Electronic Transactions on Numerical Analysis.
Volume 40, pp. 36-57, 2013.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

# INEXACT AND TRUNCATED PARAREAL-IN-TIME KRYLOV SUBSPACE METHODS FOR PARABOLIC OPTIMAL CONTROL PROBLEMS*

XIUHONG DU†, MARCUS SARKIS‡, CHRISTIAN E. SCHAERER§, AND DANIEL B. SZYLD¶

**Abstract.** We study the use of inexact and truncated Krylov subspace methods for the solution of the linear systems arising in the discretized solution of the optimal control of a parabolic partial differential equation. An all-at-once temporal discretization and a reduction approach are used to obtain a symmetric positive definite system for the control variables only, where a Conjugate Gradient (CG) method can be used at the cost of the solution of two very large linear systems in each iteration. We propose to use inexact Krylov subspace methods, in which the solution of the two large linear systems are not solved exactly, and their approximate solutions can be progressively less exact. The option we propose is the use of the parareal-in-time algorithm for approximating the solution of these two linear systems. The use of less parareal iterations makes it possible to reduce the time integration costs and to improve the time parallel scalability. We also show that truncated methods could be used without much delay in convergence but with important savings in storage. Spectral bounds are provided and numerical experiments with inexact versions of CG, the full orthogonalization method (FOM), and of truncated FOM are presented, illustrating the potential of the proposed methods.

**Key words.** parabolic optimal control, reduced system, saddle point problem, inexact Krylov subspace methods, truncated Krylov subspace methods, parareal approximation, spectral bounds

**AMS subject classifications.** 65F10, 65F50, 65N22, 35B37, 15A42, 35A15

## 1. Introduction.

An important class of problems in many fields including electromagnetic inversion, diffraction tomography, and optimal design are solved using optimization with partial differential equations as constraints. A common approach for the solution of this constrained optimization problem consists of introducing Lagrange multipliers and solving for the stationary point of the Lagrangian. This approach yields a KKT system with a saddle point form; see, e.g., [3, 4, 13, 14, 15, 21]. In this paper, we consider the solution of a large saddle point (or KKT) system of the form

$$\begin{bmatrix} \mathbf{K} & \mathbf{0} & \mathbf{E}^T \\ \mathbf{0} & \mathbf{G} & \mathbf{N}^T \\ \mathbf{E} & \mathbf{N} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{0} \\ \mathbf{f}_3 \end{bmatrix}, \tag{1.1}$$

where the vector $\mathbf{y}$ is the state, the vector $\mathbf{u}$ is the control, and $\mathbf{p}$ is a vector of Lagrange multipliers. We are particularly interested in problems that evolve from time dependent PDE's where $\mathbf{E}$ and $\mathbf{N}$ are space-time discretizations of some time dependent operators and the vectors $\mathbf{y}$, $\mathbf{u}$, and $\mathbf{p}$ are the discrete space-time solutions. In many of the applications mentioned above, the control is time invariant and thus has a much smaller dimension than when it is time dependent.

*Received March 22, 2012. Accepted January 14, 2013. Published online on March 8, 2013. Recommended by L. Reichel.

†Department of Mathematics, Alfred University, Alfred, NY 14802, USA (du@alfred.edu).

‡Department of Mathematics, Worcester Polytechnic Institute - WPI, 100 Institute Road, Worcester, MA 01609, USA and Instituto de Matemática Pura e Aplicada - IMPA, Estrada Dona Castorina 110, Rio de Janeiro, RJ 22460, Brazil (msarkis@wpi.edu).

§Department of Computer Science, Polytechnic School, National University of Asuncion, P.O.Box: 2111 SL, San Lorenzo, Paraguay (cschaer@pol.una.py). Supported in part by Paraguay CONACYT under Program 1698OC/PR.

¶Department of Mathematics, Temple University (038- 16), 1805 N. Broad Street, Philadelphia, PA 19122-6094, USA (szyld@temple.edu). Supported in part by the U.S. Department of Energy under grant DE-FG02-05ER25672 and the U.S. National Science Foundation under grant DMS-1115520.

ETNA
Kent State University
http://etna.math.kent.edu



Although it is possible to tackle the KKT system head on, it requires storage for all space-time vectors. In this case, it is possible to use the reduced Hessian approach (see, e.g., [19, 20, 22, 23]) that yields a much smaller linear system

$$\mathbf{H}\mathbf{u} = \mathbf{b}, \tag{1.2}$$

where the matrix $\mathbf{H} := \mathbf{G} + \mathbf{N}^T\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}\mathbf{N}$ is symmetric positive definite, and it is often referred to as the reduced Hessian.

Other approaches can be used as well (see, e.g., the survey [2]), however, here we focus on the reduced Hessian approach for two reasons. We already mentioned that the size of the problem (1.1) which comes from a space and time discretization is such that the Hessian approach gives a considerable smaller system. Second, the reduced Hessian method is the method of choice for nonlinear problems Sequential Quadratic Programming codes (see [23]) and as such is well developed. We mention that there is recent work that allows for inexactness in the solution of the KKT system [6], but the reduced Hessian is still the dominant approach in nonlinear programming. See also [1] for an additional discussion on the merits of the reduced Hessian approach and further references therein.

One important feature of the reduced Hessian approach applied to our problem is that it leads to a symmetric positive definite system and thus could be solved using the conjugate gradient method (CG). However, the main disadvantage of the reduced Hessian method is that each matrix-vector product is very expensive. The CG at each iteration requires only one matrix-vector product with the matrix $\mathbf{H}$. Note however that each of these matrix-vector products requires the solution of two very large linear systems, one with $\mathbf{E}$ and one with $\mathbf{E}^T$, say

$$\mathbf{E}\mathbf{z} = \mathbf{s} \quad \text{and} \quad \mathbf{E}^T\mathbf{w} = \mathbf{v}, \tag{1.3}$$

and traditionally these are expected to be solved accurately. This means that we need to solve two discretized time dependent partial differential equations (PDEs) per CG step, and since the problem under consideration is large, an iterative method has to be employed for the solution of these two discretized PDEs. In practice, the solution of the linear systems (1.3) are performed iteratively, using suitable preconditioners, up to a certain given tolerance. Thus an iterative solver is embedded within an outer one (that is, the one used for the solution of the reduced Hessian system). The cost of these inner computations, of course, may be considerable. However, if the calculations are performed inexactly, there may be significant savings in computational effort. In other words, relaxing the accuracy of these inner matrix-vector products would decrease the cost of the overall calculations and thus make the reduced Hessian method attractive.

There are two likely scenarios where solving the Equations (1.3) approximately may bring considerable computational advantages. The first corresponds to spatial parallelization and it appears for instance when $\mathbf{E}^{-1}$ and $\mathbf{E}^{-T}$ involve an implicit temporal discretization and a domain decomposition. The second case, which is the one we analyze in this paper, comes from a temporal parallelization, where for instance an exact solver is used in each time step, however, the parareal method [18] is applied in order to speed up total CPU time. We note that both spatial and temporal parallelization could also be considered in the inexact and truncated Krylov subspace framework developed below.

For the approximate solution of (1.3) we introduce the use of inexact parareal approximations. The parareal method [18] is a parallel-in-time iterative method for solving an evolution based on a decomposition of its time domain. The operations $\mathbf{E}^{-1}\mathbf{s}$ and $\mathbf{E}^{-T}\mathbf{v}$ represent the discrete forward and reversed in time evolution of the parabolic equation, and even though the

ETNA
Kent State University
http://etna.math.kent.edu



time direction seems intrinsically sequential, the combination of coarse and fine solution procedures have proven to converge and allow for more rapid solution if parallel architectures are available. Due to coarse granularity and time parallelization of the method, an inexact parareal with a fixed number of parareal iterations was considered in [22] for constructing a preconditioner for all-at-once KKT large systems arising in parabolic optimal control problems. The goal here is different, it is to develop inexact parareal approximations for the reduced Hessian method. While in [22] the main mathematical concern was to establish condition number estimates of the preconditioned system, here the concern is in how to measure the inexactness in the computation of the Schur complement (Hessian) system (1.2) in terms of the number of parareal iterations, therefore, new theoretical results are required (see Theorem 3.5).

The natural question that arises then is how inexact these inner matrix-vector multiplications are allowed to be performed in order to ensure the convergence of the outer iterations. In the context of nonlinear optimization it is a common practice to solve the linear equations at each step of a Newton method inaccurately as long as we are far from the solution, but as we get closer, we need to increase the accuracy if we wish to achieve quadratic convergence [16]. But in the context of linear systems, such as the one with reduced Hessian, it was shown in [28] that it is actually beneficial to perform the calculations in an increasingly inexact way as the iteration progresses; see also [5, 12, 31, 32]. In fact, in [7] experiments are shown where increasing the accuracy in the linear systems degrades the performance of the method.

The inexactness introduced when solving the systems (1.3) up to a certain tolerance can be understood as performing instead of an exact matrix-vector multiplication $\mathbf{Hv}$, an inexact matrix-vector multiplication given by

$$\mathcal{H}\mathbf{v} := (\mathbf{H} + \mathbf{D})\,\mathbf{v} \tag{1.4}$$

where $\mathbf{D}$ is an error (or discrepancy) matrix which usually changes from one iteration to the next.

Studies of inexact Krylov subspace methods, where matrix-vector products are of the form (1.4), indicate that $\|\mathbf{D}\|$ can be allowed to grow as the iterations progress; see [5, 28, 31]. In our context this means that the (inner) tolerance with which the systems (1.3) are solved can increase, with the associated computational savings. We describe this in detail in Section 3.1.

When using inexact matrix-vector products, the three-term recurrence of CG does not guarantee the orthogonality of the basis of the Krylov subspace. In [25] MINRES is used and the problem is assumed to maintain symmetry. Furthermore, if we use different number of inner iterations or different number of sweeps for the approximation for the two systems (1.3), then the resulting matrix $\mathcal{H}$ is not symmetric. In other words, we may gain computational time, but we lose symmetry, and thus we need a Krylov subspace method without a three-term recurrence. In this paper, we use the full orthogonalization method (FOM) [26, 27, 30], which reduces to CG when the coefficient matrix is symmetric and does not change from one iteration to the next.

To mitigate the need for additional storage, we explore the use of truncated FOM (TFOM), namely we only store the last $m_T$ vectors and only orthogonalize new basis vectors with respect to those $m_T$ vectors (two sets of $m_T$ vectors are computed and stored, and the approximate solution can be computed progressively without the need to store the whole basis); see, e.g., [26, 27, 30]. Usually, the truncated methods have a "delay" in convergence, i.e., the lack of full orthogonality translates into taking more iterations to converge to the same accuracy. The theory developed in [29] indicates that the delay experienced in truncated methods does not have to be significant. This delay in convergence with its associated computational

ETNA
Kent State University
http://etna.math.kent.edu



cost is of course offset by the tremendous storage savings that one can obtain. Thus, in this paper we use inexact and truncated FOM (TIFOM) for the solution of the reduced Hessian system (1.2). We believe that this is the first time that both inexactness and truncation are used simultaneously. In the special case when only $m_T = 2$ vectors are kept, then TIFOM is sometimes called inexact CG (ICG). We note that *Flexible Conjugate Gradients* (FCG) in [24] uses a different but related approach; it starts from CG and considers the storage of additional vectors for further "local orthogonalization". We note that inexact Krylov methods have also been studied for singular matrices [8], and therefore they can also be applied to ill-conditioned problems.

In the next section we describe the general parabolic control problems that we consider, and then specify a class of problems on which we illustrate our approach: a classical distributed control problem. In Section 3, we discuss the inexactness in the computation of the Schur complement (Hessian) system (1.2) as well as the conditions on the approximation of $\mathbf{E}$ and $\mathbf{E}^T$ for using the parareal approximation of $\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}$; see Equation (1.2). In Section 4, we report numerical experiments using inexact FOM and its truncated variants. The results show that considerable savings in time and memory requirements are obtained when the proposed truncated and inexact methods are used.

**2. A parabolic optimal control problem.** Let $\Omega \subset \mathbb{R}^d$ be an interval ($d = 1$) or a polygonal ($d = 2$) domain of size of $O(1)$ and let $\mathcal{A}$ be a coercive map from a Hilbert space $L^2(t_0, t_f; Y)$ to $L^2(t_0, t_f; Y')$, where $Y'$ is the dual of $Y$ with respect to the pivot space $H = L^2(\Omega)$. Denote the state variable space as

$$\mathcal{Y} = \{z \in L^2(t_0, t_f; Y) : z_t \in L^2(t_0, t_f; Y')\}.$$

Given $y_0 \in H$, we consider the following state equation on $(t_0, t_f)$ with $z \in \mathcal{Y}$:

$$\left\{ \begin{array}{rcll} z_t + \mathcal{A}z & = & v & \text{in } x \in \Omega, \\ z & = & y_D & \text{on } x \in \partial\Omega, \\ z(0) & = & y_0. & \end{array} \right. \tag{2.1}$$

In this paper we consider the following control problem:

> The distributed control problem, where the distributed control $v$ belongs to an admissible space $\mathcal{V} = L^2(t_0, t_f; V)$, where in our application $V = L^2(\Omega)$. We consider $\mathcal{A} = -\Delta$ (minus the Laplacian), and without loss of generality we assume homogeneous Dirichlet boundary conditions $y_D = 0$ (equivalently $Y = H_0^1(\Omega)$), and we indicate the dependence of $z$ on $v \in \mathcal{V}$ using the notation $z(v)$.

