The Meshless Kernel–Based Method of Lines for Solving Nonlinear Evolution Equations

Robert Schaback

June 2, 2008

Abstract

If kernel–based trial spaces are used, it is well–known since at least 1990 that the Method of Lines can be successfully applied to solve time–dependent partial differential equations, but a thorough mathematical analysis is still missing. For well–posed nonlinear evolution equations allowing arbitrary linear spatial differential operators and sufficiently smooth solutions, this paper proves that the Method of Lines leads to solvable ODE systems and small errors, provided that the spatial discretizations are fine enough and if sufficiently smooth kernels are used to generate the trial spaces. An example for the Burgers equation illustrates the theory.

1 Introduction

Following the early survey [1] of Belytschko et.al., meshless methods are characterized by representing trial functions *"entirely in terms of nodes"*. If fixed nodes $\mathbf{x}_j$ are taken in a solution domain $\Omega \subset \mathbb{R}^d$, then trial functions $w(\mathbf{x}, t)$ for a time–dependent partial differential equation should be written in the form

$$w(\mathbf{x}, t) = \sum_j v_j(\mathbf{x}) w(\mathbf{x}_j, t) \tag{1}$$

in the simplest case where only function values at nodes are relevant, and where the purely spatial "*shape*" functions v_j satisfy the standard Lagrange conditions

$$v_j(\mathbf{x}_k) = \delta_{jk}. \tag{2}$$

Among other things, this has the advantage of allowing an easy evaluation of linear spatial differential operators L on a trial function w via

$$(Lw)(\mathbf{x},t)=\sum_j w(\mathbf{x}_j,t)(Lv_j)(\mathbf{x})$$

provided that the functions v_j are sufficiently smooth and explicitly available. This is the case if the trial functions are generated from spaces spanned by translates of smooth kernels like the Gaussian. Consequently, there is no need for any special discretization of purely spatial differential operators. Representations like (1) and their error behavior including differentiation have a fully developed theory. We suggest the book [9] of H. Wendland for filling in the details.

When applied to solving nonlinear evolution equations like the Burgers equation

$$u_t(x,t)=\nu u_{xx}(x,t)-u(x,t)u_x(x,t),$$

with a positive parameter ν one can use the above representations to get

$$\begin{aligned} & \sum_j v_j(x)w'(x_j,t) \\ \approx\ & \nu\sum_j v_j''(x)w(x_j,t) \\ & -\left(\sum_j v_j(x)w(x_j,t)\right)\left(\sum_k v_k'(x)w(x_k,t)\right). \end{aligned}$$

Taken at a spatial node x_ℓ and using the Lagrange conditions (2) this gives a system

$$w'(x_\ell,t) \;=\; \nu\sum_j v_j''(x_\ell)w(x_j,t)-w(x_\ell,t)\left(\sum_k v_k'(x_\ell)w(x_k,t)\right)$$

of ordinary differential equations for the unknown scalar functions $w(x_k,t)$. This is a special instance of the well–known *Method of Lines*. Note how the Lagrange conditions (2) serve to make the system explicit. There are many successful applications of this technique starting from an early paper of E. Kansa [5] in 1990, but without any theoretical foundation, so far.

This paper provides a rigorous mathematical analysis of the Method of Lines in conjunction with kernel–based meshless trial spaces. Since this will be quite technical, we summarize the major building blocks in short and simplified terms here.

1. We admit nonlinear evolution equations which are first–order in time and depend smoothly on an arbitrary number of linear multivariate spatial operators. The equations (including boundary conditions) are assumed to be well–posed in a bounded time domain $[0, T]$, and should have a smooth unique solution

   $$u \,:\, \Omega \times [0, T] \to \mathbb{R}$$

   on a bounded spatial domain $\Omega \subset \mathbb{R}^d$. If the equations are written as e.g.

   $$u_t = F(t, u, \nabla u, \Delta u, \mathcal{L}u, \ldots),$$

   with certain spatial linear operators, the nonlinear function F should be smooth and well–defined on a domain containing the trajectory of the true solution u and some leeway around it.

2. We require smooth initial values specified for $t = 0$ on the domain Ω.

3. We allow time–dependent boundary conditions specified by evaluations of arbitrary linear spatial operators on the boundary $\Gamma = \partial\Omega$ of Ω or a part thereof, provided that well–posedness and solvability of the evolution problem is guaranteed.

4. The above smoothness assumptions must be such that the solution $u(\cdot, t)$ lies in a Sobolev space $W_2^\tau(\Omega)$ for each time t. For the final results, we need

   $$\tau > k + 3d/2 \tag{3}$$

   where d is the spatial dimension and k is the maximal order of spatial differential operators in the right–hand side of the evolution equation.

5. To define the trial space, we take a positive definite kernel. It should be reproducing in some associated “native” Hilbert space H which is continuously embedded in $W_2^\tau(\Omega)$, and we require the solution $u(\cdot, t)$ to be in H for all t.

6. Then we employ a single fine–grained and nondegenerate spatial discretization for both the given initial and boundary data and the values of trial functions in the interior of Ω. If $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are the nodes in Ω, the granularity of the discretization is measured by the fill distance

   $$h := \sup_{x \in \Omega} \min_{1 \le i \le m} \|\mathbf{x} - \mathbf{x}_i\|_2.$$

7. On this discretization, we use our smooth kernel to provide generalized Hermite interpolation of both boundary and interior values. This yields an affine trial space satisfying the discretized boundary conditions by Hermite interpolation, and its linear part is parameterized by a time–dependent vector
$$\mathbf{w}(t) = (w(\mathbf{x}_1, t), \dots, w(\mathbf{x}_m, t))^T \tag{4}$$
of values at nodes $\mathbf{x}_1, \dots, \mathbf{x}_m$ in the interior of Ω.

8. The linear spatial operators occurring in the evolution equation are applied analytically to trial functions, and each result is again expressed by $\mathbf{w}(t)$.

9. In terms of these parameters, the Method of Lines then yields a nonlinear ODE system of the standard explicit form
$$\mathbf{w}'(t) = \mathbf{G}(\mathbf{w}(t), t),\ \mathbf{w}(0) = \mathbf{w}_0 \tag{5}$$
for the vector $\mathbf{w}(t)$. Initial values are taken from the discretization of the the initial values of the evolution equation. The boundary conditions are built into the trial space via item (7) above.

Now for the results of the paper, in equally simplified form:

1. The ODE system (5) of the Method of Lines is solvable in $[0, T]$, if the fill distance h of the spatial discretization is small enough to satisfy
$$Ch^{\tau-d/2-k} T \exp(LT) \|u\| \leq R \tag{6}$$
and $h < h_0$ for a certain positive number h_0 depending on the domain, the PDE and the smoothness assumptions, where
    - L is a Lipschitz–type constant like in the standard Picard–Lindelöf theory,
    - $\|u\|$ is the sup of all $\|u(\cdot, t)\|_H$ over $t \in [0, T]$,
    - R is a parameter increasing with the leeway of F around the exact solution.

2. If $w(\mathbf{x}, t)$ is the trial function (1) interpolating the values (4) given by solving the ODE system (5), there is an error bound of the form
$$\|(u - w)(\cdot, t)\|_{\infty,\Omega} \leq Ch^{\tau-3d/2-k}(1 + t \exp(Lt))\|u\|.$$

3. Time–stepping methods using kernel–based trial spaces for discretization of the spatial variables coincide with time-stepping for the ODE system of the Method of Lines. Thus a separate convergence analysis of time–stepping methods is not necessary.

