SIAM J. Sci. Comput.
Vol. 36, No. 1, pp. B1–B13

TIME-DECOUPLED HIGH ORDER CONTINUOUS SPACE-TIME FINITE ELEMENT SCHEMES FOR THE HEAT EQUATION[*]

CAROLA KRUSE[†] AND SIMON SHAW[‡]

Abstract. In [*Comput. Methods Appl. Mech. Engrg.*, 190 (2001), pp. 6685–6708] Werder et al. demonstrated that time discretizations of the heat equation by a temporally discontinuous Galerkin finite element method could be decoupled by diagonalizing the temporal Gram matrices. In this article we propose a companion approach for the heat equation by using a *continuous* Galerkin time discretization. As a result, if piecewise polynomials of degree d are used as the trial functions in time and the spatial discretization produces systems of dimension M, then, after decoupling, d systems of size M need to be solved rather than a single system of size Md. These decoupled systems require complex arithmetic, as did the Werder et al. technique, but are amenable to parallel solutions on modern multicore architectures. We give numerical tests for temporal polynomial degrees up to six for three different model test problems, using both Galerkin and spectral element spatial discretizations, and show convergence and temporal superconvergence rates that accord with the bounds given by Aziz and Monk [*Math. Comp.*, 52 (1989), pp. 255–274]. We also interpret error as a function of computational time and see that our high order schemes may offer greater efficiency than the Crank–Nicolson method in terms of accuracy per unit of computational time—although in a multicore world, with highly tuned iterative solvers, one has to be cautious with such claims. We close with a speculation on the application of these ideas to the Navier–Stokes equations for incompressible fluids.

Key words. continuous Galerkin finite element method, spectral element method, space-time finite elements, high order methods

AMS subject classifications. 65M60, 15A21, 35K05

DOI. 10.1137/130914589

1. Introduction.

This short article discusses the design and implementation of moderately high order space-time continuous finite element approximations of parabolic problems. With an overdot denoting partial time differentiation we focus specifically on the canonical problem of finding $u\colon I \to \mathbb{R}$ such that

$$\varrho\dot{u} - \nabla\cdot\sigma\nabla u = f \qquad \text{in } \Omega\times I, \tag{1.1}$$

$$u = 0 \qquad \text{on } \Gamma_D\times\bar{I}, \tag{1.2}$$

$$\widehat{\boldsymbol{n}}\cdot\sigma\nabla u = g \qquad \text{on } \Gamma_N\times\bar{I}, \tag{1.3}$$

$$u = \breve{u} \qquad \text{in } \Omega\times\{0\}, \tag{1.4}$$

where $I := (0, T]$ is a finite time interval and $\Omega \subset \mathbb{R}^n$ for $n = 1, 2,$ or 3 is an open bounded connected domain with, for simplicity, a polygonal ($n = 2$) or polyhedral ($n = 3$) boundary $\partial\Omega$. Furthermore Γ_N and Γ_D form a time independent partition of $\partial\Omega$ with either allowed to be empty and $\widehat{\boldsymbol{n}}$ is the unit outward normal vector defined almost everywhere on Γ_N. The coefficients ρ and σ are assumed sufficiently smooth, time independent, and positive valued.

[*]Submitted to the journal's Computational Methods in Science and Engineering section March 27, 2013; accepted for publication (in revised form) October 16, 2013; published electronically January 2, 2014. This work was supported by the Engineering and Physical Sciences Research Council, EP/H011072/1.
http://www.siam.org/journals/sisc/36-1/91458.html

[†]Institut National de Recherche en Informatique et Automatique, Rocquencourt, France (carola.kruse@inria.fr).

[‡]BICOM, Brunel University, Uxbridge, UB8 3PH, England (simon.shaw@brunel.ac.uk, www.brunel.ac.uk/bicom).

The material presented below has its roots in a 2001 article by Werder et al. [7]. They used a temporally discontinuous Galerkin (DG) method on the problem above and demonstrated its practicality for high order polynomials in time. This approach has subsequently been extended to second order hyperbolic problems in [6] and our goal here is to develop the methodology for continuous Galerkin (CG) finite element temporal approximation of (1.1) using polynomials of moderately high order. (We return to this point in the conclusions.)

To describe the context of what follows recall that once a spatial finite element approximation of (1.1) has been made, we have a coupled system of M (say) ordinary differential equations to solve. The well-known and popular implicit Euler and Crank–Nicolson (CN) methods are "single step" and, for the time stepping, require only one system solve per time step. It is well known that the CN scheme can (with time averaged rather than pointwise data) be interpreted as a temporally CG finite element approximation using a trial space of piecewise linears (and a test space of piecewise constants). However, we need not stop at that since the Galerkin methodology gives a clear recipe for generating higher order temporal approximations by using higher order piecewise polynomials; see Aziz and Monk [1] for a very complete theoretical treatment of this setup.

Moving to higher orders in time means that each temporal basis function has as a coefficient a set of M unknowns associated with the spatial discretization. An obvious difficulty with the resulting high order temporal scheme is that a naïve implementation would produce temporal inner products that are not *diagonalized*. As a result the $M \times M$ systems would get coupled through the temporal basis and produce a $d \times d$ block matrix system with each block being of size $M \times M$, where d denotes the temporal polynomial degree of the trial functions. Given the power of modern computer architectures this in one space dimension (as implemented in [1]) would not prove too onerous even for quite large values of d but in two space dimensions it soon becomes impractical. For three space dimensions such an implementation is unlikely to be of any practical use.

So, following the DG scheme presented by Werder et al. in [7], our goal here is to present a similar diagonalized formulation for CG. The result will be d systems of size M to solve at each time level, rather than one system of size Md, and the "price" will be the introduction of complex arithmetic (as it also was in [7]). The advantage is that high order CG finite element time discretizations become practical for problems like (1.1) and, due to the decoupling, the d systems of size M can be solved in parallel. We describe one possible parallel implementation strategy later in the conclusions.

As mentioned above, the CG method for the time (as well as space) discretization of the heat equation has been extensively studied by Aziz and Monk in [1] and so we have not seen a need to include any further analysis here. Rather we focus tightly on the implementation in section 2 and its performance on some test problems in section 3. Our means of diagonalizing the temporal systems rely partly on assuming that a certain matrix is diagonalizable (which, by demonstration, it is for the cases considered) and partly on exact integration of the temporal inner products by using the matching Gauss–Lobatto quadrature rule. The temporal trial and test bases are supported on the Gauss–Lobatto nodes and so we see an immediate connection to the "mass-lumping" diagonalization often used in spatial discretizations: so-called spectral element methods (e.g., [3, 4]). To reinforce this connection we consider in our test problems both Galerkin and this spectral element spatial discretization. We note that this "spectral element in time" method is sigificantly different from the method in [7] in that it does not rely on exact orthogonality. Time dependent coefficients, σ, can in principle be handled with ease, and this includes nonlinearities.

