Efficient Adaptive Stochastic Collocation Strategies for Advection-Diffusion Problems with Uncertain Inputs

DOI:
10.1007/s10915-023-02247-w

Document Version
Accepted author manuscript

Springer manuscript No.
(will be inserted by the editor)

Efficient Adaptive Stochastic Collocation Strategies for Advection-Diffusion Problems with Uncertain Inputs

Benjamin M. Kent · Catherine E. Powell · David J. Silvester · Małgorzata J. Zimoń

Received: date / Accepted: date

Abstract Physical models with uncertain inputs are commonly represented as parametric partial differential equations (PDEs). That is, PDEs with inputs that are expressed as functions of parameters with an associated probability distribution. Developing efficient and accurate solution strategies that account for errors on the space, time and parameter domains simultaneously is highly challenging. Indeed, it is well known that standard polynomial-based approximations on the parameter domain can incur errors that grow in time. In this work, we focus on advection–diffusion problems with parameter-dependent wind fields. A novel adaptive solution strategy is proposed that allows users to combine stochastic collocation on the parameter domain with off-the-shelf adaptive timestepping algorithms with local error control. This is a non-intrusive strategy that builds a polynomial-based surrogate that is adapted sequentially in time. The algorithm is driven by a so-called hierarchical estimator for the parametric error and balances this against an estimate for the global timestepping error which is derived from a scaling argument.

Keywords parametric PDEs, stochastic collocation, adaptivity, error estimation

Mathematics Subject Classification (2020) 65C20 · 65M22 · 65M15 · 65M70

This work was partially supported by the EPSRC grants EP/V048376/1 and EP/W033801/1.

B.M. Kent
Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK E-mail: benjamin.kent@mancehster.ac.uk

C.E. Powell
Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK E-mail: catherine.powell@manchester.ac.uk

D.J. Silvester
Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK E-mail: david.silvester@manchester.ac.uk

M.J. Zimoń
IBM Research Europe, Daresbury Laboratory, Keckwick Lane, Warrington, WA4 4AD, UK E-mail: malgorzata.zimon@uk.ibm.com

1 Introduction

Parametric partial differential equations (PDEs) are commonly used to model physical problems with uncertain inputs that are represented as random variables. Briefly, if the uncertain inputs can be expressed as functions of a finite set of independent random variables $\xi_i : \Omega \to \Gamma_i \subset \mathbb{R}$, $i = 1,\dots,d$ on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, then with the change of variables $y_i = \xi_i(\omega)$, $i = 1,\dots d$, we may recast the inputs and PDE solution as functions of the parameter vector $\mathbf{y} = [y_1,\dots,y_d]^\top \in \Gamma$ where $\Gamma := \Gamma_1 \times \cdots \times \Gamma_d \subset \mathbb{R}^d$ denotes the parameter domain. Denoting the Borel σ-algebra by $\mathscr{B}(\Gamma)$ and the joint probability density by $\rho(\mathbf{y}) = \prod_{i=1}^d \rho_i(y_i)$, where $\rho_i(y_i)$ is the probability density of ξ_i we can formulate the problem and compute statistical quantities of interest with respect to the equivalent probability space $(\Gamma, \mathscr{B}(\Gamma), \rho(\mathbf{y})d\mathbf{y})$.

In this work, we are interested in the efficient and accurate numerical solution of parametric time-dependent advection-diffusion problems of the form

$$\frac{\partial}{\partial t}u(\mathbf{x},t,\mathbf{y}) - \varepsilon \Delta u(\mathbf{x},t,\mathbf{y}) + \mathbf{w}(\mathbf{x},\mathbf{y}) \cdot \nabla u(\mathbf{x},t,\mathbf{y}) = 0, \tag{1}$$

where the d input parameters y_i are images of random variables that encode our uncertainty about the spatially-varying advection field or 'wind' $\mathbf{w}$. Such time-dependent parametric PDEs are more challenging to solve than their steady-state counterparts. Different choices of $\mathbf{y}$ yield different space-time solutions that may be pushed further away from each other in time. This makes $u(\mathbf{x},t,\mathbf{y})$ harder to approximate as a function of $\mathbf{y}$ as time $t \to \infty$. This is not a new observation, for example it is discussed in [38], where the authors investigate a transport problem with uncertain velocity.

Numerical solution strategies for parametric PDEs can be categorised as either non-intrusive methods, which involve the solution of decoupled deterministic problems for particular choices of $\mathbf{y}$ or intrusive methods, based on Galerkin projection, which require the solution of a single large coupled problem. Among these, polynomial-based approximation schemes such as stochastic collocation [39, 27, 1] and generalised polynomial chaos (gPC) approximation [40, 2] that lead to surrogates that can be cheaply evaluated for any $\mathbf{y} \in \Gamma$ are useful for tasks like uncertainty quantification, optimisation and design. Non-intrusive schemes that allow the reuse of existing space-time solvers are especially appealing for practitioners.

Both intrusive and non-intrusive solution strategies have been extensively studied for parametric elliptic PDEs and state-of-the-art adaptive solvers exist for such problems. Anisotropic [26] and adaptive [25] collocation schemes have been proposed for problems with a moderately large number of parameters. Adaptive variants of sparse grid collocation typically use a strategy based on Gerstner and Griebel's quadrature algorithm [18], creating an operator that is a sum of appropriately chosen hierarchical surplus operators. More recently, residual based a posteriori error estimation strategies have been developed for stochastic collocation approximations of solutions to parametric elliptic PDEs [20] and the convergence of adaptive algorithms driven by such estimators has been analysed [10, 15]. Less work has been done for time-dependent parametric PDEs. However, a residual-based error estimator was used to drive an adaptive stochastic collocation algorithm for the heat equation in [28] and time-adapted bases were used in the context of dynamical low rank approximation

for parametric parabolic problems in [22]. For gPC approximations, the use of polynomial bases that are adapted in time is discussed in [16].

We will use sparse grid stochastic collocation [37,4,29] for the parametric approximation of (1). This is appealing due to its non-intrusive nature and the fact that it leads to a polynomial surrogate. The main challenge we address is how to combine such a method with an adaptive timestepping scheme so that the distinct contributions to the total error remain balanced. For this, an efficient a posteriori error estimator is required. The strategy we propose is different to the one proposed in [28]. Our estimator is not residual-based. Moreover, we use adaptive timestepping methods in a black-box fashion and adapt the polynomial approximation *sequentially* in time.

In Sect. 2 we introduce the parametric problem of interest in more detail and describe the numerical solution approach, combining sparse grid stochastic collocation on Γ and an adaptive timestepping algorithm with local error control. In Sect. 3, we consider a related ODE test problem and motivate the need to adapt the parametric approximation in time. An a posteriori error estimator combining a Gerstner and Griebel type strategy on the parameter domain and a global timestepping error estimation strategy is defined in Sect. 4. A novel adaptive algorithm that is driven by the error estimator is described in Sect. 5 and numerical results are presented in Sect. 6 and Sect. 7.

2 Problem Statement and Approximation Scheme

Let $D \subset \mathbb{R}^n$ $(n = 2,3)$ be an open polygonal spatial domain with Lipschitz boundary ∂D and let $\mathbf{y} = [y_1, \ldots, y_d]^\top$ be a vector of d parameters taking values in a parameter domain $\Gamma = \Gamma_1 \times \cdots \times \Gamma_d \subset \mathbb{R}^d$, with associated joint probability density $\rho(\mathbf{y}) = \prod_{i=1}^d \rho_i(y_i)$. We consider an advection-diffusion problem with a fixed diffusion coefficient $\varepsilon > 0$ and a parameter-dependent, spatially divergence-free advection field $\mathbf{w} : D \times \Gamma \to \mathbb{R}^n$. That is, we seek a solution $u : D \times [0,T] \times \Gamma \to \mathbb{R}$ satisfying

$$\begin{cases} \dfrac{\partial}{\partial t} u(\mathbf{x},t,\mathbf{y}) - \varepsilon \Delta u(\mathbf{x},t,\mathbf{y}) + \mathbf{w}(\mathbf{x},\mathbf{y}) \cdot \nabla u(\mathbf{x},t,\mathbf{y}) = 0, & (\mathbf{x},t) \in D \times (0,T], \\ u(\mathbf{x},0,\mathbf{y}) = u_0(\mathbf{x}), & \mathbf{x} \in D, \end{cases} \tag{2}$$

ρ-almost surely on Γ subject to the Dirichlet boundary condition

$$u(\mathbf{x},t,\mathbf{y}) = u_{\partial D}(\mathbf{x},t), \quad (\mathbf{x},t,\mathbf{y}) \in \partial D \times [0,T] \times \Gamma. \tag{3}$$

For the weak formulation, we define $H^1_{\partial D}(D) := \{v \in H^1(D) \text{ s.t. } \operatorname{Tr} v = u_{\partial D}\}$ and seek $u(\cdot,t,\mathbf{y}) \in H^1_{\partial D}(D)$ with $\frac{\partial u}{\partial t}(\cdot,t,\mathbf{y}) \in H^{-1}(D)$ that solves

$$\begin{cases} \displaystyle\int_D \frac{\partial}{\partial t} u(\mathbf{x},t,\mathbf{y}) v(\mathbf{x}) + \varepsilon \nabla u(\mathbf{x},t,\mathbf{y}) \cdot \nabla v(\mathbf{x}) \\ \qquad + \mathbf{w}(\mathbf{x},\mathbf{y}) \cdot \nabla u(\mathbf{x},t,\mathbf{y}) v(\mathbf{x}) \mathrm{d}\mathbf{x} = 0, & t \in (0,T], \\ \displaystyle\int_D u(\mathbf{x},0,\mathbf{y}) v(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_D u_0(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x}, \end{cases} \tag{4}$$

ρ-almost surely on Γ for all test functions $v \in H_0^1(D) := \{v \in H^1(D) \text{ s.t. } \mathrm{Tr}\, v = 0\}$. Conditions for existence and uniqueness of a solution for a fixed $\mathbf{y}$ are outlined in Evans [14, Section 7.1.2]. In particular, we need to assume that $\varepsilon > 0$ is finite and $\mathbf{w}(\mathbf{x},\mathbf{y}) = [w^1(\mathbf{x},\mathbf{y}), w^2(\mathbf{x},\mathbf{y})]^\top$ where $w^1, w^2 \in L^\infty(D)$, ρ-almost surely on Γ. In our numerical experiments we will choose wind fields of the form

$$\mathbf{w}(\mathbf{x},\mathbf{y}) := \mathbf{w}_0(\mathbf{x}) + \sum_{i=1}^{d} y_i \mathbf{w}_i(\mathbf{x}) \tag{5}$$

where $\{\mathbf{w}_i\}_{i=0}^d$ are spatial fields with $w_i^1, w_i^2 \in L^\infty(D)$ for each $i = 0, ..., d$ and the parameters $y_i \in [-1, 1]$ (i.e., the associated random variables are bounded).

For the spatial discretisation, we apply finite element approximation. We do not specify a particular method here but assume that the chosen approximation is globally continuous and defined piecewise with respect to a spatial mesh of non-overlapping triangular (tetrahedral if $n = 3$) or rectangular (hexahedral) elements with characteristic edge length h. The associated nodes are partitioned into two sets $\mathcal{N}^h = \{\mathbf{x}_i\}_{i=1}^{m_D} \cup \{\mathbf{x}_i\}_{i=m_D+1}^{m_D+m_\partial}$ corresponding to interior and boundary nodes, respectively, as are the associated sets of nodal basis functions $\{\phi_i^h\}_{i=1}^{m_D} \cup \{\phi_i^h\}_{i=m_D+1}^{m_D+m_\partial}$. For each $(t,\mathbf{y}) \in [0,T] \times \Gamma$ we seek an approximation of the form

$$u^h(\mathbf{x},t,\mathbf{y}) = \sum_{i=1}^{m_D} u_i^h(t,\mathbf{y})\phi_i^h(\mathbf{x}) + \underbrace{\sum_{i=1}^{m_\partial} u_{\partial D}(\mathbf{x}_{m_D+i},t)\phi_{m_D+i}^h(\mathbf{x})}_{=:u_{\partial D}^h(\mathbf{x},t)} \tag{6}$$

where the coefficients associated with the boundary nodes are fixed. Substituting this into (4) and using the test space $S_0^h = \{\phi_i^h\}_{i=1}^{m_D} \subset H_0^1(D)$ yields a parametric system of ODEs for the vector of coefficients $\mathbf{u}^h(t,\mathbf{y}) := [u_1^h(t,\mathbf{y}), ..., u_{m_D}^h(t,\mathbf{y})]^\top$. To formulate this system, we define a mass matrix Q^h, a diffusion matrix D^h and a parameter-dependent advection matrix $A^h(\mathbf{y})$ associated with the interior nodes by

$$\begin{aligned}
\left[Q^h\right]_{i,j} &= \int_D \phi_i^h(\mathbf{x})\phi_j^h(\mathbf{x})\mathrm{d}\mathbf{x}, \\
\left[D^h\right]_{i,j} &= \int_D \nabla\phi_i^h(\mathbf{x}) \cdot \nabla\phi_j^h(\mathbf{x})\mathrm{d}\mathbf{x}, \\
\left[A^h(\mathbf{y})\right]_{i,j} &= \int_D \left(\mathbf{w}(\mathbf{x},\mathbf{y}) \cdot \nabla\phi_j^h(\mathbf{x})\right)\phi_i^h(\mathbf{x})\mathrm{d}\mathbf{x}
\end{aligned} \tag{7}$$

for $i, j = 1, ..., m_D$. We also define analogous matrices $Q_{\partial D}^h, D_{\partial D}^h$ and $A_{\partial D}^h(\mathbf{y})$ that account for boundary interactions, for $i = 1, ..., m_D$ and $j = 1, ..., m_\partial$, as follows

$$\begin{aligned}
\left[Q_{\partial D}^h\right]_{i,j} &= \int_D \phi_i^h(\mathbf{x})\phi_{m_D+j}^h(\mathbf{x})\mathrm{d}\mathbf{x}, \\
\left[D_{\partial D}^h\right]_{i,j} &= \int_D \nabla\phi_i^h(\mathbf{x}) \cdot \nabla\phi_{m_D+j}^h(\mathbf{x})\mathrm{d}\mathbf{x}, \\
\left[A_{\partial D}^h(\mathbf{y})\right]_{i,j} &= \int_D \left(\mathbf{w}(\mathbf{x},\mathbf{y}) \cdot \nabla\phi_{m_D+j}^h(\mathbf{x})\right)\phi_i^h(\mathbf{x})\mathrm{d}\mathbf{x}.
\end{aligned} \tag{8}$$

