Fast tensor-product solvers: Part II: Spectral discretization in space and time

Yvon Maday* , and Einar M. Rønquist†

July 23, 2007

Abstract

We consider the numerical solution of an unsteady convection-diffusion equation using high order polynomial approximations both in space and time. General boundary conditions and initial conditions can be imposed. The method is fully implicit and enjoys exponential convergence in time and space for analytic solutions. This is confirmed by numerical experiments (in one space dimension) using a spectral element approach in time and a pure spectral method in space. A fast tensor-product solver has been developed to solve the coupled system of algebraic equations for the $\mathcal{O}(N^d)$ unknown nodal values within a single space-time spectral element. This solver has a fixed complexity of $\mathcal{O}(N^{d+1})$ floating point operations and a memory requirement of $\mathcal{O}(N^d)$ floating point numbers. An alternative solution method, allowing for a parallel implementation and more easily extendable to more complex problems is sketched out at the end of the paper.

1 Introduction

There have been a number of research efforts in the past with the goal of using high order approximations both in time and space for the numerical solution of time-dependent partial differential equations. The first contribution we are aware of was done more than 25 years ago [7, 8], while the most recent work we have found on this topic is within the last few months [13, 9]. While the use of high order (spectral) approximations in space is currently a well developed field, the same is not true for the time direction. Encouraging progress has been made over the past couple of decades [11, 12, 1, 10, 13, 9], however, a significant advance, appropriate for solving real applications, is still missing.

From a pure approximation point of view, the motivation for using high order approximations in time is at least as strong as for using such methods in space. Indeed, for many problems the regularity in time is at least as high as the regularity in space. For parabolic problems, there may exist singularities at $t = 0$, however, this situation is simpler to analyze than the various spatial singularities resulting from the presence of vertices, edges and faces. Furthermore, while the solution may exhibit sharp boundary layers in space, this may often not be the case in the time direction. Of course, there may be problems where the regularity of the solution with respect to time can be expected to be low; nonetheless, the motivation for using high order approximations in time is quite significant.

*Université Pierre et Marie Curie-Paris6, UMR 7598, Laboratoire J.-L. Lions, Paris, F-75005 France and Division of Applied Mathematics, Brown University, 182 George Street, Providence, RI 02912, USA

†Norwegian University of Science and Technology, Department of Mathematical Sciences, N-7491 Trondheim, Norway, fax: +47 73 59 35 24, phone: +47 73 59 35 47, email: ronquist@math.ntnu.no

What has prevented the adoption of high order methods in time has fundamentally been related to computational complexity. In general, such an approach will result in implicit systems of algebraic equations. If the number of degrees-of-freedom in space is already large, multiplying this number with a large factor in the time direction poses significant challenges in terms of solver technology. Unless the computational cost for the solvers is close to optimal, i.e., the cost is approximately proportional with respect to the number of degrees-of-freedom (a condition partially alleviated when high order approximations are used), the method will not be competitive with a more conventional time-advancing discretization based on low order finite difference approximations where the solution is advanced one small time step at a time.

There are a number of reasons for revisiting the approach used for the temporal treatment. First, the computers are now becoming quite powerful with many (thousands of) processors and with a large aggregate memory. Second, there have recently been several alternative methods proposed to exploit more parallelism in the time direction [1, 6]. Third, a number of fast solution algorithms are currently available, and some of these may be exploited with respect to the time direction [2].

In this paper, we first consider the numerical discretization of the time-dependent convection-diffusion equation in one space dimension based on high order polynomial approximations in time and space. For simplicity, we assume a polynomial approximation of degree N in each direction. We then develop a fast tensor-product solver for the resulting system of algebraic equations; this solver can be classified as a direct solver. We also present numerical results demonstrating the convergence properties of the method.

We will discuss the extension to multiple space dimensions and nonlinear problems in a future paper. The focus of this paper is the use of spectral elements in time, and exploring the possibilities of exploiting the tensor-product feature between space and time in order to construct fast solution algorithms.

2 The heat equation

For the sake of simplicity, let us consider first the one-dimensional heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + f(x,t) \quad \text{in} \quad \omega_x = (0, L), \tag{1}$$

subject to the homogeneous boundary conditions

$$\begin{aligned} u(x=0,t) &= 0, \quad \forall\, t \in \omega_t = [0,T], \\ u(x=L,t) &= 0, \quad \forall\, t \in \omega_t, \end{aligned} \tag{2}$$

and subject to the initial condition

$$u(x, t=0) = s(x). \tag{3}$$

Here, $u(x,t)$ is the solution (the temperature), κ is the thermal diffusivity, and $f(x,t)$ is a thermal heat source that depends on space and time. We assume that the thermal diffusivity, κ, is a constant.

A common approach when discretizing this equation is to discretize space and time separately. In particular, when applying a spectral discretization, it is common to only consider a spectral discretization in space; discretization in time is typically done using a multi-stage method or a multi-step method [4, 5].