We finish with some observations in section 4 and would like to point out here that the material below is presented only in prototype form. There are many other avenues of study that can be pursued but we have deliberately kept this article short and to the point in order to communicate the method in an efficient manner. Lastly here, we remark that our notation is standard and is introduced as necessary.

2. The diagonalised numerical scheme.

Let $V = \{v \in H^1(\Omega) \colon v = 0 \text{ on } \Gamma_D\}$ and then we can introduce the weak formulation of (1.1) in the usual way: find a smooth map $u \colon I \to V$ such that

$$(\varrho\dot{u}(t), v) + a(u(t), v) = \langle L(t), v\rangle \qquad \forall v \in V, \tag{2.1}$$

$$(\varrho u(0), v) = (\varrho\breve{u}, v) \qquad \forall v \in V, \tag{2.2}$$

where, here and below, we suppress dependence on $\boldsymbol{x} \in \Omega$ to make the notation simpler, $(\cdot,\cdot)$ is the standard $L_2(\Omega)$ inner product, $a(\cdot,\cdot) := (\sigma\nabla\cdot, \nabla\cdot)$, and $L \colon I \to V'$ is given by $\langle L(t), \cdot\rangle := (f(t), \cdot) + (g(t), \cdot)_{\Gamma_N}$ with $(\cdot,\cdot)_{\Gamma_N}$ the $L_2(\Gamma_N)$ inner product. The bilinear form $a(\cdot,\cdot)$ is an inner product only if Γ_D has positive surface measure but, nevertheless, we will write $\|v\|_V := a(v,v)^{1/2}$ with the understanding that if $\Gamma_D = \varnothing$, then this is only a seminorm. (However, in this case we can define $w = \exp(-\lambda t)u$ and substitute into (1.1) to get $\varrho\dot{w} - \nabla\cdot\sigma\nabla w + \varrho\lambda w = e^{-\lambda t} f$. Since ϱ is positive, Gårding's inequality then guarantees that this problem has a coercive bilinear form for large enough λ. See, for example, Wloka [8].)

We used the word "smooth" above because we want to avoid too many technical details. Clearly the solution smoothness should be borne in mind before selecting the degree of time approximation, but here for approximations of degree d we would interpret "smooth" as meaning $C^{d+1}(\bar{I}; V)$ in order to realize the optimal convergence rates. We refer to Aziz and Monk [1] for the detailed error bounds, as well as for the alterations needed for the superconvergent behavior.

Choosing $v = u$ (and assuming Γ_D has positive surface measure for simplicity) we easily obtain

$$\frac{d}{dt}\|\varrho^{1/2}u(t)\|^2 + \|u(t)\|_V^2 \leqslant \|L(t)\|_{V'}^2,$$

where $\|\cdot\|$ denotes the $L_2(\Omega)$ norm. Hence we arrive at the standard stability estimate,

$$\|\varrho^{1/2}u(t)\|^2 + \int_0^t \|u(s)\|_V^2\, ds \leqslant \|\varrho^{1/2}\breve{u}\|^2 + \int_0^t \|L(s)\|_{V'}^2\, ds,$$

and this will motivate the choice of the "h-scaled energy" norm (see (3.1)), used later to demonstrate the performance of the scheme.

Next we partition $\bar{I}$ via an increasing sequence of discrete times $0 = t_0 < t_1 < \cdots < t_N = T$, define time intervals $I_j := [t_{j-1}, t_j]$ for $1 \leqslant j \leqslant N$, and let $\mathbb{P}_d(I_j)$ denote the space of polynomials (in *time*) of degree d on I_j. Restricting our attention to a generic time interval I_j, and letting $((\cdot,\cdot))$ denote the $L_2(I_j; L_2(\Omega))$ inner product with the obvious extension of notation to $a((\cdot,\cdot))$ and $\langle\langle L, \cdot\rangle\rangle$, we then have a time-discrete form of (2.1) as follows: for each $j = 1, 2, 3, \ldots, N$ find $U \in \mathbb{P}_d(I_j; V)$ such that

$$((\varrho\dot{U}, v)) + a((U, v)) = \langle\langle L, v\rangle\rangle \qquad \forall v \in \mathbb{P}_{d-1}(I_j; V) \tag{2.3}$$

for any $d \in \mathbb{N}$ and where continuity is enforced on I_j by defining $U(t_{j-1})$ from the computation on the previous time step or from $\breve{u}$ on the first time step. That is, $U(t_{j-1})|_{I_j} := U(t_{j-1})|_{I_{j-1}}$ with $U(0)$ obtained by $L_2(\Omega)$ projection of $\breve{u}$, as in (2.2).

To effect the spatial discretization we let $V^h \subset V$ be a finite dimensional subspace of piecewise polynomials with respect to some partition, or mesh, of $\bar{\Omega}$ and follow the standard procedure with the standard assumptions (see, for example, [5, 2]). The resulting fully discrete problem is to find $U^h \in \mathbb{P}_d(I_j; V^h)$ such that

$$((\varrho\dot{U}^h, v)) + a((U^h, v)) = \langle\langle L, v\rangle\rangle \qquad \forall v \in \mathbb{P}_{d-1}(I_j; V^h). \tag{2.4}$$

Next we introduce the approximation ansatz,

$$u\Big|_{I_j} \approx U^h(t) = \sum_{m=0}^{d} u_m \phi_m(t), \tag{2.5}$$

where each $u_m \in V^h$ and $\{\phi_m\}$ is the Lagrange polynomial basis for $\mathbb{P}_d(I_j)$ with respect to the $d+1$ nodes given by the $d+1$ point Gauss–Lobatto integration rule for I_j. These Gauss–Lobatto nodes are denoted by $\{\tau_q\}$ with $t_{j-1} = \tau_0 < \tau_1 < \tau_2 \cdots < \tau_d = t_j$, and we recall that such a rule integrates polynomials of degree $2d-1$ exactly. Note that in this ansatz we have not referred to the time level index j on the right-hand side—this is because our description will be confined to just one time level and so its omission makes for simpler notation. In general, of course, these quantities, as well as several below, have an implicit j dependence.

It is obvious that $\phi_m(\tau_q) = \delta_{mq}$ (the Kronecker delta) and easy to see that $U^h(t_{j-1}) = u_0$, which will always be known from the computation on the previous time step or, to begin with, derived from $\breve{u}$.

