinates, $\mathbf{u} = [u_x(x, y), u_y(x, y)]^T$ and

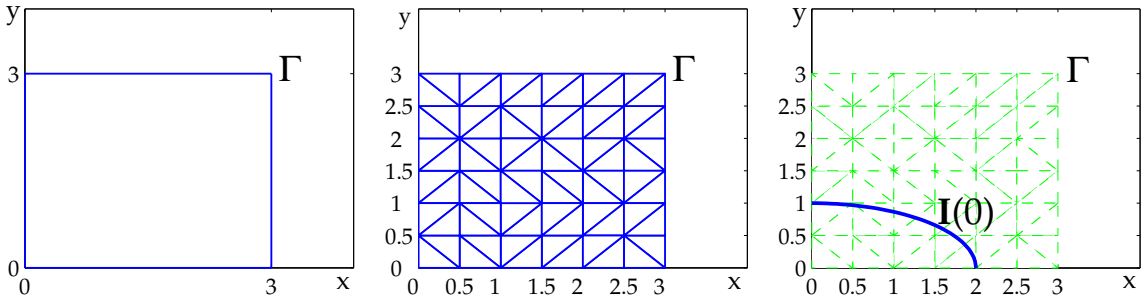
$$\mathbf{F}(x, y) = -\cos y - x \sin y. \tag{5.5}$$

The domain is a rectangle (see Figure 5.5a) defined by

$$\Gamma := \{(x, y) \,|\, 0 \le x \le 3,\ 0 \le y \le 3\}. \tag{5.6}$$

Let the boundary conditions be given by

$$\mathbf{u}(\mathbf{x})\,|_{\partial\Gamma} = \begin{cases} [0,\ 0]^T, & x = 0, \\ \left[-\frac{1}{2}x^2,\ 0\right]^T, & y = 0, \\ \left[-\frac{9}{2}\cos y,\ 3\sin y\right]^T, & x = 3, \\ \left[-\frac{\cos 3}{2}x^2,\ x \sin 3\right]^T, & y = 3. \end{cases} \tag{5.7}$$



(a) The domain Γ

(b) The triangular grid

(c) The flow initial condition

Figure 5.5: The boundary problem domain

To obtain the numerical solution of this problem one needs to discretise (5.4). To avoid a discussion on error contamination due to this discretisation we will use the exact solution of the boundary value problem, which we happen to know in this case. Indeed, we find a solution of (5.4), (5.5), (5.6) and (5.7)

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}x^2 \cos y \\ x \sin y \end{bmatrix}. \tag{5.8}$$

We will assume this to be known at a set of grid points $\{\mathbf{x}_k\}_{k=1}^n \in \Gamma^*$ only. Now, let us define the flow problem

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{5.9}$$

with initial condition $\mathbf{x}(0) \in \mathbf{I}(0)$, where $\mathbf{I}(0)$ is a quarter of an ellipse defined by

$$\mathbf{I}(0) = \left\{ (x, y) \,\middle|\, \frac{x^2}{4} + y^2 \leq 1,\ x, y \geq 0 \right\}. \tag{5.10}$$

Clearly $\mathbf{I}(0) \subset \Gamma$ as shown in Figure 5.5c and it is discretised (as pointed out in Section 2) by a set of points $\{\mathbf{x}_j\}_{j=1}^q$. Assuming well-posedness these points can be placed at the boundary of the flow only. Now we apply the flow method, defined by (2.4) and (2.12), and compute the flow evolution in time, see Figure 5.6. Here we take $h = 0.01$, $q = 10$ and $T_f = 2.0$. To compare the results we also compute the solution of the "standard" IMR method with the same time step size. By taking the $\infty-$norm of the vector of differences between results for both coordinates we have the measure of the additional error (coming from interpolation) introduced by the flow method. We perform the numerical experiment by taking $n = 6 \times 6$, 11×11, 21×21, 41×41, 81×81, for which we have the corresponding spatial step (for both coordinates) $\Delta x = 0.6, 0.3, 0.15, 0.075, 0.0375$