The coefficient vector $\mathbf{u}^h : [0,T] \times \Gamma \to \mathbb{R}^{m_D}$ then satisfies the ODE system

$$\begin{cases} Q^h \frac{\partial}{\partial t}\mathbf{u}^h(t,\mathbf{y}) + \varepsilon D^h \mathbf{u}^h(t,\mathbf{y}) + A^h(\mathbf{y})\mathbf{u}^h(t,\mathbf{y}) = \mathbf{f}^h(t,\mathbf{y}), \quad t \in (0,T], \\ \mathbf{u}^h(0,\mathbf{y}) = \mathbf{u}_0^h = [u_0(\mathbf{x}_1),...,u_0(\mathbf{x}_{m_D})]^\top, \end{cases} \tag{9}$$

ρ-almost surely on Γ subject to the forcing term

$$\mathbf{f}^h(t,\mathbf{y}) := -Q^h_{\partial D}\frac{\mathrm{d}\mathbf{u}^h_{\partial D}(t)}{\mathrm{d}t} - \left(\varepsilon D^h_{\partial D} + A^h_{\partial D}(\mathbf{y})\right)\mathbf{u}^h_{\partial D}(t), \tag{10}$$

where $\mathbf{u}^h_{\partial D}(t) := [u_{\partial D}(\mathbf{x}_{m_D+1},t),...,u_{\partial D}(\mathbf{x}_{m_D+m_\partial},t)]^\top$. After solving this system, the spatially-discrete approximation $u^h : D \times [0,T] \times \Gamma \to \mathbb{R}$ can be reconstructed as

$$u^h(\mathbf{x},t,\mathbf{y}) := \sum_{i=1}^{m_D}\left[\mathbf{u}^h(t,\mathbf{y})\right]_i \phi_i^h(\mathbf{x}) + u^h_{\partial D}(\mathbf{x},t). \tag{11}$$

Note that in general, $u^h_{\partial D} \neq u_{\partial D}$ unless the boundary condition is a piecewise polynomial of an appropriate degree. However, we will ignore this error here. Our focus is on approximating the solution of the ODE system (9), or equivalently the function u^h in (11), that arises after applying the spatial discretisation.

For the parametric approximation, we will construct a global polynomial approximation on Γ that interpolates u^h at a set of collocation points. Specifically, we use the sparse grid construction proposed by Smolyak [37] and developed by Barthelmann, Novak and Ritter [4]. The reader is referred to the extensive literature (e.g., see [29, 17, 18]) for more details. Sparse grid stochastic collocation has been used extensively for solving parametric elliptic and parabolic PDEs [1, 27, 39]. Here, we give only a brief outline to set up notation.

First, we require a set of one-dimensional interpolation operators. Let $\{Z^\alpha\}_{\alpha=1}^\infty$ be a sequence of sets of n_α distinct collocation points with $Z^\alpha = \{z_i\}_{i=1}^{n_\alpha} \subset \Gamma_1 = [-1,1]$. For each 'level' number $\alpha \in \mathbb{N}$ and for continuous functions $f : \Gamma_1 \to \mathbb{R}$, we define an interpolation operator $\mathrm{I}^\alpha : C^0(\Gamma_1) \to \mathscr{P}^{n_\alpha - 1}(\Gamma_1)$ in the usual way by

$$\mathrm{I}^\alpha[f](y) := \sum_{z \in Z^\alpha} f(z) L_z^\alpha(y), \tag{12}$$

where $L_z^\alpha \in \mathscr{P}^{n_\alpha-1}(\Gamma_1)$ is the Lagrange polynomial of degree $n_\alpha - 1$ that takes the value one at $y = z$ and zero at all the other points in Z^α. If we set $\Delta^1 := \mathrm{I}^1$ and define the *detail operator* $\Delta^\alpha : C^0(\Gamma_1) \to \mathscr{P}^{n_\alpha-1}(\Gamma_1)$ for $\alpha > 1$ as

$$\Delta^\alpha := \mathrm{I}^\alpha - \mathrm{I}^{\alpha-1} \tag{13}$$

then clearly $\mathrm{I}^\alpha = \sum_{i=1}^\alpha \Delta^i$.

Next, switching to d dimensions, for any multi-index $\underline{\alpha} \in \mathbb{N}^d$ we can define the so-called *hierarchical surplus* (or multi-dimensional detail operator)

$$\Delta^{\underline{\alpha}} := \bigotimes_{i=1}^d \Delta^{\alpha_i}. \tag{14}$$

Given an *admissible* multi-index set $I \subset \mathbb{N}^d$, that is, a set I such that for every $\underline{\alpha} \in I$,

$$\underline{\alpha} - \underline{e}_j \in I \text{ for } j = 1, ..., d \text{ where } \alpha_j > 1 \tag{15}$$

and $\underline{e}_j := [0, ..., \underbrace{1}_{j\text{-th}}, ..., 0]^\top$, we define the sparse grid operator I^I as follows

$$\mathrm{I}^I[f](\mathbf{y}) := \sum_{\underline{\alpha} \in I} \Delta^{\underline{\alpha}}[f](\mathbf{y}). \tag{16}$$

This is an interpolant if the sets of one-dimensional points are nested, i.e., $Z^1 \subset Z^2 \subset Z^3 ...$ [4, Proposition 6]. In that case, we can rewrite (16) as

$$\mathrm{I}^I[f](\mathbf{y}) := \sum_{\mathbf{z} \in Z^I} f(\mathbf{z}) L^I_{\mathbf{z}}(\mathbf{y}) \tag{17}$$

for appropriate interpolation polynomials $L^I_{\mathbf{z}}$ and interpolation points $Z^I \subset \Gamma$. We will use Clenshaw–Curtis points.

Definition 1 (Clenshaw–Curtis points [8]) On the interval $\Gamma_1 = [-1, 1]$ the set of n Clenshaw–Curtis (CC) points is defined by

$$z_i = -\cos\left(\pi \frac{i-1}{n-1}\right), \qquad i = 1, ..., n, \tag{18}$$

for $n > 1$ and we set $z_1 = 0$ for $n = 1$.

To construct nested sets of n_α CC points we set $n_1 = 1$ and choose

$$n_\alpha = 2^{\alpha - 1} + 1 \text{ for } \alpha > 1. \tag{19}$$

Remark 1 (Use of sparse grids) The motivation for using sparse grids is to reduce the growth in computational cost as the number d of parameters increases. For tensor product grids the number of points required to achieve a fixed total degree of polynomial exactness grows exponentially. The analogous asymptotic result for sparse grids shows an optimal polynomial growth [4, Remark 5].

Applying the sparse grid interpolation operator to u^h in (11) gives

$$u^{h,I}(\mathbf{x}, t, \mathbf{y}) := \mathrm{I}^I[u^h(\mathbf{x}, t, \mathbf{y})]. \tag{20}$$

Re-writing (20) as a linear combination of Lagrange polynomials as in (17) reveals that we need to solve the ODE system (9) for each of the chosen collocation points $\mathbf{z} \in Z^I$. The solution to each of these deterministic ODE systems needs to be further approximated using an appropriate timestepping algorithm. If an *adaptive* scheme is used, then for each $\mathbf{z} \in Z^I$ the timestepping method will select a (potentially different) set of timesteps $\{\Delta t_k\}_{k=0}^{n-1}$ and generate a sequence of corresponding approximations

$$\mathbf{u}_k^{h,\delta}(\mathbf{z}) \approx \mathbf{u}^h(t_k; \mathbf{z}) \tag{21}$$

to the ODE solution for times $\{t_k\}_{k=1}^n = \{\sum_{i=0}^{k-1} \Delta t_i\}_{k=1}^n$ where $t_n \geq T$. We will consider stable, convergent, implicit methods with local error control. Local error control means that we select the timestep Δt_k so that the local error $\mathscr{E}_k^{\text{local}}(\mathbf{z})$ satisfies

$$\mathscr{E}_k^{\text{local}}(\mathbf{z}) := \|\mathbf{u}_{local}^h(t_k + \Delta t_k, \mathbf{z}) - \mathbf{u}_{k+1}^{h,\delta}(\mathbf{z})\|_{Q^h} \approx \delta, \tag{22}$$

where δ is a user-chosen tolerance, $\mathbf{u}_{local}^h(t,\mathbf{z})$ is the ODE solution on $[t_k, t_{k+1}]$ with initial data $\mathbf{u}_{local}^h(t_k,\mathbf{z}) := \mathbf{u}_k^{h,\delta}(\mathbf{z})$ and $\|\cdot\|_{Q^h}$ is the discrete analogue of $\|\cdot\|_{L^2(D)}$.

After applying a timestepping method for each collocation point $\mathbf{z} \in Z^I$, a space-time approximation $u^{h,\delta}(\mathbf{x},t;\mathbf{z})$ can be defined in a similar way to (11) by first constructing continuous-in-time coefficients from the approximations $\mathbf{u}_k^{h,\delta}(\mathbf{z})$ using linear interpolation on each subinterval $[t_k, t_{k+1})$ as follows

$$\mathbf{u}^{h,\delta}(t;\mathbf{z}) := \mathbf{u}_k^{h,\delta}(\mathbf{z}) + \frac{t - t_k}{t_{k+1} - t_k} \left(\mathbf{u}_{k+1}^{h,\delta}(\mathbf{z}) - \mathbf{u}_k^{h,\delta}(\mathbf{z}) \right). \tag{23}$$

The final approximation to the solution of the parametric PDE (2) is then

$$u^{h,I,\delta}(\mathbf{x},t,\mathbf{y}) := \sum_{\mathbf{z} \in Z^I} u^{h,\delta}(\mathbf{x},t;\mathbf{z}) L_{\mathbf{z}}^I(\mathbf{y}). \tag{24}$$

The simple example in Sect. 3 motivates the need to adapt the multi-index set Z^I in time as well as use adaptive timestepping.

3 Motivating Example

Applying the Fourier transform in space to the advection-diffusion problem on $D = \mathbb{R}$ gives a system of ODEs with complex coefficients that depend on the wavenumber. We consider a single ODE of this form with a parameter-dependent coefficient $\alpha(y)$ where y is the image of a uniform random variable on $\Gamma = [-1,1]$ with probability density $\rho(y) = \frac{1}{2}$. That is, we wish to approximate $u : [0,T] \times \Gamma \to \mathbb{C}$ satisfying

$$\begin{cases} \frac{\partial}{\partial t} u(t,y) = \alpha(y) u(t,y) \quad t \in (0,T] \\ u(0,y) = u_0 \end{cases} \tag{25}$$

ρ-almost surely on Γ where $\alpha : \Gamma \to \mathbb{C}$ is defined by

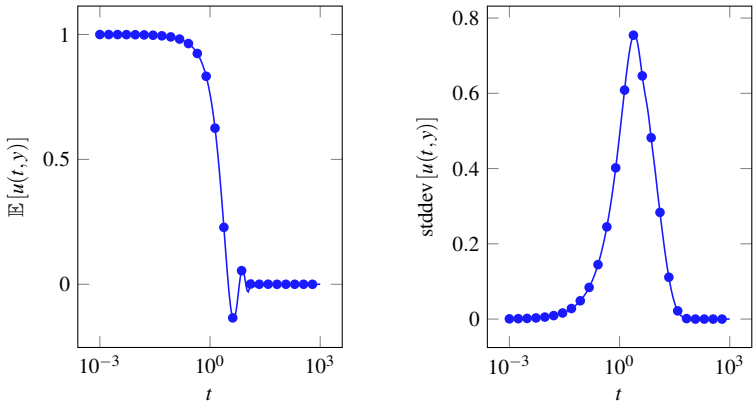
$$\alpha(y) = (-\varepsilon + \mathrm{i}y). \tag{26}$$

Here, $\Re[\alpha(y)] = -\varepsilon$ is associated with diffusion and damps the solution in time and $\Im[\alpha(y)] = y$ represents an uncertain advective term. The exact solution to (25) is

$$u(t,y) = u_0 \exp(-\varepsilon t) \exp(\mathrm{i}yt) \tag{27}$$

with mean

$$\mathbb{E}[u(t,y)] = u_0 \exp(-\varepsilon t) \operatorname{sinc}(t) \tag{28}$$



(a) mean of $u(t,y)$ (b) standard deviation of $u(t,y)$

Fig. 1: Mean and standard deviation of (27) for $u_0 = 1$ and $\varepsilon = 0.1$

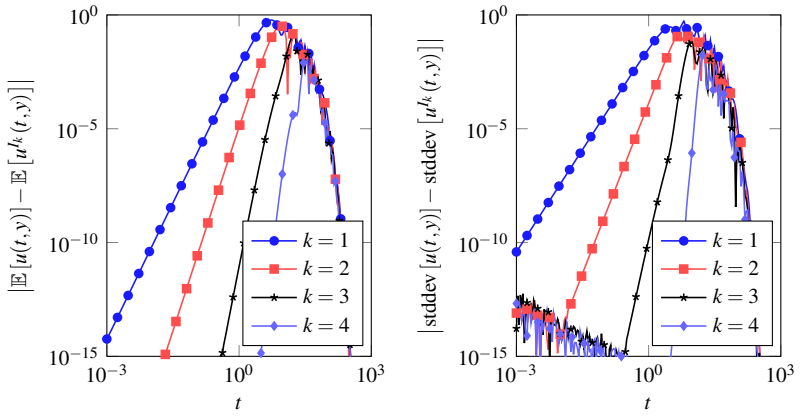
and standard deviation

$$\text{stddev}\,[u(t,y)] = u_0 \exp(-\varepsilon t)\sqrt{1-\text{sinc}(t)^2}. \tag{29}$$

These functions are plotted in Figs. 1a and 1b for the case $u_0 = 1$ and $\varepsilon = 0.1$. The mean decays in time with modulation due to the sinc function. More interestingly, the standard deviation *grows* in time initially. The solution (27) becomes more oscillatory as a function of the parameter y as time passes and hence will become harder to approximate in a fixed polynomial approximation space. In the long time limit the exponential decay dominates and the standard deviation decays to zero. We now apply the polynomial-based approximation scheme outlined in Sect. 2 and highlight the issue of long time integration for parametric problems. Issues associated with polynomial chaos approximation for a similar ODE are discussed in [38].