2.1 Weak formulation

We consider here a spectral discretization in both space and time. The discretization will be based upon a weak space time formulation of the problem: Find $u \in X$ such that

$$\int_0^T \int_0^L \frac{\partial u}{\partial t}\, v\, dx\, dt = -\kappa \int_0^T \int_0^L \frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\, dx\, dt + \int_0^T \int_0^L f\, v\, dx\, dt, \quad \forall v \in X^0. \tag{4}$$

with

$$X = H^1(\omega_t, L^2(\omega_x)) \cap L^2(\omega_t, H_0^1(\omega_x))$$

and e.g.

$$X^0 = \{w \in X, w(x,0) = 0\},$$

together with the initial condition $u(x,0) = s(x)$ satisfied over ω_x.

This is similar to a standard variational procedure, except for the fact that we here integrate the governing equation over both space and time.

Next, we consider the (space-time) domain

$$\Omega = (0,L) \times (0,T) = \omega_x \times \omega_t$$

as an affine mapping of the reference domain

$$\Phi : \widehat{\Omega} = (-1,1) \times (-1,1) \longrightarrow \Omega.$$

In the following, we denote a point in the reference domain as (ξ, η), and use the convention that a "hat" refers to a quantity defined in terms of the reference variables, e.g.,

$$\hat{u} = u \circ \Phi.$$

The weak formulation can then be expressed in the reference variables as: Find $\hat{u} \in \widehat{X}$ such that

$$\int_{\hat{\Omega}} \frac{\partial \hat{u}}{\partial \eta}\, \hat{v}\, d\xi\, d\eta + \gamma_1 \int_{\hat{\Omega}} \frac{\partial \hat{u}}{\partial \xi}\frac{\partial \hat{v}}{\partial \xi}\, d\xi\, d\eta = \gamma_2 \int_{\hat{\Omega}} \hat{f}\, \hat{v}\, d\xi\, d\eta, \quad \forall \hat{v} \in \widehat{X}^0, \tag{5}$$

where

$$\widehat{X} = \{\hat{w}, \quad \hat{w} \circ \Phi^{-1} \in X\}, \qquad \widehat{X}^0 = \{\hat{w}, \quad \hat{w} \circ \Phi^{-1} \in X^0\},$$

and

$$\gamma_1 = 2\,\kappa\,\frac{T}{L^2}, \qquad \gamma_2 = \frac{T}{2}.$$

2.2 A spectral discretization in space and time

We now consider a spectral discretization of the continuous problem (5) in space and time. In particular, we consider a discretization based on high-order polynomials. We define our discrete spaces as

$$\widehat{X}_N = \widehat{X} \cap \mathbb{P}_N(\hat{\Omega}), \qquad \widehat{X}_N^0 = \widehat{X}^0 \cap \mathbb{P}_N(\hat{\Omega}),$$

where $\mathbb{P}_N(\widehat{\Omega})$ is the space of all functions which are polynomials of degree less than or equal to N in each direction ξ and η. The Galerkin approximation of (5) is : Find $\hat{u}_N \in \widehat{X}_N$ such that

$$\int_{\hat{\Omega}} \frac{\partial \hat{u}_N}{\partial \eta}\, \hat{v}_N \, d\xi \, d\eta + \gamma_1 \int_{\hat{\Omega}} \frac{\partial \hat{u}_N}{\partial \xi} \frac{\partial \hat{v}_N}{\partial \xi} \, d\xi \, d\eta = \gamma_2 \int_{\hat{\Omega}} \hat{f}\, \hat{v}_N \, d\xi \, d\eta, \quad \forall \hat{v}_N \in \widehat{X}_N^0, \tag{6}$$

In order to represent an element in the linear space $\widehat{X}_N^0$, we choose a tensor-product, Lagrangian interpolant basis through the Gauss-Lobatto Legendre (GLL) points, $\xi_i, i = 0, \ldots, N$. Specifically, $\forall \hat{v}_N \in \widehat{X}_N^0$, we write

$$\hat{v}_N(\xi, \eta) = \sum_{i=0}^{N} \sum_{j=0}^{N} v_{ij}\, \ell_i(\xi)\, \ell_j(\eta), \tag{7}$$

where $v_{ij} = \hat{v}_N(\xi_i, \xi_j) = v_N(x(\xi_i), y(\xi_j))$. The Lagrangian interpolants satisfy the properties

$$\begin{aligned} \ell_k(\xi) &\in \mathbb{P}_N(\hat{\omega}), \quad \forall\, k \in \{0, \ldots, N\}, \\ \ell_k(\xi_l) &= \delta_{kl}, \qquad \forall\, k, l \in \{0, \ldots, N\}^2. \end{aligned} \tag{8}$$

Note that some of the basis coefficients in (7) are explicitly imposed by the boundary conditions and the intial condition. In particular, $v_{i0} = 0$ for $i = 0, ..., N$, $v_{0j} = 0$ for $j = 0, ..., N$, and $v_{Nj} = 0$ for $j = 0, ..., N$.