Consequently, the Method of Lines has an extremely wide application scope and allows a surprisingly general convergence analysis which by the previous argument also gives some insight into corresponding time–stepping techniques. For illustration, we give a simple example at the end of the paper. It considers the Burgers equation and is reproducing the experimental results by Hon and Mao [4].

The kernel–based Method of Lines is not suggested here as the method of choice for solving evolution equations numerically, in particular not in the crude form we analyze it here. But there are some advantages of the Method of Lines that should be mentioned and that deserve further study, both theoretically and numerically.

There is no time discretization on the PDE level, and there is just a single spatial discretization leading to a space of smooth and explicitly available trial functions. Spatial derivatives thereof need no additional discretization, because they can be applied directly and analytically to the trial functions. This simplifies both the analysis and the numerical implementation via a large choice of admissible kernel–based trial spaces, and it completely avoids spatial numerical integration. This is in sharp contrast to finite element or finite–volume methods.

The analysis given here applies directly to the ODE system to be solved by the Method of Lines. This makes arguments about time–stepping and stability unnecessary, because these are now part of the ODE solver chosen. Note that the majority of applications like [5, 4] perform some kind of time discretization. Consistency is no issue either, due to the way the ODE system is constructed directly from the PDE.

Since at each time t one has a smooth non–discrete approximation of the solution, it is easy to implement changes in the spatial discretization. This would be called “re–meshing” in standard techniques, but here we have a meshless method which does not make assumptions about connectivity, so that changes in the discretization can easily be made by re–interpolation on another discretization. One can also implement certain forms of adaptivity without serious problems (see an early case by Hon and Mao [4]), but all of these numerical enhancements should be left to subsequent papers for a thorough theoretical study.

Since there are kernels generating divergence–free vector fields (see Narcowich and Ward [7]), it should be possible to extend the results to the Navier–Stokes equations. A paper concerning this is in preparation.

However, some natural shortcomings of the above results should be pointed out, too. First, the theory and the method are only applicable in cases of high regularity, since they completely ignore the notion of weak solutions. Low regularity cases will need a different treatment, but then they usually require spatial numerical integration.

There will be no problems for simple cases with small Lipschitz constants of F on large domains. But if the true solution on $[0, T]$ runs close to a singularity or a shock (this is to be expected even for simple spatially univariate cases like the Burgers equation), one has to expect large Lipschitz constants along the solution, for instance when terms like $u \cdot u_x$ are considered. Thus one cannot use large R to get much singularity–free leeway in the condition (6). Furthermore, the Lipschitz constant L must be valid also on the full additional leeway needed via R for solvability of the ODE system up to T, and thus users cannot pick a large R unless accepting huge values of L, spoiling the gain in (6) for large R. These considerations lead to the conclusion that unrealistically fine spatial discretizations will be needed to resolve solutions close to singularities, in particular if they are far away in time, i.e. when the $\exp(Lt)$ factor in the error bound hurts most. This is quite natural and will be applicable to many other methods as well. The ODE system generated by the method of Lines will be stiff in such situations, but sophisticated future users will switch from a coarse to a fine discretization when coming close to a singularity.

Users might think that there is a way out by using extremely large τ, i.e. extremely smooth kernels and assuming extremely smooth solutions. But the constants above will also depend heavily on τ (though not on h), and certain proof ingredients will only be valid for all $h \leq h_0$ below a small h_0 which also depends on τ. This also enforces rather fine spatial discretizations through the back door. Experts in kernel techniques might suggest to use analytic kernels with exponential convergence rates, but some of the proof ingredients are not yet established for this situation. This requires further research.

The rest of the paper starts with setting up the time–dependent partial differential equations we admit, and then it describes the kernel–based discretizations with their special technique for handling boundary conditions. Using this discretization, the ODE system for the Method of Lines is derived, and it turns out to be a perturbation of another ODE system solved by the values $u(\mathbf{x}_j, t)$ of the solution u of the PDE. Then perturbation theory of ODEs is invoked to produce the stated results, and a numerical example for the Burgers equation finishes the paper.

2 Nonlinear Evolution Equations

Consider a spatial domain $\Omega \subset \mathbb{R}^d$ and various linear operators $\mathcal{L}_i$, $i \in I$ acting on smooth functions on Ω. We write the spatial variables as vectors $\mathbf{x}$, $\mathbf{y}, \ldots \in \Omega$, and we assume that the operators $\mathcal{L}_i$ always act with respect to the spatial variables, even if a time variable t is present later.

On a time domain $[0, T]$ we look for a scalar function

$$u \,:\, \Omega \times [0, T] \to \mathbb{R}$$

satisfying a nonlinear evolution equation

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = F(\mathcal{L}_i(u)(\mathbf{x}, t), \mathbf{x}, t), \tag{7}$$

where for notational convenience we assume that the arguments of the right–hand side F consist of all $\mathcal{L}_i(u)$ for indices i in some index set I. This allows examples like

$$\begin{array}{rcl} \dfrac{\partial u}{\partial t} & = & F(u, \mathbf{x}, t) \\ \dfrac{\partial u}{\partial t} & = & F(u, \nabla u, \mathbf{x}, t) \\ \dfrac{\partial u}{\partial t} & = & F(u, \nabla u, \Delta u, \mathbf{x}, t) \end{array}$$

with F being smooth and nonlinear, while all spatial operators must be linear.

Though we shall fix initial or boundary conditions later, we assume strong solvability of the problem (7) on $[0, T]$. This means that u and F have enough smoothness to let (7) to be satisfied pointwise on $\Omega \times [0, T]$. Furthermore, we assume that the time derivatives of the functions $\mathcal{L}_i u(\mathbf{x}, t)$ are still smooth.

3 Spatial Kernel Interpolation

Let us forget the time variable for a moment and focus on the spatial variables. We take a smooth symmetric positive definite kernel function

$$K \ : \Omega \times \Omega \to \mathbb{R}$$

like the Gaussian

$$K(\mathbf{x}, \mathbf{y}) := \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2), \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$$

or one of Wendland's [8] compactly supported radial basis functions. Each such kernel defines a "native" Hilbert space H of functions on Ω and with an inner product $(.,.)_H$ in which the kernel K is reproducing in the sense

$$f(\mathbf{x}) = (f, K(\mathbf{x}, .))_H \text{ for all } \mathbf{x} \in \Omega, \ f \in H.$$

More generally,

$$\lambda(f) = (f, \lambda^{\mathbf{x}} K(\mathbf{x}, .))_H$$

for all $f \in H$ and all continuous linear functionals λ from the dual space H^*. The upper index at $\lambda^{\mathbf{x}}$ indicates that λ acts with respect to $\mathbf{x}$.

We assume the kernel to be smooth enough for what we need, and for reasons coming up later we assume that its native space H can be continuously embedded into some Sobolev space $W_2^\tau(\Omega)$ for some τ with (3).