Letting $(\cdot,\cdot)_j$ denote the $L_2(I_j)$ inner product we recall the assumption that ϱ and σ are time independent and use the ansatz (2.5) to rewrite (2.4) as

$$\sum_{m=0}^{d} \Big[(\varrho u_m, v)(\dot{\phi}_m, \psi_n)_j + a(u_m, v)(\phi_m, \psi_n)_j\Big] = \langle (L, \psi_n)_j, v\rangle \tag{2.6}$$

for all $v \in V^h$ and for each $n \in \{1, 2, \ldots, d\}$, where $\{\psi_n\}_{n=1}^d$ is the Lagrange polynomial basis for $\mathbb{P}_{d-1}(I_j)$ with respect to the nodes $\{\tau_q\}_{q=1}^d$. It is then clear that $\psi_n(\tau_q) = \delta_{nq}$ for $n = 1, \ldots, d$ and $q = 1, \ldots, d$.

At this stage it is evident that the spatial systems are coupled together because the temporal basis is not orthogonal. To diagonalize this system we introduce a linear mapping $\chi\colon t \in I_j \to s \in [-1, 1]$ as $\chi(t) = \big(2t - (t_j + t_{j-1})\big)/k_j$, where $k_j = t_j - t_{j-1}$ is the time step, and let $\{\varpi_q\}_{q=0}^d$ be the $d+1$ Gauss–Lobatto integration weights for the interval $(-1, 1)$ with corresponding nodes $s_q = \chi(\tau_q)$. Then, in general, if $f \in \mathbb{P}_{2d-1}(I_j)$ we have

$$\int_{t_{j-1}}^{t_j} f(t)\,dt = \frac{k_j}{2}\int_{-1}^{1} f(\chi^{-1}(s))\,ds = \frac{k_j}{2}\sum_{q=0}^{d} \varpi_q f(\chi^{-1}(s_q)) = \frac{k_j}{2}\sum_{q=0}^{d} \varpi_q f(\tau_q).$$

Therefore, replacing ψ_n in (2.6) with $\varpi_n^{-1}\psi_n$ and noting that $\phi_m\psi_n \in \mathbb{P}_{2d-1}(I_j)$ we can apply this quadrature without committing a variational crime to obtain

$$\frac{1}{\varpi_n}(\phi_m, \psi_n)_j = \frac{k_j}{2\varpi_n}\sum_{q=0}^{d} \varpi_q \phi_m(\tau_q)\psi_n(\tau_q) = \frac{k_j\varpi_0}{2\varpi_n}\delta_{m0}\psi_n(t_{j-1}) + \frac{k_j}{2}\delta_{mn}.$$

Using this in (2.6) then gives

$$\sum_{m=1}^{d} \frac{1}{\varpi_n}(\varrho u_m, v)(\dot{\phi}_m, \psi_n)_j + \frac{k_j}{2} a(u_n, v) = \mathscr{L}_n(\psi_n, v), \tag{2.7}$$

where

$$\mathscr{L}_n(\psi_n, v) = \frac{1}{\varpi_n}\langle (L, \psi_n)_j, v\rangle - \frac{1}{\varpi_n}(\varrho u_0, v)(\dot{\phi}_0, \psi_n)_j - \frac{k_j \varpi_0}{2\varpi_n}\psi_n(t_{j-1}) a(u_0, v). \tag{2.8}$$

The next step follows the same path as in [7] except with our different test space. We define the square matrix A by $\mathsf{A}_{nm} = \varpi_n^{-1}(\dot{\phi}_m, \psi_n)_j$ (for $1 \leqslant n, m \leqslant d$ and where n indexes the rows) and assume that there exists a diagonal matrix $\mathsf{D} = \ulcorner \lambda_1\ \lambda_2\ \ddots\ \lambda_d \lrcorner$ of eigenvalues A and an invertible matrix Q such that $\mathsf{QD} = \mathsf{AQ}$. That this assumption (with the factorization over $\mathbb{C}$) is justified will be demonstrated later in section 3, when we show some numerical results. The limitations of this assumption will be discussed in section 4.

To make the next manipulation a little less heavy on notation we temporarily adopt the summation convention for repeated indices. Then (2.7) can be written as

$$\mathsf{A}_{nm}(\varrho u_m, v) + \frac{k_j}{2} a(u_n, v) = \mathscr{L}_n,$$

and if we let $\{w_\ell\}$ be the unique solution of $u_m = \mathsf{Q}_{m\ell} w_\ell$ we have

$$\mathsf{A}_{nm}\mathsf{Q}_{m\ell}(\varrho w_\ell, v) + \frac{k_j}{2}\mathsf{Q}_{n\ell} a(w_\ell, v) = \mathscr{L}_n.$$

Taking linear combinations of this using each row of $\mathsf{R} = \mathsf{Q}^{-1}$ then gives

$$\mathsf{R}_{pn}\mathsf{A}_{nm}\mathsf{Q}_{m\ell}(\varrho w_\ell, v) + \frac{k_j}{2}\mathsf{R}_{pn}\mathsf{Q}_{n\ell} a(w_\ell, v) = \mathsf{R}_{pn}\mathscr{L}_n,$$

and noting first that $\mathsf{R}_{pn}\mathsf{Q}_{n\ell} = \delta_{p\ell}$ and second that $\mathsf{R}_{pn}\mathsf{A}_{nm}\mathsf{Q}_{m\ell} = \delta_{p\ell}\lambda_p$ we arrive at a family of d *time-decoupled* problems.

Before stating these explicitly we note that in matrix form these last three equations can also be written as follows: first,

$$(\varrho \mathsf{Au}, v) + \frac{k_j}{2} a(\mathsf{u}, v) = \mathsf{L},$$

where $\mathsf{u} = (u_1, \ldots, u_d)^T$ and $(\varrho\mathsf{Au}, v)$ is the vector with nth component given by $(\varrho(\mathsf{Au})_n, v)$; second,

$$(\varrho \mathsf{AQw}, v) + \frac{k_j}{2} a(\mathsf{Qw}, v) = \mathsf{L};$$

and third,

$$(\varrho \mathsf{Dw}, v) + \frac{k_j}{2} a(\mathsf{w}, v) = \mathsf{RL}.$$

Returning now to the component form, and with summation no longer implied the time-decoupled problems are as follows: for $p = 1, \ldots, d$ find $w_p \in V^h \oplus iV^h$ such that

$$\lambda_p(\varrho w_p, v) + \frac{k_j}{2} a(w_p, v) = \sum_{n=1}^{d} \mathsf{R}_{pn}\mathscr{L}_n(\psi_n, v) \qquad \forall v \in V^h \text{ and } 1 \leqslant n \leqslant d \tag{2.9}$$

with $\mathscr{L}_n$ given by (2.8) and, in that, u_0 obtained from $\breve{u}$ when $j = 1$ or from $U(t_{j-1})$ when $j > 1$.