We mention that many of the considerations we present for the above problem can also be applied to the boundary control problem, which is an ill-posed inverse problem, and where the interest consist in recovering the boundary conditions; see, e.g., [1, 7].

We describe now our approach. To define an optimal control problem, we consider a time interval $(t_0, t_f)$, a given target function $\tilde{y}$ in $L^2(t_0, t_f; Y)$, parameters $\alpha \geq 0$, $\beta \geq 0$, and $\gamma > 0$, and we employ the following performance function which we associate with the state equation (2.1)

$$\begin{aligned} J(z(v), v) := & \frac{\alpha}{2} \int_{t_0}^{t_f} \|z(v)(t, \cdot) - \tilde{y}(t, \cdot)\|^2_{L^2(\Omega)} \\ & + \frac{\beta}{2} \|z(v)(t_f, \cdot) - \tilde{y}(t_f, \cdot)\|^2_{L^2(\Omega)} + \frac{\gamma}{2} \int_{t_0}^{t_f} \|v(t, \cdot)\|^2_{L^2(\Omega)}. \end{aligned} \tag{2.2}$$

**ETNA**
Kent State University
http://etna.math.kent.edu



For simplicity, we assume that $y_0 \in Y$ and $\tilde{y} \in L^2(t_0, t_f; Y)$. Following [17], we consider the optimal control problem for Equation (2.1), which is equivalent to finding a control $v$ which *minimizes* the cost function (2.2).

To discretize the state equation (2.1) in space, we apply the finite element method to its weak formulation for each fixed $t \in (t_0, t_f)$. We choose a quasi-uniform triangulation $\mathcal{T}_h(\Omega)$ of $\Omega$, and employ the $\mathcal{P}_1$ conforming finite element space $Y_h \subset Y$ for approximating $z(t,\cdot)$, and the $\mathcal{P}_0$ finite element space $V_h \subset V$ for approximating $v(t,\cdot)$. Let $\{\phi_j\}_{j=1}^{\widehat{q}}$ and $\{\psi_j\}_{j=1}^{\widehat{p}}$ denote the standard basis functions for $Y_h$ and $V_h$, respectively. Throughout the paper we use the same notation $z \in Y_h$ and $z \in \mathbb{R}^{\widehat{q}}$, or $v \in V_h$ and $v \in \mathbb{R}^{\widehat{p}}$, to denote both a finite element function in space and its corresponding vector representation, and to indicate their time dependence, we denote them as $\underline{z}$ and $\underline{v}$, respectively.

A discretization in space of the continuous time linear-quadratic optimal control problem will seek to minimize the following quadratic functional

$$
\begin{aligned}
J_h(\underline{z}, \underline{v}) :=& \frac{\alpha}{2} \int_{t_0}^{t_f} (\underline{z} - \underline{\tilde{y}}^h)^T(t) M_h (\underline{z} - \underline{\tilde{y}})(t)\, dt \\
&+ \frac{\beta}{2} (\underline{z}(t_f) - \underline{\tilde{y}}(t_f))^T M_h (\underline{z}(t_f) - \underline{\tilde{y}}(t_f)) + \frac{\gamma}{2} \int_{t_0}^{t_f} \underline{v}^T(t) R_h \underline{v}(t)\, dt
\end{aligned} \tag{2.3}
$$

subject to the *constraint* that $\underline{z}$ satisfies the discrete equation of state:

$$
M_h \dot{\underline{z}} + A_h\, \underline{z} = B_h \underline{v}, \quad \text{for } t_0 < t < t_f; \quad \text{and } \underline{z}(t_0) = y_0^h. \tag{2.4}
$$

Here $(\underline{z} - \underline{\tilde{y}}^h)(t)$ and $(\underline{z}(t_f) - \underline{\tilde{y}}(t_f))$ denote the tracking and the final error. The functions $\underline{\tilde{y}}^h(t)$ and $y_0^h$ belong to $Y_h$ and are approximations to $\underline{\tilde{y}}(t)$ and $y_0$, respectively (for instance, consider $L^2(\Omega)$-projections into $Y_h$). The matrices $M_h, A_h \in \mathbb{R}^{\widehat{q}\times\widehat{q}}$, $B_h \in \mathbb{R}^{\widehat{q}\times\widehat{p}}$, and $R_h \in \mathbb{R}^{\widehat{p}\times\widehat{p}}$ have entries $(M_h)_{ij} := (\phi_i, \phi_j)$, $(A_h)_{ij} := (\phi_i, \mathcal{A}\phi_j)$, $(B_h)_{ij} := (\phi_i, \psi_j)$, and $(R_h)_{ij} := (\psi_i, \psi_j)$, where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product.

To obtain a temporal discretization of (2.3) and (2.4), we partition $[t_0,\ t_f]$ into $\widehat{l}$ equal sub-intervals with time step size $\tau = (t_f - t_0)/\widehat{l}$. We denote $t_l = t_0 + l\,\tau$ for $0 \le l \le \widehat{l}$. Associated with this partition, we assume that the state variable $\underline{z}$ is continuous in $[t_0, t_f]$ and linear in each sub-interval $[t_{l-1}, t_l]$, $1 \le l \le \widehat{l}$, with associated basis functions $\{\vartheta_l\}_{l=0}^{\widehat{l}}$. Denoting by $z_l \in \mathbb{R}^{\widehat{q}}$ the nodal representation of $\underline{z}(t_l)$, we have $\underline{z}(t) = \sum_{l=0}^{\widehat{l}} z_l \vartheta_l(t)$. The control variable $\underline{v}$ is assumed to be time discontinuous and constant in each sub-interval $(t_{l-1}, t_l)$ with basis functions $\{\chi_l\}_{l=1}^{\widehat{l}}$. Denoting $v_l \in \mathbb{R}^{\widehat{p}}$ as the nodal representation of $\underline{v}(t_l - (\tau/2))$ yields $\underline{v}(t) = \sum_{l=1}^{\widehat{l}} v_l \chi_l(t)$.

The corresponding discretization of the expression (2.3) yields:

$$
J_h^\tau(\mathbf{z}, \mathbf{v}) = \frac{1}{2} (\mathbf{z} - \tilde{\mathbf{y}})^T \mathbf{K} (\mathbf{z} - \tilde{\mathbf{y}}) + \frac{1}{2} \mathbf{v}^T \mathbf{G} \mathbf{v} + (\mathbf{z} - \tilde{\mathbf{y}})^T \mathbf{g}, \tag{2.5}
$$

where the block vectors $\mathbf{z} := [z_1^T, \ldots, z_{\widehat{l}}^T]^T \in \mathbb{R}^{\widehat{l}\widehat{q}}$ and $\mathbf{v} := [v_1^T, \ldots, v_{\widehat{l}}^T]^T \in \mathbb{R}^{\widehat{l}\widehat{p}}$ denote the state and control variables, respectively, at all the discrete times; the discrete target is $\tilde{\mathbf{y}} := [\tilde{y}_1^T, \ldots, \tilde{y}_{\widehat{l}}^T]^T \in \mathbb{R}^{\widehat{l}\widehat{q}}$ with target error $e_l = (z_l - \tilde{y}_l^h)$ for $0 \le l \le \widehat{l}$, where $z_0 := y_0^h$; the matrix $\mathbf{K} := \mathbf{Z} + \mathbf{\Gamma}$ with $\mathbf{Z}, \mathbf{\Gamma} \in \mathbb{R}^{(\widehat{l}\widehat{q})\times(\widehat{l}\widehat{q})}$, $\mathbf{\Gamma} = \beta\, \mathrm{diag}(0, 0, ..., M_h)$ and $\mathbf{Z} = \alpha D_\tau \otimes M_h$, $D_\tau \in \mathbb{R}^{\widehat{l}\times\widehat{l}}$ with entries $(D_\tau)_{ij} := \int_{t_0}^{t_f} \vartheta_i(t)\vartheta_j(t)dt$, for $1 \le i, j \le \widehat{l}$, and $\otimes$ stands for the Kronecker product; the matrix $\mathbf{G} = \gamma\tau I_{\widehat{l}} \otimes R_h \in \mathbb{R}^{(\widehat{l}\widehat{p})\times(\widehat{l}\widehat{p})}$ and $I_{\widehat{l}} \in \mathbb{R}^{\widehat{l}\times\widehat{l}}$ is an identity matrix; and the vector $\mathbf{g} = (g_1^T, 0^T, \ldots, 0^T)^T$, where $g_1 = \alpha\frac{\tau}{6} M_h e_0$ and where we

ETNA
Kent State University
http://etna.math.kent.edu



have used $\frac{\tau}{6} = \int_{t_0}^{t_f} \vartheta_0(t)\vartheta_1(t)dt$. Note that $g_1$ does not necessarily vanish because it is not assumed that $\tilde{y}_0^h = z_0$.

Employing the backward Euler discretization in time, the Equation (2.4) takes the form

$$F_1 \underline{z}_{i+1} = F_0 \underline{z}_i + \tau B_h \underline{v}_{i+1} \quad \text{for} \quad t_0 < t < t_f; \quad \text{and} \quad \underline{z}(t_0) = y_0^h, \tag{2.6}$$

where $F_0, F_1 \in \mathbb{R}^{\widehat{q}\times\widehat{q}}$ are (fixed) matrices given by $F_0 := M_h$ and $F_1 := M_h + \tau A_h$. Using a full discretization in time, Equation (2.4) has the matrix form

$$\mathbf{E}\,\mathbf{z} + \mathbf{N}\,\mathbf{v} = \mathbf{f}_3, \tag{2.7}$$

where the input vector is $\mathbf{f}_3 := [(F_0 y_0^h)^T, 0^T, ..., 0^T]^T \in \mathbb{R}^{\widehat{lq}}$. The block lower bidiagonal matrix $\mathbf{E} \in \mathbb{R}^{(\widehat{lq})\times(\widehat{lq})}$ is given by

$$\mathbf{E} = \begin{bmatrix} F_1 & & & \\ -F_0 & F_1 & & \\ & \ddots & \ddots & \\ & & -F_0 & F_1 \end{bmatrix},$$

and the block diagonal matrix $\mathbf{N} \in \mathbb{R}^{(\widehat{lq})\times(\widehat{lp})}$ is given by $\mathbf{N} = -\tau I_{\widehat{l}} \otimes B_h$.

**3. Inexact Krylov subspace methods for the Schur complement system.** The Lagrangian functional $\mathcal{L}_h^\tau(\mathbf{z}, \mathbf{v}, \mathbf{q})$ for minimizing (2.5) subject to the constraint (2.7) is

$$\mathcal{L}_h^\tau(\mathbf{z}, \mathbf{v}, \mathbf{q}) = J_h^\tau(\mathbf{z}, \mathbf{v}) + \mathbf{q}^T(\mathbf{E}\mathbf{z} + \mathbf{N}\mathbf{v} - \mathbf{f}_3). \tag{3.1}$$

To obtain a discrete saddle point formulation of (3.1), we apply the optimality conditions for $\mathcal{L}_h^\tau(\cdot,\cdot,\cdot)$. This yields the symmetric indefinite linear system (1.1), where $\mathbf{f}_1 := \mathbf{K}\tilde{\mathbf{y}} - \mathbf{g}$ and $\tilde{\mathbf{y}} := [(\tilde{y}_1^h)^T, \ldots, (\tilde{y}_{\widehat{l}}^h)^T]^T \in \mathbb{R}^{\widehat{lq}}$.

Eliminating $\mathbf{y} = \mathbf{E}^{-1}(\mathbf{f}_3 - \mathbf{N}\,\mathbf{u})$ and $\mathbf{p} = \mathbf{E}^{-T}(\mathbf{f}_1 - \mathbf{K}\,\mathbf{y})$ in (1.1) yields the *reduced* Schur complement system:

$$\mathbf{H}\mathbf{u} := (\mathbf{G} + \mathbf{N}^T\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}\mathbf{N})\mathbf{u} = \mathbf{b} \tag{3.2}$$

(see [19, 21]), where $\mathbf{b} := \mathbf{N}^T\mathbf{E}^{-T}\left(\mathbf{K}\mathbf{E}^{-1}\mathbf{f_3} - \mathbf{f_1}\right)$ is pre-computed. The matrix $\mathbf{H}$ is symmetric positive definite, and in addition we have that

$$(\mathbf{v}, \mathbf{G}\mathbf{v}) \le (\mathbf{v}, \mathbf{H}\mathbf{v}) \le \mu(\mathbf{v}, \mathbf{G}\mathbf{v}),$$

where $\mu$ is estimated later in (3.27). As a result, the (preconditioned) Conjugate Gradient method can be used to solve (3.2), but each matrix-vector product with $\mathbf{H}$ requires the solution of two linear systems, one with $\mathbf{E}$ and one with $\mathbf{E}^T$.