The discrete solution $\hat{u}_N$ will be an element of $\widehat{X}_N$. Similar to (7), we express the discrete solution in the reference variables as

$$\hat{u}_N(\xi, \eta) = \sum_{m=0}^{N} \sum_{n=0}^{N} u_{mn}\, \ell_m(\xi)\, \ell_n(\eta). \tag{9}$$

The following basis coefficients are given by the boundary conditions: $u_{0n} = 0$ for $n = 0, \ldots, N$, and $u_{Nn} = 0$ for $n = 0, \ldots, N$. The initial condition is given by setting $u_{m0} = s_m = s(x(\xi_m))$ for $m = 0, \ldots, N$. Hence, the initial condition is approximated by the Nth order polynomial interpolant through the GLL points. Approximating the given heat source f also as a high order interpolant, the discrete problem can be expressed as: Find $\hat{u}_N \in \widehat{X}_N$ such that

$$\begin{aligned} &\int_{-1}^{1} \int_{-1}^{1} [\, \sum_{m=0}^{N} \sum_{n=0}^{N} u_{mn}\, \ell_m(\xi)\, \ell_n'(\eta)\,]\, \ell_i(\xi)\, \ell_j(\eta)\, d\xi\, d\eta + \\ \gamma_1 &\int_{-1}^{1} \int_{-1}^{1} [\, \sum_{m=0}^{N} \sum_{n=0}^{N} u_{mn}\, \ell_m'(\xi)\, \ell_n(\eta)\,]\, \ell_i'(\xi)\, \ell_j(\eta)\, d\xi\, d\eta = \\ \gamma_2 &\int_{-1}^{1} \int_{-1}^{1} [\, \sum_{m=0}^{N} \sum_{n=0}^{N} f_{mn}\, \ell_m(\xi)\, \ell_n(\eta)\,]\, \ell_i(\xi)\, \ell_j(\eta)\, d\xi\, d\eta \ , \\ &\qquad \forall i \in \{1, \ldots, N-1\}, \quad \forall j \in \{1, \ldots, N\}. \end{aligned} \tag{10}$$

We have here chosen $\hat{v}_N$ to be $\ell_i(\xi)\, \ell_j(\eta)$ for all $i \in \{1, \ldots, N-1\}$, and for all $j \in \{1, \ldots, N\}$. This will give us enough equations to determine the unknown basis coefficients u_{mn} for all $m \in \{1, \ldots, N-1\}$, and for all $n \in \{1, \ldots, N\}$.

If we introduce the notation that, $\forall v, w \in L^2(-1, 1)$,

$$(v, w) = \int_{-1}^{1} v(\xi)\, w(\xi)\, d\xi, \tag{11}$$

we can also write the discrete problem (10) as: Find $\hat{u}_N \in \widehat{X}_N$ such that

$$\begin{aligned} \sum_{m=0}^{N}\sum_{n=0}^{N} \; (\ell_i, \ell_m)\; (\ell_j, \ell_n')\; u_{mn} + \gamma_1 \sum_{m=0}^{N}\sum_{n=0}^{N} \; (\ell_i', \ell_m')\; (\ell_j, \ell_n)\; u_{mn} = \\ \gamma_2 \sum_{m=0}^{N}\sum_{n=0}^{N} \; (\ell_i, \ell_m)\; (\ell_j, \ell_n)\, f_{mn} \\ \forall i \in \{1, \ldots, N-1\}, \quad \forall j \in \{1, \ldots, N\}. \end{aligned} \tag{12}$$

The final step in the discretization process is to evaluate (or approximate) the L^2 innerproducts $(\cdot,\cdot)$ with numerical quadrature. A common and natural choice is to use Gauss-Lobatto Legendre (GLL) quadrature, e.g.,

$$(v, w) = \int_{-1}^{1} v(\xi)\, w(\xi)\, d\xi \approx \sum_{\alpha=0}^{N} \rho_\alpha\, v(\xi_\alpha)\, w(\xi_\alpha) \equiv (v, w)_G. \tag{13}$$

Here, $\{\rho_\alpha\}_{\alpha=0}^{N}$ and $\{\xi_\alpha\}_{\alpha=0}^{N}$ are the GLL weights and points, respectively, and $(\cdot,\cdot)_G$ represents the discrete L^2 innerproduct. We recall that the GLL quadrature is exact for all integrands in $\mathbb{P}_{2N-1}(-1, 1)$.

We now introduce the one-dimensional mass matrix $\underline{\widehat{B}}$ with elements

$$\widehat{B}_{kl} = (\ell_k, \ell_l)_G \;=\; \rho_k\, \delta_{kl}, \tag{14}$$

the one-dimensional convection matrix $\underline{\widehat{C}}$ with elements

$$\widehat{C}_{kl} = (\ell_k, \ell_l')_G \;=\; \rho_k\, D_{kl}, \tag{15}$$

and the one-dimensional stiffness matrix $\underline{\widehat{A}}$ with elements

$$\widehat{A}_{kl} = (\ell_k', \ell_l')_G \;=\; \sum_{\alpha=0}^{N} \rho_\alpha\, D_{\alpha k}\, D_{\alpha l}. \tag{16}$$

Here, we have introduced the derivative matrix

$$D_{\alpha k} \equiv \ell_k'(\xi_\alpha). \tag{17}$$

We remark that the mass matrix is diagonal, while the discrete innerproducts defining the convection matrix and the stiffness matrix are, in fact, exact.