In the next section we will see some numerical experiments that will demonstrate the convergence behavior of the scheme. To close this section we note that

$$\mathsf{A}_{nm} = \frac{1}{\varpi_n}\int_{t_{j-1}}^{t_j} \dot{\phi}_m(t)\psi_n(t)\,dt = \frac{k_j}{2}\frac{1}{\varpi_n}\int_{-1}^{1} \dot{\Phi}_m(s)\Psi_n(s)\,ds,$$

where $\Phi_m(s) = \Phi_m(\chi(t)) := \phi_m(t)$ and $\Psi_n(s) = \Psi_n(\chi(t)) := \psi_n(t)$ are the corresponding Lagrange polynomials on the reference time element $(-1, 1)$. It is obvious that aside from the scalar multiple $k_j/2$, the matrix A is essentially both problem and time step independent. It follows that the decomposition need be computed only once.

Remark 2.1. After spatial discretization the decoupled system (2.9) can be written, for each p, as $\lambda_p \boldsymbol{D}\boldsymbol{w}_p + 2^{-1}k_j\boldsymbol{A}\boldsymbol{w}_p = \boldsymbol{F}_p$ for a (real) mass matrix, $\boldsymbol{D}$, and a (real) stiffness matrix, $\boldsymbol{A}$. If we write $\lambda_p = \alpha + i\beta$, $\boldsymbol{w}_p = \boldsymbol{u} + i\boldsymbol{v}$, and $\beta^{-1}\boldsymbol{F} = i\boldsymbol{N} - \boldsymbol{M}$, then we get (for each p)

$$\boldsymbol{D}\boldsymbol{u} + \left(\frac{\alpha}{\beta}\boldsymbol{D} + \frac{k_j}{2\beta}\boldsymbol{A}\right)\boldsymbol{v} = \boldsymbol{N},$$
$$\boldsymbol{D}\boldsymbol{v} - \left(\frac{\alpha}{\beta}\boldsymbol{D} + \frac{k_j}{2\beta}\boldsymbol{A}\right)\boldsymbol{u} = \boldsymbol{M}.$$

Setting $\boldsymbol{y} = \boldsymbol{D}^{1/2}\boldsymbol{u}$, $\boldsymbol{z} = \boldsymbol{D}^{1/2}\boldsymbol{v}$ and premultiplying by $\boldsymbol{D}^{-1/2}$ then gives

$$\boldsymbol{y} + \boldsymbol{C}\boldsymbol{z} = \boldsymbol{D}^{-1/2}\boldsymbol{N},$$
$$\boldsymbol{z} - \boldsymbol{C}\boldsymbol{y} = \boldsymbol{D}^{-1/2}\boldsymbol{M},$$

where $\boldsymbol{C} := \frac{\alpha}{\beta}\boldsymbol{I} + \frac{k_j}{2\beta}\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ is real symmetric. It follows that

$$\boldsymbol{D}^{1/2}\boldsymbol{u} = \boldsymbol{y} = (\boldsymbol{I} + \boldsymbol{C}^2)^{-1}\Big(\boldsymbol{D}^{-1/2}\boldsymbol{N} - \boldsymbol{C}\boldsymbol{D}^{-1/2}\boldsymbol{M}\Big),$$
$$\boldsymbol{D}^{1/2}\boldsymbol{v} = \boldsymbol{z} = (\boldsymbol{I} + \boldsymbol{C}^2)^{-1}\Big(\boldsymbol{D}^{-1/2}\boldsymbol{M} + \boldsymbol{C}\boldsymbol{D}^{-1/2}\boldsymbol{N}\Big).$$

These calculations can be implemented in real arithmetic and, if the spectral element method is used, the matrix $\boldsymbol{D}$ is diagonal positive definite.

3. Numerical experiments.

Our goal in this section is simply to demonstrate that the approach just described is practical, in that the necessary spectral decompostion of A exists for polynomial degrees of moderate size and that the expected convergence rates are achieved for some test problems with manufactured solutions. We also give some indication of the dependence of error on computation time and see that high order schemes may, in this respect, be preferred, but we are also aware that timing is a subtle issue that depends heavily on implementation and hardware features. So here, to prototype this code, we used MATLAB with only the standard functionality and show timing data only in terms of *relative time*, where, in each case, the wall-clock times for each of the runs was normalized with respect to the longest run's duration. There is no deep reason for choosing that particular normalization—it just seems a clean way of presenting results that can be easily compared.

We take $T = 1$ and consider a unit square spatial domain $\Omega = (0, 1) \times (0, 1)$ in $\mathbb{R}^2$. We set $\varrho = \sigma = 1$ and either $\Gamma_N = \partial\Omega$ or $\Gamma_D = \partial\Omega$ depending on the problem.

We use N equal width time intervals to step from $t = 0$ to T in time steps of width $k = T/N$ and, by dividing the x and y direction boundary edges into M_x and M_y equal width intervals, we generate a uniform mesh of $M = M_xM_y$ quadrilaterals on Ω in the obvious way. A tensor product finite element space of polynomials is then defined with respect to this mesh. The procedure is completely standard except that we use (the tensor product of) the Gauss–Lobatto nodes on each of the quadrilaterals. This is so that we can use high order Gauss–Legendre quadrature to approximate a Galerkin finite element discretization in space, but also so that we can use a Gauss–Lobatto quadrature to produce a spectral element method. Note that in this spectral method we only use the Gauss–Lobatto rules on the system (mass and stiffness) matrix entries. All other spatial integrals are computed with high order Gauss–Legendre quadrature so as to minimize the effect of unwanted variational crimes. To choose the order of this Gauss–Legendre rule for both space and time integrals, we suppose that d is the degree of the polynomial basis in question and recall that an n point Gauss–Legendre rule is exact for polynomial degree $2n-1$. In our code we insist that $2n-1 = 2d+3$ so that the rule can deal with three degrees higher than the products of basis functions of degree d. This means that we use an $n = d+2$ point rule in space (time) when the spatial (temporal) basis is of degree d.