3.1 Interpolation

Since we have a one-dimensional parameter domain, we define the sequence of multi-index sets $I_k := \{\alpha \in \mathbb{N} \text{ s.t. } \alpha \leq 1+k\}$ for $k = 1,2,\ldots$. We construct interpolants using nested sets of CC collocation points with the doubling relation (19) so that for a fixed k the resulting polynomial is of degree 2^k. The exact solution is available at each collocation point so timestepping is not necessary and here we simply denote approximations by $u^{I_k} := \mathrm{I}^{I_k}[u]$, for $k = 1,2,\ldots$. Errors in the approximation of the mean and the standard deviation are plotted in Figs. 2a and 2b for $k = 1,2,3,4$. These plots illustrate two important points. Firstly, for a *fixed* approximation space (fixed k) the error grows with time (at least initially). Secondly, if we want to suppress the error we must grow the multi-index set I_k in time, increasing the degree of the polynomial approximation. In Fig. 3 the approximation error $\|u(t,y) - u^{I_k}(t,y)\|_{L^2_\rho(\Gamma)}$ is estimated using the n_{11}-point CC quadrature rule. Again the error is seen to grow in



(a) error in mean (b) error in standard deviation

Fig. 2: Error in mean and standard deviation of (27) for $u_0 = 1$ and $\varepsilon = 0.1$ and increasing polynomial degree 2^k

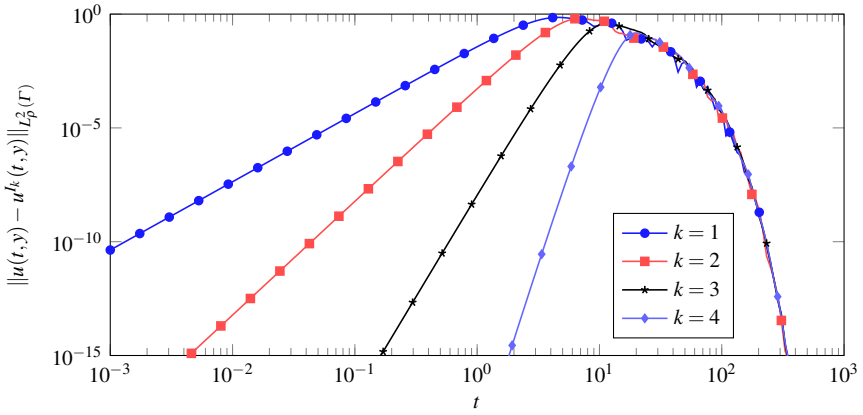
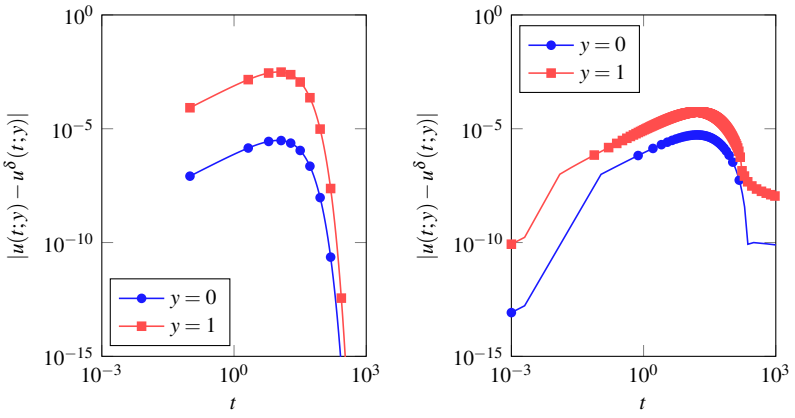


Fig. 3: Approximation error $\|u(t,y) - u^{I_k}(t,y)\|_{L^2_\rho(\Gamma)}$ for the solution of (25) with $u_0 = 1$ and $\varepsilon = 0.1$ and increasing polynomial degree 2^k

time for a fixed polynomial approximation space and can be suppressed by increasing the degree of the polynomial approximation. It is clear that adapting the set I_k in time would be advantageous.

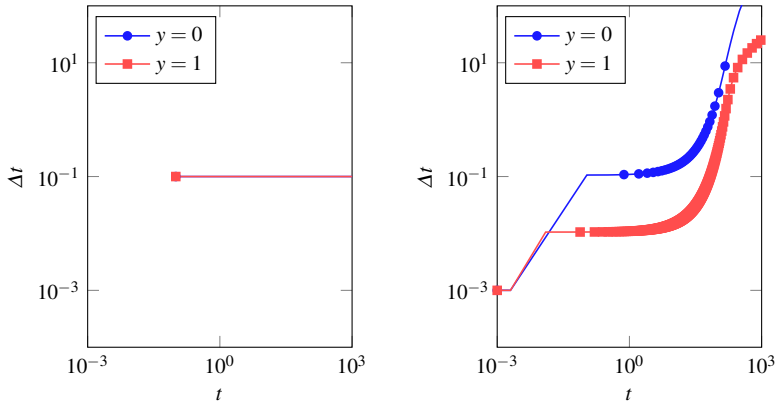
3.2 Timestepping

The ODE test problem (25) is now solved numerically using the standard trapezoidal rule (TR) and the `stabtr` implementation of the TR-AB2 algorithm [21, Chapter 5], [19]. TR-AB2 pairs TR with the two-step Adams–Bashforth (AB2) method to



(a) TR with $\Delta t = 10^{-1}$ (b) TR-AB2 with $\delta = 10^{-7}$

Fig. 4: Global timestepping error for approximation of (25) with $y = 0, 1$



(a) TR with $\Delta t = 10^{-1}$ (b) TR-AB2 with $\delta = 10^{-7}$

Fig. 5: Timesteps for approximation of (25) with $y = 0, 1$

estimate and control the local timestepping error. We consider the deterministic ODEs associated with the fixed values $y = 0$ (corresponding to $\alpha(0) = \mathbb{E}[\alpha(y)]$) and $y = 1$ (which gives the maximal imaginary component of α) and investigate the *global* timestepping error

$$|u(t;y) - u^\delta(t;y)| \tag{30}$$

associated with the chosen schemes. The TR algorithm is applied using a fixed timestep of $\Delta t = 10^{-1}$. For TR-AB2, we choose a local error tolerance $\delta = 10^{-7}$. The plots in Fig. 4 demonstrate that for $y = 0$ the two algorithms give comparable global errors for times up to $\mathcal{O}\left(10^2\right)$. In the long time limit the TR-AB2 error stagnates — but it is already very small by this time. To reach the final time $T = 10^3$, the TR method required 10^4 steps but TR-AB2 required only 296. This computational saving motivates

the need for adaptive timestepping; we attain a comparable global error at a significantly reduced cost. Figure 5 illustrates why this saving is possible. It can be seen that in the long time limit the timestep with TR-AB2 is allowed to grow much larger than 10^{-1}. Similar results are seen for $y = 1$. In fact, a notably better global error is obtained using TR-AB2 with $\delta = 10^{-7}$, requiring only 2907 timesteps, compared to the 10^4 timesteps used in standard TR with $\Delta t = 10^{-1}$.

Returning to (24), we now consider how to adapt the approximation in time to ensure that the interpolation and timestepping errors remain balanced. Key to this is an efficient a posteriori error estimator. We discuss this next.

4 Error Estimation

Error estimation will be performed in the norm $\|\cdot\|_{L^2_\rho(\Gamma;L^2(D))}$ where

$$\|u(\mathbf{x},t,\mathbf{y})\|_{L^2_\rho(\Gamma;L^2(D))} := \left(\int_\Gamma \int_D (u(\mathbf{x},t,\mathbf{y}))^2 \,\mathrm{d}\mathbf{x}\, \rho(\mathbf{y})\mathrm{d}\mathbf{y}\right)^{\frac{1}{2}}. \tag{31}$$

The $L^2(D)$ norm in space is chosen as it is the most natural choice in timestepping algorithms for local error control. The $L^2_\rho(\Gamma)$ norm is chosen as we consider problems with uncertain inputs and solutions that have finite variance.

To simplify the analysis the *spatial* contribution to the error is not estimated — we are interested in approximating the solution to the parametric ODE system (9). The focus is on estimating and balancing the interpolation and timestepping errors so we consider the splitting of the error

$$\begin{aligned}\|u^h - u^{h,\delta,I}\|_{L^2_\rho(\Gamma;L^2(D))} &\leq \|u^h - u^{h,I}\|_{L^2_\rho(\Gamma;L^2(D))} + \|u^{h,I} - u^{h,I,\delta}\|_{L^2_\rho(\Gamma;L^2(D))} \\ &=: \mathscr{E}^I(t) + \mathscr{E}^\delta(t).\end{aligned} \tag{32}$$

The interpolation error $\mathscr{E}^I(t)$ is estimated using a hierarchical approach by comparing the approximation $u^{h,I}$ to an *enhanced approximation* and appealing to a saturation assumption. This approach follows the adaptive algorithm first proposed by Gerstner and Griebel [18]. To define an enhanced approximation, we first define two additional multi-index sets. The *margin* M_I of I is defined as the set of all neighbouring multi-indices,

$$M_I := \bigcup_{\underline{\alpha}\in I} \{\underline{\alpha} + \underline{e}_j \text{ for } j \in 1,...,d\}. \tag{33}$$

The *reduced margin* R_I is defined as the set of multi-indices in M_I that can be individually added to I whilst maintaining the admissibility property. That is, the set

$$R_I := \{\underline{\alpha} \in M_I \text{ s.t. } I \cup \underline{\alpha} \text{ is admissible}\}. \tag{34}$$

The enhanced multi-index set is defined as $I^* := I \cup R_I$ and is used to construct the enhanced approximation u^{h,I^*}. The error estimation strategy we adopt is underpinned by the following assumption.

Assumption 1 (Saturation Assumption) *Let $I^* := I \cup R_I$ be the enhanced multi-index set. There exists a saturation constant $\gamma(t) \in [0,1)$ for each $t \in [0,T]$ such that*

$$\begin{aligned} &\|u^h(\cdot,t,\cdot) - u^{h,I^*}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))} \\ &\qquad \leq \gamma(t)\|u^h(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))}. \end{aligned} \tag{35}$$

A saturation assumption for the interpolation error estimation is a standard ingredient in a posteriori error analysis of elliptic PDEs with random data, see e.g. [5, Section 4]. The assumption can be justified by considering the analyticity of the solution with respect to $\mathbf{y}$, see e.g. [15].

Corollary 1 (Interpolation Error Bound) *If Assumption 1 holds, then the following upper bound holds for the interpolation error,*

$$\mathscr{E}^I(t) \leq \frac{1}{1-\gamma(t)}\|u^{h,I^*}(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))}. \tag{36}$$

Proof The result follows by writing

$$\begin{aligned} &\|u^h(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))} = \\ &\qquad \|u^h(\cdot,t,\cdot) - u^{h,I^*}(\cdot,t,\cdot) + u^{h,I^*}(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))} \end{aligned} \tag{37}$$

and applying the triangle inequality and Assumption 1.

The upper bound in (36) is not computable — the saturation constant is unknown and the approximations $u^{h,I}(\mathbf{x},t,\mathbf{y})$ and $u^{h,I^*}(\mathbf{x},t,\mathbf{y})$ are only known *after* further approximation with timestepping. Hence, we consider the further splitting

$$\begin{aligned} &\|u^{h,I^*}(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))} \\ &\quad \leq \underbrace{\|u^{h,I^*,\delta}(\cdot,t,\cdot) - u^{h,I,\delta}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))}}_{\pi_I(t)} \\ &\qquad + \underbrace{\|u^{h,I^*}(\cdot,t,\cdot) - u^{h,I^*,\delta}(\cdot,t,\cdot) - u^{h,I}(\cdot,t,\cdot) + u^{h,I,\delta}(\cdot,t,\cdot)\|_{L^2_\rho(\Gamma;L^2(D))}}_{\textit{correction}} \end{aligned} \tag{38}$$

where $\pi_I : [0,T] \to \mathbb{R}$ is computable but there is now an additional correction term to estimate. We will call $\pi_I(t)$ the interpolation error estimate.

Remark 2 (Interpolation error indicators) We can bound the interpolation error estimate π_I as

$$\pi_I(t) \leq \sum_{\underline{\alpha} \in R_I} \pi_{I,\underline{\alpha}}(t) \tag{39}$$

where

$$\pi_{I,\underline{\alpha}}(t) := \left\| \sum_{z \in Z^{I \cup \underline{\alpha}}} u^{h,\delta}(\mathbf{x},t;\mathbf{z}) L_{\mathbf{z}}^{I \cup \underline{\alpha}}(\mathbf{y}) - \sum_{z \in Z^I} u^{h,\delta}(\mathbf{x},t;\mathbf{z}) L_{\mathbf{z}}^{I}(\mathbf{y}) \right\|_{L^2_\rho(\Gamma;L^2(D))}. \tag{40}$$

These indicators will guide the adaptive algorithm introduced in Sect. 5.