The discrete problem, including quadrature, can thus be expressed as: Find $\hat{u}_N \in \widehat{X}_N$ such that, $\forall i \in \{1, \ldots, N-1\}$, $\forall j \in \{1, \ldots, N\}$,

$$\begin{aligned} \sum_{m=1}^{N-1}\sum_{n=1}^{N} \left(\widehat{B}_{im}\, \widehat{C}_{jn} + \gamma_1\, \widehat{A}_{im}\, \widehat{B}_{jn} \right) u_{mn} = \\ \gamma_2 \sum_{m=1}^{N-1}\sum_{n=1}^{N} \widehat{B}_{im}\, \widehat{B}_{jn}\, f_{mn} \;-\; \sum_{m=1}^{N-1} \delta_{im}\, \widehat{C}_{j0}\, s_m. \end{aligned} \tag{18}$$

We remark that the last term on the right hand side represents the imposition of the initial condition via high order polynomial interpolation, with $s_m = s(x(\xi_m))$.

Next, we define

$$g_{ij} = \gamma_2 \sum_{m=1}^{N-1} \sum_{n=1}^{N} \widehat{B}_{im}\, \widehat{B}_{jn}\, f_{mn} \; - \; \sum_{m=1}^{N-1} \delta_{im}\, \widehat{C}_{j0}\, s_m. \tag{19}$$

$\forall i \in \{1, \ldots, N-1\}$, $\forall j \in \{1, \ldots, N\}$; this expression incorporates all the known data for this problem, including the initial condition.

Once we have computed g_{ij}, we can write the algebraic problem as: Find $\hat{u}_N \in \widehat{X}_N$ such that, $\forall i \in \{1, \ldots, N-1\}$, $\forall j \in \{1, \ldots, N\}$,

$$\sum_{m=1}^{N-1} \sum_{n=1}^{N} \left(\; \widehat{B}_{im}\, \widehat{C}_{jn} \; + \gamma_1\, \widehat{A}_{im}\, \widehat{B}_{jn} \; \right) u_{mn} = g_{ij}. \tag{20}$$

Instead of representing the unknown basis coefficients u_{mn} as a multidimensional array (i.e., with one, local index for each direction ξ and η, we can, of course, also represent the unknown basis coefficients as a single, long vector $\underline{u}$; the associated local-to-global mapping is done by using the convention that the first index (i.e., along ξ) "runs faster" than the second index (i.e., along η). Note that $\underline{u} = \mathbb{R}^{(N-1)N}$.

We can then write the system (20) using tensor-product forms: Find $\underline{u} = \mathbb{R}^{(N-1)N}$ such that

$$\left(\; \underline{\widehat{C}}_2 \otimes \underline{\widehat{B}}_1 \; + \; \gamma_1\, \underline{\widehat{B}}_2 \otimes \underline{\widehat{A}}_1 \; \right) \underline{u} \; = \; \underline{g}. \tag{21}$$

Here, $\underline{g}$ is the global vector corresponding to the multi-dimensional array g_{ij}, $\underline{\widehat{C}}$ is the one-dimensional convection operator defined in (15), $\underline{\widehat{A}}$ is the one-dimensional diffusion operator defined in (16), and $\underline{\widehat{B}}$ is the one-dimensional mass matrix defined in (14). Note that we use subscript 1 and 2 to differentiate between the directions ξ and η, respectively. For example, $\underline{\widehat{B}}_1$ and $\underline{\widehat{A}}_1$ are square matrices of size $(N-1) \times (N-1)$, corresponding to Dirichlet boundary conditions at the end points $\xi = \pm 1$. In contrast, $\underline{\widehat{B}}_2$ and $\underline{\widehat{C}}_2$ are square matrices of size $N \times N$, corresponding to Dirichlet boundary conditions at $\eta = -1$.

Our objective now is to develop a fast tensor-product solver for (21). To this end, we need to diagonalize the operators. First, we solve the generalized eigenvalue problem

$$\underline{\widehat{A}}_1\, \underline{Q}_1 = \underline{\widehat{B}}_1\, \underline{Q}_1\, \underline{\Lambda}_1$$

associated with direction 1, i.e., ξ (or space). Here, $\underline{Q}_1$ is the eigenvector matrix, and $\underline{\Lambda}_1$ is a diagonal matrix containing the eigenvalues. Since $\underline{\widehat{A}}_1$ and $\underline{\widehat{B}}_1$ are both symmetric and positive definite, all the eigenvalues are real and positive, and the eigenvectors are ortogonal with respect to the mass matrix $\underline{\widehat{B}}_1$. In fact, if we normalize the eigenvectors so that

$$\underline{Q}_1^T\, \underline{\widehat{B}}_1\, \underline{Q}_1 = \underline{I}_1,$$

we see that

$$\underline{Q}_1^T\, \underline{\widehat{A}}_1\, \underline{Q}_1 = \underline{\Lambda}_1.$$

Hence, we can express $\underline{\widehat{A}}_1$ and $\underline{\widehat{B}}_1$ in (21) as

$$\begin{aligned} \underline{\widehat{A}}_1 &= \underline{Q}_1^{-T}\, \underline{\Lambda}_1\, \underline{Q}_1^{-1}, \\ \underline{\widehat{B}}_1 &= \underline{Q}_1^{-T}\, \underline{Q}_1^{-1}. \end{aligned} \tag{22}$$