There are three sets of results. In subsection 3.1 we examine the error, $e = u-U$, due only to time discretization, while in subsections 3.2 and 3.3 we look at the space-time error. The error analysis in [1, Thm. 3.2 and 3.3] tells us that we can expect

$$\|e(T)\|_{H^r(\Omega)} \leqslant C(h^{p+1-r} + k^{d+1}) \qquad \text{for } r = 0, 1$$

when using polynomials of degree (p, d) in (space, time) under fairly standard assumptions. Motivated by this, and also by the stability estimate given earlier, our errors are measured in the "h-scaled energy norm," which we define by

$$\|e\|_E := \|e(T)\| + \frac{1}{\sqrt{M_xM_y}} \left(\int_0^T \|\nabla e(s)\|^2 \, ds \right)^{1/2}. \tag{3.1}$$

The scaling is introduced so as not to mix the different powers of $h = \max\{M_x^{-1}, M_y^{-1}\}$ that arise from mixing $L_2(\Omega)$ norms of e and ∇e. For example, we expect ∇e to be of order $O(h^p)$ in the $L_2(\Omega)$ norm, whereas e will be of order $O(h^{p+1})$. This scale factor has the effect of making these two error terms of the same order of magnitude. In fact below we will just take $M_x = M_y \sim h^{-1}$ for all the examples.

Below we refer to the CG-in-time scheme with piecewise polynomials of degree d as CG(d) and we recall that CG(1) corresponds, in essence, to the CN scheme. Henceforth we will refer to CG(1) as CN so that this familiar and popular scheme can be used as a comparator for the performance of the high order schemes.

Under more restrictive assumptions a more specialized result is also available [1, Thm. 4.2], which suggests superconvergence in time:

$$\|e(T)\|_{L_2(\Omega)} \leqslant C(h^{p+1} + k^{2d}).$$

This is illustrated in in subsection 3.3. We note that this is not a superconvergent result for CN because $2d = d+1$ when $d = 1$.

3.1. Convergence behavior for time errors. For these tests we take $u = x(x-1)y(y-1)\cos(t^3)$ and specify load and boundary data so that this is the exact solution of (1.1) with $\Gamma_N = \partial\Omega$. We use biquadratic finite elements in space and

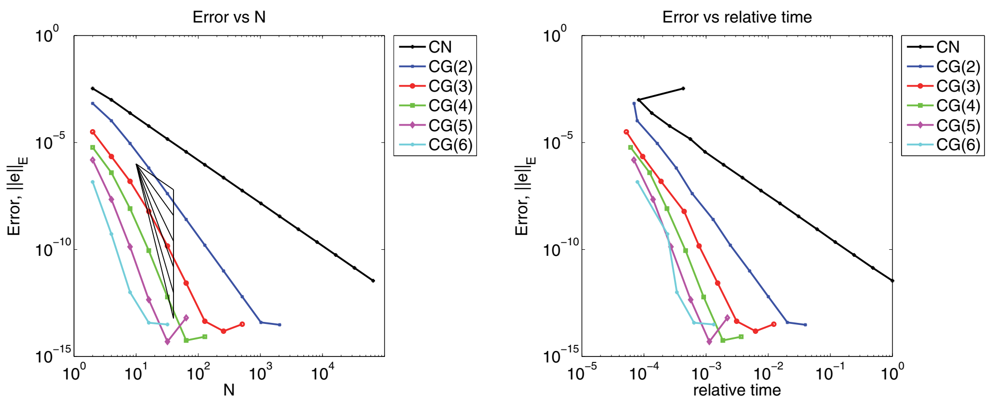

Fig. 3.1. *Time errors for CN and CG(d), $d = 2,3,4,5,6$, resulting from the test problem described in subsection* 3.1*. The dependence of error on the number of time steps is shown on the left (with the triangular fan slopes indicating convergence rates of* $2d$*). The relationship between error and relative computational time is shown on the right.*

need only take $M_x = M_y = 1$ because there will be no spatial discretization error. The results are shown in Figure 3.1 for the Galerkin (as opposed to spectral) spatial discretization.

The figure shows the dependence of error on the number of time steps N, and on the relative solution time, for temporal polynomial degrees $d \in \{1, 2, \ldots, 6\}$. The triangle "fan" on the error versus N plots has slopes that indicate convergence rates of $2, 4, 6, 8, 10, 12$.

3.2. Convergence behavior for space-time errors. In this subsection we specify (1.1) and its data so that $u = \sin(\pi x)\cos(2\pi y)\cos(3\pi t)$ with $\Gamma_N = \partial\Omega$, and we use polynomials of the same degree in both space and time with CN corresponding to bilinears in space and linears in time. The results are shown in Figure 3.2 for the Galerkin spatial discretization (Gauss–Legendre rules everywhere) and for the spectral element discretization (Gauss–Lobatto rules for the system matrices). The fan in this case indicates convergence rates of $2, 3, 4, 5, 6, 7$.

3.3. Superconvergence of time errors. Although we have already seen the temporal superconvergence in subsection 3.1 it is also of interest to examine the more realistic case where there are both space and time discretization errors. For this we take $\Gamma_D = \partial\Omega$ and all data such that $u = e^{-2\pi^2 t}\sin(\pi x)\sin(\pi y)$ is the exact solution. In this case $f = 0$ because $u_t = \nabla^2 u$ and the smoothness conditions required in [1, Thm. 4.2] are satisfied.

The computations are set up so that $h^{-1} = M_x = M_y = N$, the number of time steps, and we choose p and d to satisfy $p = 2d - 1$ with the expectation that we will observe $\|e\|_E = O(h^{p+1} + N^{-2d}) = O(N^{-2d})$. Again, CN corresponds to using bilinears in space and linears in time and the results are shown in Figure 3.3 for the Galerkin spatial discretization (Gauss–Legendre rules everywhere) and for the spectral element discretization (Gauss–Lobatto rules for the system matrices). The indicated rates shown by the fan are $2d$ for $d = 1, \ldots, 6$.

4. Conclusions and discussion. This brief article has described a means of obtaining high order time stepping schemes in the space-time variational framework offered by continuous Galerkin finite element methods and its spectral element variant.