As already stated, our aim is to use inexact Krylov subspace methods for the approximation to the solution of (3.2). We propose the use of a Truncated Full Orthogonalization Method (TFOM) and its inexact version, which we call TIFOM. Again, we favor the use of versions of FOM here, since they reduce to CG if the solutions of the two linear systems in (1.3) are exact.

**3.1. The truncated full orthogonalization method.** In the FOM algorithm, at each iteration one would require a matrix-vector product with the matrix $\mathbf{H}$ defined in (3.2), say $\mathbf{H}\mathbf{s}$ with $\|\mathbf{s}\| = 1$. This product would proceed as follows:

ETNA
Kent State University
http://etna.math.kent.edu



ALGORITHM 1: Matrix-vector product $\mathbf{Hs}$

1. Multiply $\mathbf{Ns}$ and $\mathbf{Gs}$.
2. Solve $\mathbf{Ez} = \mathbf{Ns}$.
3. Multiply $\mathbf{Kz}$.
4. Solve $\mathbf{E}^T\mathbf{w} = \mathbf{Kz}$.
5. Compute $\mathbf{N}^T\mathbf{w} + \mathbf{Gs}$.

The idea we are exploring is to replace steps 2 and 4 in Algorithm 1 above using an approximation (later on in the paper, this approximation will be obtained using the parareal algorithm studied, e.g., in [11]). We are interested in approximating the solutions of these two linear systems using as less accuracy as possible, while obtaining a good solution to (3.2). To that end, we first review some results available in the recent literature on inexact Krylov subspace methods; see, [5, 28, 31].

We begin by mentioning two results from [28] dealing with inexact FOM and its truncated version (Theorems 3.1 and 3.2 below). The Full Orthogonalization Method (FOM) is a Krylov subspace method for nonsymmetric linear systems, say of the form $\mathbf{Hu} = \mathbf{b}$ with initial vector $\mathbf{u}_0$, which after $m$ iterations builds an orthogonal basis of the usual Krylov subspace using the Arnoldi method and collects these vectors in a matrix $\mathbf{V}_m$. Then, the approximation $\mathbf{u}_m = \mathbf{u}_0 + \mathbf{V}_m\mathbf{x}_m$ is computed, where $\mathbf{x}_m$ is the solution of the linear system

$$\mathbf{H}_m\mathbf{x} = \beta\mathbf{e}_1, \tag{3.3}$$

with $\beta = \|\mathbf{r}_0\|$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{Hu}_0$, $\mathbf{H}_m = \mathbf{V}_m^T\mathbf{HV}_m$ is an $m \times m$ upper Hessenberg matrix, and $\mathbf{e}_1$ is the first Euclidean vector; see, e.g., [9, 26, 27, 30] for more details on FOM. The truncated version of FOM consists of computing a basis collected in $\mathbf{V}_m$ where the last vector $\mathbf{v}_m$ is only orthogonalized with respect to the previous $m_T$ vectors say. In this manner, only $m_T + 1$ vectors are needed to be kept in storage, and the resulting matrix $\mathbf{H}_m = \mathbf{V}_m^T\mathbf{HV}_m$ is $m \times m$ upper Hessenberg and banded with (upper) semi-bandwidth $m_T - 1$*. In the extreme case, if $m_T = 2$ and if $\mathbf{H}$ is symmetric positive definite, FOM reduces to CG, and $\mathbf{H}_m$ is tridiagonal.

As indicated in the introduction, when we refer to the inexact Arnoldi method, we simply mean that at the $k$th step of the Arnoldi method, the matrix-vector product $\mathbf{Hv}_{k-1}$ is not exact. Instead we have $(\mathbf{H} + \mathbf{D}_k)\mathbf{v}_{k-1}$ for some *discrepancy* matrix $\mathbf{D}_k$, which is usually a different matrix at each different step $k$. A natural question is how large $\|\mathbf{D}_k\|$ is allowed to grow and how we assure a residual norm below a prescribed tolerance.

Using the Arnoldi decomposition $\mathbf{HV}_k = \mathbf{V}_{k+1}\widehat{\mathbf{H}}_k$ and since the principal square part of $\widehat{\mathbf{H}}_k$ is given by $\mathbf{H}_k = [I_k, 0]\,\widehat{\mathbf{H}}_k$, the next theorem guarantees overall convergence below a given tolerance $\epsilon$.

THEOREM 3.1. [28] *Assume that $m$ steps of the inexact Arnoldi method have been carried out, and let $\mathbf{x}_m$ be the solution of (3.3). Let $\mathbf{r}_k = \mathbf{b} - \mathbf{Hu}_k = \mathbf{r}_0 - \mathbf{HV}_k\mathbf{x}_k$ be the true residual, and $\tilde{\mathbf{r}}_k = \mathbf{r}_0 - \mathbf{V}_{k+1}\widehat{\mathbf{H}}_k\mathbf{y}_k$ be the computed residual at the $k$th FOM iteration,*

*Truncated FOM is called IOM in [27], and it can be implemented in such a way that $\mathbf{u}_m = \mathbf{u}_0 + \mathbf{V}_m\mathbf{x}_m$ is computed directly from $\mathbf{u}_{m-1}$ without the need to store all the vectors in $\mathbf{V}_m$. This implementation is called DIOM in [27].

ETNA
Kent State University
http://etna.math.kent.edu



*respectively. Let* $\epsilon > 0$*, and let*

$$\ell_m = \sigma_m(\mathbf{H}_m)/m, \tag{3.4}$$

*where* $\sigma_m(\mathbf{H}_m)$ *is the smallest singular value of* $\mathbf{H}_m$*. If for every* $k < m$*,*

$$\|\mathbf{D}_k\| \leq \ell_m \frac{\epsilon}{\|\tilde{\mathbf{r}}_{k-1}\|}, \tag{3.5}$$

*then,*

$$\|\mathbf{r}_m - \tilde{\mathbf{r}}_m\| < \epsilon \ \text{ and } \ \|\mathbf{V}_m^T \mathbf{r}_m\| < \epsilon. \tag{3.6}$$

An equivalent result for the inexact truncated FOM with a truncation parameter $m_T$ is shown in the following theorem.

THEOREM 3.2. [28] *Assume that* $m$ *steps of the inexact truncated Arnoldi method have been carried out (with truncation parameter* $m_T$*). Let the hypothesis of Theorem 3.1 hold, and let here* $\ell_m$ *be*

$$\ell_m = \sigma_m(\mathbf{V}_m)\sigma_m(\mathbf{H}_m)/m. \tag{3.7}$$

*If (3.5) holds for every* $k < m$*, then one has that (3.6) holds.*

REMARK 3.3. As mentioned earlier, the advantage of truncated methods is that fewer vectors need to be kept in storage. The price one pays is that the matrix $\mathbf{V}_m$ with the basis vectors does not have orthogonal columns. In the case of full FOM (i.e., with no truncation) the quantity $\sigma_m(\mathbf{V}_m) = 1$, while in the truncated case it decreases as the truncation parameter $m_T$ decreases. Therefore, the value of $\ell_m$ in (3.7) is smaller than that in (3.4), and furthermore the smaller the truncation parameter $m_T$ is, the more restrictive the condition (3.5) is. In other words, we can allow less inexactness when we have more truncation.

Remark 3.3 applies in particular to the extreme case of $m_T = 2$, i.e., to inexact CG. We mention that the convergence bound of FCG in [24, Theorem 3.1] is of a different kind than (3.6), but nevertheless, the essence of Remark 3.3 also applies: the smaller the truncation parameter $m_T$ is, the smaller the discrepancy needs to be to maintain convergence.

Returning to Algorithm 1, we now consider the situation when for the matrix vector product $\mathbf{Hs}$, we approximate the solution of each of the linear systems in steps 2 and 4. We consider that the approximate solution $\widehat{\mathbf{z}}$ to $\mathbf{Ez} = \mathbf{Ns}$ in step 2 is obtained via an iterative method. In particular, in the next section we describe the parareal method represented by $\widehat{\mathbf{z}} = \mathbf{E}_{n_1}^{-1}\mathbf{Ns}$, where $\mathbf{E}_{n_1}$ corresponds to $n_1$ applications (or sweeps) of the parareal method.

**3.1.1. Parareal approximation $\mathbf{E}_n^{-T}\widehat{\mathbf{K}}\mathbf{E}_n^{-1}$.** The parareal method is a parallel iterative method for solving an evolution equation based on a *decomposition* of its *temporal* domain $[t_0, t_f]$ into $\widehat{k}$ *coarse* sub-intervals of length $\Delta T = (t_f - t_0)/\widehat{k}$, setting $T_0 = t_0$ and $T_k = t_0 + k\Delta T$ for $1 \leq k \leq \widehat{k}$; see, e.g., [18]. It determines the solution at the times $T_k$ for $1 \leq k \leq \widehat{k}$ by using a *multiple-shooting* technique which requires solving the parabolic equation on each interval $(T_{k-1}, T_k)$ in *parallel*. To speed up the multiple shooting iteration, the residual equations are "preconditioned" by solving a "coarse" time-grid discretization of the parabolic equation using the time step $\Delta T$.

We define the matrix $\widehat{\mathbf{K}} := \widehat{\mathbf{Z}} + \widehat{\mathbf{\Gamma}}$ with $\widehat{\mathbf{Z}}, \widehat{\mathbf{\Gamma}} \in \mathbb{R}^{((\widehat{l}+\widehat{k}-1)\widehat{q})\times((\widehat{l}+\widehat{k}-1)\widehat{q})}$, where $\widehat{\mathbf{\Gamma}} = \beta\,\mathrm{diag}(0, 0, ..., M_h)$. Here, $\widehat{\mathbf{Z}} = \alpha\widehat{D}_\tau \otimes M_h$, $\widehat{D}_\tau := \mathrm{blockdiag}(\widehat{D}_\tau^1, \ldots, \widehat{D}_\tau^{\widehat{k}})$, $\widehat{D}_\tau^1 \in \mathbb{R}^{(\widehat{m})\times(\widehat{m})}$, and $\widehat{D}_\tau^k \in \mathbb{R}^{((\widehat{m}+1))\times((\widehat{m}+1))}$, for $2 \leq k \leq \widehat{k}$, are the time mass matrices associated to the sub-intervals $[T_{k-1}, T_k]$, where $\widehat{m} = (T_k - T_{k-1})/\tau$. Note that $\widehat{\mathbf{K}}$ is a

**ETNA**
Kent State University
http://etna.math.kent.edu



block diagonal (in time) matrix, and it is easy to see that $(\widehat{l}+\widehat{k}-1)\widehat{q} = \widehat{m}\widehat{q}+(\widehat{k}-1)(\widehat{m}+1)\widehat{q}$. Note that $\mathbf{K}$ can be obtained by assembling $\widehat{\mathbf{K}}$ at the times $T_k, 1 \le k \le \hat{k}-1$. In order to simplify notation, from now on we denote the operation $\mathbf{w}^T\mathbf{K}\mathbf{z}$ by $\mathbf{w}^T\widehat{\mathbf{K}}\mathbf{z}$, where the vectors $\mathbf{w}, \mathbf{z} \in \mathbb{R}^{(\widehat{l}\widehat{q})}$ are mapped to vectors in $\mathbb{R}^{((\widehat{l}+\widehat{k}-1)\widehat{q})}$, also denoted by $\mathbf{w}$ and $\mathbf{z}$, where their nodal values corresponding to the times $T_k, 1 \le k \le \widehat{k}-1$ are duplicated.

In this section we formulate a preconditioner $\mathbf{E}_n$ for $\mathbf{E}$ based on $n$ Richardson iterations of the parareal algorithm; cf. [32] where Richardson is used as an outer iteration for a different Schur complement problem. Using $\mathbf{E}_n$, an application of $\mathbf{E}_n^{-T}\widehat{\mathbf{K}}\mathbf{E}_n^{-1}$ to a vector $\mathbf{v} = [v_1^T, \ldots, v_{\widehat{l}}^T]^T \in \mathbb{R}^{\widehat{l}\widehat{q}}$ can be computed in three steps.

**Step I**, apply $\mathbf{E}_n^{-1}\mathbf{v} :\rightarrow \widehat{\mathbf{z}}_n$ using $n$ applications of the parareal method (described in more detail below).
**Step II**, multiply $\widehat{\mathbf{K}}\widehat{\mathbf{z}}_n :\rightarrow \widehat{\mathbf{t}}_n$ (see below).
**Step III**, apply $\mathbf{E}_n^{-T}\widehat{\mathbf{t}}_n :\rightarrow \mathbf{w}_n$, i.e., the transpose of **Step I**.