Second, we would like to diagonalize the convection operator $\underline{\widehat{C}}_2$ with respect to the mass matrix $\underline{\widehat{B}}_2$, i.e., we consider the eigenvalue problem

$$\underline{\widehat{C}}_2 \, \underline{S}_2 = \underline{\widehat{B}}_2 \, \underline{S}_2 \, \underline{\Lambda}_2, \tag{23}$$

where $\underline{S}$ is a matrix containing the eigenvectors as columns. Now, since Dirichlet conditions are only specified at $\eta = -1$, the convection operator $\underline{\widehat{C}}_2$ is *not* skew-symmetric and therefore not normal. Hence, at a fundamental level, we do not know whether we are able to diagonalize $\underline{\widehat{C}}_2$. Even if we can diagonalize $\underline{\widehat{C}}_2$, the eigenvectors will certainly not be orthogonal with respect to the mass matrix, and this fact can cause numerical difficulties. However, as discussed in [3], we expect to be able to diagonalize $\underline{\widehat{C}}_2$ for $N \leq 32$ without any significant round-off errors. This limitation on N should not pose any major constraint since we do not (at least currently) forsee using a too high value of N in the time direction.

In the following, we therefore assume that we are in this regime, i.e., $N \leq 32$. The eigenvalues and eigenvectors are complex, and we can express $\underline{\widehat{C}}_2$ and $\underline{\widehat{B}}_2$ in (21) as

$$\begin{aligned} \underline{\widehat{C}}_2 &= \underline{\widehat{B}}_2 \, \underline{S}_2 \, \underline{\Lambda}_2 \, \underline{S}_2^{-1}, \\ \underline{\widehat{B}}_2 &= \underline{\widehat{B}}_2 \, \underline{S}_2 \, \underline{S}_2^{-1}. \end{aligned} \tag{24}$$

Substituting for $\underline{\widehat{B}}_1$, $\underline{\widehat{A}}_1$, $\underline{\widehat{C}}_2$ and $\underline{\widehat{B}}_2$ in (21) and using the usual rules for tensor products, the system (21) can be expressed as

$$(\ \underline{\widehat{B}}_2 \, \underline{S}_2 \otimes \underline{Q}_1^{-T}\ )(\ \underline{\Lambda}_2 \otimes \underline{I}_1 \ + \ \gamma_1 \, \underline{I}_2 \otimes \underline{\Lambda}_1\ )(\ \underline{S}_2^{-1} \otimes \underline{Q}_1^{-1}\ )\ \underline{u} \ = \ \underline{g}. \tag{25}$$

We thus see that that the solution $\underline{u}$ can be expressed explicitly as

$$\underline{u} \ = \ (\ \underline{S}_2 \otimes \underline{Q}_1\ )(\ \underline{\Lambda}_2 \otimes \underline{I}_1 \ + \ \gamma_1 \, \underline{I}_2 \otimes \underline{\Lambda}_1\ )^{-1}(\ \underline{S}_2^{-1} \, \underline{\widehat{B}}_2^{-1} \ \otimes \underline{Q}_1^T\ )\ \underline{g}\ . \tag{26}$$

At this point, several comments are in order. First, we have demonstrated that it is possible to develop a fast, direct solver for the problem at hand, at least for a polynomial approximation $N \leq 32$ in the time direction. Second, the expression (26) will not be used directly in its current form. Indeed, we will revert back to a multi-dimensional array representation, and perform the necessary operations using matrix-matrix products. This will give an overall computational complexity of $\mathcal{O}(N^3)$. In contrast, if we explicitly form the global matrices, the computational complexity would be $\mathcal{O}(N^4)$. In higher space dimensions, the difference between these two approaches will be much higher. For example, it is possible to extend the methodology presented here to two and three space dimensions when the domain is tensorized. With three space dimensions and one time dimension, the complexity of the solver will be $\mathcal{O}(N^5)$ for $\mathcal{O}(N^4)$ unknowns (assuming for simplicity that we use the same polynomial degree in each spatial direction as well as in the time direction for each spectral element in time). Finally, note that the solution we arrive at represents a fully implicit, high-order polynomial approximation (high order in both space and time).

3 The unsteady convection-diffusion equation

We now consider the one-dimensional, unsteady convection-diffusion equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \kappa \, \frac{\partial^2 u}{\partial x^2} + f(x,t) \quad \text{in } \ \omega_x = (0, L), \tag{27}$$

subject to the same boundary conditions and initial condition as before.