Now consider $u^{h,I}, u^{h,I,\delta}, u^{h,I^*}$ and $u^{h,I^*,\delta}$ in the correction term in (38). Rewriting these terms as linear combinations of Lagrange polynomials as in (17) gives

$$\begin{aligned}&\left(u^{h,I^*}(\mathbf{x},t,\mathbf{y}) - u^{h,I^*,\delta}(\mathbf{x},t,\mathbf{y})\right) - \left(u^{h,I}(\mathbf{x},t,\mathbf{y}) - u^{h,I,\delta}(\mathbf{x},t,\mathbf{y})\right)\\ &= \sum_{z\in Z^{I^*}\setminus Z^I} e^{h,\delta}(\mathbf{x},t;\mathbf{z}) L_{\mathbf{z}}^{I^*}(\mathbf{y}) + \sum_{z\in Z^I} e^{h,\delta}(\mathbf{x},t;\mathbf{z})\left(L_{\mathbf{z}}^{I^*}(\mathbf{y}) - L_{\mathbf{z}}^{I}(\mathbf{y})\right),\end{aligned} \tag{41}$$

where we have now introduced the *global timestepping error*

$$e^{h,\delta}(\mathbf{x},t;\mathbf{z}) := u^h(\mathbf{x},t;\mathbf{z}) - u^{h,\delta}(\mathbf{x},t;\mathbf{z}) \tag{42}$$

for each collocation point $\mathbf{z}$ at all times $t\in[0,T]$. Note that this error represents both timestepping error and an interpolation error due to the extension of the approximation from discrete times $\{t_k\}_{k=0}^n$ to $[0,t_n]$ in (23). Let $\mathscr{E}^{\mathrm{ge}}(t;\mathbf{z}) := \|e^{h,\delta}(\mathbf{x},t;\mathbf{z})\|_{L^2(D)}$. An upper bound for the correction term is then given by

$$\begin{aligned}&\|u^{h,I^*}(\mathbf{x},t,\mathbf{y}) - u^{h,I^*,\delta}(\mathbf{x},t,\mathbf{y}) + u^{h,I}(\mathbf{x},t,\mathbf{y}) - u^{h,I,\delta}(\mathbf{x},t,\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma;L^2(D))}\\ &\quad\le \sum_{z\in Z^{I^*}\setminus Z^I} \mathscr{E}^{\mathrm{ge}}(t;\mathbf{z})\|L_{\mathbf{z}}^{I^*}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}\\ &\qquad + \sum_{z\in Z^I} \mathscr{E}^{\mathrm{ge}}(t;\mathbf{z})\|L_{\mathbf{z}}^{I^*}(\mathbf{y}) - L_{\mathbf{z}}^{I}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}\\ &\qquad =: E^{I,\delta}(t).\end{aligned} \tag{43}$$

A similar approach can be used to bound the timestepping error in (32) as follows

$$\mathscr{E}^{\delta}(t) \le E^{\delta}(t) := \sum_{z\in Z^I} \mathscr{E}^{\mathrm{ge}}(t;\mathbf{z})\|L_{\mathbf{z}}^{I}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}. \tag{44}$$

For each $\mathbf{z}\in Z^{I^*}$, let $\pi_{ge}(t;\mathbf{z}):[0,T]\to\mathbb{R}$ be a global timestepping error estimate,

$$\pi_{\mathrm{ge}}(t;\mathbf{z}) \approx \mathscr{E}^{\mathrm{ge}}(t;\mathbf{z}). \tag{45}$$

Then, using the set of error estimates $\{\pi_{\mathrm{ge}}(t;\mathbf{z})\}_{\mathbf{z}\in Z^{I^*}}$ the correction bound (43) can be estimated by

$$\begin{aligned}\pi_{I,\delta}(t) := &\sum_{z\in Z^{I^*}\setminus Z^I} \pi_{ge}(t;\mathbf{z})\|L_{\mathbf{z}}^{I^*}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}\\ &+ \sum_{z\in Z^I} \pi_{ge}(t;\mathbf{z})\|L_{\mathbf{z}}^{I^*}(\mathbf{y}) - L_{\mathbf{z}}^{I}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}\end{aligned} \tag{46}$$

and the timestepping bound (44) can be estimated by

$$\pi_{\delta}(t) := \sum_{z\in Z^I} \pi_{\mathrm{ge}}(t;\mathbf{z})\|L_{\mathbf{z}}^{I}(\mathbf{y})\|_{L^2_{\hat\rho}(\Gamma)}. \tag{47}$$

Returning to the full bound (32) we can write

$$\|u^h - u^{h,I,\delta}\|_{L^2_{\hat\rho}(\Gamma;L^2(D))} \le \frac{1}{1-\gamma(t)}\left(\pi_I(t) + E^{I,\delta}(t)\right) + E^{\delta}(t). \tag{48}$$

Dropping the unknown $(1-\gamma(t))^{-1}$ and replacing $E^{I,\delta}(t)$ and $E^{\delta}(t)$ with the computable estimates $\pi_{I,\delta}(t)$ and $\pi_{\delta}(t)$, we arrive at the final error estimator

$$\pi(t) := \pi_I(t) + \pi_{I,\delta}(t) + \pi_{\delta}(t), \qquad t\in[0,T]. \tag{49}$$

4.1 Global Timestepping Error Estimation

The error estimators $\pi_{I,\delta}$ and π_δ defined in (46) and (47) require a global timestepping error estimate for each collocation point $\mathbf{z} \in Z^{I^*}$. Relating the global error incurred by timestepping algorithms to the local error is a challenging problem with an array of literature on the subject [34,36,7]. A simple approach is to compare the approximation computed with local error tolerance δ to a reference approximation computed with a smaller local error tolerance $\delta^* \ll \delta$ and estimate the global error as follows [36]

$$\pi_{ge}(t;\mathbf{z}) := \|u^{h,\delta^*}(\mathbf{x},t;\mathbf{z}) - u^{h,\delta}(\mathbf{x},t;\mathbf{z})\|_{L^2(D)}. \tag{50}$$

This is an expensive method — the enhanced approximation is even more expensive to compute than the original one. The cost can be reduced by using a scaling argument which we describe next.

We assume that we are using a stable, convergent, order $p \geq 1$ time integrator with local error control and that the ODE system (9) rewritten in the form

$$\frac{\mathrm{d}\mathbf{u}^h}{\mathrm{d}t}(t;\mathbf{z}) = \mathbf{g}\left(t, \mathbf{u}^h(t;\mathbf{z})\right) \tag{51}$$

satisfies the Lipschitz condition where the solution is $p+1$ times continuously differentiable. The global error is then $\mathscr{E}^{\mathrm{ge}}(t;\mathbf{z}) = \mathscr{O}(\Delta t^p)$ [6, Definition 2.3]. Adaptive timestepping methods generally select timesteps based on the relationship between the local truncation error and timestep. For an order p method the local truncation error is $\mathscr{O}\left(\Delta t^{p+1}\right)$ and for a fixed local error tolerance δ the timesteps are chosen so that $\Delta t \propto \delta^{\frac{1}{p+1}}$ [32,19]. Hence,

$$\mathscr{E}^{\mathrm{ge}}(t;\mathbf{z}) = \mathscr{O}\left(\delta^{\frac{p}{p+1}}\right). \tag{52}$$

Using (52), for two local error tolerances δ and $\delta_0 \gg \delta$ we may assume the crude relationship

$$\frac{\mathscr{E}^{\mathrm{ge},\delta}(t;\mathbf{z})}{\mathscr{E}^{\mathrm{ge},\delta_0}(t;\mathbf{z})} \approx \left(\frac{\delta}{\delta_0}\right)^{\frac{p}{p+1}}, \tag{53}$$

where the superscript denotes the global error corresponding to the local error tolerance δ, and similarly for δ_0. Therefore,

$$\mathscr{E}^{\mathrm{ge},\delta}(t;\mathbf{z}) \approx \left(\frac{\delta}{\delta_0}\right)^{\frac{p}{p+1}} \mathscr{E}^{\mathrm{ge},\delta_0}(t;\mathbf{z}). \tag{54}$$

Using a low fidelity approximation u^{h,δ_0} with tolerance $\delta_0 \gg \delta$ and comparing to the computed approximation $u^{h,\delta}$ means a global error estimate for the tolerance δ_0 can be formed as

$$\pi_{ge,\delta_0}(t,\mathbf{z}) := \|u^{h,\delta}(\mathbf{x},t;\mathbf{z}) - u^{h,\delta_0}(\mathbf{x},t;\mathbf{z})\|_{L^2(D)} \approx \mathscr{E}^{\mathrm{ge},\delta_0}(t,\mathbf{z}). \tag{55}$$

Combining this with (54) yields a cheap to compute global error estimate for approximations computed with the local error tolerance δ. That is,

$$\begin{aligned}\pi_{\mathrm{ge}}(t;\mathbf{z}) &:= \left(\frac{\delta}{\delta_0}\right)^{\frac{p}{p+1}} \pi_{\mathrm{ge},\delta_0}(t,\mathbf{z}) \\ &= \left(\frac{\delta}{\delta_0}\right)^{\frac{p}{p+1}} \|u^{h,\delta}(\mathbf{x},t;\mathbf{z}) - u^{h,\delta_0}(\mathbf{x},t;\mathbf{z})\|_{L^2(D)}.\end{aligned} \tag{56}$$

We can then use this to compute our error estimates $\pi_{I,\delta}(t)$ and $\pi_{\delta}(t)$ in (46) and (47).

Remark 3 (Choice of timestepping method) In our numerical experiments we will use the `stabtr` implementation of TR-AB2 that is included in the MATLAB software package IFISS 3.6 [12]. This is an order $p = 2$ method. Many alternative algorithms with local error control exist, including the built-in MATLAB integrators described in [33]. Ideally, we would use a time integration method that includes global error estimation but these are rarely implemented and used in practice. An example of such a method is GERK [35] but this is both an explicit method and more expensive than the TR-AB2 integrator.

5 Adaptive Strategy

To efficiently construct an approximation to the solution of the semi-discrete parametric advection–diffusion problem (9), we will *adapt* the multi-index set I in time whilst allowing the timestepping algorithm to adaptively choose timesteps for the deterministic systems associated with individual collocation points. The construction (24) only requires a small modification to allow this. The approximation is now constructed with a time-dependent multi-index set

$$I : [0,T] \to \{\text{admissible multi-index sets in } d \text{ parameters}\} \tag{57}$$

with the adaptive approximation defined pointwise in time as

$$u^{\mathrm{adapt}}(\mathbf{x},t,\mathbf{y}) := \sum_{\mathbf{z}\in Z^{I(t)}} u^{h,\delta}(\mathbf{x},t;\mathbf{z}) L_{\mathbf{z}}^{I(t)}(\mathbf{y}). \tag{58}$$

To construct such an approximation we use a solve–estimate–mark–refine type loop in the spirit of Gerstner and Griebel to select an appropriate multi-index set [18]. At carefully selected times, we estimate the interpolation error by $\pi_I(t)$ as defined in (38), noting that the multi-index set I can now change through time. If the estimated interpolation error is greater than a chosen tolerance tol at time t, we mark a set of multi-indices $\mathscr{J} \subset R_{I(t)}$ based on the error indicators $\{\pi_{I,\underline{\alpha}}(t)\}_{\underline{\alpha}\in R_{I(t)}}$ defined in (40). The approximation (58) is then refined by adding the marked multi-indices to the construction of the approximation.

The time dependence in the problem provides an additional challenge as this adaptive strategy must be embedded inside a time propagation loop. The approximation at each collocation point is constructed independently with different discrete

timesteps. To compute error estimates, the set of approximations must be synchronised at a common set of *synchronisation times*. Such a set of synchronization times could be specified a priori, but would be better generated dynamically. This can be done by starting at time $t = 0$ with an initial synchronization timestep $\tau := \tau_0$. The approximation (58) and the error estimators π_I, $\pi_{I,\delta}$ and π_δ defined in (38), (46) and (47) would then be computed at the new time $t + \tau$. At this new time we must decide if the multi-index set needs to be refined by comparing $\pi_I(t+\tau)$ to a chosen tolerance $\mathrm{tol}(t+\tau)$. Given that the goal is to balance the error contributions then a sensible strategy is to link the interpolation error tolerance to the correction error estimated by (46). Specifically, we define

$$\mathrm{tol}(t) := c_{\mathrm{tol}} \cdot \pi_{I,\delta}(t), \tag{59}$$

with a *safety factor* $c_{\mathrm{tol}} > 1$. If the computed interpolation error estimate $\pi_I(t+\tau)$ is less than $\mathrm{tol}(t+\tau)$ then the synchronisation step would be *accepted*. In this case, the time is reset $t \leftarrow t + \tau$ and the synchronization timestep is increased $\tau \leftarrow c_+ \tau$, where $c_+ \geq 1$ is a user-defined input. Otherwise, the step is *rejected*. In this case the interpolant would be refined so that $I \leftarrow I \cup \mathscr{J}$ and the synchronization timestep is decreased $\tau \leftarrow c_- \tau$, where $c_- \in (0,1)$ is a second user-defined input. The attraction of this construction is that it generates an approximation in which the interpolation error is *balanced* with the timestepping error. The strategy does not attempt to *control* the global timestepping error.

In our experiments, time integration over each interval $[t, t+\tau]$ is done by running TR–AB2 adaptive timestepping with a local error tolerance δ. The TR–AB2 algorithm outputs a sequence of spatial approximations and the corresponding intermediate timesteps for every collocation point. On each interval, for each collocation point, the adaptive timestepping will be initialised with the corresponding solution approximation at the previous two time integrator steps. This is necessary as AB2 is not self-starting. Global error estimates are computed using (56) as discussed in Sect. 4.1.

The complete algorithm specification will also require a marking strategy for the multi-index sets. A generic strategy, similar to that used in many related algorithms [9], is specified in Algorithm 1. The inputs are the current reduced margin R_I, the corresponding error indicators $\{\pi_{I,\underline{\alpha}_i}\}_{i=1}^n$ and a threshold parameter θ. The algorithm returns a set of marked multi-indices $\mathscr{J} \subset R_I$ that will be used to refine the approximation (58).