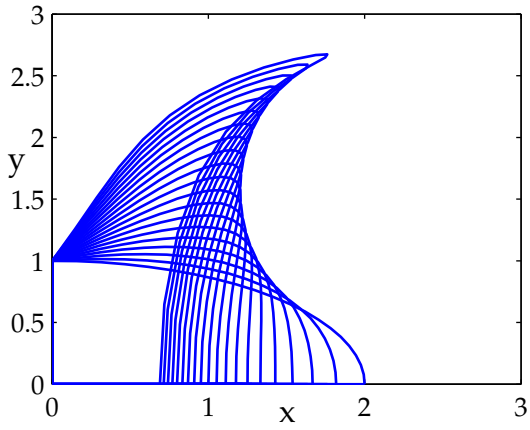


Figure 5.6: Quarter of ellipse time evolution

and $a = \sqrt{2}\Delta x$. Now we can observe how the interpolation error behaves in time, depending on the size of the spatial step used, see Figure 5.7a. Here, the ticker the line the smaller the grid step size. Note that even for a relatively large spatial step size ($\Delta x \leq 0.3$), the order of the error is smaller then $O(h^2)$, which is the order of IMR, even for the relatively long time scale introduced here. This means that we can compute the flow evolution relatively cheaply (without using extreme number of grid points) and yet keep the interpolation error negligible comparing to the local discretisation error for a large number of time steps. The interpolation error is again quadratically dependent on the diameter of the simplex, which can be shown by computing the error at the end of the computational time for different values of a. Again we obtain the relation, which is close to quadratic function $q(a) = Ca^2$, see Figure 5.7b (where $C = 0.00075$). For the volume preservation results one should see [11], where this example was introduced.

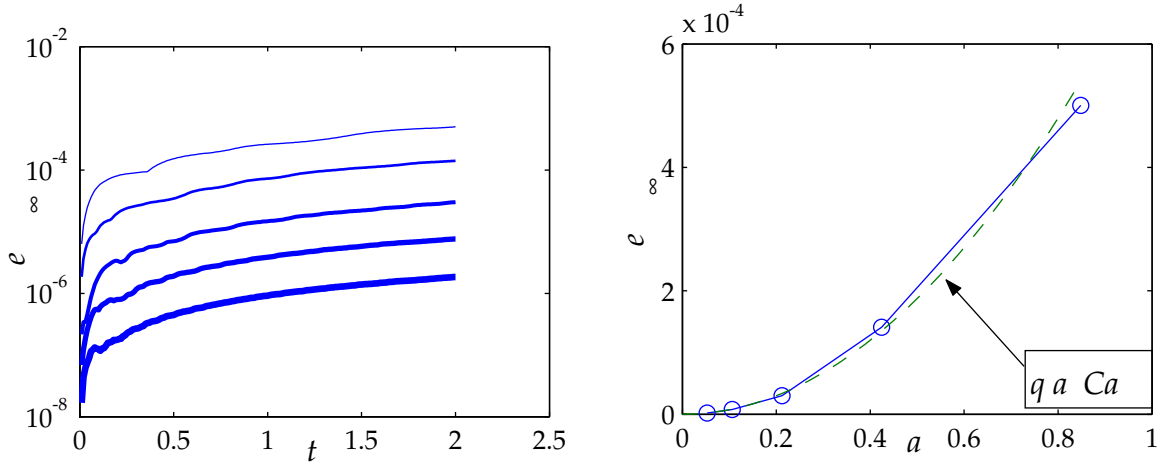
Acknowledgment: The authors would like to thank H. G. ter Morsche for his valuable advice on the interpolation error subject.

References

[1] T. Aubin. *A Course in Differential Geometry*. American Mathematical Society, 2001.

[2] O. Axelsson and V. A. Barker. *Finite Element Solution of Boundary Value Problems*. Academic Press, Inc., 1984.

[3] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, Inc., 1975.