If one takes a finite set $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_{|X|}\}$ of points in Ω, one can define a trial space

$$S_X := \text{ span } \{K(\mathbf{x}_k, \cdot), \ \mathbf{x}_k \in X\}$$

which is a subspace of H. Then it is a standard result of meshless kernel–based techniques [9] that any function $f \in H$ can be interpolated on X by a function s in the trial space S_X, i.e.

$$s(\mathbf{x}_k) = f(\mathbf{x}_k) \text{ for all } \mathbf{x}_k \in X. \tag{8}$$

In fact, under our assumptions the symmetric "kernel matrix" $\mathbf{A}$ with entries

$$K(\mathbf{x}_j, \mathbf{x}_k) \text{ for all } \mathbf{x}_j, \mathbf{x}_k \in X \tag{9}$$

is nonsingular, and if we represent s as

$$s(\mathbf{x}) := \sum_{\mathbf{x}_j \in X} a_j K(\mathbf{x}, \mathbf{x}_j)$$

the interpolation (8) is solved via the linear system

$$\sum_{\mathbf{x}_j \in X} a_j K(\mathbf{x}_j, \mathbf{x}_k) = f(\mathbf{x}_k) \text{ for all } \mathbf{x}_k \in X. \tag{10}$$

For later use, we cite a standard result [6, 9] on the error behavior.

Theorem 1. *Assume that the kernel K is smooth enough such that its native Hilbert space is continuously embedded in Sobolev space $W_2^\tau(\Omega)$, and let the bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ have an interior cone condition. Then there are positive constants C and h_0 such that for all $f \in W_2^\tau(\Omega)$ and all interpolants s_{f,X_h} on any discrete set $X_h \subset \Omega$ with* fill distance

$$h := \sup_{y \in \Omega} \min_{x \in X_h} \|x - y\|_2 \leq h_0$$

the error has a bound

$$\|D^\alpha f - D^\alpha s_{f,X_h}\|_{\infty,\Omega} \leq C h^{\tau - |\alpha| - d/2} \|f\|_{W_2^\tau(\Omega)} \tag{11}$$

including derivatives up to order $|\alpha| < \tau - d/2$. □

The assumption (3) gives us enough leeway here for taking $|\alpha| \leq k$, the maximal order of spatial differential operators we have to care for.

4 Boundary Conditions

But in order to handle differential equations with boundary conditions, we need more. To see what is needed, we imagine that systems like (10) have to be posed for each time t. But already for the easy case of time–constant Dirichlet conditions, we have to fix certain boundary data of trial functions while the rest will be time–dependent. If we want to calculate via the coefficients a_j, these will all be time–dependent but subject to additional constraints imposed by the Dirichlet conditions. Furthermore, we also want to allow more general boundary conditions than Dirichlet conditions.

Summarizing, we have to decouple “fixed” and “free” parts of the trial space, and we have to incorporate general boundary conditions into the fixed part, admitting derivatives. Thus we generalize the interpolation problem as follows. We assume that our trial functions $s(\mathbf{x})$ on Ω are partially restricted by a linear spatial “boundary” operator B and certain conditions

$$(Bs)(\mathbf{x}) = g(\mathbf{x}) \text{ for all } \mathbf{x} \in \Gamma \subseteq \partial\Omega$$

with a given function $g(\mathbf{x})$ on a part Γ of the boundary $\partial\Omega$ of the domain Ω. It is allowed that B consists of several parts, e.g. Dirichlet conditions on one part and Neumann conditions on another part of the boundary. Then we discretize B by the conditions

$$\lambda_i^{\mathbf{x}}(s(\mathbf{x})) := (Bs)(\mathbf{y}_i) = g(\mathbf{y}_i)$$

for points $\mathbf{y}_i \in \Gamma \subseteq \partial\Omega$, $1 \leq i \leq n$. Thus we assume n continuous and linearly independent functionals $\lambda_1, \ldots, \lambda_n \in H^*$ to be given which will resemble discretized boundary conditions as above. They may take the form of point evaluations in case of Dirichlet data, but they may also evaluate a normal derivative, for instance, and these types can be mixed.

In addition to this, and in order to have enough degrees of freedom, we take a finite set $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_{|X|}\}$ of points in Ω as above, avoiding the above points, if they concern Dirichlet boundary data. Our free parameters will consist of function values at these nodes. To facilitate the notation, we add the linear functionals $\delta_{\mathbf{x}_k}$ for $\mathbf{x}_k \in X$ to the other functionals, giving us a total of $n + |X| =: N$ linearly independent functionals which we rename as $\mu_1, \ldots, \mu_N \in H^*$.

To facilitate the notation, we introduce the following evaluation operators

$$\begin{array}{rcl} E_\Lambda & : & f \mapsto (\lambda_1(f), \ldots, \lambda_n(f))^T \\ E_M & : & f \mapsto (\mu_1(f), \ldots, \mu_N(f))^T \\ E_X & : & f \mapsto (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{|X|})^T \\ E_M & = & (E_\Lambda^T, E_X^T)^T \end{array}$$

generating column vectors, while their transposes generate rows. We use these operators to generate matrices and vectors. For instance, the matrix of (10) is generated by

$$\mathbf{A} = E_X^{\mathbf{x}}(E_X^{\mathbf{y}})^T K(\mathbf{x}, \mathbf{y})$$

where the upper index indicates the variable to be acted on. Similarly, our boundary conditions are

$$E_\Lambda^{\mathbf{y}} s(\mathbf{y}) = (E_Y B)^{\mathbf{y}} s(\mathbf{y}) = E_Y^{\mathbf{y}} g(\mathbf{y}).$$

Our final notational example is the trial function

$$(E_M^{\mathbf{x}})^T K(\mathbf{x}, \cdot) a = \sum_{j=1}^{N} a_j \mu_j^{\mathbf{x}} K(\mathbf{x}, \cdot)$$

defined for any vector $a \in \mathbb{R}^N$.

Following a paper [11] by Z.M. Wu we apply the theory of Hermite interpolation by kernels. We define the trial space

$$S_M := \{(E_M^{\mathbf{x}})^T K(\mathbf{x}, \cdot)a \;:\; a \in \mathbb{R}^N\}$$

and rewrite each function

$$s := (E_M^{\mathbf{x}})^T K(\mathbf{x}, \cdot)a \in S_M$$

with $a \in \mathbb{R}^N$ as its own Hermite interpolant solving the system

$$E_M^{\mathbf{y}}(E_M^{\mathbf{x}})^T K(\mathbf{x}, \mathbf{y})a = E_M^{\mathbf{y}} s(\mathbf{y})$$

with the symmetric positive definite generalized kernel matrix

$$\mathbf{A} := E_M^{\mathbf{y}}(E_M^{\mathbf{x}})^T K(\mathbf{x}, \mathbf{y}) \tag{12}$$

which generalizes the matrix $\mathbf{A}$ of (10) and will be needed in all further calculations. But we can avoid the kernel coefficients by going over to a Lagrange form via

$$E_M^{\mathbf{y}}(E_M^{\mathbf{x}})^T K(\mathbf{x}, \mathbf{y})V(\mathbf{z}) = \mathbf{A}V(\mathbf{z}) = E_M^{\mathbf{y}} K(\mathbf{y}, \mathbf{z})$$

which is solvable for a vector

$$V(\mathbf{z}) := (v_1(\mathbf{z}), \ldots, v_N(\mathbf{z}))^T = \mathbf{A}^{-1} E_M^{\mathbf{y}} K(\mathbf{y}, \mathbf{z}) \tag{13}$$

satisfying the Lagrange conditions

$$(E_M^{\mathbf{z}})^T V(\mathbf{z}) = I_N$$

with the $N \times N$ unit matrix I_N. We shall need later that this splits into four matrix identities

$$\begin{array}{rclrcl} (E_\Lambda^{\mathbf{z}})^T V_\Lambda(\mathbf{z}) & = & I_{|\Lambda|}, & (E_\Lambda^{\mathbf{z}})^T V_X(\mathbf{z}) & = & 0, \\ (E_X^{\mathbf{z}})^T V_\Lambda(\mathbf{z}) & = & 0, & (E_X^{\mathbf{z}})^T V_X(\mathbf{z}) & = & I_{|X|}. \end{array} \tag{14}$$