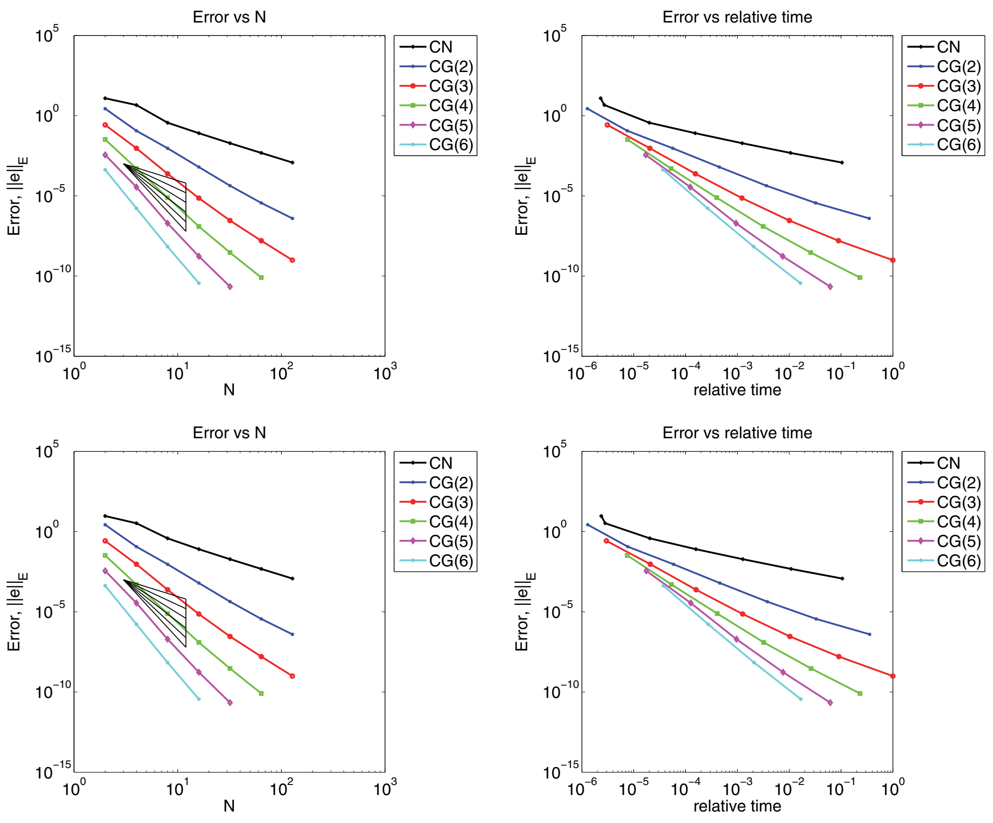

FIG. 3.2. *Space-time errors for CN and CG*(d)*,* $d = 2, 3, 4, 5, 6$*, resulting from Galerkin* (*top two*) *and spectral element* (*bottom two*) *approximation of the test problem described in subsection* 3.2*. The dependence of error on the number of time steps is shown on the left* (*with the triangular fan slopes indicating convergence rates of* $2, 3, \ldots, 7$)*. The relationship between error and relative computational time is shown on the right.*

In the absence of spatial discretization error, Figure 3.1 suggests a temporal convergence rate for the scaled energy norm, (3.1), of N^{-2d}, where N is the number of time steps and d the temporal polynomial degree.

For the more realistic case where both spatial and temporal discretization errors are present we see in Figure 3.2 optimal order convergence in the scaled energy norm, (3.1), and in Figure 3.3 we see the superconvergence carried over to an example where there is both space and time error.

Our results are consistent with the error bounds given by Aziz and Monk in [1]. Moreover, the plots of error against relative computational time suggest, fairly strongly, that this methodology could be profitably deployed on parabolic problems despite the introduction of complex arithmetic. As compared to CN it seems clear that the high order schemes deliver greater accuracy per unit of computational time or, equivalently, can produce a given accuracy in less time.

We also would like to remark that other possibilities exist for investigating the decoupling potential of high order CG-in-time discretizations. As an obvious one we mention that the method used by Werder et al. in [7] (based on orthogonal polynomials) could be used. Less obvious but worth mentioning is the use of a Gauss–Radau quadrature, with a node chosen at the right endpoint of the interval of integration.

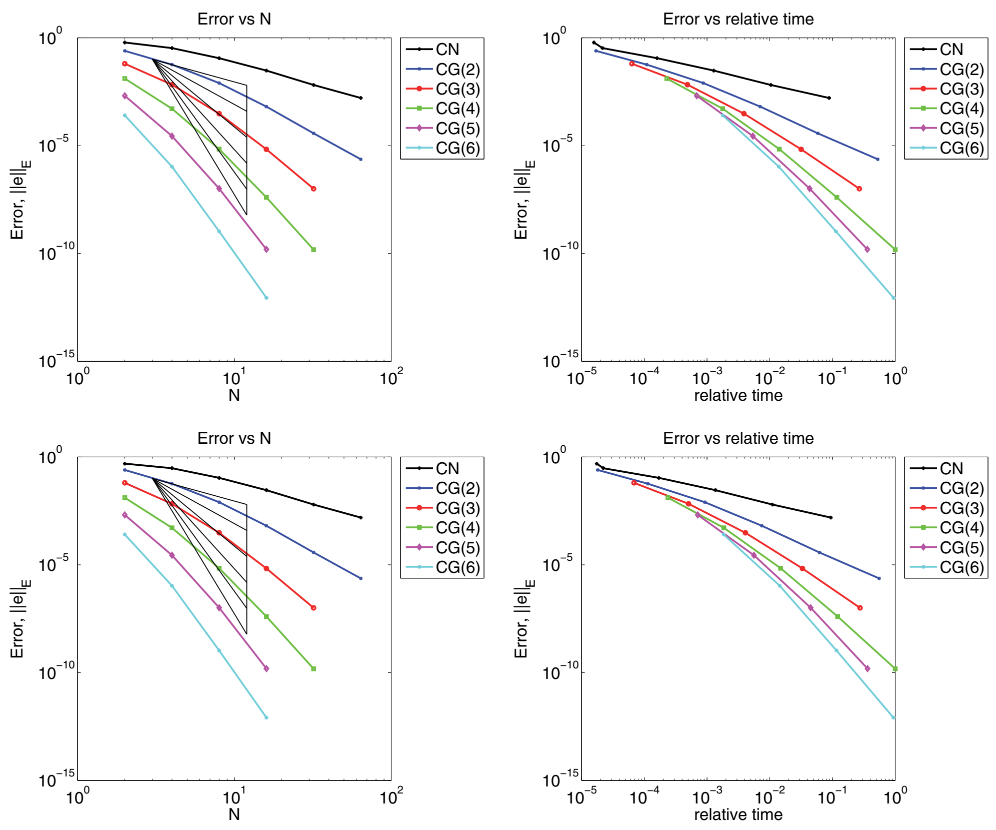

FIG. 3.3. *Time errors for CN and CG*(d), $d = 2, 3, 4, 5, 6$, *resulting from Galerkin* (*top two*) *and spectral element* (*bottom two*) *approximation of the test problem described in subsection* 3.3. *The dependence of error on the number of time steps is shown on the left* (*with the triangular fan slopes indicating convergence rates of* $2d$). *The relationship between error and computational time is shown on the right.*