Algorithm 1 Dörfler marking

procedure MARK(R_I, $\{\pi_{I,\underline{\alpha}_i}\}_{i=1}^n$, θ)
  Order $\underline{\alpha} \in R_I$ such that $\pi_{I,\underline{\alpha}_1} \geq \pi_{I,\underline{\alpha}_2} \geq \cdots \geq \pi_{I,\underline{\alpha}_n}$.
  $\eta = 0, m = 0$
  while $\eta < (1-\theta)$ **do**
    $m \leftarrow m + 1$
    $\eta \leftarrow \sum_{i=1}^m \pi_{I,\underline{\alpha}_i} / \sum_{i=1}^n \pi_{I,\underline{\alpha}_i}$
  end while
  return $\mathscr{J} = \{\underline{\alpha}_i\}_{i=1}^m$
end procedure

Alternative marking schemes can be designed, for example incorporating computational cost to give a measure of error per unit of computational work. This would order multi-indices by the profit indicators $\pi_{I,\underline{\alpha}}/n_{\underline{\alpha}}$ where $n_{\underline{\alpha}}$ is the number of additional collocation points required; for further details see, for example, [25,20,10].

The complete solution algorithm is given in Algorithm 2. The input parameters are the local error tolerance δ, the safety factor c_{tol}, and the marking parameter θ. The other parameters on Line 2 are the spatial discretisation parameter h, the initial timestep Δt_0, initial synchronisation timestep τ_0, and the scaling factors $c_{\pm}$. They are distinguished from the inputs. While they affect the constructed approximation, they have little influence on the resulting approximation error.

The performance of Algorithm 2 will be illustrated on two test problems in Sect. 6 and Sect. 7.

Algorithm 2 Adaptive SC-FEM for parametric advection–diffusion

1: **procedure** ADAPTIVE SC-FEM($\delta, c_{\mathrm{tol}}, \theta$)
2: Fix $h, \Delta t_0, \{\tau_0, c_-, c_+\}$. ▷ Setup
3: Initialise multi-index sets $I \leftarrow \{\underline{1}\}$, R_I, $I^* \leftarrow I \cup R_I$.
4: Construct Q^h,D^h, $A^h(\mathbf{z})$, $\mathbf{f}^h(t,\mathbf{z})$ as defined in (7) and (10) for all $\mathbf{z} \in Z^{I^*}$.
5: Initialise $t = 0, \tau = \tau_0$ and $u^{h,\delta}_{local}(\mathbf{x};\mathbf{z}) = u^h_0(\mathbf{x}), \Delta t_{\mathbf{z}} = \Delta t_0$ for all $\mathbf{z} \in Z^{I^*}$.
6:
7: **while** $t < T$ **do**
8: **for** $\mathbf{z} \in Z^{I^*}$ **do** ▷ Solve
9: $[\{u^{h,\delta}(\mathbf{x},t_k;\mathbf{z})\}_{k=1}^n, \{\Delta t_k\}_{k=1}^n] = \textsc{Timestepping}(\delta, u^{h,\delta}_{local}(\mathbf{x};\mathbf{z}), t, t+\tau, \Delta t_{\mathbf{z}})$.
10: Interpolate $u^{h,\delta}(\mathbf{x},s;\mathbf{z})$ in time for $s \in [t, t+\tau]$.
11: Update $u^{h,\delta}_{local}(\mathbf{x};\mathbf{z}) \leftarrow u^{h,\delta}(\mathbf{x},t_n;\mathbf{z})$, $\Delta t_{\mathbf{z}} \leftarrow \Delta t_n$.
12: **end for**
13: Construct $u^{\mathrm{adapt}}(\mathbf{x},s,\mathbf{y}) = \sum_{\mathbf{z}\in I} u^{h,\delta}(\mathbf{x},s;\mathbf{z}) L^I_{\mathbf{z}}(\mathbf{y})$ for $s \in [t, t+\tau]$.
14: Compute $\pi_I(t+\tau), \pi_{I,\delta}(t+\tau), \pi_\delta(t+\tau)$. ▷ Estimate
15: $\mathrm{tol} \leftarrow c_{\mathrm{tol}} \cdot \pi_{I,\delta}(t+\tau)$.
16:
17: **if** $\pi_I(t+\tau) > \mathrm{tol}(t+\tau)$ **then** ▷ Reject
18: Compute $\{\pi_{I,\underline{\alpha}}(t+\tau)\}$ for $\underline{\alpha} \in R_I$.
19: $\mathscr{J} = \textsc{Mark}(R_I, \{\pi_{I,\underline{\alpha}}(t+\tau)\}_{\underline{\alpha}\in R_I}, \theta)$. ▷ Mark
20: **for** $\mathbf{z} \in Z^{I^*\cup\mathscr{J}} \setminus Z^{I^*}$ **do** ▷ Refine
21: Initialise $u^{h,\delta}_{local}(\mathbf{x};\mathbf{z}) = u^h_0(\mathbf{x}), \Delta t_{\mathbf{z}} = \Delta t_0$ and $A^h(\mathbf{z})$, $\mathbf{f}^h(t,\mathbf{z})$.
22: $[\{u^{h,\delta}(\mathbf{x},t_k;\mathbf{z})\}_{k=1}^n, \{\Delta t_k\}_{k=1}^n] = \textsc{Timestepping}(\delta, u^{h,\delta}_{local}(\mathbf{x},\mathbf{z}), 0, t, \Delta t_{\mathbf{z}})$.
23: Update $u^{h,\delta}_{local}(\mathbf{x};\mathbf{z}) \leftarrow u^{h,\delta}(\mathbf{x},t_n;\mathbf{z})$, $\Delta t_{\mathbf{z}} \leftarrow \Delta t_n$.
24: **end for**
25: $I \leftarrow I \cup \mathscr{J}$
26: $I^* \leftarrow I \cup R_I$
27: $\tau \leftarrow c_- \tau$
28: **else** ▷ Accept
29: $t \leftarrow t + \tau$
30: $\tau \leftarrow c_+ \tau$
31: **end if**
32: **end while**
33:
34: **return** $u^{\mathrm{adapt}}(\mathbf{x},t,\mathbf{y})$ for $(\mathbf{x},t,\mathbf{y}) \in D \times [0,T] \times \Gamma$
35: **end procedure**

Remark 4 (Shrinking the multi-index set) The diffusive term will smooth the solution through time as illustrated in the motivational example in Sect. 3. This suggests that it may be desirable to simplify the parametric approximation by shrinking the multi-index set. This could be achieved by computing the error indicators (40) for each $\underline{\alpha} \in I$ and defining a pruning algorithm to remove the smallest contributions. Care would be required to ensure the resulting multi-index set is still admissible. In practice, the computational savings may not be significant. At later times when the diffusive smoothing is the dominant effect, the timesteps will have already grown large as seen in Fig. 5b. Advancing the approximation in time, even with a large number of collocation points, is then cheap to compute.

6 Computational experiment: $d = 4$

We first consider a low dimensional problem for which an accurate reference solution can be computed to test the efficiency and reliability of the error estimator in Algorithm 2. We consider a generalization of the double glazing problem [11]. The problem is solved on a square domain $D = (-1,1)^2$ with Dirichlet boundary data

$$u_{\partial D}(\mathbf{x},t) := \begin{cases} 0 & \mathbf{x} \in \partial D_0 \\ (1-x_2^4)\,(1-\exp(-t/T_\partial)) & \mathbf{x} \in \partial D_1 \end{cases}, \tag{60}$$

where $\partial D_1 = \{\mathbf{x} \in \partial D \text{ s.t. } x_1 = 1\}$ and $\partial D_0 := \partial D \setminus \partial D_1$. The parameter T_∂ in (60) controls the rate at which the right-hand boundary wall 'heats up'. The initial condition is identically zero,

$$u(\mathbf{x},0,\mathbf{y}) = u_0(\mathbf{x},\mathbf{y}) = 0. \tag{61}$$

The mean wind field is given by

$$\mathbf{w}_0(x_1,x_2) = \big(2x_2(1-x_1^2), -2x_1(1-x_2^2)\big) \tag{62}$$

and uncertainty will be introduced by perturbing $\mathbf{w}_0$.

The parameter domain is given by $\Gamma = [-1,1]^4$ and we set $\rho(\mathbf{y}) = (0.5)^4$. The spatial domain D is subdivided into four quadrants

$$D_i = [a_i - 0.5, a_i + 0.5] \times [b_i - 0.5, b_i + 0.5], \tag{63}$$

where $\mathbf{a} = [-0.5,-0.5,0.5,0.5]^\top$, $\mathbf{b} = [-0.5,0.5,-0.5,0.5]^\top$. The parametric wind field $\mathbf{w} : D \times \Gamma \to \mathbb{R}^2$ is then defined in piecewise fashion so that

$$\mathbf{w}(\mathbf{x},\mathbf{y}) = \mathbf{w}_0(\mathbf{x}) + \sigma \sum_{i=1}^{4} y_i\, \mathbf{w}_i(\mathbf{x}) \tag{64}$$

where $\sigma = 0.5$, with smaller locally divergence-free recirculations given by

$$\mathbf{w}_i(\mathbf{x}) := \begin{cases} \mathbf{w}_0(z_1,z_2), & z_1 = 2(x_1 - a_i), z_2 = 2(x_2 - b_i), \quad \mathbf{x} \in D_i \\ 0, & \text{otherwise.} \end{cases} \tag{65}$$

Note that, while the linearity of (64) ensures that the perturbed wind field remains divergence-free, the tangential components of the perturbed wind are discontinuous along the interior subdomain boundaries. By design, the non-uniform finite element mesh is aligned with these lines of discontinuity.

The spatial discretization is generated by IFISS 3.6 using continuous piecewise bilinear approximation. This gives a parametric system of ODEs as specified in (9). Timestepping is performed using `stabtr` as defined in [19]. Approximation input options are specified in Table 1. Nested sets of CC collocation points are generated using SPARSE GRIDS MATLAB KIT (V22.02 CALIFORNIA) [3,31].

Table 1: IFISS 3.6 and `stabtr` input options used for approximating the parametric double glazing problem with $d = 4$

IFISS 3.6 option	value
Hot wall time constant T_∂	10^{-1}
Grid parameter ℓ ($2^{2\ell}$ elements)	4
Stretched grid	yes
Viscosity parameter ε	10^{-1}

`stabtr` option	value
Averaging counter n_*	no averaging
Initial timestep Δt_0	10^{-9}
Local error tolerance δ	10^{-5}
Final time T	100

The error in the ADAPTIVE SC-FEM approximation will be evaluated at a set of fifty logarithmically-spaced *reference times* $\mathscr{T}$ between $T_\partial \log\left(0.9^{-1}\right)$ and 10^2 and computed with respect to a *reference* approximation $u^{\rm ref}$ as

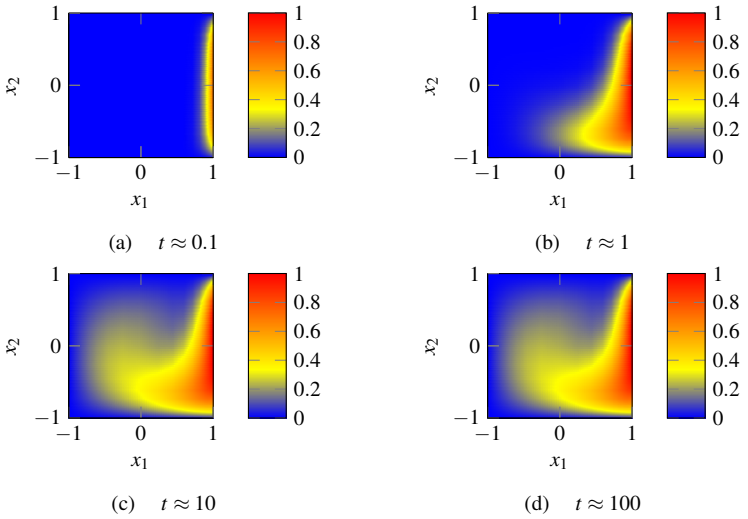
$$\|u^{\rm ref} - u^{\rm adapt}\|_{L^2_\rho(\Gamma;L^2(D))} \approx \mathscr{E}^{\rm adapt} := \|u^h - u^{\rm adapt}\|_{L^2_\rho(\Gamma;L^2(D))}. \tag{66}$$

The reference approximation $u^{\rm ref}$ is constructed using a fixed multi-index set together with high-fidelity timestepping with the specific input options listed in Table 2.

Table 2: Reference approximation input options used for the parametric double glazing problem with $d = 4$

option	value
Reference multi-index set $I_{\rm ref}$	$\{\underline{\alpha}$ s.t. $\|\underline{\alpha}\|_1 \le 4+5\}$
Reference local error tolerance $\delta_{\rm ref}$	10^{-7}

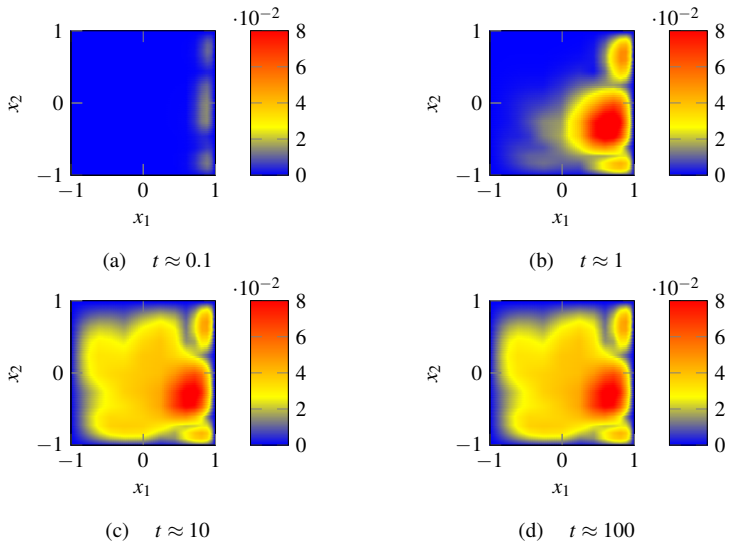
The mean of the reference approximation $\mathbb{E}\left[u^{\rm ref}\right]$ is shown in Fig. 6. The snapshots plotted demonstrate the expected advection–diffusion behaviour: at short times a diffusive boundary layer forms on the "hot wall", as time passes heat is advected clockwise around the domain before eventually reaching a steady state. The standard deviation of the reference approximation is illustrated in Fig. 7. As for the ODE problem in Sect. 3.1, the standard deviation at each point in space will initially increase in magnitude as time passes: this suggests that the multi-index set in an adaptive parametric approximation will need to grow in time to keep the interpolation error under control. In the long time limit the standard deviation reaches a steady, non-zero state.