Let $\widehat{m} = (T_k - T_{k-1})/\tau$ and $j_{k-1} = (T_{k-1} - T_0)/\tau$. Consider the solution $Z_k$ at time $T_k$ defined by marching from time $T_{k-1}$ to time $T_k$ using the backward Euler discretization scheme on the fine time mesh (characterized by $\tau$) with an initial data $Z_{k-1}$ at $T_{k-1}$ with forcing term $[v_{j_{k-1}+1}^T, \ldots, v_{j_{k-1}+\widehat{m}}^T]^T$. It is easy to see that

$$F_1\, Z_k = F_0^{\Delta}\, Z_{k-1} + S_k,$$

where $F_0^{\Delta} := (F_0 F_1^{-1})^{\widehat{m}-1} F_0 \in \mathbb{R}^{\widehat{q}\times\widehat{q}}$, $S_k := \sum_{m=1}^{\widehat{m}} \left(F_0 F_1^{-1}\right)^{\widehat{m}-m} v_{j_{k-1}+m}$, $Z_0 = 0$, and $F_0$ and $F_1$ as in (2.6). Imposing continuity $F_1\, Z_k - F_0^{\Delta}\, Z_{k-1} - S_k = 0$ at times $T_k$, for $1 \le k \le \widehat{k}$, yields

$$\mathbf{C}\,\mathbf{Z} := \begin{bmatrix} F_1 & & & \\ -F_0^{\Delta} & F_1 & & \\ & \ddots & \ddots & \\ & & -F_0^{\Delta} & F_1 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{\widehat{k}} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{\widehat{k}} \end{bmatrix} =: \mathbf{S}. \tag{3.8}$$

In this paper we consider the case where the coarse solution at $T_k$ with initial data $Z_{k-1} \in \mathbb{R}^{\widehat{q}}$ at $T_{k-1}$ is obtained by applying one coarse time step of the backward Euler method $G_1 Z_k = G_0\, Z_{k-1}$, where the matrix $G_1 := (M_h + A_h \Delta T)$ and $G_0 := M_h \in \mathbb{R}^{\widehat{q}\times\widehat{q}}$.

In the parareal algorithm, the following coarse propagator based on $G_0$ and $G_1$ is employed to precondition the system (3.8) via:

$$\begin{bmatrix} Z_1^{i+1} \\ Z_2^{i+1} \\ \vdots \\ Z_{\widehat{k}}^{i+1} \end{bmatrix} = \begin{bmatrix} Z_1^{i} \\ Z_2^{i} \\ \vdots \\ Z_{\widehat{k}}^{i} \end{bmatrix} + \begin{bmatrix} G_1 & & & \\ -G_0 & G_1 & & \\ & \ddots & \ddots & \\ & & -G_0 & G_1 \end{bmatrix}^{-1} \begin{bmatrix} R_1^{i} \\ R_2^{i} \\ \vdots \\ R_{\widehat{k}}^{i} \end{bmatrix},$$

for $0 \le i \le n-1$, where the residual $\mathbf{R}^i := \left[{R_1^i}^T, \ldots, {R_{\widehat{k}}^i}^T\right]^T \in \mathbb{R}^{\widehat{k}\widehat{q}}$ in (3.8) is defined as $\mathbf{R}^i := \mathbf{S} - \mathbf{C}\,\mathbf{Z}^i$, where $\mathbf{Z}^i := \left[{Z_1^i}^T, \ldots, {Z_{\widehat{k}}^i}^T\right]^T \in \mathbb{R}^{\widehat{k}\widehat{q}}$, and $\mathbf{Z}^0 := \left[0^T, \ldots, 0^T\right]^T$.

We now define $\widehat{\mathbf{z}}_n := \mathbf{E}_n^{-1}\mathbf{s}$. Let $\widehat{\mathbf{z}}_n$ be the nodal representation of a piecewise linear function $\widehat{\underline{z}}^n$ in time with respect to the fine triangulation parameterized by $\tau$ on $[t_0, t_f]$, and continuous inside each coarse sub-interval $[T_{k-1}, T_k]$, i.e., the function $\widehat{\underline{z}}_n$ can be discontinuous across the coarse points $T_k$, $1 \le k \le \widehat{k}-1$, therefore, $\widehat{\mathbf{z}}_n \in \mathbb{R}^{(\widehat{l}+\widehat{k}-1)\widehat{q}}$. On each sub-interval $[T_{k-1}, T_k]$, $\widehat{\underline{z}}_n$ is defined marching from time $T_{k-1}$ to time $T_k$ using the backward Euler scheme with fine times steps $\tau$ and initial data $Z_{k-1}^n$ at $T_{k-1}$.

ETNA
Kent State University
http://etna.math.kent.edu



**3.1.2. Conditions on the approximation of $\mathbf{E}^{-1}$ and $\mathbf{E}^{-T}$.** We return to Algorithm 1 and analyze the situation when for the matrix vector product $\mathbf{Hs}$, we approximate the solution of each of the linear systems in steps 2 and 4. We mention that studies of inexact Krylov subspace methods such as FOM applied to some standard Schur complements (i.e., with only one inverse) can be found in [28, 32].

Let $\widehat{\mathbf{z}}$ be the approximate solution to $\mathbf{Ez} = \mathbf{Ns}$ and let $\mathbf{q}_1 = \mathbf{E}\widehat{\mathbf{z}} - \mathbf{Ns}$ be its residual. Now step 3 has the form $\widehat{\mathbf{K}}\widehat{\mathbf{z}}$. Let $\widehat{\mathbf{w}}$ be the approximate solution to $\mathbf{E}^T\mathbf{w} = \widehat{\mathbf{K}}\widehat{\mathbf{z}}$ and let $\mathbf{q}_2 = \mathbf{E}^T\widehat{\mathbf{w}} - \widehat{\mathbf{K}}\widehat{\mathbf{z}}$ be its residual. Therefore, we have that $\widehat{\mathbf{z}} = \mathbf{E}^{-1}\mathbf{q}_1 + \mathbf{E}^{-1}\mathbf{Ns}$ and $\widehat{\mathbf{w}} = \mathbf{E}^{-T}\mathbf{q}_2 + \mathbf{E}^{-T}\widehat{\mathbf{K}}\widehat{\mathbf{z}}$. Thus, in step 5 we have

$$\begin{aligned}\mathbf{N}^T\widehat{\mathbf{w}} &= \mathbf{N}^T\mathbf{E}^{-T}\mathbf{q}_2 + \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\left(\mathbf{E}^{-1}\mathbf{q}_1 + \mathbf{E}^{-1}\mathbf{Ns}\right),\\ &= \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{Ns} + \left(\mathbf{N}^T\mathbf{E}^{-T}\mathbf{q}_2 + \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{q}_1\right).\end{aligned}$$

Thus, the inexact matrix-vector product $\mathbf{Gs} + \mathbf{N}^T\widehat{\mathbf{w}}$ differs from the exact matrix-vector product $\mathbf{Gs} + \mathbf{N}^T\mathbf{w}$ exactly by the discrepancy vector

$$\mathbf{d} = \mathbf{N}^T\mathbf{E}^{-T}\mathbf{q}_2 + \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{q}_1.$$

Let us define the discrepancy matrix as

$$\mathbf{D} := \mathbf{N}^T\mathbf{E}^{-T}\mathbf{q}_2\mathbf{s}^T + \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{q}_1\mathbf{s}^T, \tag{3.9}$$

where $\|\mathbf{s}\| = 1$. Our goal is to satisfy a condition of the form (3.5). To that end, observe that from (3.9), we have

$$\|\mathbf{D}\| \le \|\mathbf{N}^T\mathbf{E}^{-T}\|\,\|\mathbf{q}_2\| + \|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\|\,\|\mathbf{q}_1\|.$$

Therefore, to achieve (3.5), it suffices to require that

$$\|\mathbf{q}_2\| \le \eta\,\frac{\ell_m}{\|\mathbf{N}^T\mathbf{E}^{-T}\|}\,\frac{\epsilon}{\|\mathbf{r}_{m-1}\|} := \ell_m^{(1)}\frac{\epsilon}{\|\mathbf{r}_{m-1}\|} \tag{3.10}$$

and that

$$\|\mathbf{q}_1\| \le (1-\eta)\,\frac{\ell_m}{\|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\|}\,\frac{\epsilon}{\|\mathbf{r}_{m-1}\|} := \ell_m^{(2)}\frac{\epsilon}{\|\mathbf{r}_{m-1}\|}, \tag{3.11}$$

for a given $0 < \eta < 1$.

In the expressions (3.10) and (3.11), the parameter $\eta$ can the fixed (e.g., $\eta = 1/2$), or it may vary from one step to the next.

**3.1.3. The convergence of inexact parareal.** We consider that the approximate solution $\widehat{\mathbf{z}}$ is obtained via an iterative method. In particular, we use the parareal method represented by $\widehat{\mathbf{z}} = \mathbf{E}_{n_1}^{-1}\mathbf{Ns}$ as described in Section 3.1.1. In a similar manner, $\widehat{\mathbf{w}}$ is obtained via $\widehat{\mathbf{w}} = \mathbf{E}_{n_2}^{-T}\widehat{\mathbf{K}}\widehat{\mathbf{z}}$. Notice that $n_1$ is not necessarily equal to $n_2$. In case $n_1 = n_2$, it symmetrizes the matrix $\mathbf{E}_{n_2}^{-T}\widehat{\mathbf{K}}\mathbf{E}_{n_1}^{-1}$; this property will be explored further. The residual at step 2 of Algorithm 1 in terms of the parareal method is given by

$$\mathbf{q}_1 = \mathbf{E}\widehat{\mathbf{z}} - \mathbf{Ns} = \mathbf{E}\mathbf{E}_{n_1}^{-1}\mathbf{Ns} - \mathbf{Ns}$$

and, correspondingly, the residual in step 4 is:

$$\mathbf{q}_2 = \mathbf{E}^T\widehat{\mathbf{w}} - \widehat{\mathbf{K}}\widehat{\mathbf{z}} = \mathbf{E}^T\left(\mathbf{E}_{n_2}^{-T}\widehat{\mathbf{K}}\mathbf{E}_{n_1}^{-1}\right)\mathbf{Ns} - \widehat{\mathbf{K}}\mathbf{E}_{n_1}^{-1}\mathbf{Ns}.$$

ETNA
Kent State University
http://etna.math.kent.edu



As a result the discrepancy matrix defined in the expression (3.9) takes the form

$$\mathbf{D} = \mathbf{N}^T\mathbf{E}_{n_2}^{-T}\widehat{\mathbf{K}}\mathbf{E}_{n_1}^{-1}\mathbf{N}\mathbf{s}\mathbf{s}^T - \mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\mathbf{s}\mathbf{s}^T. \tag{3.12}$$

Theorem 3.5 below shows that the norm of the discrepancy matrix $\mathbf{D}$ converges geometrically to zero as we increase the number $n := \min\{n_1, n_2\}$ of applications of the parareal method, that is, $\|\mathbf{D}\| \leq C\rho_n\|\mathbf{G}\|$ where $\rho_n \leq 0.2984256075^n$. The convergence rate $0.2984256075$ holds when the backward Euler scheme is applied to both time steps $\tau$ and $\Delta T$; see [10, 21, 22] on how to establish convergence rates for parareal methods. Before we prove Theorem 3.5, we next prove the following intermediate result.

LEMMA 3.4. *Let $\rho_n$ denote the convergence factor for $n$ applications of the parareal method. Then for any $\mathbf{w} \in \mathbb{R}^{(\widehat{l}\widehat{q})\times(\widehat{l}\widehat{q})}$ and $\mathbf{z} := \mathbf{E}^{-1}\mathbf{w}$ with $\underline{z}(t)$ indicating its time dependence, we have that*

$$\left((\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{w}, \widehat{\mathbf{K}}(\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{w}\right) \leq (\alpha(t_f - t_0) + \beta)\rho_n^2 \sum_{k=1}^{\hat{k}} \|\underline{z}(T_k)\|_{L^2(\Omega)}. \tag{3.13}$$

*Proof.* Let $A_h$ and $M_h$ be the $\widehat{q} \times \widehat{q}$ symmetric positive definite matrices introduced in (2.4). Let $X_h := [x_1, ..., x_{\widehat{q}}]$ and $\Lambda_h := \text{diag}\{\lambda_1, ..., \lambda_{\widehat{q}}\}$ be the generalized eigenvectors and eigenvalues of $A_h$ with respect to $M_h$, i.e., $A_h = M_h X_h \Lambda_h X_h^{-1}$. Let $\mathbf{z} := \mathbf{E}^{-1}\mathbf{w}$ with $\underline{z}(t) = \sum_{q=1}^{\hat{q}} \phi_q(t)x_q$ and $\hat{\mathbf{z}}_n := \mathbf{E}_n^{-1}\mathbf{w}$ with $\underline{\hat{z}}_n(t) = \sum_{q=1}^{\hat{q}} \phi_q^n(t)x_q$. We note that $\phi_q^n$ might be discontinuous across the coarse points $T_k$. Then

$$\begin{aligned}\left((\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{w}, \widehat{\mathbf{K}}(\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{w}\right) &\\ = \alpha\|\underline{\hat{z}}_n - \underline{z}\|^2_{L^2(t_0,t_f;L^2(\Omega))} + \beta\|\underline{\hat{z}}_n(t_f) - \underline{z}(t_f)\|^2_{L^2(\Omega)} &\\ = \sum_{q=1}^{\hat{q}} \alpha\|\phi_q^n - \phi_q\|^2_{L^2(t_0,t_f)} + \beta|\phi_q^n(t_f) - \phi_q(t_f)|^2.&\end{aligned}$$