We can now rewrite s in Lagrange form as

$$s(\mathbf{x}) := V(\mathbf{x})^T E_M^{\mathbf{y}} s(\mathbf{y}).$$

If we split V into

$$V(\mathbf{z}) := (v_1(\mathbf{z}), \ldots, v_N(\mathbf{z}))^T =: (V_\Lambda^T(\mathbf{z}), V_X^T(\mathbf{z}))^T$$

we can write

$$\begin{aligned} s(\mathbf{x}) &= V_\Lambda(\mathbf{x})^T E^{\mathbf{y}}_\Lambda s(\mathbf{y}) + V_X(\mathbf{x})^T E^{\mathbf{y}}_X s(\mathbf{y}) \\ &= V_\Lambda(\mathbf{x})^T E^{\mathbf{y}}_Y g(\mathbf{y}) + V_X(\mathbf{x})^T E^{\mathbf{y}}_X s(\mathbf{y}) \end{aligned}$$

to split the representation of the trial function s in a way we can use later, because $E^{\mathbf{y}}_\Lambda s(\mathbf{y}) = E^{\mathbf{y}}_Y g(\mathbf{y})$ is fixed by boundary conditions, while $E^{\mathbf{y}}_X s(\mathbf{y})$ is free and will be entering into the differential equation. For numerical purposes, we shall parameterize functions s from our trial space including boundary conditions by the vector $E^{\mathbf{y}}_X s(\mathbf{y}) \in \mathbb{R}^{|X|}$ only.

But we also want to use the fact that we have a spatial discretization which allows us to evaluate spatial differential operators analytically on our trial functions, without any additional discretization for taking derivatives. Let L be a linear operator acting on functions in Ω. Then we have

$$Ls(\mathbf{x}) = (LV(\mathbf{x}))^T E^{\mathbf{y}}_M s(\mathbf{y}) = (LV_\Lambda(\mathbf{x}))^T E^{\mathbf{y}}_Y g(\mathbf{y}) + (LV_X(\mathbf{x}))^T E^{\mathbf{y}}_X s(\mathbf{y}) \quad (15)$$

and want to re–express the vector $LV(\mathbf{x})$ in terms of the data, like in spectral methods. But this is easy via (13) because of

$$\begin{aligned} LV(\mathbf{z}) &= (Lv_1(\mathbf{z}), \ldots, Lv_N(\mathbf{z}))^T \\ &= \mathbf{A}^{-1} L^{\mathbf{z}} E^{\mathbf{y}}_M K(\mathbf{y}, \mathbf{z}) \\ &= \mathbf{A}^{-1} E^{\mathbf{y}}_M L^{\mathbf{z}} K(\mathbf{y}, \mathbf{z}) \\ &= (LV_\Lambda(\mathbf{z})^T, LV_X(\mathbf{z})^T)^T. \end{aligned}$$

We only have to take our kernel to be smooth enough to allow L, which is guaranteed by (3), and we take analytical derivatives of the kernel to generate the matrices we need. Note that identities like

$$\begin{aligned} E^{\mathbf{x}}_X Ls(\mathbf{x}) &= E^{\mathbf{x}}_X (LV(\mathbf{x}))^T E^{\mathbf{y}}_M s(\mathbf{y}) \\ &= E^{\mathbf{x}}_X (LV_\Lambda(\mathbf{x}))^T E^{\mathbf{y}}_Y g(\mathbf{y}) + E^{\mathbf{x}}_X (LV_X(\mathbf{x}))^T E^{\mathbf{y}}_X s(\mathbf{y}) \end{aligned}$$

arising when evaluating (15) on X yield “differentiation matrices” in the context of pseudo-spectral methods if the boundary conditions are not present. See the paper [3] of G. Fasshauer for the connection to pseudo-spectral methods.

But we still need an extension of the error bound (11) to the case of additional boundary conditions. We go back to the terminology there, but we add certain interpolation conditions we do not need to specify in detail, provided that interpolation in points of a sufficiently dense subset X_h is maintained. If $s_{f,h}$ is such an extended interpolant to some function $f \in W_2^\tau(\Omega)$, the usual minimum–norm property of interpolants implies $\|s_{f,h}\| \leq \|f\|$ in the native space norm. We then invoke results of Narcowich, Ward, and Wendland [6, 9] to get

$$\begin{array}{rl} & \|D^\alpha f - D^\alpha s_{f,h}\|_{\infty,\Omega} \\ \leq & Ch^{\tau-|\alpha|-d/2}\|f - s_{f,h}\|_{W_2^\tau(\Omega)} \\ \leq & Ch^{\tau-|\alpha|-d/2}\left(\|f\|_{W_2^\tau(\Omega)} + \|s_{f,h}\|_{W_2^\tau(\Omega)}\right) \\ \leq & Ch^{\tau-|\alpha|-d/2}\left(\|f\|_{W_2^\tau(\Omega)} + \|s_{f,h}\|\right) \\ \leq & Ch^{\tau-|\alpha|-d/2}\left(\|f\|_{W_2^\tau(\Omega)} + \|f\|\right) \\ \leq & Ch^{\tau-|\alpha|-d/2}\|f\| \end{array} \tag{16}$$

with generic constants involving the embedding of the native Hilbert space into Sobolev space. This means that the error behavior still is dependent on the standard Lagrange data in the domain, and the additional boundary conditions can be ignored within the error analysis.

5 Time–Dependent Interpolation

We assume that a function $u(\mathbf{x}, t)$ on $\Omega \times [0, T]$ is partially restricted by a spatial linear boundary operator B and the conditions

$$B^{\mathbf{x}} u(\mathbf{x}, t) = g(\mathbf{x}, t) \text{ for all } \mathbf{x} \in \Gamma \subseteq \partial\Omega,\ t \in [0, T]$$

with a given function $g(\mathbf{x}, t)$. Like in the previous section, we discretize B by the conditions

$$\lambda_i^{\mathbf{x}}(u(\mathbf{x}, t)) := (B^{\mathbf{x}} u)(\mathbf{y}_i, t) = g(\mathbf{y}_i, t)$$

for points $\mathbf{y}_i \in \Gamma \subseteq \partial\Omega$, $1 \leq i \leq n$. Using our standard notation, this is

$$E_\Lambda^{\mathbf{y}} u(\mathbf{y}, t) = (E_Y B)^{\mathbf{y}} u(\mathbf{y}, t) = E_Y^{\mathbf{y}} g(\mathbf{y}, t).$$

Interpolating u with the methods of the previous section will give a function $s(\mathbf{x}, t)$ with

$$\begin{array}{rcl} s(\mathbf{x}, t) & = & V_\Lambda(\mathbf{x})^T E_\Lambda^{\mathbf{y}} u(\mathbf{y}, t)) + V_X(\mathbf{x})^T E_X^{\mathbf{y}} u(\mathbf{y}, t) \\ & = & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + V_X(\mathbf{x})^T E_X^{\mathbf{y}} u(\mathbf{y}, t) \\ & = & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y}, t) \end{array} \tag{17}$$

which now is parameterized via the unknown quantities $E_X^{\mathbf{y}} u(\mathbf{y}, t) = E_X^{\mathbf{y}} s(\mathbf{y}, t)$, while the boundary data are determined.