In terms of discretization, we follow an identical approach as for the heat equation. We thus arrive at the discrete problem: Find $\hat{u}_N \in \widehat{X}_N$ such that

$$(\underline{\widehat{C}}_2 \otimes \underline{\widehat{B}}_1 \; + \; \beta \, \underline{\widehat{B}}_2 \otimes \underline{\widehat{C}}_1 \; + \; \gamma_1 \, \underline{\widehat{B}}_2 \otimes \underline{\widehat{A}}_1 \;) \; \underline{u} \; = \; \underline{g}. \tag{28}$$

where $\underline{\widehat{C}}_1$ is the usual convection operator in space, $\underline{\widehat{B}}_2$ is the mass matrix associated with the time direction, and the constant $\beta = \frac{T}{L}$. Note that $\underline{\widehat{C}}_1$ is a matrix of size $(N-1) \times (N-1)$, while $\underline{\widehat{C}}_2$ is a matrix of size $N \times N$. Furthermore, $\underline{\widehat{C}}_1$ is a skew-symmetric matrix (for the given boundary conditions), while $\underline{\widehat{C}}_2$ is not.

In the following, our objective is to develop a fast, tensor product solver for (28); we now present two alternatives for achieving this.

3.1 Alternative 1

Similar to the unsteady heat equation, we again rely on being able to solve the eigenvalue problem (23) associated with the time direction. We recall that a fundamental issue here is the fact that the matrix $\underline{\widehat{C}}_2$ is not normal, and that the eigenvalue problem (23) may not have a solution. However, as discussed in [3], the eigenvalues and eigenvectors can be computed without any significant round-off errors for $N \leq 32$.

For the space direction, we propose to solve the following eigenvalue problem:

$$(\beta \, \underline{\widehat{C}}_1 + \gamma_1 \, \underline{\widehat{A}}_1) \; \underline{S}_1 = \underline{\widehat{B}}_1 \, \underline{S}_1 \, \underline{\Lambda}_1. \tag{29}$$

This allows us to write

$$\begin{aligned} (\beta \, \underline{\widehat{C}}_1 + \gamma_1 \, \underline{\widehat{A}}_1) &= \underline{\widehat{B}}_1 \, \underline{S}_1 \, \underline{\Lambda}_1 \, \underline{S}_1^{-1}, \\ \underline{\widehat{B}}_1 &= \underline{\widehat{B}}_1 \, \underline{S}_1 \, \underline{S}_1^{-1}. \end{aligned} \tag{30}$$

Using the above information, the system (28) can be then expressed as

$$(\underline{\widehat{B}}_2 \, \underline{S}_2 \otimes \underline{\widehat{B}}_1 \, \underline{S}_1 \;)(\; \underline{\Lambda}_2 \otimes \underline{I}_1 \; + \; \underline{I}_2 \otimes \underline{\Lambda}_1 \;)(\; \underline{S}_2^{-1} \otimes \underline{S}_1^{-1} \;) \; \underline{u} \; = \; \underline{g}. \tag{31}$$

We thus see that that the solution $\underline{u}$ can be expressed explicitly as

$$\underline{u} \; = \; (\; \underline{S}_2 \otimes \underline{S}_1 \;)(\; \underline{\Lambda}_2 \otimes \underline{I}_1 \; + \; \underline{I}_2 \otimes \underline{\Lambda}_1 \;)^{-1} (\; \underline{S}_2^{-1} \, \underline{\widehat{B}}_2^{-1} \; \otimes \underline{S}_1^{-1} \, \underline{\widehat{B}}_1^{-1} \;) \; \underline{g}. \tag{32}$$

For efficiency reasons, a local numbering scheme should be used to represent the right hand side as well as the solution.

This alternative thus relies on being able to solve the eigenvalue problem (29) in addition to the eigenvalue problem (23). However, since the discrete convection-diffusion operator in (29) is not normal either, (29) may not have a solution even if we have not experienced any numerical difficulty so far; see the numerical experiments reported in the next section.

3.2 Alternative 2

This alternative, which we shall develop further in Part III of the series, only relies on diagonalizing the first order operator in the time direction. Combining (24) and (28), and using the rules for tensor-products, we end up with

$$(\; \underline{\widehat{B}}_2 \, \underline{S}_2 \otimes \underline{I}_1 \;)(\; \underline{\Lambda}_2 \otimes \underline{\widehat{B}}_1 \; + \; \underline{I}_2 \otimes (\beta \underline{\widehat{C}}_1 + \gamma_1 \underline{\widehat{A}}_1)) (\; \underline{S}_2^{-1} \otimes \underline{I}_1 \;) \; \underline{u} \; = \; \underline{g}. \tag{33}$$

Introducing the variables

$$\underline{\tilde{u}} = (\ \underline{S}_2^{-1} \otimes \underline{I}_1\ )\ \underline{u}, \tag{34}$$

$$\underline{\tilde{g}} = (\ \underline{S}_2^{-1} \underline{\widehat{B}}_2^{-1} \otimes \underline{I}_1\ )\ \underline{g}, \tag{35}$$

we can express (33) as

$$(\ \underline{\Lambda}_2 \otimes \underline{\widehat{B}}_1\ +\ \underline{I}_2 \otimes (\beta \underline{\widehat{C}}_1 + \gamma_1 \underline{\widehat{A}}_1))\ \underline{\tilde{u}} = \underline{\tilde{g}}. \tag{36}$$

Reverting back to the local numbering scheme, we can alternatively write (36) as

$$\sum_{m=1}^{N-1} [(\gamma_1\, \widehat{A}_1 + \beta\, \widehat{C}_1)_{im} + \lambda_j (\widehat{B}_1)_{im}]\, \tilde{u}_{mj} = \tilde{g}_{ij}, \qquad i = 1, \ldots N-1, \quad j = 1, \ldots, N. \tag{37}$$

For each value of j, we solve a steady convection-diffusion-type system. Each of these systems is of dimension $N-1$. In the linear case, the N systems we end up with can be solved completely independently. This is similar to the approach presented in [2], but here involving a non-symmetric operator instead of a symmetric one. Once the solution $\underline{\tilde{u}}$ has been computed, we can find $\underline{u}$ via the transformation (34); in practice, we compute this transformation using a local numbering scheme; see [2].