*First part* (Estimation of $\alpha\|\phi_q^n - \phi_q\|^2_{L^2(t_0,t_f)}$). For each $t_l \in [T_{k-1}, T_k]$ we have

$$|\phi_q^n(t_l) - \phi_q(t_l)| = \left((1 + \tau\lambda_q)^{-1}\right)^{(t_l - T_{k-1})/\tau} |\phi_q^n(T_{k-1}) - \phi_q(T_{k-1})|,$$

and since $\lambda_q > 0$ implies $\left((1 + \tau\lambda_q)^{-1}\right)^{(t_l - T_{k-1})/\tau} \leq 1$, we obtain

$$\|\phi_q^n - \phi_q\|^2_{L^2(T_{k-1},T_k)} \leq \Delta T|\phi_q^n(T_{k-1}) - \phi_q(T_{k-1})|^2.$$

Hence,

$$\|\phi_q^n - \phi_q\|^2_{L^2(t_0,t_f)} \leq (t_f - t_0) \max_{1\leq k\leq\hat{k}} |\phi_q^n(T_k) - \phi_q(T_k)|^2.$$

Using [22, Lemma 4.3] with $\phi_q(T_0) = 0$ and initial value $\phi_q^0(T_k) = 0$, we obtain

$$\max_{1\leq k\leq\hat{k}} |\phi_q^n(T_k) - \phi_q(T_k)|^2 \leq \rho_n^2 \max_{1\leq k\leq\hat{k}} |\phi_q(T_k)|^2 \leq \rho_n^2 \sum_{k=1}^{\hat{k}} |\phi_q(T_k)|^2. \tag{3.14}$$

We just have established the upper bound for $\alpha\|\phi_q^n - \phi_q\|^2_{L^2(t_0,t_f)}$ given by

$$\alpha\|\phi_q^n - \phi_q\|^2_{L^2(t_0,t_f)} \leq \alpha(t_f - t_0)\rho_n^2 \sum_{k=1}^{\hat{k}} |\phi_q(T_k)|^2. \tag{3.15}$$

ETNA
Kent State University
http://etna.math.kent.edu



*Second part* (Estimation of $\beta|\phi_q^n(t_f) - \phi_q^n(t_f)|^2$). It follows from (3.14) that

$$\beta|\phi_q^n(t_f) - \phi_q^n(t_f)|^2 \leq \beta\rho_n^2 \sum_{k=1}^{\hat{k}} |\phi_q(T_k)|^2. \tag{3.16}$$

Using the expressions (3.15) and (3.16), and the identity

$$\sum_{q=1}^{\hat{q}} |\phi_q(T_k)|^2 = \|\underline{z}(T_k)\|^2_{L^2(\Omega)}$$

yields the upper bound (3.13).
This completes the proof. □

THEOREM 3.5. *Let $\hat{k} = (t_f - t_0)/\Delta T$, $\mathbf{D}$ be as in (3.12) and $\rho_n$ the rate of convergence of the parareal in consideration. Then*

$$\|\mathbf{D}\| \leq 4(t_l - t_0)\frac{\alpha(t_f - t_0) + \beta}{\gamma}\left(\hat{k}\rho_{n_1}\rho_{n_2} + \hat{k}^{\frac{1}{2}}(\rho_{n_1} + \rho_{n_2})\right)\|\mathbf{G}\|. \tag{3.17}$$

*Proof.* Using

$$\begin{aligned}\mathbf{E}_{n_2}^{-T}\widehat{\mathbf{K}}\mathbf{E}_{n_1}^{-1} - \mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1} &= (\mathbf{E}_{n_2}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}(\mathbf{E}_{n_1}^{-1} - \mathbf{E}^{-1}) \\ &\quad + (\mathbf{E}_{n_2}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}\mathbf{E}^{-1} + \mathbf{E}^{-T}\widehat{\mathbf{K}}(\mathbf{E}_{n_1}^{-1} - \mathbf{E}^{-1}),\end{aligned}$$

$\|\mathbf{s}\mathbf{s}^T\| = 1$, and the symmetry and positive definiteness of $\widehat{\mathbf{K}}$, we obtain

$$\begin{aligned}&\|\mathbf{D}\| \\ &\leq \|\mathbf{N}^T(\mathbf{E}_{n_2}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}(\mathbf{E}_{n_2}^{-1} - \mathbf{E}^{-1})\,\mathbf{N}\|^{1/2}\|\mathbf{N}^T(\mathbf{E}_{n_1}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}(\mathbf{E}_{n_1}^{-1} - \mathbf{E}^{-1})\mathbf{N}\|^{1/2} \\ &\quad + \|\mathbf{N}^T(\mathbf{E}_{n_2}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}(\mathbf{E}_{n_2}^{-1} - \mathbf{E}^{-1})\mathbf{N}\|^{1/2}\,\|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\|^{1/2} \\ &\quad + \|\mathbf{N}^T(\mathbf{E}_{n_1}^{-T} - \mathbf{E}^{-T})\widehat{\mathbf{K}}(\mathbf{E}_{n_1}^{-1} - \mathbf{E}^{-1})\mathbf{N}\|^{1/2}\,\|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\|^{1/2}.\end{aligned}$$

Let us first bound the term $\|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\|^{1/2}$. Note that if for any $\mathbf{v} \in \mathbb{R}^{\widehat{\mathbf{lp}}}$

$$(\mathbf{E}^{-1}\mathbf{N}\mathbf{v}, \widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\mathbf{v}) \leq \lambda(\mathbf{v}, \mathbf{G}\mathbf{v}), \tag{3.18}$$

then $\|\mathbf{N}^T\mathbf{E}^{-T}\widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\| \leq \lambda\|\mathbf{G}\|$. The next goal is to find an upper bound for $\lambda$. As before, let $\mathbf{z} = \mathbf{E}^{-1}\mathbf{N}\mathbf{v}$. The continuous version of (3.18) can be described as how to bound $\alpha\|z\|^2_{(t_f,t_0;L^2(\Omega))} + \beta\|z(t_f)\|^2_{L^2(\Omega)}$ by $\lambda\gamma\|v\|^2_{L^2(\Omega)}$, where $z$ and $v$ satisfy the state equation (2.1). This can be obtained by using the energy method, that is, multiply (2.1) by $z(t)$, integrate on $\Omega$, and use the coerciveness of $\mathcal{A}$ to obtain

$$\frac{1}{2}\frac{d}{dt}\|z(t)\|^2_{L^2(\Omega)} \leq (v(t), z(t))_{L^2(\Omega)}. \tag{3.19}$$

Integrating in time and applying a Young inequality we obtain

$$\|z(t)\|^2_{L^2(\Omega)} \leq 2(t - t_0)\|v\|^2_{(t_0,t;L^2(\Omega))} + \frac{1}{2(t - t_0)}\|z\|^2_{(t_0,t;L^2(\Omega))}, \tag{3.20}$$

ETNA
Kent State University
http://etna.math.kent.edu



and integrating in time again we obtain

$$\|z\|^2_{(t-t_0;L^2(\Omega))} \le 4(t-t_0)^2\|v\|^2_{(t_0,t;L^2(\Omega))} \tag{3.21}$$

and

$$\|z(t)\|^2_{L^2(\Omega)} \le 4(t-t_0)\|v\|^2_{(t_0,t;L^2(\Omega))}. \tag{3.22}$$

We now consider the discrete counterparts of (3.19)–(3.22) to the backward Euler scheme. Let us denote $\mathbf{z} = [z_1^T, \ldots, z_{\hat{l}}^T]^T \in \mathbb{R}^{\widehat{lq}}$ and $\mathbf{v} = [v_1^T, \ldots, v_{\hat{l}}^T]^T \in \mathbb{R}^{\widehat{lp}}$, and let $t_l = t_0 + \tau l$. It is easy to show that the counterparts of (3.21) and (3.22) are given by

$$\tau \sum_{i=1}^{l} (z_i, M_h z_i) \le 4(t_l - t_0)^2 \tau \sum_{i=1}^{l} (v_i, R_h v_i) \tag{3.23}$$

and

$$(z_l, M_h z_l) \le 4(t_l - t_0)\tau \sum_{i=1}^{l} (v_i, R_h v_i). \tag{3.24}$$

We note that

$$\tau \sum_{i=1}^{l} (v_i, R_h v_i) \le \tau \sum_{i=1}^{\hat{l}} (v_i, R_h v_i) = (\mathbf{v}, \mathbf{G}\mathbf{v}), \tag{3.25}$$

and using properties of the mass matrix of piecewise linear functions in time we have

$$\|\underline{z}\|^2_{(t_0,t_l;L^2(\Omega))} \le \tau \sum_{i=1}^{l} (z_i, M_h z_i). \tag{3.26}$$

Hence, using (3.23)–(3.26) we obtain

$$(\mathbf{E}^{-1}\mathbf{N}\mathbf{v}, \widehat{\mathbf{K}}\mathbf{E}^{-1}\mathbf{N}\mathbf{v}) \le 4(t_f - t_0)\frac{\alpha(t_f - t_0) + \beta}{\gamma}(\mathbf{v}, \mathbf{G}\mathbf{v}). \tag{3.27}$$

Similarly, and using Lemma 3.4, we obtain

$$\begin{aligned}\Big((\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{N}\mathbf{v}, \widehat{\mathbf{K}}(\mathbf{E}_n^{-1} - \mathbf{E}^{-1})\mathbf{N}\mathbf{v}\Big) \\ \le 4(t_l - t_0)\frac{\alpha(t_f - t_0) + \beta}{\gamma}(\hat{k}\rho_n^2)(\mathbf{v}, \mathbf{G}\mathbf{v}).\end{aligned} \tag{3.28}$$

Combining the inequalities (3.27) and (3.28) with $n = n_1$ or $n = n_2$, yields the bound (3.17). This completes the proof. □

REMARK 3.6. We discuss a variant of the performance function (2.2). It consists of modifying its first term to

$$\begin{aligned}J(z(v), v) := \frac{\alpha}{2}\sum_{k=1}^{\hat{k}} \Delta\|z(v)(T_k, \cdot) - \tilde{y}(T_k, \cdot)\|^2_{L^2(\Omega)} \\ + \frac{\gamma}{2}\int_{t_0}^{t_f} \|v(t, \cdot)\|^2_{L^2(\Omega)}\, dt + \frac{\beta}{2}\|z(v)(t_f, \cdot) - \tilde{y}(t_f, \cdot)\|^2_{L^2(\Omega)}.\end{aligned} \tag{3.29}$$

ETNA
Kent State University
http://etna.math.kent.edu



This means that the discrepancy between $z(v)$ and $\tilde{y}$ is measured only at certain specific points $T_k$. In particular we force, although this is not necessary, that the points $T_k$ be in correspondence with the coarse time mesh of the parareal method. This simplifies the implementation of the cost function, and more importantly, a large savings in memory allocation is achieved since we do not need to store $z(v)$ at all fine time mesh points only at those of the coarse time mesh. It is not hard to see that Lemma 3.4 holds for this case with the same constant in (3.13). Additionally, Theorem 3.5 holds for this case with the same constant in (3.17).

We end this section with a comment on the practical use of our conditions (3.10) and (3.11). While the values of the problem dependent constants $\ell_m^{(1)}$ and $\ell_m^{(2)}$ can in principle be computed, this is not computational feasible for our kind of problems. Therefore, since we know from Theorem 3.5 that these constants exist, we try some initial values, and if need be, we modify them. We mention also that the bounds used in the proofs of Theorems 3.1 and 3.2 that give rise to this constants are by no means tight. Thus, there is a wide latitude in choosing these constants.

**4. Numerical experiments.** In this section, we describe numerical results on tests of an optimal control problem involving the following 2D-heat equation:

$$\begin{cases} z_t - \Delta z &= v, \quad x \in \Omega, \qquad 0 < t \\ z(t,0) &= 0, \quad x \in \partial\Omega, \quad 0 \le t \\ z(0,x) &= 0, \quad x \in \partial\Omega, \end{cases}$$

where $\Omega = [0,1] \times [0,1]$. We choose the performance target function:

$$\tilde{y}(x) = x_1(1-x_1)e^{-x_1}x_2(1-x_2)e^{-x_2} \quad \text{for } t \in [0,1]. \tag{4.1}$$

We choose as stopping criterion for the iterative solvers for the outer iteration $\epsilon = \|\mathbf{r}_m\|/\|\mathbf{r}_0\| \le 10^{-6}$, where $\mathbf{r}_m$ denotes the residual at the $m$th iteration. We implemented inexact FOM (IFOM) and its truncated variant TIFOM($m_T$). We concentrate on the inexactness arising from the internal tolerance of the parareal method. The stopping criteria for the inner applications of the (parareal scheme) is given by expressions (3.10) and (3.11):

$$\epsilon_{inner}^{(i)} = \ell_m^{(i)} \frac{\epsilon}{\|\mathbf{r}_{m-1}\|}, \qquad i = 1, 2. \tag{4.2}$$

Since we want a relative residual below the prescribed tolerance $\epsilon$, instead of (4.2), we use

$$\epsilon_{inner}^{(i)} = \ell_m^{(i)} \frac{\epsilon\|\mathbf{r}_0\|}{\|\mathbf{r}_{m-1}\|}, \qquad i = 1, 2. \tag{4.3}$$

REMARK 4.1. The number of applications of the parareal scheme depends on the expression (4.3). As a consequence, the number of inner iterations of the parareal method need not be equal from one outer iteration to the next.