For convenient programming, we write the action of an operator L on $s(\mathbf{x},t)$ with values in X as

$$\begin{array}{rcl}(E_X L)^{\mathbf{x}} s(\mathbf{x},t) & = & (E_X L)^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},t) \\ & & +(E_X L)^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y},t) \\ & = & \mathbf{B}_L E_Y^{\mathbf{y}} g(\mathbf{y},t) + \mathbf{C}_L E_X^{\mathbf{y}} s(\mathbf{y},t)\end{array}$$

where the matrices $\mathbf{B}_L$ and $\mathbf{C}_L$ follow from

$$\begin{array}{rcl}\mathbf{D}_L \mathbf{A}^{-1} & = & (\mathbf{B}_L, \mathbf{C}_L) \\ \mathbf{D}_L & := & (E_X L)^{\mathbf{x}} (E_M^{\mathbf{y}})^T K(\mathbf{y},\mathbf{x}).\end{array} \tag{18}$$

Note that all matrices are independent of t and can be calculated and stored on startup. Numerical calculation will focus on the $|X|$ time–dependent scalar functions in $E_X^{\mathbf{y}} s(\mathbf{y},t)$ only. They will arise later in the ODE system of the Method of Lines.

6 Euler Method

Since many readers may be more familiar with time–stepping techniques than with the Method of Lines, here is a short detour.

We use the representation (17) for our approximate solution at time t. We assume that the starting values at time $t = 0$ can be fully obtained as $E_X^{\mathbf{y}} s(\mathbf{y},0) = E_X^{\mathbf{y}} u(\mathbf{y},0)$ such that we have

$$s(\mathbf{x},0) = V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},0) + V_X(\mathbf{x})^T E_X^{\mathbf{y}} u(\mathbf{y},0)$$

at startup. We assume the differential equation at t to be (7), and we insert our approximation into the right–hand side as a function. This gives

$$\begin{array}{rl} & F(\mathcal{L}_i(s),\mathbf{x},t) \\ = & F\left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y},t),\mathbf{x},t\right)\end{array}$$

in terms of our free parameters $E_X^{\mathbf{y}} s(\mathbf{y},t)$. We evaluate this right–hand side using the time–independent matrices of the previous section, and interpret the result as

$$\frac{\partial s(\mathbf{x},t)}{\partial t} \approx \frac{s(\mathbf{x},t+\delta) - s(\mathbf{x},t)}{\delta}.$$

Let us go for the plain Euler method and use the spatial function

$$s^+(\mathbf{x}) := s(\mathbf{x},t) + \delta F\left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y},t),\mathbf{x},t\right)$$

as a candidate for $s(\mathbf{x}, t + \delta)$. But this is not in the trial space, and it fails to satisfy the boundary conditions. Thus we interpolate $s^+(\mathbf{x})$ spatially and enforce boundary conditions by defining

$$s(\mathbf{x}, t + \delta) := V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + V_X(\mathbf{x})^T E_X^{\mathbf{x}} s^+(\mathbf{x}).$$

Note that this means

$$\begin{array}{rl} & s(\mathbf{x}, t + \delta) \\ = & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) \\ & + V_X(\mathbf{x})^T E_X^{\mathbf{x}} s(\mathbf{x}, t) \\ & + \delta V_X(\mathbf{x})^T E_X^{\mathbf{x}} F \left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y}, t), \mathbf{x}, t \right) \\ = & s(\mathbf{x}, t) \\ & + \delta V_X(\mathbf{x})^T E_X^{\mathbf{x}} F \left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y}, t), \mathbf{x}, t \right) \end{array}$$

as an iteration in the trial space, and by taking the values at X we arrive at the discrete form

$$\begin{array}{rl} & E_X^{\mathbf{y}} s(\mathbf{y}, t + \delta) \\ = & E_X^{\mathbf{y}} s(\mathbf{y}, t) \\ & + \delta E_X^{\mathbf{x}} F \left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y}, t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y}, t), \mathbf{x}, t \right) \end{array}$$

of this iteration, as it could be implemented. One can introduce the shorthand notation

$$\begin{array}{rcl} \mathbf{w}(t) & := & E_X^{\mathbf{y}} s(\mathbf{y}, t) \\ \mathbf{g}(t) & := & E_Y^{\mathbf{y}} g(\mathbf{y}, t) \end{array}$$

to bring this into the form

$$\mathbf{w}(t + \delta) = \mathbf{w}(t) + \delta E_X^{\mathbf{x}} F \left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{w}(t), \mathbf{x}, t \right)$$

which can be viewed as an Euler step applied to the ODE system

$$\mathbf{w}'(t) = E_X^{\mathbf{x}} F \left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{w}(t), \mathbf{x}, t \right)$$

we shall encounter in the next section, deriving it differently.

7 Method of Lines

Our version of the Method of Lines first replaces the evolution equation (7) by one on a time–dependent trial space, i.e.:

$$\frac{\partial s(\mathbf{x}, t)}{\partial t} = F(\mathcal{L}_i(s)(\mathbf{x}, t), \mathbf{x}, t)$$

and then discretizes this on a set of spatial points in order to get a system of ordinary differential equations. We use the representation (17) again, but take the time derivative to get

$$\begin{aligned} & \frac{\partial}{\partial t} s(\mathbf{x},t) =: s_t(\mathbf{x},t) \\ = \ & V_\Lambda(\mathbf{x})^T \frac{\partial}{\partial t} E_Y^{\mathbf{y}} g(\mathbf{y},t) + V_X(\mathbf{x})^T \frac{\partial}{\partial t} E_X^{\mathbf{y}} s(\mathbf{y},t) \\ = \ & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g_t(\mathbf{y},t) \\ & + V_X(\mathbf{x})^T E_X^{\mathbf{y}} \frac{\partial}{\partial t} \left(V_\Lambda(\mathbf{y})^T E_Y^{\mathbf{z}} g(\mathbf{z},t) + V_X(\mathbf{y})^T E_X^{\mathbf{z}} s(\mathbf{z},t) \right) \\ = \ & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g_t(\mathbf{y},t) + V_X(\mathbf{x})^T E_X^{\mathbf{z}} \frac{\partial}{\partial t} s(\mathbf{z},t). \end{aligned} \tag{19}$$