4 Numerical results

We consider here the unsteady convection-diffusion equation (27) in $\omega_x = (0, 1)$ (i.e., $L = 1$), with an exact solution

$$u(x, t) = \sin(\pi t)\, \sin(\pi x).$$

We apply the proposed discretization scheme and integrate the discrete equations until a final time $T_{\text{final}} = 40$ corresponding to 20 full periods in time. The time interval is split up into K equal subintervals of size T. The tensor-product solver (32) is then used to solve the system of algebraic equations associated with each space-time spectral element.

We first use a time step (or spectral element size in time) equal to $T = 2$, corresponding to one full period in time. The numerical solution corresponding to the last time interval is shown in Figure 1.

Note that, with this choice of time step ($T = 2$), the initial condition for each time interval corresponds to a homogeneous condition. This is by no means necessary. In Figure 2, we again show the numerical solution corresponding to the last time interval, but now choosing a smaller time step $T = 4/3$ (independent of the period of the solution); with this choice it takes $K = 30$ time steps to arrive at the same final time $T_{\text{final}} = 40$.

Finally, we measure the error in the discrete L^2 norm as a function of the polynomial degree N for a fixed time step $T = 2$. The error is measured over the last time interval (i.e., the last space-time spectral element), and the convergence results are reported in Figure 3. As expected, exponential convergence is achieved. Note that the odd values of N do not give any reduction in the error; this is due to the fact that the solution is even with respect to both space and time over each whole period in time.

5 Conclusions

For a one dimensional example, we have presented a method for solving the linear convection-diffusion equation using high order polynomial approximations in both time and space. General boundary conditions and initial conditions can be imposed. The method is fully implicit and enjoys exponential convergence in time and space for analytic solutions. This is confirmed by numerical experiments using a spectral element approach in time and a pure spectral method in space (in one space dimension).

A fast tensor-product solver has been developed to solve the coupled system of algebraic equations for the $\mathcal{O}(N^d)$ unknown nodal values within a single space-time spectral element. For simplicity, we have here assumed a polynomial approximation of degree N in each direction. This solver has a fixed complexity of $\mathcal{O}(N^{d+1})$ floating point operations and a memory requirement of $\mathcal{O}(N^d)$ floating point numbers. The polynomial approximation in the time direction should be chosen to be less than or equal to 32 in order to avoid any round-off errors.

6 Future work

Future work will focus on solving more complex time-dependent partial differential equations (multiple space dimensions; linear and nonlinear equations). For example, the proposed methods could be interesting to use for the approximation of the time-dependent Maxwell equations.

In the case of the unsteady convection-diffusion equations, the extension to multiple space dimensions will be based on a similar approach as presented in [2] in the context of solving elliptic problems in partially deformed three-dimensional domains. In particular, we will exploit the tensor-product representation that always exists between the single time direction and the (generally deformed) physical domain. This will always allow us to achieve parallelism through the solution of independent steady convection-diffusion-type problems as in Alternative 2.

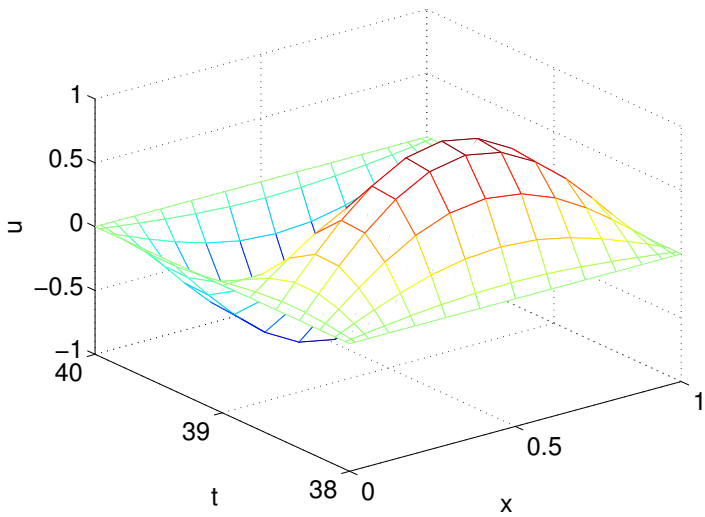

Figure 1: The numerical solution computed at the last time interval, i.e., for $38 \leq t \leq 40$ ($\Delta T = 2$). The polynomial degree in both the spatial and temporal direction is $N = 15$. The error in the discrete L^2 norm is $2.1 \cdot 10^{-9}$.