**Experiment 1**. For this experiment, we consider $\alpha = 1$, $\beta = 12$, $\gamma = 10^{-5}$. The 2D domain $\Omega$ is discretized in a $15 \times 15$ grid ($\hat{q} = 13^2$, $h = 1/14$, and $\hat{p} = 14 \times 14 \times 2 - 2 = 390$) and the time discretization of $[0,1]$, $\tau = 1/512$ ($\hat{l} = 512$). In all cases, we use the parareal method described in Section 3.1.1 as a preconditioner with $\widehat{k} = 32$ coarse time intervals. For this problem the size of the matrix $\mathbf{G}$ is $199680 \times 199680$ (almost $200000 \times 200000$) corresponding to $\hat{p}\,\hat{l} = 390 \times 512$. The matrices $\mathbf{E}$, $\mathbf{N}$ are of size $(13^2 \times 512) \times (13^2 \times 512)$ and $(13^2 \times 512) \times (390 \times 512)$, respectively. The results (number of outer iterations and

**ETNA**
Kent State University
http://etna.math.kent.edu



TABLE 4.1
*Outer (inner)-iterations comparison between* IFOM *and* TIFOM($m_T$). *Outer tolerance* $\epsilon = 10^{-6}$, $\ell_m^{(1)} = \ell_m^{(2)}$, $\alpha = 1$, $\beta = 12$, $\gamma = 10^{-5}$, *mesh grid size of* $15 \times 15$, $\tau = 1/512$, $\hat{k} = 32$, $\Delta T/\tau = 16$, *and n.c. means that the algorithm does not converge for* 100 *iterations.*

| $\ell_m^{(i)}\epsilon$ | IFOM | | TIFOM | | | |
|---|---|---|---|---|---|---|
| | | | $m_T = 2$ | | $m_T = 4$ | |
| | Eq. (4.1) | Eq. (4.4) | Eq. (4.1) | Eq. (4.4) | Eq. (4.1) | Eq. (4.4) |
| $10^{-12}$ | 15(576) | 15(566) | 16(610) | 16(600) | 16(608) | 16(596) |
| $10^{-10}$ | 15(482) | 15(476) | 17(532) | 17(526) | 17(528) | 17(524) |
| $10^{-8}$ | 15(388) | 15(378) | 17(426) | 16(402) | 17(426) | 16(400) |
| $10^{-7}$ | 15(340) | 15(328) | 18(394) | 18(374) | 17(374) | 16(350) |
| $10^{-6}$ | 15(288) | 15(284) | 19(340) | 17(316) | 19(338) | 19(334) |
| $10^{-5}$ | 17(238) | 16(222) | 24(298) | 22(272) | 21(266) | 26(284) |
| $10^{-4}$ | 17(180) | 16(174) | n.c. | 48(286) | 22(210) | 70(406) |
| $\ell_m^{(i)}\epsilon$ | | | $m_T = 8$ | | $m_T = 12$ | |
| | | | Eq. (4.1) | Eq. (4.4) | Eq. (4.1) | Eq. (4.4) |
| $10^{-12}$ | | | 15(576) | 15(566) | 15(576) | 15(566) |
| $10^{-10}$ | | | 16(504) | 16(498) | 15(482) | 15(476) |
| $10^{-8}$ | | | 17(420) | 17(412) | 15(388) | 15(378) |
| $10^{-7}$ | | | 17(368) | 17(354) | 15(340) | 15(328) |
| $10^{-6}$ | | | 18(320) | 16(296) | 16(298) | 16(294) |
| $10^{-5}$ | | | 19(258) | 19(244) | 19(242) | 19(244) |
| $10^{-4}$ | | | 28(254) | 20(182) | 20(192) | 20(188) |

number of applications of the parareal method) corresponding to IFOM and TIFOM($m_T$) with $m_T = 2$, $m_T = 8$, and $m_T = 12$ orthogonal vectors are presented in the Table 4.1. We force the same $\ell_m^{(i)}\epsilon$ for each application of the parareal method, i.e., $\ell_m^{(1)} = \ell_m^{(2)}$ in (4.2). Observe that as expected, in all cases, IFOM converges to the prescribed tolerance in fewer outer iterations than the truncated versions. However, note that TIFOM does converge in all cases with a relatively small increase in the number of total iterations, i.e., the delay is small but with the concomitant savings in storage. This is further illustrated in Figure 4.2(a) and (b), where a fixed outer tolerance of $10^{-6}$ and $\ell_m^{(i)}\epsilon = 10^{-5}$ is considered. Observe that the computed residual converges below the outer tolerance $\epsilon$, and $\ell_m^{(i)}\epsilon = 10^{-5}$ roughly specifies the accuracy of the true solution $\mathbf{r} = \mathbf{b} - \mathbf{H}\mathbf{u}$.

Note also that if one compares in Table 4.1 the rows corresponding to the $\ell_m^{(i)}\epsilon = 10^{-12}$ (closer to exact FOM) with that of $\ell_m^{(i)}\epsilon = 10^{-6}$, for example, one can appreciate the savings of almost 50% in total computational effort. This is consistent with the savings shown in [28, 32] for other problems.

In Figure 4.1 we show the contour plot of two slices corresponding to times $t = 0.5$ and $t = 1$ of the exact ((a) and (d)) and inexact solution ((b) and (e)) and the difference between them ((c) and (f)). We use the TIFOM(8) and $\ell_m^{(i)}\epsilon = 10^{-5}$. The comparison between the difference between the exact and inexact solution (see for example plot 4.1(f)) reveals that the worst case difference is attained at $t = 1$ being of order of $10^{-7}$.

In general, when $\ell_m^{(i)}\epsilon$ decreases from $10^{-7}$ to $10^{-3}$, the true residual deteriorates (see Figure 4.2). For the cases reported, the true residual stagnates, being the stagnation point directly dependent on $\ell_m^{(i)}\epsilon$. In principle, although reducing the internal tolerance is recommended to save computational time, there are limits. We have reported experiments where we can not guarantee to satisfy the hypotheses of Theorem 3.2 and therefore no longer possible guarantee the convergence of the inexact method.

ETNA
Kent State University
http://etna.math.kent.edu



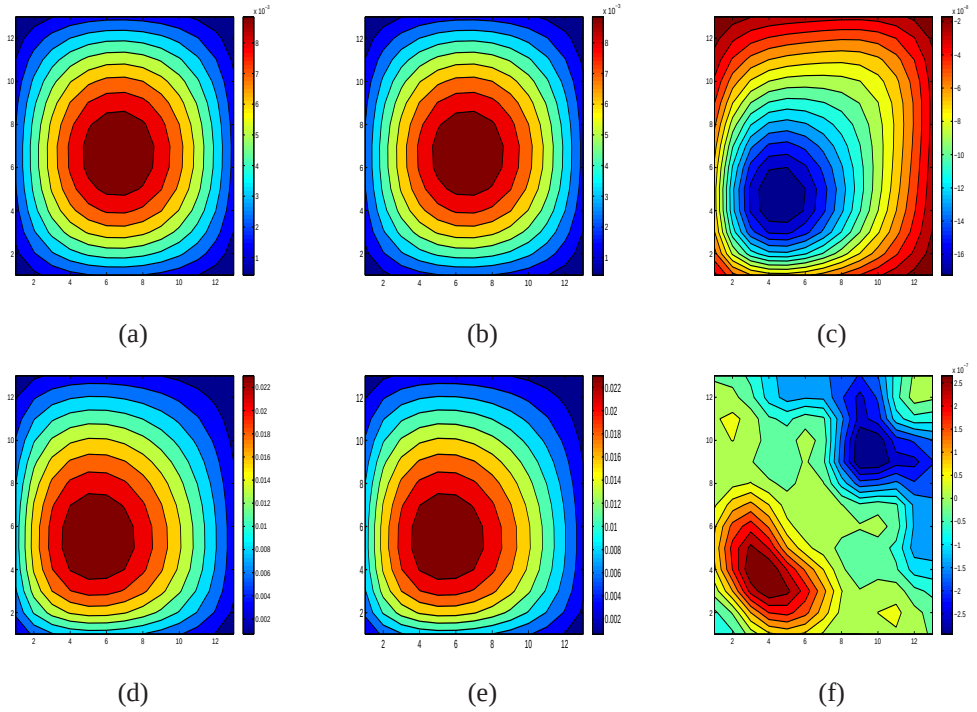


FIG. 4.1. *Contour plot of slice at time $t = 0.5$:* (a) *exact solution,* (b) *inexact solution, and* (c) *difference between the exact and inexact solution. Contour plot of slice at time $t = 1$:* (d) *exact solution,* (e) *inexact solution, and* (f) *difference between the exact and inexact solution. Truncation parameter $m_T = 8$ and outer tolerance $10^{-6}$ and $\ell_m^{(i)}\epsilon = 10^{-5}$.*

To illustrate the robustness of the proposed method for different smoothness of the target functions, we run the same experiments with a target function which is not smooth in time like (4.1), namely the following discontinuous in time target function:

$$\begin{cases} \tilde{y}(x) = x_1(1-x_1)e^{-x_1}x_2(1-x_2)e^{-x_2} & \text{for} \quad t \in [0, 0.5], \\ \tilde{y}(x) = 2x_1(1-x_1)e^{-x_1}x_2(1-x_2)e^{-x_2} & \text{for} \quad t \in (0.5, 1]. \end{cases} \tag{4.4}$$

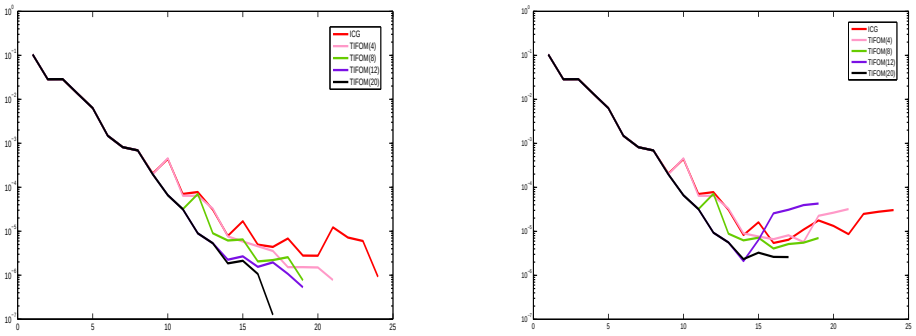
The results are shown at Table 4.1. It can be observed that the conclusions obtained with the target function (4.1) remain valid with the target function (4.4). Therefore, there is no special bias in performing our numerical tests using the smooth target function (4.1). We do this for the rest of the paper.

**Experiment 2. Scalability.** Here we consider the same problem as in Experiment 1 to study the convergence, mainly the variation in the number of iterations with respect to the discretization parameters $\tau$, $\widehat{k}$, and $\widehat{q}$ in terms of the strong and weak scalability of TIFOM(8) when parareal is used in $\mathbf{E}_n^{-T}\widehat{\mathbf{K}}\mathbf{E}_n^{-1}$. The results are summarized at Tables 4.2 and 4.3.

In Table 4.2 we list the number of outer (inner) iterations required to solve the system (3.2) varying the values of $\ell_m^{(i)}\epsilon$ to be $10^{-7}$, $10^{-6}$, and $10^{-5}$. Different mesh grid corresponding to different space variables, i.e., $\widehat{q} = 6^2$, $\widehat{q} = 9^2$, and $\widehat{q} = 13^2$ (corresponding to 36, 81 and 169 space variables) and different coarse time steps $\Delta T$ are tested with fixed $\tau = 1/512$ and parameters $\alpha = 1$, $\beta = 12$, $\gamma = 10^{-5}$.Table 4.2 shows how the number of iterations varies with the respect to $\widehat{k}$ for a fixed total problem size. Observe that for a variety of grid sizes, the number of outer and inner iterations remain approximately constant

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**






(a) (b)

FIG. 4.2. (a) *Computed residual and* (b) *true residual. TIFOM with* $m_T = 20$ *(black),* $m_T = 12$ *(blue),* $m_T = 8$ *(green),* $m_T = 4$ *(pink), and ICG (red). The outer tolerance is* $10^{-6}$ *and* $\ell_m^{(i)}\epsilon = 10^{-5}$.