This is parameterized by the $|X|$ scalar functions of time in the vector

$$\mathbf{w}'(t) := E_X^{\mathbf{z}} \frac{\partial}{\partial t} s(\mathbf{z},t)$$

which is the time derivative of

$$\mathbf{w}(t) := E_X^{\mathbf{z}} s(\mathbf{z},t)$$

parameterizing our trial functions $s(\mathbf{x},t)$. Still without discretizing the $\mathbf{x}$ variable, we write down the system

$$\begin{aligned} & s_t(\mathbf{x},t) \\ = \ & V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g_t(\mathbf{y},t) + V_X(\mathbf{x})^T E_X^{\mathbf{z}} \frac{\partial}{\partial t} s(\mathbf{z},t) \\ = \ & F(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T E_X^{\mathbf{y}} s(\mathbf{y},t), \mathbf{x}, t) \end{aligned}$$

of ODEs making up the semi–discrete version of the Method of Lines. It can be rewritten as

$$\begin{aligned} & V_X(\mathbf{x})^T \mathbf{w}'(t) \\ = \ & F(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{w}(t), \mathbf{x}, t) - V_\Lambda(\mathbf{x})^T \mathbf{g}'(t) \end{aligned}$$

and using (14) it takes a particularly simple explicit form

$$\mathbf{w}'(t) = E_X^{\mathbf{x}} F(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{w}(t), \mathbf{x}, t) \tag{20}$$

if the evaluation operator $E_X^{\mathbf{x}}$ is applied. We already encountered this system at the end of the previous section, and we want to simplify it to

$$\mathbf{w}'(t) \ = \ \mathbf{F}(\mathbf{w}(t), 0, t) \tag{21}$$

with the right–hand side

$$\begin{array}{rl} & \mathbf{F}(\mathbf{z}, v(\cdot,\cdot), t) \\ := & E_X^{\mathbf{x}} F(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{z} + \mathcal{L}_i^{\mathbf{x}} v(\mathbf{x}, t), \mathbf{x}, t) \end{array} \tag{22}$$

acting partially on functions v defined and smooth on $\Omega \times [0, T]$. The role of the "perturbation" v which is zero in (21) will become apparent in the next section.

Note that the previous section derived an Euler step for solving the ODE system of the Method of Lines. Other time discretizations like Crank–Nicholson can also be viewed as time–stepping techniques for the ODE system of the Method of Lines. Thus we do not need to pursue analysis of time–stepping methods in the standard Lax–Richtmyer style. Instead, we directly go for a theory concerning the Method of Lines system directly, without time discretization.

8 Discrete Error Analysis

For comparison, let us look at equations solved by the true solution $u(\mathbf{x}, t)$ if discretized at the points of X via

$$\mathbf{u}(t) := E_X^{\mathbf{x}} u(\mathbf{x}, t).$$

The system (7) gives

$$\mathbf{u}'(t) = E_X^{\mathbf{x}} F(\mathcal{L}_i^{\mathbf{x}}(u)(\mathbf{x}, t), \mathbf{x}, t), \tag{23}$$

but the right–hand side is not a function of $\mathbf{u}(t)$, and thus the above system is not in ODE form. Since

$$s(\mathbf{x}, t) = V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + V_X(\mathbf{x})^T \mathbf{u}(t)$$

is the interpolant to u from (17), we can approximate the right–hand side of (23) by something we can handle somewhat easier:

$$\begin{array}{rl} & \mathcal{L}_i^{\mathbf{x}}(u)(\mathbf{x}, t) \\ = & \mathcal{L}_i^{\mathbf{x}} s(\mathbf{x}, t) + \mathcal{L}_i^{\mathbf{x}}(u(\mathbf{x}, t) - s(\mathbf{x}, t)) \\ = & \mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{u}(t) + \mathcal{L}_i^{\mathbf{x}}(u(\mathbf{x}, t) - s(\mathbf{x}, t)). \end{array}$$

This turns (23) into

$$\begin{array}{rl} & \mathbf{u}'(t) \\ = & E_X^{\mathbf{x}} F\left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T \mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T \mathbf{u}(t) + \mathcal{L}_i^{\mathbf{x}}(u(\mathbf{x}, t) - s(\mathbf{x}, t)), \mathbf{x}, t\right). \end{array}$$

Consequently, the vector function $\mathbf{u}$ satisfies the ODE system

$$\mathbf{u}'(t) = \mathbf{F}(\mathbf{u}(t), \epsilon, t)$$

where we treat the error function

$$\epsilon(\mathbf{x}, t) := u(\mathbf{x}, t) - s(\mathbf{x}, t)$$

as a fixed perturbation in the right–hand side (22) of the system. Thus we can now compare the systems

$$\mathbf{w}'(t) \;=\; \mathbf{F}(\mathbf{w}(t), 0, t), \quad \mathbf{u}'(t) \;=\; \mathbf{F}(\mathbf{u}(t), \epsilon, t)$$

with identical initial values

$$\mathbf{w}(0) = \mathbf{u}(0) = E_X^{\mathbf{x}} u(\mathbf{x}, 0),$$

where the second system is solvable in $[0, T]$.

At this point, we invoke standard perturbation theory of ODE systems (e.g. Satz 3.1.6 of the textbook [10] of H. Werner and H. Arndt, p. 76–77). For convenience of the reader, here is what we need:

Theorem 2. *Let $\Omega \subset \mathbb{R}^{n+1}$ be a bounded domain, and let*

$$f = f(t, y) \;:\; \Omega \to \mathbb{R}^n$$

be Lipschitz continuous with respect to y with Lipschitz constant L on Ω. Furthermore, let u be a continuous and piecewise differentiable function on $[0, T]$ with values in $\mathbb{R}^n$ such that

$$\begin{array}{rcl} (t, u(t)) & \in & \Omega \\ \|f(t, u(t)) - u'(t)\| & \leq & \epsilon_1 \text{ where } u'(t) \text{ is defined} \end{array}$$

for all $t \in [0, T]$. Furthermore, let $(0, y_0) \in \mathbb{R}^{n+1}$ be a point of Ω, and assume

$$\|y_0 - u(0)\| \;\leq\; \epsilon_0.$$

Finally, let t_1 be the largest point in $[0, T]$ such that the set

$$\{(t, y) \in \mathbb{R}^{n+1} \;:\; t \in [0, t_1],\; \|y - u(t)\| \leq (\epsilon_0 + t \cdot \epsilon_1) \exp(Lt)\}$$

is contained in Ω. Then there is a unique solution to the system

$$y'(t) = f(t, y(t)),\; y(0) = y_0$$

in $[0, t_1]$ and there is a bound

$$\|u(t) - y(t)\| \leq (\epsilon_0 + t \cdot \epsilon_1) \exp(Lt)$$

for all $t \in [0, t_1]$.