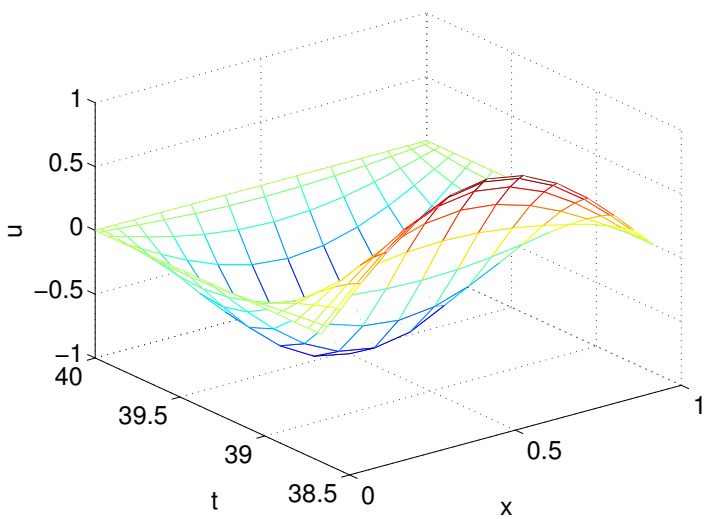

Figure 2: The numerical solution computed at the last time interval, i.e., for $38.6666 \leq t \leq 40$ ($T = 4/3$). The polynomial degree in both the spatial and temporal direction is $N = 15$. The error in the discrete L^2 norm is $3.6 \cdot 10^{-12}$.

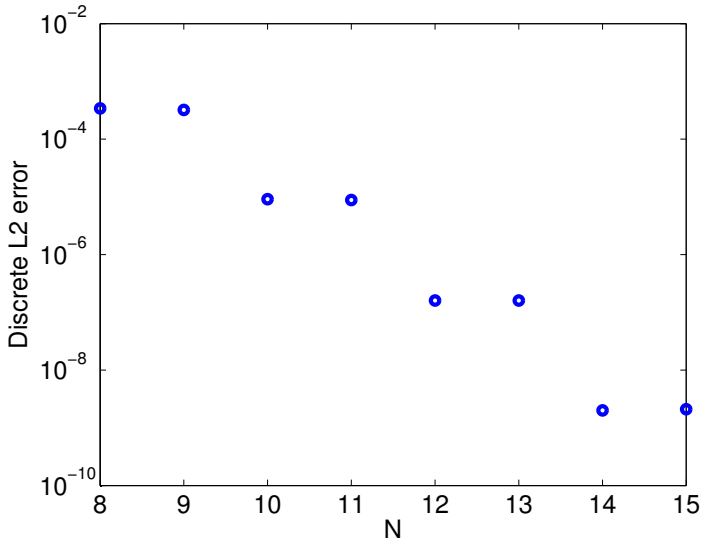

Figure 3: Convergence results for the convection-diffusion problem. The size of each spectral element in the time direction is $T = 2$.

References

[1] Bar-Yoseph, P., Moses, E., Zrahia, U., and Yarin, A. (1995). Space-time spectral element methods for one-dimensional nonlinear advection-diffusion problems. *J. Comput. Phys.*, **119**, 62-74.

[2] Bjøntegaard, T., Maday, Y., and Rønquist, E.M. (submitted, 2007). Fast tensor-product solvers. Part I: Partially deformed three-dimensional domains.

[3] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A. (2006). *Spectral Methods*, Springer-Verlag.

[4] C.W. Gear (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall.

[5] E. Hairer, S.P. Nørsett and G. Wanner (1993). *Solving Ordinary Differential Equations I*, Springer.

[6] Lions, J.-L., Maday, Y., and Turninici, G. (2001). Résolution d'EDP par un schéma en temps pararéel. *C.R. Acad Sci. Paris Sér. I Math.*, **332**, 661-668.

[7] Morchoisne, Y. (1981). Pseudo-spectral space-time calculations of incompressible viscous flows. *AIAA Paper*, No. 81-0109.

[8] Morchoisne, Y. (1984). Inhomogeneous flow calculations by spectral methods: Mono-domain and multi-domain techniques. In Voigt, R.G, Gottlieb, D., and Hussaini, M.Y. (eds.) *Spectral Methods for Partial Differential Equations*, SIAM, Philadelphia.

[9] Shen, J. and Wang, L.-L. (2007). Fourierization of the Legendre-Galerkin method and a new space-time spectral method. *Appl. Numer. Math.*, **57**, 710-720.

[10] Shukla, P. and Eswaran, V. (1996). A fully spectral method for hyperbolic equations. *Int. J. Numer. Meth. Engrg.*, **39**, 67-79.

[11] Tal-Ezer, H. (1986). Spectral methods in time for hyperbolic problems. *SIAM J. Numer. Anal.*, **23**, 11-26.

[12] Tal-Ezer, H. (1989). Spectral methods in time for parabolic problems. *SIAM J. Numer. Anal.*, **26**, 1-11.

[13] Tang, J.-G. and Ma, H.-P. (2007). A Legendre spectral method in time for first-order hyperbolic equations. *Appl. Numer. Math.*, **57**, 1-11.