Global continuity in time is then enforced by using the left endpoint along with d quadrature points to build piecewise polynomials of degree d with the CG-in-time FEM described earlier. The test space will be of degree $d-1$ and we recall that the d-point Gauss–Radau rule integrates polynomials of degree $2d-2$ exactly. It follows that, after discretization, $(\dot{u}, v)$ will be of degree $2d-2$ and integrated exactly, while $(\nabla u, \nabla v)$ will be of degree $2d-1$ and, therefore, require a variational crime with error of size $O(k^{2d})$ (recall that k denotes the time step). Comparing this to the expected $L_2(L_2(\Omega))$ Galerkin error of size $O(k^{d+1})$ we see that optimality should not be destroyed so long as $2d \geqslant d+1$, or $d \geqslant 1$. For high order approximation this is automatically satisfied. With these considerations it then seems worth investigating the diagonalization properties of the A-matrix induced by this rule. We leave this for another time.

There seem to be several more avenues worthy of further exploration related to the ideas presented earlier. We finish with just four.

First, with regard to Remark 2.1, it is clearly of interest to better understand the invertibility properties of $\boldsymbol{I} + \boldsymbol{C}^2$. We note that since $\boldsymbol{C}^2 = (\alpha/\beta)^2\boldsymbol{I} + O(k_j)$ we can at least expect the inverse to exist for small enough k_j but, of course, for a high order

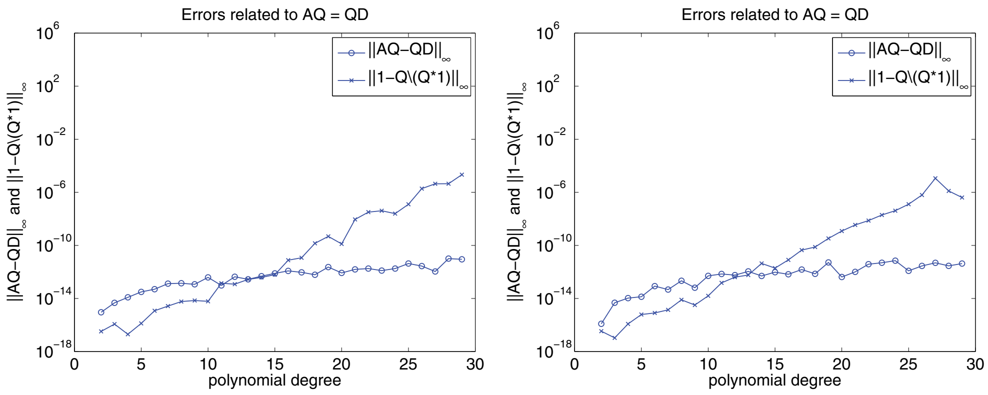

FIG. 4.1. *Computed values of the matrix norms* $\|\mathsf{AQ} - \mathsf{QD}\|_\infty$ *and* $\|\mathbf{1} - \mathsf{Q}\backslash\mathsf{Q}\mathbf{1}\|_\infty$ *for temporal polynomial degrees up to* 29 *using, on the left, the Matlab command* `[Q V]=eig(A)` *and, on the right, the command* `[Q V]=eig(A, 'nobalance')`.

method we would like to be able to take k_j quite large and obtain accuracy through the high polynomial degree.

Second, in the foregoing we have consistently referred to this method being useful for *moderately* high order temporal polynomials. The reason for this is that the arithmetic related to the decomposition $\mathsf{AQ} = \mathsf{QD}$ may not be uniformly stable across all polynomial degrees. This was first noticed when some initial calculations using this diagonalized method were carried out in an exploratory M.Sc. thesis supervised by S. Shaw [9] for a variety of simpler problems.

To illustrate this apparent ill-conditioning we plot the quantities $\|\mathsf{AQ} - \mathsf{QD}\|_\infty$ and $\|\mathbf{1} - \mathsf{Q}\backslash\mathsf{Q}\mathbf{1}\|_\infty$ against polynomial degrees $1, 2, \ldots, 29$ in Figure 4.1. Here the backslash refers to MATLAB's "solve" operator and $\mathbf{1}$ is a vector of ones. (Both normed quantities should, of course, be zero.) The calculations were carried out using `eig` in MATLAB (32-bit R2009b running on 64-bit Windows 7) with and without balancing. These quantities were chosen because the decomposition and subsequent solve involving Q are the key steps in the decoupling procedure.

It seems evident that the decompsition is stable across all degress of practical interest (we did not code for degree 30 and above), but the matrix solve involving Q becomes more and more seriously ill-conditioned as the degree rises above around 15. The `nobalance` option appears to be having a mitigating effect at high degree but the trend is not clear from these data. Both sets of results have used two sweeps of iterative refinement. These had a minor benefit but more sweeps had no visible beneficial effect. We conclude that while degrees less than, say, 15 ought to produce good results, implementation for degrees higher than that may run into trouble due to matrix-solve errors (at least with this routine).

Nevertheless, even with this limitation, the formulation presented above still represents a potentially useful and efficient way of deriving high order time stepping schemes. Furthermore, it is of course possible that enhanced precision calculations using, say, ARPREC (see http://crd-legacy.lbl.gov/~dhbailey/mpdist), could produce high-quality results at degrees greater than 15, but we see this as future work.

Third, we offer a speculation on the application of this methodology to the Navier–Stokes equations for the flow of an incompressible and homogeneous fluid. If the fluid has density ϱ and viscosity μ, then, in the context of section 2 and with $\boldsymbol{w} = \dot{\boldsymbol{u}}$, these are

$$\int_{t_{j-1}}^{t_j} (\dot{\boldsymbol{u}}, \boldsymbol{v})\,dt = \int_{t_{j-1}}^{t_j} (\boldsymbol{w}, \boldsymbol{v})\,dt,$$
$$\int_{t_{j-1}}^{t_j} (\boldsymbol{w}, \boldsymbol{v}) + (\boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}) + (\nu\nabla\boldsymbol{u}, \nabla\boldsymbol{v})\,dt = \mathscr{F}(\boldsymbol{v})$$

for $\nu = \mu/\varrho$ and $\mathscr{F}(\cdot) := \int_{t_{j-1}}^{t_j} (\varrho^{-1}\boldsymbol{f}, \cdot) - (\varrho^{-1}\nabla p, \cdot)\,dt$. The only term here that is new in the context of this article is the trilinear form. Applying the ansatz (2.5) and the Gauss–Lobatto approximation to the time integral of this form produces (in the earlier notation)