TABLE 4.2
$TIFOM(8)$ *number of outer (inner) iterations. The parameters are outer tolerance* $\epsilon = 10^{-6}$, $\alpha = 1$, $\beta = 12$, *and* $\gamma = 10^{-5}$. *Backward Euler discretization is used with* $\tau = 1/512$ *and backward Euler coarse propagator with* $\hat{k} = 1/(\Delta T)$. $\ell_m^{(1)} = \ell_m^{(2)}$ *and the* $\ell_m^{(i)}\epsilon$ *values are* $10^{-7}, 10^{-6}$, *and* $10^{-5}$.

| $\hat{k}$ | 8 | 16 |
|---|---|---|
| $\Delta T/\tau$ | 64 | 32 |
| $\widehat{q} = 6^2$ | 25(398) 28(396) 31(344) | 25(480) 27(434) 44(468) |
| $\widehat{q} = 9^2$ | 22(358) 22(326) 25(278) | 21(416) 22(378) 24(308) |
| $\widehat{q} = 13^2$ | 17(286) 17(262) 19(226) | 17(354) 17(308) 19(264) |
| $\hat{k}$ | 32 | 64 |
| $\Delta T/\tau$ | 16 | 8 |
| $\widehat{q} = 6^2$ | 23(466) 25(412) 36(380) | 23(436) 25(380) 197(1494) |
| $\widehat{q} = 9^2$ | 22(436) 22(376) 35(384) | 21(314) 22(336) 51(434) |
| $\widehat{q} = 13^2$ | 17(368) 18(320) 19(258) | 17(330) 17(274) 18(226) |

if an inner tolerance $\ell_m^{(i)}\epsilon < 10^{-5}$ is used. This indicates that TIFOM(8) when combined with the parareal method for approximating $\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}$, i.e., $\mathbf{E}_n^{-T}\widehat{\mathbf{K}}\mathbf{E}_n^{-1}$, is independent of the coarse grid discretization if an adequate $\ell_m^{(i)}\epsilon$ is taken.

In Table 4.3, we analyze how the number of iterations varies with respect to $\widehat{k}$ for a fixed problem size, i.e., in this case the number of fine temporal subintervals inside each coarse temporal subintervals is set to $\Delta T/\tau = 16$. Different grid sizes, coarse time steps $\Delta T$, and $\ell_m^{(i)}\epsilon$ are tested. Observe that the TIFOM(8) with the parareal method is robust and scalable for the $\ell_m^{(i)}\epsilon$ tested when the size of the problem is increased maintaining fixed the size of each problem inside each subdomain.

**Experiment 3.** $\ell_m^{(1)}\epsilon \neq \ell_m^{(2)}\epsilon$**.** Here we consider the effect of forcing that $\ell_m^{(1)} \neq \ell_m^{(2)}$ to test the applicability of the method to the case where the two systems in (1.3) are solved with different (inner) tolerances, resulting in a nonsymmetric matrix $\mathcal{H}$. In the experiments we consider the same problem as in Experiment 1.

In Table 4.4 the results of imposing different $\ell_m^{(i)}$ are shown. We present the number of outer and inner (number of applications of the parareal method) iterations corresponding to IFOM, TIFOM($m_T$) with $m_T = 2$, $m_T = 8$, and $m_T = 12$ orthogonal vectors. From these

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**



TABLE 4.3
*$TIFOM(8)$ number of outer (inner) iterations. The parameters are outer tolerance $\epsilon = 10^{-6}$, $\alpha = 1$, $\beta = 12$, and $\gamma = 10^{-5}$. Backward Euler discretization is used with $\tau = 1/\hat{l}$. $\ell_m^{(1)} = \ell_m^{(2)}$, and the $\ell_m^{(i)}\epsilon$ values are $10^{-7}, 10^{-6}$, and $10^{-5}$. The number of time intervals in each subdomain is $\Delta T/\tau = 16$.*

| $\hat{k}$ | 8 | 16 | 32 |
|---|---|---|---|
| $\hat{l}$ | 128 | 256 | 512 |
| $\widehat{q} = 6^2$ | 19(306) 23(316) 25(258) | 22(424) 27(410) 40(422) | 23(466) 25(412) 36(380) |
| $\widehat{q} = 9^2$ | 17(274) 19(272) 21(228) | 21(383) 20(364) 29(318) | 22(436) 22(376) 30(384) |
| $\widehat{q} = 13^2$ | 14(230) 15(224) 17(202) | 16(328) 16(286) 17(242) | 17(368) 18(320) 19(258) |

TABLE 4.4
*Comparison between the IFOM and TIFOM($m_T$). Outer tolerance $\epsilon = 10^{-6}$, $\ell_m^{(1)}\epsilon \neq \ell_m^{(2)}\epsilon$, $\alpha = 1$, $\beta = 12$, and $\gamma = 10^{-5}$. Mesh grid size of $15 \times 15$, $\tau = 1/512$, $\hat{k} = 32$, $\Delta T/\tau = 16$, and n.c. means that the algorithm doest not converge for 100 outer iterations.*

| $\ell_m^{(1)}\epsilon$ | $\ell_m^{(2)}\epsilon$ | IFOM | TIFOM o-iter. (i-iter.) | | | |
|---|---|---|---|---|---|---|
| | | o-iter. (i-iter) | $m_T = 2$ | $m_T = 4$ | $m_T = 8$ | $m_T = 12$ |
| $10^{-7}$ | $10^{-7}$ | 15(340) | 18(394) | 17(334) | 17(368) | 15(340) |
| $10^{-7}$ | $10^{-6}$ | 15(340) | 19(408) | 19(400) | 18(382) | 15(340) |
| $10^{-7}$ | $10^{-5}$ | 16(352) | 22(454) | 22(442) | 20(404) | 17(364) |
| $10^{-7}$ | $10^{-4}$ | 16(354) | n.c. | 39(656) | 20(424) | 17(368) |
| $10^{-6}$ | $10^{-7}$ | 15(288) | 18(330) | 19(338) | 17(310) | 16(298) |
| $10^{-6}$ | $10^{-6}$ | 15(288) | 19(340) | 19(338) | 18(320) | 16(298) |
| $10^{-6}$ | $10^{-5}$ | 16(300) | 23(390) | 20(346) | 19(334) | 17(310) |
| $10^{-6}$ | $10^{-4}$ | 16(300) | 51(710) | 37(520) | 20(352) | 17(310) |
| $10^{-5}$ | $10^{-7}$ | 15(234) | 19(270) | 27(280) | 18(254) | 21(246) |
| $10^{-5}$ | $10^{-6}$ | 15(234) | 19(270) | 29(284) | 18(254) | 19(242) |
| $10^{-5}$ | $10^{-5}$ | 17(238) | 24(298) | 21(266) | 19(258) | 19(242) |
| $10^{-5}$ | $10^{-4}$ | 17(240) | 30(334) | 31(356) | 20(270) | 19(244) |

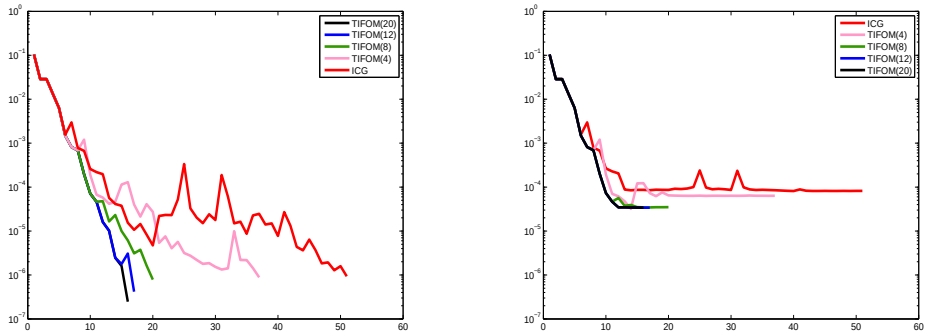
experiments, it can be observed that more than two vectors are needed to attain convergence, i.e., in some situations (when $m_T = 2$), the number of storage vectors do not suffice to guarantee the convergence of the method. In fact, when the difference between $\ell_m^{(1)}\epsilon$ and $\ell_m^{(2)}\epsilon$ is large, then ICG is not expected to work well. In all cases, however, the convergence of TIFOM is attained with storage savings for $m_T = 4$, $m_T = 8$, and $m_T = 12$. Observing TIFOM, we note that the more asymmetric the matrix $\mathcal{H}$ is, more orthogonal vectors are required to obtain the same result, i.e., $m_T$ must be increased.

In Figure 4.3 we report the computed and true residual behavior for $\ell_m^{(1)}\epsilon = 10^{-6}$ and $\ell_m^{(2)}\epsilon = 10^{-4}$. The influence of $\ell_m^{(2)}\epsilon > \ell_m^{(1)}\epsilon$ can be observed since the true residual stagnates around $\ell_m^{(2)}\epsilon = 10^{-4}$. Observe also in the computed residual the delay in the convergence of the ICG when $\ell_m^{(1)} \neq \ell_m^{(2)}$ corroborating the results of Table 4.4.

In Figure 4.4 we show the contour plot of two slices corresponding to times $t = 0.5$ and $t = 1$ of the exact ((a) and (d)) and inexact solution ((b) and (e)) and the difference between them ((c) and (f)). We use the TIFOM(8), $\ell_m^{(1)}\epsilon = 10^{-6}$ and $\ell_m^{(2)}\epsilon = 10^{-4}$. The comparison between the difference between the exact and inexact solution (see for example plot 4.4(f)) reveals that the worst case difference is being of order of $10^{-6}$.

ETNA
Kent State University
http://etna.math.kent.edu






(a) (b)

FIG. 4.3. *(a) Computed residual and (b) true residual for TIFOM with $m_T = 20$ (black), $m_T = 12$ (blue), $m_T = 8$ (green), $m_T = 4$ (pink), and ICG (red). The outer tolerance is $10^{-6}$ and $\ell_m^{(1)}\epsilon = 10^{-6}$ and $\ell_m^{(2)}\epsilon = 10^{-4}$.*

TABLE 4.5
*Comparison between* IFOM *and* TIFOM($m_T$). *Outer tolerance $\epsilon = 10^{-6}$, $\ell_m^{(1)}\epsilon = \ell_m^{(2)}\epsilon$, functional (3.29), $\tilde{\alpha} = 1$, $\tilde{\beta} = 12$, mesh grid size $15 \times 15$, $\tau = 1/512$, $\hat{k} = 32$, $\Delta T/\tau = 16$, and n.c. means that the algorithm does not converge for 100 iterations.*

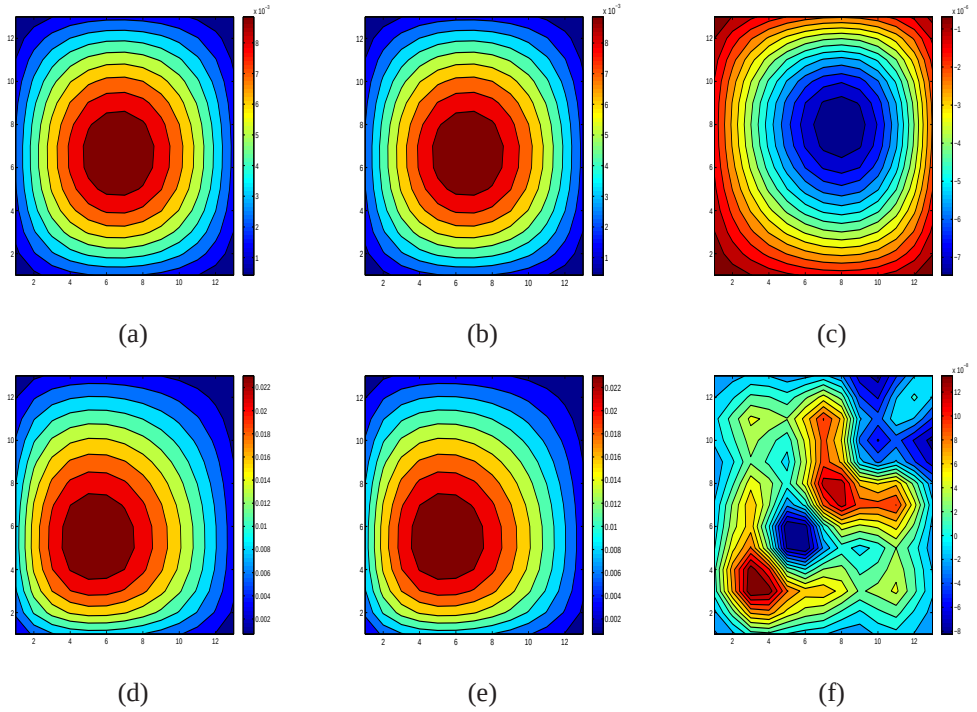
| $\ell_m^{(i)}\epsilon$ | IFOM | | TIFOM | |
|---|---|---|---|---|
| | | | $m_T = 2$ | |
| | o-iter (i-iter) | | o-iter (i-iter) | |
| | $\tilde{\gamma} = 10^{-3}$ | $\tilde{\gamma} = 10^{-5}$ | $\tilde{\gamma} = 10^{-3}$ | $\tilde{\gamma} = 10^{-5}$ |
| $10^{-7}$ | 5(112) | 15(316) | 5(112) | 17(356) |
| $10^{-6}$ | 5( 92) | 15(162) | 5( 92) | 17(300) |
| $10^{-5}$ | 5( 74) | 17( 90) | 5( 74) | n.c. |
| $\ell_m^{(i)}\epsilon$ | TIFOM | | | |
| | $m_T = 4$ | | $m_T = 8$ | |
| | o-iter (i-iter) | | o-iter (i-iter) | |
| | $\tilde{\gamma} = 10^{-3}$ | $\tilde{\gamma} = 10^{-5}$ | $\tilde{\gamma} = 10^{-3}$ | $\tilde{\gamma} = 10^{-5}$ |
| $10^{-7}$ | 5(112) | 16(340) | 5(112) | 17(338) |
| $10^{-6}$ | 5( 92) | 18(310) | 5( 92) | 16(282) |
| $10^{-5}$ | 5( 74) | 27(296) | 5( 74) | 18(230) |

**Experiment 4. Functional (3.29).** Here we are interested in the analysis of the functional (3.29) introduced in Remark 3.6. To this end we take the same problem as in the Experiment 1 but now with the functional (3.29) and we perform variations on its parameters $\tilde{\alpha}$, $\tilde{\beta}$, and $\tilde{\gamma}$. We first analyze the influence of $\tilde{\gamma}$ in the solution determined by (3.29) since it is associated to the regularization term. In Table 4.5, we take $\tilde{\alpha} = 1$, $\tilde{\beta} = 12$, and $\tilde{\gamma}$ with values $10^{-3}$ and $10^{-5}$. Observe that when $\tilde{\gamma}$ is reduced, then the number of iterations increases. This shows the sensitivity of the problem to $\tilde{\gamma}$. These results are in accordance with previous published works related to this problem (see [22]) and with Theorem 3.5.