We apply this to $u(t) := \mathbf{u}(t)$ as follows. Let R be a fixed positive constant, and define Ω to be the union of all sets

$$\{(t,y)\ :\ \|y-\mathbf{u}(t)\|_\infty \le R\}.$$

This gives us an "R–tube" around the existing solution, and we assume that $f(t,y) := \mathbf{F}(y,0,t)$ is defined and Lipschitz continuous on Ω with Lipschitz constant L. Since we use $y_0 := \mathbf{u}(0)$, we have $\epsilon_0 = 0$, but we have to calculate ϵ_1. With a common Lipschitz constant L_F for the relevant arguments of F we get

$$\begin{array}{rl} & \|\mathbf{F}(\mathbf{z},\epsilon,t)-\mathbf{F}(\mathbf{z},0,t)\|_\infty \\ = & \big\|E_X^{\mathbf{x}} F\left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T\mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T\mathbf{z} + \mathcal{L}_i^{\mathbf{x}}\epsilon(\mathbf{x},t),\mathbf{x},t\right) \\ & -E_X^{\mathbf{x}} F\left(\mathcal{L}_i^{\mathbf{x}} V_\Lambda(\mathbf{x})^T\mathbf{g}(t) + \mathcal{L}_i^{\mathbf{x}} V_X(\mathbf{x})^T\mathbf{z},\mathbf{x},t\right)\big\|_\infty \\ \le & L_F \max_i \|\mathcal{L}_i^{\mathbf{x}}\epsilon(\mathbf{x},t)\|_\infty. \end{array}$$

If k is the maximal order of the differential operators $\mathcal{L}_i$, and if the norm of $u(\cdot,t)$ is taken in the native space of the kernel, we can invoke (16) to get

$$\begin{array}{rl} & \|\mathbf{F}(\mathbf{z},\epsilon,t)-\mathbf{F}(\mathbf{z},0,t)\|_\infty \\ \le & L_F \max_i \|\mathcal{L}_i^{\mathbf{x}}\epsilon(\mathbf{x},t)\|_\infty \\ \le & L_F h^{\tau-k-d/2}\|u(\cdot,t)\| =: \epsilon_1. \end{array}$$

Now we can choose t_1 to be the maximal $t \le T$ satisfying

$$\epsilon_1 t \exp(Lt) \le R$$

and get solvability of the system (21) on $[0,t_1]$ with an error bound

$$\|\mathbf{u}(t)-\mathbf{w}(t)\|_\infty \le L_F h^{\tau-k-d/2} t \exp(Lt)\|u(\cdot,t)\|. \qquad (24)$$

Theorem 3. *If the spatial discretization is fine enough in the sense of (6), the Method of Lines reaches the full solvability range and an arbitrarily small discrete error (24).* □

9 Global Error Analysis

We now want to bound the error between the spatial discretization points. Using he discrete values obtained from the Method of Lines, we define

$$w(\mathbf{x},t) := V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},t) + V_X(\mathbf{x})^T\mathbf{w}(t)$$

as a spatial interpolant to the discrete values in the vector $\mathbf{w}(t)$ and to the discretized boundary conditions. As an intermediate function, we introduce $v(\cdot,t)$ as the spatial interpolant to the values $\mathbf{u}(t)$, i.e.

$$v(\mathbf{x},t) := V_\Lambda(\mathbf{x})^T E_Y^{\mathbf{y}} g(\mathbf{y},t) + V_X(\mathbf{x})^T \mathbf{u}(t).$$

Since v interpolates u and is in the native space H for the kernel, we can apply standard convergence results like (11) to get

$$\begin{array}{rcl} \|(u-v)(\cdot,t)\|_{\infty,\Omega} & \leq & Ch^{\tau-d/2}\|u(\cdot,t)\|_{W_2^\tau(\Omega)} \\ & \leq & Ch^{\tau-d/2}\|u(\cdot,t)\| \end{array} \tag{25}$$

leaving us with

$$(w-v)(\mathbf{x},t) = V_X(\mathbf{x})^T \left(\mathbf{w}(t) - \mathbf{u}(t)\right).$$

The technique of [2] yields boundedness of Lagrange functions for well–positioned scattered data, and the proof there extends to cases where additional homogeneous data are satisfied on the boundary, provided that the spatial interpolation points stay away from the boundary by the separation distance. This is done by replacing a bound like (11) by (16) in [2].

Thus we have to expect an additional error of size $|X| = \mathcal{O}(h^{-d})$ to arrive at

$$\begin{array}{rl} & \|(u-w)(\cdot,t)\|_{\infty,\Omega} \\ \leq & \|(u-v)(\cdot,t)\|_{\infty,\Omega} + \|(v-w)(\cdot,t)\|_{\infty,\Omega} \\ \leq & Ch^{\tau-d/2}\|u(\cdot,t)\| + Ch^{-d}\|\mathbf{w}(t)-\mathbf{u}(t)\|_{\infty,\Omega} \\ \leq & Ch^{\tau-d/2}\|u(\cdot,t)\| + Ch^{-d}L_F h^{\tau-k-d/2} t\exp(Lt)\|u(\cdot,t)\|_\tau \\ \leq & Ch^{\tau-3d/2-k}(1+t\exp(Lt))\|u(\cdot,t)\|. \end{array} \tag{26}$$

Theorem 4. *Under the above assumptions, the global error behavior of the Method of Lines is described by (26).* □

10 Example

We do not claim that the Method of Lines in this primitive form is to be preferred over other methods, but we want to show that it can be set to work. Numerical improvements are left to future research.

Consider the Burgers equation

$$\begin{array}{rcll} u_t(x,t) & = & -u(x,t)u_x(x,t) + \nu u_{xx}(x,t), & -1 \leq x \leq 1,\ t \geq 0 \\ u(x,0) & = & 1-x^2, & -1 \leq x \leq 1 \\ u(1,t) & = & 0, & t \geq 0 \\ u(-1,t) & = & 0, & t \geq 0 \end{array}$$

which is known to develop a discontinuity near the boundary $x = 1$ if the Reynolds number $Re = 1/\nu$ is large. As a kernel we take the scaled Gaussian

$$K(x, y) = \exp(-(x - y)^2/c^2) \text{ for all } x, y \in \mathbb{R}.$$

Th spatial discretization will simply consist of $m + 1$ equally spaced points

$$x_j = -1 + j * h,\ 0 \leq j \leq m \text{ with } h = \frac{2}{m}.$$

Since the outermost points carry the boundary conditions, the terminology of the previous sections is satisfied by defining

$$\begin{array}{rcl} X & := & \{x_1, \ldots, x_{m-1}\} \subset (-1, 1) \\ \mu_j(f) & := & \delta_{x_j}(f) = f(x_j),\ 0 \leq j \leq m \\ \Lambda & := & \{\mu_0, \mu_m\}. \end{array}$$

We then form the kernel matrix (12) as

$$\mathbf{A} := E_M^y (E_M^x)^T K(x, y) = \big(\exp(-(x_j - x_k)^2/c^2)\big)_{0 \leq j,k \leq m}$$

and pick a scale c such that the condition is still reasonable. Then we have

$$E_M^y K(y, z) = (K(z, x_1), \ldots, K(z, x_{m-1}))^T$$

and get a full Lagrange basis $v_0, \ldots, v_m$ via (13) with

$$(v_0(z), \ldots, v_m(z))^T = (K(z, x_0), \ldots, K(z, x_m))^T \mathbf{A}^{-1} \tag{27}$$

satisfying the Lagrange conditions

$$v_j(x_k) = \delta_{jk},\ 0 \leq j, k \leq m.$$

Note that the zero boundary conditions at $x_0 = -1$ and $x_m = 1$ are satisfied if we use the span of the functions $v_1, \ldots, v_{m-1}$ as our trial space. Thus we define

$$V_X(z) := (v_1(z), \ldots, v_{m-1}(z))^T.$$

Since we have zero boundary conditions, we do not need the V_Λ functions v_0, v_m at all.