(a) $t \approx 0.1$ (b) $t \approx 1$

(c) $t \approx 10$ (d) $t \approx 100$

Fig. 6: Snapshots in time of the mean of the reference approximation when solving the parametric double glazing problem with $d = 4$



(a) $t \approx 0.1$ (b) $t \approx 1$

(c) $t \approx 10$ (d) $t \approx 100$

Fig. 7: Snapshots in time of the standard deviation of the reference approximation when solving the parametric double glazing problem with $d = 4$

Having computed a reference solution, we are now in a position to test the reliability and efficiency of the error estimation strategy in Algorithm 2. In the first instance, the synchronisation timestep is initialised as $\tau_0 = T_\partial \log\left(0.9^{-1}\right)$ and the scaling parameters are fixed as $c_+ = 1.2$ and $c_- = 0.5$. Approximations are formed

for collocation points with an initial timestep $\Delta t_0 = 10^{-9}$. The local error tolerance is chosen to be $\delta = 10^{-3}$, the safety factor $c_{\text{tol}} = 10^1$ and the marking factor $\theta = 10^{-1}$. Different choices of these parameters will be investigated later. First, we investigate the global timestepping error estimate (56). Denoting the time-dependent set of collocation points by $Z^{\text{adapt}}(t) := Z^{I(t)}$, the estimates $\{\pi_{\text{ge}}(t;\mathbf{z})\}_{\mathbf{z}\in Z^{\text{adapt}}}$ are computed at the reference times $t \in \mathscr{T}$. The sample mean, minimum and maximum values of these estimates are plotted in Fig. 8 . These are compared to the same statistics for the errors computed with respect to the reference approximation, $\|u^{h,\delta_{\text{ref}}} - u^{h,\delta}\|_{L^2(D)} \approx \mathscr{E}^{\text{ge}}(t) = \|e^{h,\delta}\|_{L^2(D)}$. Looking at these results, we see that the true error is generally *underestimated* for short times and *overestimated* when close to the steady state. There is room for improvement with a more sophisticated global timestepping error estimator, but (56) will suffice for our purposes as we will see shortly, when we consider the total estimator, $\pi(t)$.

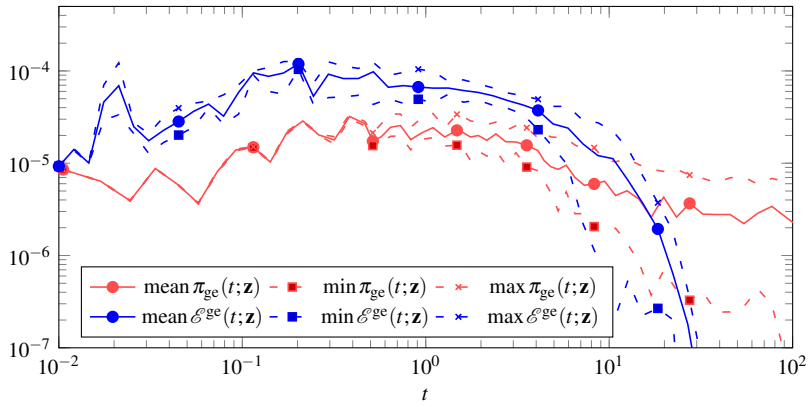
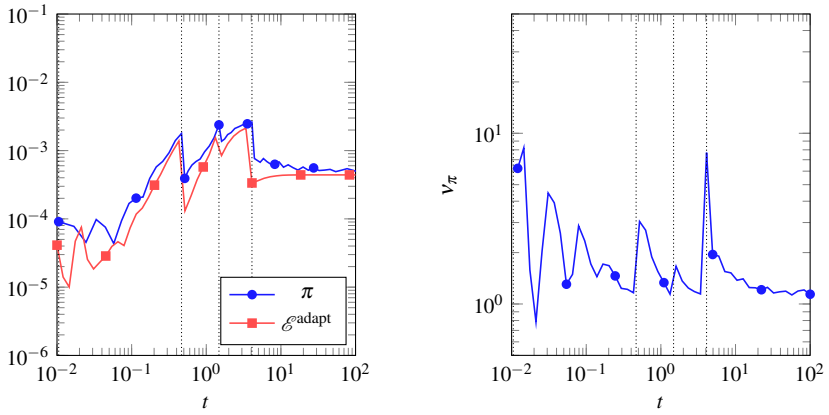


Fig. 8: Sample mean, minimum and maximum of the global timestepping error estimates $\{\pi_{\text{ge}}(t;\mathbf{z})\}_{\mathbf{z}\in Z^{\text{adapt}}}$ (in red with circle, square and cross marks, respectively) and of the reference errors $\{\mathscr{E}^{\text{ge}}(t;\mathbf{z})\}_{\mathbf{z}\in Z^{\text{adapt}}}$ (in blue with the same markers for comparison) for the parametric double glazing problem with $d = 4$

Next, we consider the quality of the total error estimator $\pi(t)$ defined in (49). Fig. 9a demonstrates that the error estimator provides a close estimate of the true error $\mathscr{E}^{\text{adapt}}$. This is reflected in the effectivity of the estimator $\nu_\pi(t)$ plotted in Fig. 9b. The error estimator generally gives a slight overestimation the true error, with the most notable spikes occurring at times where parametric refinement is done.

The corresponding error estimator components are plotted in Fig. 10. It can be seen that parametric refinement occurs when the interpolation error estimate π_I reaches the dynamic tolerance tol. In between these times, where the polynomial approximation space is *fixed*, the interpolation error grows with time. The safety factor $c_{\text{tol}} = 10$ is visible in the value of tol relative to $\pi_{I,\delta}$. A consequence of including c_{tol} is that the interpolation error is the dominant component for the majority of the time integration, therefore when we consider the error estimator effectivity, what we are actually as-



(a) Error estimator $\pi(t)$ and actual error $\mathscr{E}^{\text{adapt}}(t)$ (b) Efficiency $\nu_\pi(t) := \pi(t)/\mathscr{E}^{\text{adapt}}(t)$

Fig. 9: Error estimator $\pi(t)$ for the parametric double glazing problem with $d = 4$. Vertical dotted lines denote the parametric refinement times

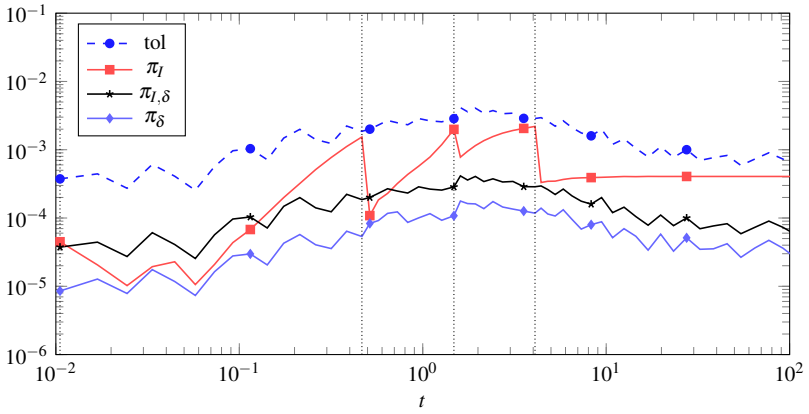
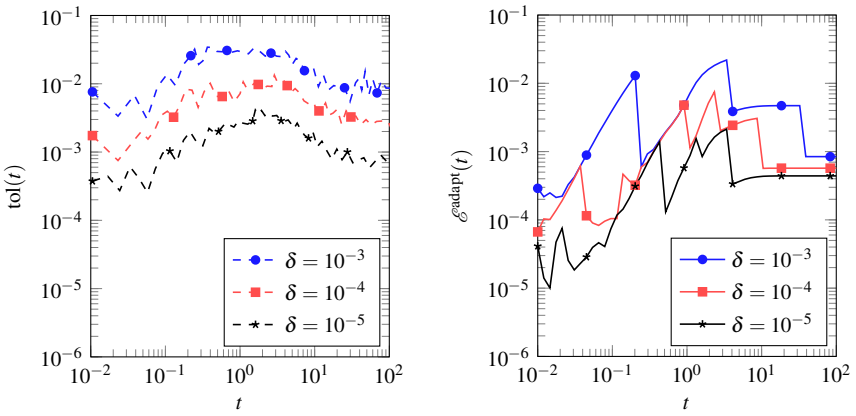


Fig. 10: Error estimator components $\pi_I(t), \pi_{I,\delta}(t), \pi_\delta(t)$ for the parametric double glazing problem with $d = 4$

sessing is the *interpolation* error estimator. In fact, referring back to Fig. 9b, the error estimator efficiency is close to one apart from when the correction term $\pi_{I,\delta}$ makes a significant contribution (generally just after parametric refinement). The resulting overestimation is likely a consequence of the application of the triangle inequality in (43). The final observation is that when parametric refinement is performed we get a reduction of the interpolation error estimate, and this reduction is by approximately one order of magnitude. This corresponds to the choice of marking factor θ.

We will now focus on the optimal selection of the inputs for Algorithm 2 . Reducing the local error tolerance δ will reduce the global timestepping error for each collocation point. The global timestepping error and the local error tolerance are re-



(a) interpolation error tolerance $\text{tol}(t)$

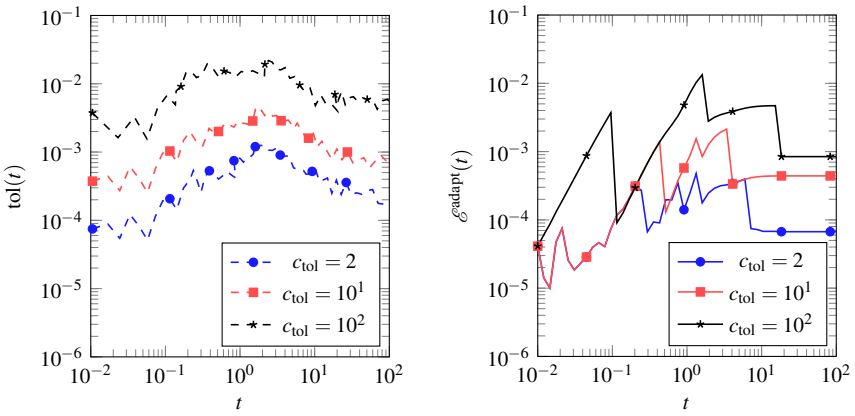
(b) approximation error $\mathscr{E}^{\text{adapt}}(t)$

Fig. 11: Interpolation error tolerance and associated approximation error evolution when varying the local error tolerance δ for the parametric double glazing problem with $d = 4$

lated through (52) with $p = 2$ when using `stabtr`. The definition of the correction estimator (46) suggests the interpolation error tolerance defined in (59) will scale in the same way, so that $\text{tol} \propto \delta^{2/3}$. This relationship is demonstrated in Fig. 11a. The plot in Fig. 11b shows that the resulting approximation error does not scale in the same way for all times. We note that the upper bound tol is briefly attained prior to the approximation being refined.

Modifying the safety factor c_{tol} affects the approximation by changing the relationship between tol and the correction estimator $\pi_{I,\delta}$. A smaller choice of c_{tol} should give an approximation with a smaller interpolation error. Figure 12a demonstrates the change in the interpolation error tolerance as c_{tol} is varied. The resulting approximation errors are plotted in Fig. 12b. We observe that the distinct peaks in the approximation error evolution are suppressed when the tolerance is tightened. Note that a minimum value of $c_{\text{tol}} = 2$ is plotted because with $c_{\text{tol}} = 1$ the interpolation error estimates begin to be limited by timestepping errors. A sensible choice is $c_{\text{tol}} = 10$.

Reducing the marking parameter θ will reduce the approximation error by a larger amount so the refinements will be less frequent. A large marking factor will give a more prudent selection of marked multi-indices. The results in Figure 13 demonstrate that a more carefully constructed multi-index set, corresponding to a larger θ, generally results in fewer collocation points being required which is computationally advantageous. The disadvantage is that more refinement steps are needed. Choosing $\theta = 0.1$ is a reasonable compromise.



(a) interpolation error tolerance $\text{tol}(t)$ (b) approximation error $\mathscr{E}^{\text{adapt}}(t)$

Fig. 12: Interpolation error tolerance and associated approximation error evolution when varying the safety factor c_{tol} for the parametric double glazing problem with $d = 4$

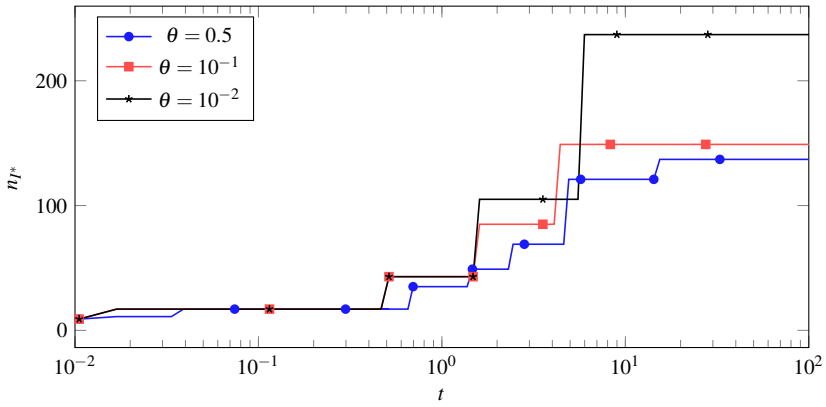


Fig. 13: Number of collocation points n_{I^*} needed for the approximation and error estimation as a function of time when varying the marking parameter θ for the parametric double glazing problem with $d = 4$

6.1 Exploiting stability for computational savings

We now discuss how the smoothing in time that is inherent in the solution of parabolic PDE problems can be exploited to give computational savings. By way of motivation, consider two ODE systems of the form (9) for a particular parameter $\mathbf{z} \in \Gamma$ with initial data $\mathbf{u}^h(t_k, \mathbf{z})$ and $\mathbf{v}^h(t_k, \mathbf{z})$ respectively at a time $t_k \in [0, T]$. The respective solutions $\mathbf{u}^h(t, \mathbf{z})$ and $\mathbf{v}^h(t, \mathbf{z})$ for $t \in [t_k, T]$ can be represented as matrix exponentials and the

difference between the two solutions $\mathbf{r}^h(t,\mathbf{z}) = \mathbf{u}^h(t,\mathbf{z}) - \mathbf{v}^h(t,\mathbf{z})$ is given by

$$\mathbf{r}^h(t,\mathbf{z}) = \exp\left(-(Q^h)^{-1}(\varepsilon D^h + A^h(\mathbf{z}))(t-t_k)\right)\mathbf{r}^h(t_k,\mathbf{z}). \tag{67}$$

If $\mathbf{r}^h(t_k,\mathbf{z})$ represents the error at time t_k, then the behaviour of the norm of this error is controlled by the real components of the eigenvalues of the matrix $(Q^h)^{-1}\left(\varepsilon D^h + A^h(\mathbf{z})\right)$. The diffusion matrix D^h is symmetric and the advection matrix $A^h(\mathbf{z})$ is skew-symmetric for all $\mathbf{z} \in \Gamma$ hence the range of the real components of the eigenvalues depends only on the diffusion matrix [23]. The diffusion matrix D^h is positive-definite hence the eigenvalues are real and positive. This means that, for some constants $C_1, C_2 > 0$ and $t \geq t_k$

$$\|\mathbf{r}^h(t,\mathbf{z})\|_{L^2(D)} \leq C_1 \exp(-C_2\varepsilon(t-t_k)). \tag{68}$$

We can interpret (68) as stating that an error at time t_k will (eventually) decay to zero through time. Note that there is no parameter dependence in the diffusion matrix, so the bound is valid for all $\mathbf{z} \in \Gamma$.