Tables 4.6 and 4.7 show the influence of $\tilde{\alpha}$ and $\tilde{\beta}$, respectively. It can be observed that the number of outer iterations almost remains invariant when $\alpha$ is modified, but the method can be sensible to converge if $\ell_m^{(i)}\epsilon$ is reduced (to values equal or lower to $10^{-5}$) and $m_T = 2$. This is due to the fact that with the functional (3.29), the condition for convergence established in Theorem 3.2 for the TIFOM(2) is no longer satisfied.

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**





(a) (b) (c)

(d) (e) (f)

FIG. 4.4. *Contour plot of slice at time $t = 0.5$: (a) exact solution, (b) inexact solution, and (c) difference between the exact and inexact solution. Contour plot of slice at time $t = 1$: (d) exact solution, (e) inexact solution, and (f) difference between the exact and inexact solution. Truncation parameter $m_T = 8$ and outer tolerance $\epsilon = 10^{-6}$ and $\ell_m^{(1)}\epsilon = 10^{-6}$ and $\ell_m^{(2)}\epsilon = 10^{-4}$.*

A similar conclusion can be reached for the sensibility of TIFOM with respect to $\beta$ in terms of $\ell_m^{(i)}\epsilon$. In this case, however, TIFOM is more sensitive to $\beta$. In general we can expect from Theorem 3.5 that when $\tilde{\alpha}$ and $\tilde{\beta}$ increases, the number of outer iterations of the TIFOM is increased.

**5. Concluding remarks.** We have proposed the use of inexact and truncated Krylov subspace methods for the solution of the linear systems arising in the discretization of parabolic control problems. We use the reduced Hessian approach, resulting in a symmetric positive definite system for which one would normally use the Conjugate Gradient (CG) method. Since the reduced Hessian is expressed as a matrix product, and two of this matrices involve solutions of very large linear systems, we only approximate their solution (leading to inexact methods), thus resulting in a nonsymmetric system. We choose inexact FOM (which would reduce to CG in the absence of nonsymmetry). The approximation of the large systems are done with the parareal method.

Our experiments show that the truncated inexact FOM can produce good results while saving in storage (because of the truncation) and computational expense (because of the inexactness). Furthermore, the number of (outer) FOM iterations remains constant for a large range of temporal and spacial discretizations, illustrating the robustness and scalability of the proposed approach.

**Acknowledgements.** We would like to thank Eldad Haber for his comments on an earlier version of the paper. We also thank the two anonymous referees whose remarks and questions led to improvements in our presentation.

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**



TABLE 4.6

*Comparison between* IFOM, *and* TIFOM($m_T$). *Outer tolerance* $\epsilon = 10^{-6}$, $\ell_m^{(1)} = \ell_m^{(2)}$, *functional (3.29) with* $\tilde{\beta} = 12$ *and* $\tilde{\gamma} = 10^{-5}$. *In addition, the mesh grid size* $15 \times 15$, $\tau = 1/512$, $\hat{k} = 32$, $\Delta T/\tau = 16$, *and n.c. means that the algorithm does not converge for 100 iterations.*

| $\ell_m^{(i)}\epsilon$ | IFOM | | | TIFOM | | |
|---|---|---|---|---|---|---|
| | | | | $m_T = 2$ | | |
| | o-iter (i-iter) | | | o-iter (i-iter) | | |
| | $\tilde{\alpha} = 0$ | $\tilde{\alpha} = 1$ | $\tilde{\alpha} = 100$ | $\tilde{\alpha} = 0$ | $\tilde{\alpha} = 1$ | $\tilde{\alpha} = 100$ |
| $10^{-7}$ | 13(220) | 15(316) | 15(338) | 15(246) | 17(356) | 17(370) |
| $10^{-6}$ | 13(178) | 15(268) | 16(174) | 16(214) | 17(300) | 17(340) |
| $10^{-5}$ | 13(154) | 15(208) | 16(104) | 20(146) | n.c | n.c |

| $\ell_m^{(i)}\epsilon$ | TIFOM | | | | | |
|---|---|---|---|---|---|---|
| | $m_T = 4$ | | | $m_T = 8$ | | |
| | o-iter (i-iter) | | | o-iter (i-iter) | | |
| | $\tilde{\alpha} = 0$ | $\tilde{\alpha} = 1$ | $\tilde{\alpha} = 100$ | $\tilde{\alpha} = 0$ | $\tilde{\alpha} = 1$ | $\tilde{\alpha} = 100$ |
| $10^{-7}$ | 15(248) | 16(340) | 16(356) | 14(232) | 17(338) | 17(362) |
| $10^{-6}$ | 17(222) | 18(310) | 19(336) | 15(196) | 16(282) | 17(314) |
| $10^{-5}$ | 19(188) | 27(296) | 23(280) | 15(164) | 18(230) | 21(282) |

TABLE 4.7

*Comparison between* IFOM, *and* TIFOM($m_T$). *Outer tolerance* $\epsilon = 10^{-6}$, $\ell_m^{(1)} = \ell_m^{(2)}$. *Functional (3.29) with* $\tilde{\alpha} = 1$ *and* $\tilde{\gamma} = 10^{-5}$. *In addition, the mesh grid size* $15 \times 15$, $\tau = 1/512$, $\hat{k} = 32$, $\Delta T/\tau = 16$, *and n.c. means that the algorithm does not converge for 100 iterations.*

| $\ell_m^{(i)}\epsilon$ | IFOM | | | TIFOM | | |
|---|---|---|---|---|---|---|
| | | | | $m_T = 2$ | | |
| | o-iter (i-iter) | | | o-iter (i-iter) | | |
| | $\tilde{\beta} = 0$ | $\tilde{\beta} = 1$ | $\tilde{\beta} = 12$ | $\tilde{\beta} = 0$ | $\tilde{\beta} = 1$ | $\tilde{\beta} = 12$ |
| $10^{-7}$ | 3(78) | 8(194) | 15(316) | 3(78) | 8(194) | 17(356) |
| $10^{-6}$ | 3(70) | 8(166) | 15(162) | 3(70) | 9(166) | 17(300) |
| $10^{-5}$ | 3(58) | 9(136) | 17( 90) | 3(58) | 9(138) | n.c. |

| $\ell_m^{(i)}\epsilon$ | TIFOM | | | | | |
|---|---|---|---|---|---|---|
| | $m_T = 4$ | | | $m_T = 8$ | | |
| | o-iter (i-iter) | | | o-iter (i-iter) | | |
| | $\tilde{\beta} = 0$ | $\tilde{\beta} = 1$ | $\tilde{\beta} = 12$ | $\tilde{\beta} = 0$ | $\tilde{\beta} = 1$ | $\tilde{\beta} = 12$ |
| $10^{-7}$ | 3(78) | 8(194) | 16(340) | 3(78) | 8(194) | 17(338) |
| $10^{-6}$ | 3(70) | 8(166) | 18(310) | 3(70) | 8(166) | 16(282) |
| $10^{-5}$ | 3(58) | 8(136) | 27(296) | 3(58) | 8(136) | 18(230) |

## REFERENCES

[1] S. S. ADAVANI AND G. BIROS, *Multigrid algorithms for inverse problems with linear parabolic PDE constraints*, SIAM J. Sci. Comput., 31 (2008), pp. 369–397.

[2] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[3] G. BIROS AND O. GHATTAS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-Constrained optimization. I. The Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[4] ———, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. II. The Lagrange-Newton solver and its application to optimal control of steady viscous flows*, SIAM J. Sci. Comput., 27 (2005), pp. 714–739.

[5] A. BOURAS AND V. FRAYSSÉ, *Inexact matrix-vector products in Krylov methods for solving linear systems: a relaxation strategy*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 660–678.

[6] F. CURTIS AND J. NOCEDAL, *Steplength selection in interior-point methods*, Applied Math. Lett., 20 (2007), pp. 516–523.

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**



[7] X. DU, E. HARBER, M. KARAMPATAKI, AND D. B. SZYLD, *Varying iteration accuracy using inexact CG in control problems governed by PDE's*, technical report, Research Report 08-06-27, Department of Mathematics, Temple University, 2008. To appear in the Proceedings of the 2nd Annual International Conference on Computational Mathematics, Computational Geometry and Statistics (CMCGS 2013).

[8] X. DU AND D. B. SZYLD, *Inexact GMRES for singular linear system*, BIT, 48 (2008), pp. 511–531.

[9] M. EIERMANN AND O. G. ERNST, *Geometric aspects in the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[10] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578.

[11] ———, *On the superlinear and linear convergence of the parareal algorithm*, in O. B. Widlund and D. E. Keyes, eds., Domain Decomposition Methods in Science and Engineering XVI, volume 55 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2007, pp. 291–298.

[12] L. GIRAUD, S. GRATTON, AND J. LANGOU, *Convergence in backward error of relaxed GMRES*, SIAM J. Sci. Comput., 29 (2007), pp. 710–728.

[13] S. GRATTON, PH. L. TOINT, AND J. TSHIMANGA ILUNGA, *Range-space variants and inexact matrix-vector products in Krylov solvers for linear systems arising from inverse problems*, SIAM. J. Matrix Anal. Appl., 32 (2011), pp. 969–986.

[14] E. HABER AND U. M. ASCHER *Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 22 (2001), pp. 1943–1961.

[15] E. HABER, U. M. ASCHER, AND D. OLDENBURG, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.

[16] C. T. KELLEY, *Iterative Methods for Optimization. Frontiers in Applied Mathematics*, SIAM, Philadelphia, 1999.

[17] J. L. LIONS, *Optimal control of systems governed by partial differential equations*, Springer, Berlin, 1971.

[18] J. L. LIONS, Y. MADAY, AND G. TURINICI, *Résolution d'edp par un schéma en temps pararéel*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.

[19] T. MATHEW, M. SARKIS, AND C. E. SCHAERER, *Analysis of block matrix preconditioners for elliptic optimal control problems*, Numer. Linear Algebra Appl., 14 (2007), pp. 257–279.

[20] ———, *Temporal domain decomposition for a linear quadratic optimal control problems*, in M. Daydé, J. M. L. M. Palma, Á. L. G. A. Couthino, E. Pacitti, and J. Correia Lopes, eds., 7th International Conference on High Performance Computing in Computational Sciences, volume 4395 of Lecture Notes in Computer Sciences, Springer, Berlin, 2007, pp. 452–465.

[21] ———, *Block diagonal parareal preconditioner for parabolic optimal control problems*, in U. Langer, M. Discacciati, D. E. Keyes, O. Widlungd, and W. Zulehner, eds., Domain Decomposition Methods in Science and Engineering XVII, volume 60 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2008, pp. 409–416.

[22] ———, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM J. Sci. Comput., 32 (2010), pp. 1180–1200.

[23] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization. Second Edition. Springer Series in Operations Research and Financial Engineering*, Springer, New York, 2006.

[24] Y. NOTAY, *Flexible conjugate gradient*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460.

[25] T. RESS, M. STOLL, AND A. WATHEN, *All-at-once solution of time-dependent PDE-constrained optimization problems*, Kybernetika, 46 (2010), pp. 341–360.

[26] Y. SAAD, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[27] ———, *Iterative Methods for Sparse Linear Systems*, 2nd. ed., SIAM, Philadelphia, 2003.

[28] V. SIMONCINI AND D. B. SZYLD, *Theory of inexact Krylov subspace mehods and applications to scientific computing*, SIAM J. Sci. Comp., 25 (2003), pp. 454–477.

[29] ———, *The effect of non-optimal bases on the convergence of Krylov subspace methods*, Numer. Math., 100 (2005), pp. 711–733.

[30] ———, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[31] J. VAN DEN ESHOF AND G. L. G. SLEIJPEN, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.

[32] J. VAN DEN ESHOF, G. L. G. SLEIJPEN, AND M. B. VAN GIJZEN, *Relaxation strategies for nested Krylov methods*, J. Comput. Appl. Math., 177 (2005), pp. 347–365.