$$\begin{aligned}
\int_{t_{j-1}}^{t_j} (\boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v})\,dt &= \int_{t_{j-1}}^{t_j} \left(\left(\sum_{m=0}^{d} \boldsymbol{u}_m \phi_m(t)\right) \cdot \left(\sum_{n=0}^{d} \nabla\boldsymbol{u}_n \phi_n(t)\right), \boldsymbol{v}(t) \right) dt,\\
&= \int_{t_{j-1}}^{t_j} \sum_{m=0}^{d}\sum_{n=0}^{d} \phi_m(t)\phi_n(t)\big(\boldsymbol{u}_m\cdot\nabla\boldsymbol{u}_n, \boldsymbol{v}(t)\big)\,dt,\\
&\approx \frac{k_j}{2}\sum_{q=0}^{d} \varpi_q \sum_{m=0}^{d}\sum_{n=0}^{d} \phi_m(\tau_q)\phi_n(\tau_q)\big(\boldsymbol{u}_m\cdot\nabla\boldsymbol{u}_n, \boldsymbol{v}(\tau_q)\big),\\
&= \frac{k_j}{2}\sum_{q=0}^{d} \varpi_q \sum_{m=0}^{d}\sum_{n=0}^{d} \delta_{mq}\delta_{nq}\big(\boldsymbol{u}_m\cdot\nabla\boldsymbol{u}_n, \boldsymbol{v}(\tau_q)\big),\\
&= \frac{k_j}{2}\sum_{q=0}^{d} \varpi_q \big(\boldsymbol{u}_q\cdot\nabla\boldsymbol{u}_q, \boldsymbol{v}(\tau_q)\big),\\
&= \frac{k_j}{2}\sum_{q=0}^{d} \frac{\varpi_q}{\varpi_n}\psi_n(\tau_q)\big(\boldsymbol{u}_q\cdot\nabla\boldsymbol{u}_q, \boldsymbol{v}\big)
\end{aligned}$$

for all $\boldsymbol{v}$ in the *spatial* test space and for each $n = 1, \ldots, d$. Therefore, we conclude

$$\int_{t_{j-1}}^{t_j} (\boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v})\,dt \approx \frac{\varpi_0 k_j}{2\varpi_n}\psi_n(t_{j-1})(\boldsymbol{u}_0\cdot\nabla\boldsymbol{u}_0, \boldsymbol{v}) + \frac{k_j}{2}(\boldsymbol{u}_n\cdot\nabla\boldsymbol{u}_n, \boldsymbol{v})$$

and slotting this into the equivalent here of (2.7) gives

$$\sum_{m=1}^{d} \frac{1}{\varpi_n}(\boldsymbol{u}_m, \boldsymbol{v})(\dot{\phi}_m, \psi_n)_j = \frac{k_j}{2}(\boldsymbol{w}_n, \boldsymbol{v}),$$
$$\frac{k_j}{2}(\boldsymbol{w}_n, \boldsymbol{v}) + \frac{k_j}{2}(\boldsymbol{u}_n\cdot\nabla\boldsymbol{u}_n, \boldsymbol{v}) + \frac{k_j}{2}(\nu\nabla\boldsymbol{u}_n, \nabla\boldsymbol{v}) = \mathscr{L}_n(\psi_n, \boldsymbol{v})$$

with a different set of $\mathscr{L}_n$'s. The first of these then diagonalizes exactly as described earlier, whereas the second is already diagonalized. While this is not yet in the form of a practical algorithm, we can conclude that for temporal polynomials of moderate degree the scheme decouples to a set of nonlinear boundary value problems. This is at the expense of a variational crime of nonnegligible order but, even so, the $d+1$ point Gauss–Lobatto temporal quadrature applied to the trilinear form is exact for polynomials of degree $2d-1$ and so will induce an error of size $O(k^{2d})$. Thus the superconvergence, if it exists for these equations, may well be preserved. Notice that it seems necessary to introduce $\boldsymbol{w} = \dot{\boldsymbol{u}}$ because the eigenvalue decomposition needed on the $\dot{\boldsymbol{u}}$ term will not have the required properties in the nonlinear term. If a spectral element method is used in space, then this $L_2(\Omega)$ projection relating $\boldsymbol{w}$ and $\dot{\boldsymbol{u}}$ is a trivial diagonal matrix inversion.

Fourth, and in closing, we point out the potential for both coarse- and fine-grained parallelism offered by this formulation. First, for temporal approximation using polynomials of degree d, recall that the d boundary value problems in (2.9) are independent of one another. Assuming, for simplicity, that we have d machines at our disposal, we can then solve for each of these problems simultaneously on different machines. If, in addition, each machine has more than one processor (e.g., an eight-core CPU), then this coarse-grained parallelism can be reinforced with a fine-grained multicore parallelism (using, for example, Open MP—see `openmp.org`) applied to each boundary value problem. One can imagine many variants, depending on the available facilities.

REFERENCES

[1] A. K. Aziz and P. Monk, *Continuous finite elements in space and time for the heat equation*, Math. Comp., 52 (1989), pp. 255–274.

[2] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag New York, 1994.

[3] G. F. Carey and E. Barragy, *Basis function selection and preconditioning high degree finite element and spectral methods*, BIT, 29 (1989), pp. 794–804.

[4] M. Durufle, P. Grob, and P. Joly, *Influence of Gauss and Gauss-Lobatto quadrature rules on the accuracy of a quadrilateral finite element method*, Numer. Methods Partial Differential Equations, 25 (2009), pp. 526–551.

[5] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Dover, Mineola, NY, 2009.

[6] C. Kruse, M. Maischak, S. Shaw, J. R. Whiteman, S. E. Greenwald, M. J. Birch, M. P. Brewin, H. T. Banks, Z. R. Kenz, and S. Hu, *High Order Space-Time Finite Element Schemes for Acoustic and Viscodynamic Wave Equations with Temporal Decoupling*, Int. J. Numer. Meth. Engng., in press, see http://onlinelibrary.wiley.com/journal/10.1002/.

[7] T. Werder, K. Gerdes, D. Schötzau, and C. Schwab, *hp-discontinuous Galerkin time stepping for parabolic problems*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6685–6708.

[8] J. Wloka, *Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1987.

[9] A. Pecka, *Time-Diagonalized Continuous Galerkin Space-Time Finite Element Approximation of the Heat Equation*, M.Sc. thesis, Brunel University, Uxbridge, UK, 2012.