The trial space is then spanned by functions

$$w(z, t) := \mathbf{w}(t)^T V_X(z) = \sum_{j=1}^{m-1} w_j(t) v_j(z)$$

with the vector

$$\mathbf{w}(t) := (w_1(t), \dots, w_{m-1}(t))^T$$

of unknown time–dependent coefficients which have the interpolation property

$$w(x_k, t) = w_k(t),\ 1 \leq k \leq m-1$$

and satisfy the boundary conditions

$$w(-1, t) = w(1, t) = 0$$

by construction. The derivatives of our trial functions can be obtained by taking derivatives of (27), but we need everything only on the points $x_1, \dots, x_{m-1}$ of X. This can be done using the above formulae and simple matrix operations based on (18) and (27) to result in $(m-1) \times (m-1)$ matrices

$$\begin{array}{rcl} \mathbf{V}_x & := & \left(v_j'(x_k)\right)_{1 \leq k,j \leq m-1} \\ \mathbf{V}_{xx} & := & \left(v_j''(x_k)\right)_{1 \leq k,j \leq m-1} \end{array}$$

where k is the row index. This yields

$$\begin{array}{rcl} \mathbf{w}_x(t) & := & (w_x(x_1, t), \dots, w_x(x_{m-1}, t))^T \\ & = & \left(\displaystyle\sum_{j=1}^{m-1} w_j(t) v_j'(x_k)\right)_{1 \leq k \leq m-1} \\ & = & \mathbf{V}_x \mathbf{w}(t) \end{array}$$

and similarly

$$\mathbf{w}_{xx}(t) = \mathbf{V}_{xx} \mathbf{w}(t).$$

The initial data are interpolated by setting

$$w_j(0) := 1 - x_j^2,\ 1 \leq j \leq m-1$$

and then we can start the Method of Lines defined like in (20) as

$$w_k'(t) \;=\; -w_k(t) \left(\sum_{j=1}^{m-1} w_j(t) v_j'(x_k)\right) + \nu \sum_{j=1}^{m-1} w_j(t) v_j''(x_k)$$

for $1 \leq k \leq m-1$. In MATLAB notation this boils down to

$$\mathbf{w}'(t) = -\mathbf{w}(t). * \mathbf{V}_x * \mathbf{w}(t) + \nu \mathbf{V}_{xx} * \mathbf{w}(t)$$

and is a simple ODE system with a quadratic right–hand side. Note that the matrices $\mathbf{V}_x$ and $\mathbf{V}_{xx}$ are calculated before the ODE system solver is started.

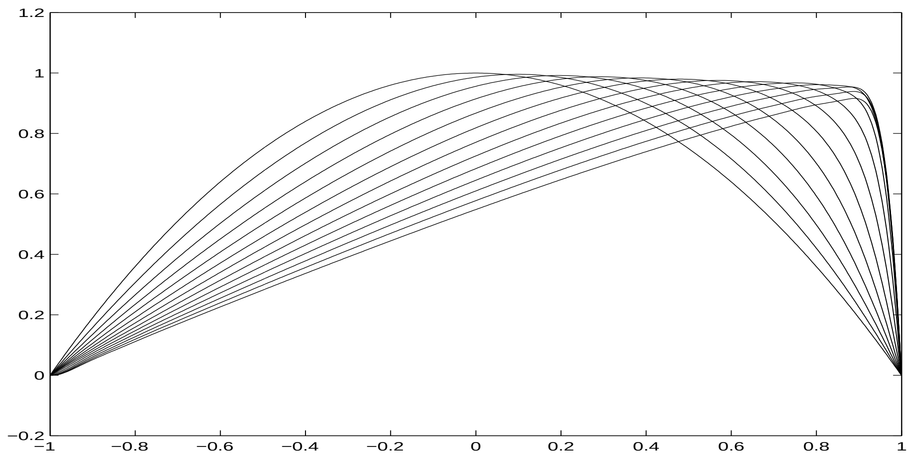

Figure 1: $m = 30,\ Re = 50$

For nontrivial future applications, these matrices should be sparse because they contain derivative stencils of a Lagrange basis.

Figures 1 and 2 show two examples. The ODE system was solved by the MATLAB© routine `ode23s`. The interpolants to the solutions were plotted for time intervals of length 0.1 between 0 and 1.2. Further experiments showed that m should grow proportional to Re for large Reynolds numbers. To ensure a stable computation without additional tricks like preconditioning, the scale of the kernel should be chosen proportional to $\nu = 1/Re$ or $1/m$. A MATLAB© program is available through the homepage of the author. Comparable results were obtained by Hon and Mao [4] using time–stepping, and they were improved by adaptive changes of the discretization close to the shock.

Acknowledgments

The results were partially obtained at City University of Hong Kong when visiting in March 2008. Special thanks go to Greg Fasshauer of Illinois Institute of Technology, Benny Y.C. Hon of City University of Hong Kong, and Leevan Ling of Hong Kong Baptist University for several useful discussions. In particular, Greg's paper [3] was very influential.

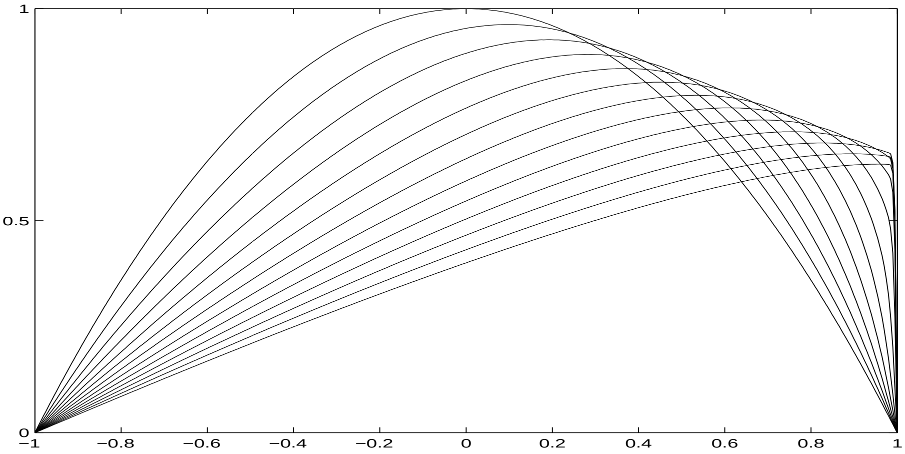

Figure 2: $m = 500,\ Re = 700$

References

[1] T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, special issue*, 139:3–47, 1996.

[2] St. De Marchi and R. Schaback. Stability of kernel-based interpolation. Preprint Göttingen, 2008.

[3] G Fasshauer. RBF collocation methods and pseudospectral methods. Preprint, Illinois Institute of Technology, 2005.

[4] Y.C. Hon and X.Z. Mao. An efficient numerical scheme for Burgers' equations. *Appl. Math. Comp.*, 95:37–50, 1998.

[5] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - II: Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19:147–161, 1990.

[6] F. J. Narcowich, J. D. Ward, and H. Wendland. Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Mathematics of Computation*, 74:743–763, 2005.

[7] F.J. Narcowich and J.D. Ward. Generalized Hermite interpolation via matrix–valued conditionally positive definite functions. *Mathematics of Computation*, 63(208):661–687, 1994.

[8] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.

[9] H. Wendland. *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, 2005.

[10] H. Werner and H. Arndt. *Gewöhnliche Differentialgleichungen.* Hochschultexte. Springer, 1985.

[11] Z. Wu. Hermite–Birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8/2:1–10, 1992.