When solving the ODE system (9) using adaptive timestepping, the accumulation of local error through time is related to the decay (68). The local error made at each step will eventually decay through time and this is why local error control can give good global error control. The bound (68) is independent of $\mathbf{z}$ which suggests that the accumulation of local error into global error may be similar for all collocation points. This is in fact the case, as demonstrated in Fig. 8. This insight suggests that it should be possible to use a single global timestepping error estimate $\pi_{\text{ge}}(t) \approx \mathscr{E}^{h,\delta}(t;\mathbf{z})$ for all collocation points $\mathbf{z} \in Z^{\text{adapt}}$ as a mechanism for reducing the overall computational cost. We could choose a fixed $\mathbf{z}^* \in \Gamma$, assume $\mathscr{E}^{h,\delta}(t;\mathbf{z}^*) \approx \mathscr{E}^{h,\delta}(t;\mathbf{z})$ for all $\mathbf{z} \in Z^{t^*}$ and then let $\pi_{\text{ge}}(t) := \pi_{\text{ge}}(t;\mathbf{z}^*)$. Since the parametrisation of the wind in (64) gives $\mathbb{E}[\mathbf{w}(\mathbf{x},\mathbf{y})] = \mathbf{w}(\mathbf{x},\mathbf{0})$, in this case a sensible choice would be

$$\pi_{\text{ge}}(t) := \pi_{\text{ge}}(t;\mathbf{0}). \tag{69}$$

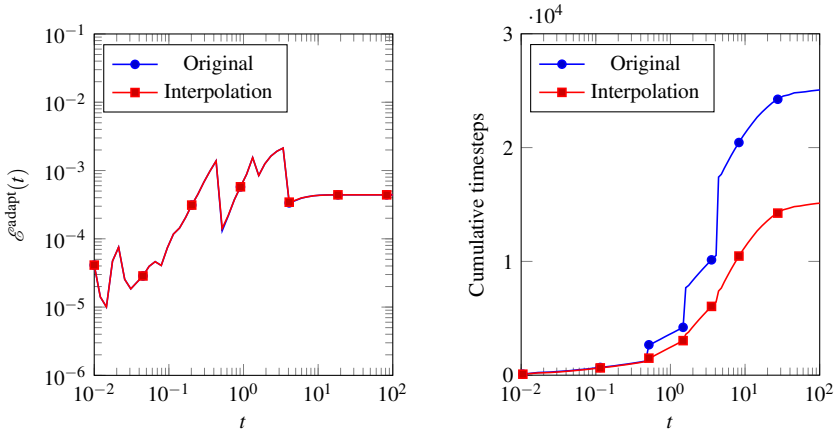
Whenever a new collocation point $\mathbf{z}$ is added at time t in Algorithm 2, we need access to the approximation $u^{h,\delta}(\mathbf{x},t;\mathbf{z})$. In Lines 21 to 23 this is achieved by applying the timestepping algorithm on the interval $[0,t]$ with the appropriate initial condition at $t = 0$. Alternatively, we could simply evaluate the current approximation $u^{\text{adapt}}(\mathbf{x},t,\mathbf{z})$ at time t at the new collocation point $\mathbf{z}$. To implement this strategy, we simply need to replace Lines 21 to 23 by

$$u^{h,\delta}_{local}(\mathbf{x},t;\mathbf{z}) \leftarrow u^{\text{adapt}}(\mathbf{x},t,\mathbf{z}) \tag{70}$$

with the corresponding timestep initialised by $\Delta t_{\mathbf{z}} \leftarrow \Delta t_0$. This modification is again justified by the decay property (68): the interpolation error introduced will decay through time. A significant computational saving will clearly result if this is done, because we do not have to solve ODE systems from time $t = 0$ for newly introduced collocation points.

The results in Fig. 14a for this modified strategy demonstrate that initialisation through interpolation gives an approximation error profile that is essentially identical to that generated using Algorithm 2. If the computational cost is measured by counting the number of timesteps used to approximate the ODE systems associated with

all the collocation points, then the results in Fig. 14b confirm that an equally accurate approximation has been constructed with a significantly reduced cost.



(a) approximation error

(b) total timesteps (all collocation points)

Fig. 14: Comparison of the performance of the original version of Algorithm 2 and the modified version obtained by replacing lines 21-23 with the interpolation step (70), for the parametric double glazing problem with $d = 4$

7 Computational Experiment: $d = 64$

We now consider a problem with a higher parametric dimension and investigate the adaptively constructed multi-index set $I(t)$ generated by Algorithm 2. The test problem is set up as in Sect. 6 but now the wind field is constructed using a parametric stream function. The method of construction is similar to that used in [30,13] and ensures $\mathbf{w}(\mathbf{x},\mathbf{y})$ is divergence-free. Specifically, $\mathbf{w}(\mathbf{x},\mathbf{y}) = \nabla \times \psi(\mathbf{x},\mathbf{y})$ where

$$\psi(\mathbf{x},\mathbf{y}) = \psi_0(\mathbf{x}) + \sum_{i=1}^{d} \sqrt{\lambda_i}\psi_i(\mathbf{x})y_i \tag{71}$$

and we choose $\mathbf{y} \in [-1,1]^d$ and $\rho(\mathbf{y}) = 0.5^d$. Here $\{(\lambda_i, \psi_i(\mathbf{x}))\}_{i=1}^d$ are chosen to be the first d eigenpairs (ordered in terms of decreasing λ_i) of an integral operator $\mathscr{C} : L^2(D) \to L^2(D)$ associated with a covariance function $C : D \times D \to \mathbb{R}$. That is,

$$(\mathscr{C}\psi_i)(\mathbf{x}) := \int_D C(\mathbf{x},\mathbf{x}')\psi_i(\mathbf{x}')\mathrm{d}\mathbf{x}' = \lambda_i\psi_i(\mathbf{x}). \tag{72}$$

Note that (71) takes the form of a truncated Karhunen–Loéve expansion [24, Section 5.4]. We choose the mean $\psi_0(\mathbf{x}) := -(1-x_1^2)(1-x_2^2)$ so that $\mathbf{w}_0(\mathbf{x}) = \nabla \times \psi_0(\mathbf{x})$ is

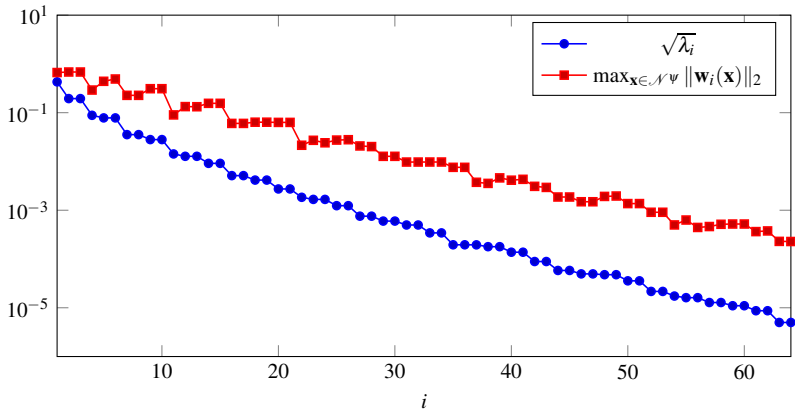


Fig. 15: Approximations of the first $d = 64$ eigenvalues defined by (72) and $\max_{\mathbf{x}\in\mathscr{N}^\Psi} \|\mathbf{w}_i(\mathbf{x})\|_2$ where $\mathbf{w}_i$ is as defined in (74) and $\mathscr{N}^\Psi$ are the spatial nodes used to approximate the eigenpairs

equal to the mean field defined in (62) and we choose the covariance function

$$C(\mathbf{x}_1,\mathbf{x}_2) := \left(\prod_{i,j=1}^{2}(1-x_{i,j}^2)\right)\sigma_0^2 \exp\left(\frac{\|\mathbf{x}_1-\mathbf{x}_2\|_2^2}{L^2}\right) \tag{73}$$

with $\sigma_0^2 = 5$ and $L = 1$. This is a modified version of the so-called Gaussian covariance function. The first term ensures that the stream function has zero variance on the boundary and therefore $\psi(\mathbf{x},\mathbf{y}) = 0$ for all $(\mathbf{x},\mathbf{y}) \in \partial D \times \Gamma$. The ensures that the wind field is always parallel to the boundary of D and there is no inflow or outflow.

The required eigenpairs are approximated using collocation and quadrature on a suitably fine spatial grid [24, Section 7.4]. The 2-norm of the wind perturbations

$$\mathbf{w}_i(\mathbf{x}) := \sqrt{\lambda_i}\,(\nabla \times \psi_i(\mathbf{x})) \tag{74}$$

is computed on the same fine spatial grid and the maximum value over the grid points is plotted for each $i = 1,2,\ldots,64$ in Fig. 15. The higher parameter dimensions should have a negligible effect on the solution as they correspond to very small perturbations. In this test problem we retain $d = 64$ terms in the expansion of the stream function and allow Algorithm 2 to identify a suitable multi-index set.

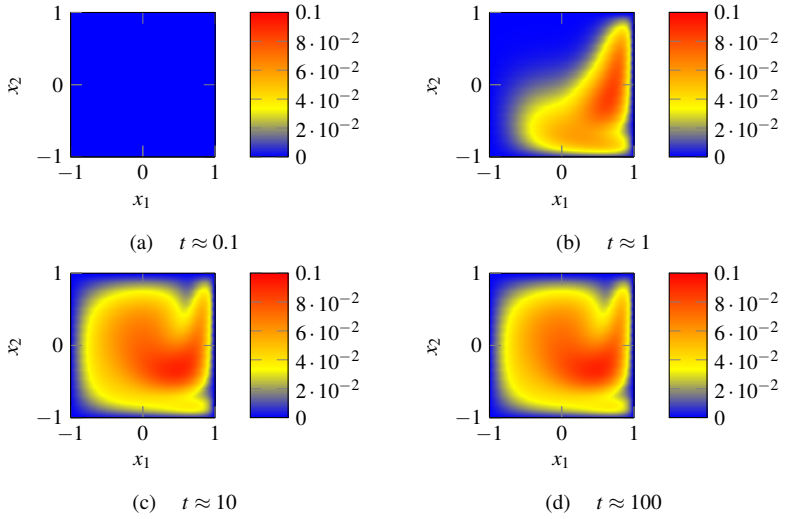
We then use Algorithm 2 to approximate the solution of the corresponding parametric advection–diffusion problem. The input options used for the experiment are summarised in Table 3 and we select the safety factor $c_{\text{tol}} = 10$ and marking parameter $\theta = 10^{-1}$. Approximations corresponding to newly introduced collocation points are added through interpolation as specified in (70).

The behaviour of the mean of the approximation is very similar to that obtained for the $d = 4$ parameter problem (see Fig. 6) so is not shown. This is not surprising as the same mean wind field was chosen in both cases. A more complicated space–time dependence is seen in the standard deviation in Fig. 16. The error estimator

Table 3: IFISS 3.6 and `stabtr` input options used for approximating the parametric double glazing problem with $d = 64$

IFISS 3.6 option	value
Hot wall time constant T_∂	10^{-1}
Grid parameter ℓ ($2^{2\ell}$ elements)	6
Stretched grid	no
Viscosity parameter ε	10^{-1}

`stabtr` option	value
Averaging counter n_*	no averaging
Initial timestep Δt_0	10^{-9}
Local error tolerance δ	10^{-6}
Final time T	100



(a) $t \approx 0.1$ (b) $t \approx 1$

(c) $t \approx 10$ (d) $t \approx 100$

Fig. 16: Snapshots in time of the standard deviation of the approximation to the solution of the parametric double glazing problem with $d = 64$

and its components are plotted in Fig. 17, demonstrating the control Algorithm 2 provides for the interpolation error. The maximum level numbers selected in each dimension to construct the sparse grid interpolant, that is $\alpha_i^{\max} := \max_{\underline{\alpha} \in I(t)} \alpha_i$ for each $i = 1, \ldots, 64$, at four stages of refinement, are shown in Fig. 18. We see that the maximum level number required grows through time and that Algorithm 2 generally chooses higher levels in the parameter dimensions associated with larger eigenvalues λ_i (corresponding to small i). Intuitively, we expect this as the lower dimensions correspond to stronger perturbations from the mean field. The final approximation is constructed using 1371 collocation points. An additional 5880 collocation points are used for the error estimator.