(a) Error in time (ticker line - smaller Δx)

(b) Error-diameter dependence

Figure 5.7: Error in time and as a function of the diameter

[4] G.D. Byrne and A.C. Hindmarsh. Stiff ODE solvers: A review of current and coming attractions. *Journal of Computational Physics*, 70:1–62, 1987.

[5] P.R. Gray, P.J. Hurst, S.H. Lewis, and R.G. Meyer. *Analysis and design of analog integrated circuits*. John Wiley & Sons, 4th edition, 2001.

[6] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, 1997.

[7] D. C. Handscomb. Errors of linear interpolation on a triangle. Technical Report Research Report NA-95/09, Oxford University, 1995.

[8] A.C. Hindmarsh and L.R. Petzold. Algorithms and software for Ordinary Differential Equations and Differential-Algebraic Equations, Part I: Euler methods and error estimation. *Computers in Physics*, 9:34–41, 1995.

[9] K. Laevsky and R.M.M. Mattheij. Determining the velocity as a kinematic boundary condition in a glass pressing problem. Technical Report RANA 01-09, Eindhoven University Of Technology, 2001.

[10] D. Lanser, J.G Blom, and J.G. Verwer. Time integration of the shallow water equations in spherical geometry. *Journal of Computational Physics*, 171:373–393, 2001.

[11] R.M.M. Mattheij and K. Laevsky. Numerical volume preservation of a divergence free fluid under symmetry. Technical Report RANA 01-11, Eindhoven University Of Technology, 2001.

[12] R.M.M. Mattheij and J. Molenaar. *Ordinary differential equations in theory and practice*. John Wiley & Sons, 1996.

[13] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, Inc., 1970.

[14] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[15] D. Ramsden and G. Holloway. Evolution of vesicles subject to adhesion. *Journal of Computational Physics*, 95:101–116, 1991.

[16] S.W. Rienstra and T.D. Chandra. Analytical approximations to the viscous glass flow problem in the mould-plunger pressing process, including an investigation of boundary conditions. *Journal of Engineering Mathematics*, 39:241–259, 2001.

[17] R. Rosso, A.M. Sonnet, and E.G. Virga. Evolution of vesicles subject to adhesion. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.*, 456:1523–1545, 2000.

[18] A. Sandu, F.A. Potra, V. Damian-Iordache, and G.R. Carmichael. Efficient implementation of fully implicit methods for atmospheric chemistry. *Journal of Computational Physics*, 129:101–110, 1996.

[19] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems*. Chapman & Hall, 1994.

[20] Yu. N. Subbotin. Dependence of estimates of a multidimensional piecewise polynomial approximation on the geometric characteristics of the triangulation. *Proc. Steklov Inst. Math.*, 189:135–159, 1990.

[21] Yu. N. Subbotin. Error of the approximation by interpolation polynomials of small degrees on n-simplices. *Math. Notes*, 48:1030–1037, 1990.

[22] B. Tasic and R.M.M.Mattheij. An explicit method for solving flows of ode. *To appear in Applied Mathematics and Computation*.

[23] R.P. Tewarson, H. Wang, J.L. Stephenson, and J.F. Jen. Efficient solution of differential equations for kidneyconcentrating mechanism analyses. *Appl. Math. Lett.*, 4:69–72, 1991.

[24] G.A.L. van de Vorst. Numerical simulation of axisymmetric viscous sintering. *Engineering Analysis with Boundary Elements*, 14:193–207, 1995.

[25] J. L. M. van Dorsselaer and M. N. Spijker. The error committed by stopping the Newton iteration in the numerical solution of stiff initial value problem. *IMA Journal of Numerical Analysis*, 14:183–209, 1994.

[26] S. Waldron. The error in linear interpolation at the vertices of a simplex. *SIAM Journal on Numerical Analysis*, 35:1191–1200, 1998.

[27] J. Williams. Existence and uniqueness of solutions of the algebraic equations in the bdf methods. Technical Report Numerical Analysis Report No. 272, University Of Manchester, 1995.