Finally, in Fig. 19 we illustrate the computational cost of applying Algorithm 2. Again this is measured as the total number of timesteps taken by the timestepping algorithm to solve all the ODE systems associated with all collocation points that are used in the computation of both the approximation and the error estimator. For short times, the bulk of the cost is in computing the interpolation error estimator. This is a fundamental issue with hierarchical error estimation and the doubling rule (19).

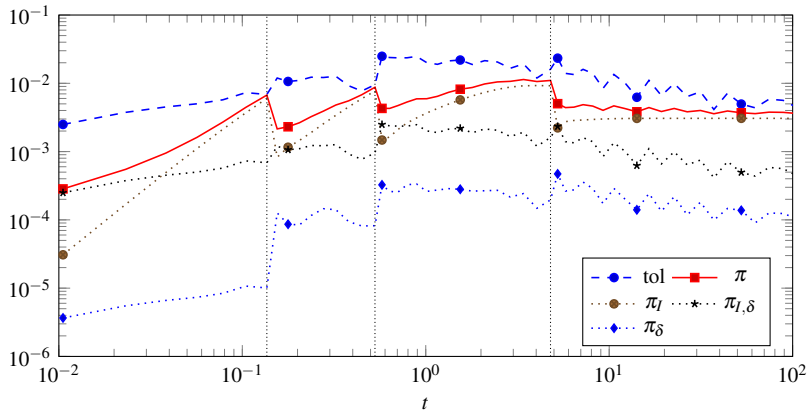


Fig. 17: Error estimator $\pi(t)$ and components $\pi_I(t), \pi_{I,\delta}(t), \pi_\delta(t)$ for the parametric double glazing problem with $d = 64$. Vertical dotted lines denote the parametric refinement times.

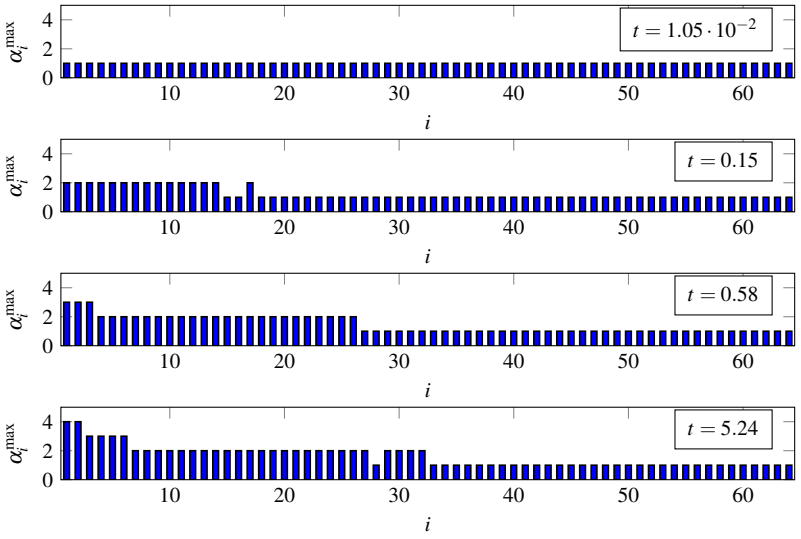


Fig. 18: Maximum level number $\alpha_i^{\max} := \max_{\underline{\alpha} \in I(t)} \alpha_i$ at four snapshots in time ($t \approx 0.01, 0.15, 0.58, 5.24$) for the parametric double glazing problem with $d = 64$

However, over time, as parametric enrichment is performed, numerical solutions to ODE systems associated with collocation points used at earlier times to compute the error estimator can be re-used to compute the approximation. In the long time limit we then see similar computational costs for the approximation and error estimator.

To highlight the advantage of constructing a parametric approximation adaptively, we consider using a fixed naively chosen Smolyak sparse grid with $n_{I_{SM}} = 8321$ points corresponding to the multi-index set $I_{SM} := \{\underline{\alpha} \text{ s.t. } \|\underline{\alpha}\|_1 \leq 64 + 2\}$. This approxima-

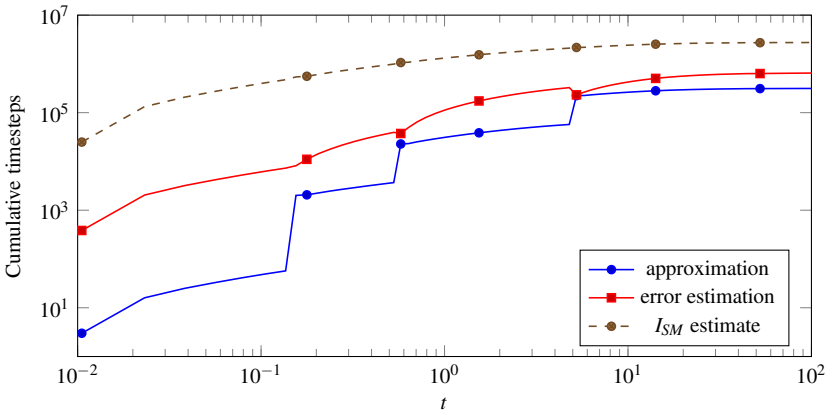


Fig. 19: Total number of timesteps used in constructing the approximation, error estimator and a naive sparse grid approximation with multi-index set $I_{SM} = \{\underline{\alpha} \text{ s.t. } \|\underline{\alpha}\|_1 \leq 64+2\}$ for the parametric double glazing problem with $d = 64$

tion would be exact for all polynomials of total degree less than or equal to 2 [4, Theorem 4]. The cost of obtaining the associated approximation can be estimated by multiplying $n_{I_{SM}}$ by the number of timesteps used by the adaptive timestepping method to solve the ODE system corresponding to the single collocation point $\mathbf{y} = \mathbf{0}$. As we can see in Fig. 19, constructing the approximation with the multi-index set I_{SM} is more expensive than with our adaptive scheme. Moreover, since the multi-index set constructed by Algorithm 2 contains higher level numbers, the associated approximation space contains polynomials of higher total degree than the one associated with I_{SM}. A naive tensor grid approximation in $d = 4$ dimensions would be impossible.

8 Conclusions

We introduced an adaptive nonintrusive stochastic collocation approximation scheme for a time-dependent, parametric PDE and developed a corresponding error estimation strategy. The algorithm is used to monitor errors relating to the timestepping and construct an adaptive approximation in which the interpolation error is controlled to a related tolerance. The efficiency of the error estimation strategy is demonstrated numerically, highlighting the necessity of adaptive stochastic collocation approximation. The adaptive approximation is able to control the error by refining the polynomial approximation space through time and sequentially constructing a tailored approximation capturing the evolving uncertainty in the solution.

The results presented demonstrate the control the algorithm inputs give over constructing an approximation. It is clear that tighter error control can be achieved by reducing the timestepping local error tolerance, but global error control in the timestepping would be required if we want rigorous control of the overall approximation error. Stability of the underlying PDE plays an important role, firstly by ensuring local error

control in the timestepping results in a good approximation and secondly by allowing the computational cost of constructing an approximation to be reduced through interpolation. The results can be shown to extend to higher dimensional parameter spaces.

References

1. Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Journal on Numerical Analysis **45**(3), 1005–1034 (2007). DOI 10.1137/050645142. URL https://doi.org/10.1137/050645142
2. Babuška, I., Tempone, R., Zouraris, G.E.: Galerkin finite element approximations of stochastic elliptic partial differential equations. SIAM Journal on Numerical Analysis **42**(2), 800–825 (2004). DOI 10.1137/S0036142902418680. URL https://doi.org/10.1137/S0036142902418680
3. Bäck, J., Nobile, F., Tamellini, L., Tempone, R.: Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In: J. Hesthaven, E. Ronquist (eds.) Spectral and High Order Methods for Partial Differential Equations, *Lecture Notes in Computational Science and Engineering*, vol. 76, pp. 43–62. Springer (2011). DOI 10.1007/978-3-642-15337-2_3. URL https://doi.org/10.1007/978-3-642-15337-2_3. Selected papers from the ICOSAHOM '09 conference, June 22-26, Trondheim, Norway
4. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Advances in Computational Mathematics **12**(4), 273–288 (2000). DOI 10.1023/A:1018977404843. URL https://doi.org/10.1023/A:1018977404843
5. Bespalov, A., Silvester, D., Xu, F.: Error estimation and adaptivity for stochastic collocation finite elements part i: single-level approximation. SIAM J. Scientific Computing **44**, A3393–A3412 (2022). DOI 10.1137/21M1446745. URL https://doi.org/10.1137/21M1446745
6. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, Ltd (2008). DOI 10.1002/9780470753767. URL https://doi.org/10.1002/9780470753767
7. Cao, Y., Petzold, L.: A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. SIAM Journal on Scientific Computing **26**(2), 359–374 (2004). DOI 10.1137/S1064827503420969. URL https://doi.org/10.1137/S1064827503420969
8. Clenshaw, C.W., Curtis, A.R.: A method for numerical integration on an automatic computer. Numerische Mathematik **2**(1), 197–205 (1960). DOI 10.1007/BF01386223. URL https://doi.org/10.1007/bf01386223
9. Dörfler, W.: A convergent adaptive algorithm for Poisson's equation. SIAM Journal on Numerical Analysis **33**(3), 1106–1124 (1996). DOI 10.1137/0733054. URL https://doi.org/10.1137/0733054
10. Eigel, M., Ernst, O.G., Sprungk, B., Tamellini, L.: On the convergence of adaptive stochastic collocation for elliptic partial differential equations with affine diffusion. SIAM Journal on Numerical Analysis **60**(2), 659–687 (2022). DOI 10.1137/20M1364722. URL https://doi.org/10.1137/20M1364722
11. Elman, H., Silvester, D., Wathen, A.: Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, second edn. Oxford University Press, Oxford, UK (2014). DOI 10.1093/acprof:oso/9780199678792.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199678792.001.0001
12. Elman, H.C., Ramage, A., Silvester, D.J.: Algorithm 866: IFISS, a MATLAB toolbox for modelling incompressible flow. ACM Trans. Math. Softw. **33**(2), 14–31 (2007). DOI 10.1145/1236463.1236469. URL https://doi.org/10.1145/1236463.1236469
13. Elman, H.C., Silvester, D.J.: Collocation methods for exploring perturbations in linear stability analysis. SIAM Journal on Scientific Computing **40**(4), A2667–A2693 (2018). DOI 10.1137/17m1117689. URL https://doi.org/10.1137/17m1117689
14. Evans, L.: Partial Differential Equations. Graduate Studies in Mathematics. American Mathematical Society (2010)
15. Feischl, M., Scaglioni, A.: Convergence of adaptive stochastic collocation with finite elements. Computers & Mathematics with Applications **98**, 139–156 (2021). DOI 10.1016/j.camwa.2021.07.001. URL https://doi.org/10.1016/j.camwa.2021.07.001
16. Gerritsma, M., van der Steen, J.B., Vos, P., Karniadakis, G.: Time-dependent generalized polynomial chaos. Journal of Computational Physics **229**(22), 8333–8363 (2010). DOI 10.1016/j.jcp.2010.07.020. URL https://doi.org/10.1016/j.jcp.2010.07.020

17. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. Numerical algorithms **18**(3), 209–232 (1998). DOI 10.1023/A:1019129717644. URL https://doi.org/10.1023/A:1019129717644
18. Gerstner, T., Griebel, M.: Dimension–adaptive tensor–product quadrature. Computing **71**(1), 65–87 (2003). DOI 10.1007/s00607-003-0015-5. URL https://doi.org/10.1007/s00607-003-0015-5
19. Gresho, P.M., Griffiths, D.F., Silvester, D.J.: Adaptive time-stepping for incompressible flow part I: Scalar advection-diffusion. SIAM Journal on Scientific Computing **30**(4), 2018–2054 (2008). DOI 10.1137/070688018. URL https://doi.org/10.1137/070688018
20. Guignard, D., Nobile, F.: A posteriori error estimation for the stochastic collocation finite element method. SIAM Journal on Numerical Analysis **56**(5), 3121–3143 (2018). DOI 10.1137/17M1155454. URL https://doi.org/10.1137/17M1155454
21. Iserles, A.: A First Course in the Numerical Analysis of Differential Equations, second edn. Cambridge Texts in Applied Mathematics. Cambridge University Press (2008). DOI 10.1017/CBO9780511995569. URL https://doi.org/10.1017/CBO9780511995569
22. Kazashi, Y., Nobile, F., Vidličková, E.: Stability properties of a projector-splitting scheme for dynamical low rank approximation of random parabolic equations. Numer. Math. **149**(4), 973–1024 (2021). DOI 10.1007/s00211-021-01241-4. URL https://doi.org/10.1007/s00211-021-01241-4
23. Loghin, D., van Gijzen, M., Jonkers, E.: Bounds on the eigenvalue range and on the field of values of non-Hermitian and indefinite finite element matrices. Journal of Computational and Applied Mathematics **189**(1), 304–323 (2006). DOI 10.1016/j.cam.2005.04.040. URL https://doi.org/10.1016/j.cam.2005.04.040. Proceedings of The 11th International Congress on Computational and Applied Mathematics
24. Lord, G.J., Powell, C.E., Shardlow, T.: An Introduction to Computational Stochastic PDEs. Cambridge Texts in Applied Mathematics. Cambridge University Press (2014). DOI 10.1017/CBO9781139017329. URL https://www.doi.org/10.1017/CBO9781139017329
25. Nobile, F., Tamellini, L., Tesei, F., Tempone, R.: An adaptive sparse grid algorithm for elliptic pdes with lognormal diffusion coefficient. In: J. Garcke, D. Pflüger (eds.) Sparse Grids and Applications - Stuttgart 2014, pp. 191–220. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-28262-6. URL https://www.doi.org/10.1007/978-3-319-28262-6
26. Nobile, F., Tempone, R., Webster, C.G.: An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. SIAM Journal on Numerical Analysis **46**(5), 2411–2442 (2008). DOI 10.2307/25663069. URL https://www.doi.org/10.2307/25663069
27. Nobile, F., Tempone, R., Webster, C.G.: A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM Journal on Numerical Analysis **46**(5), 2309–2345 (2008). DOI 10.1137/060663660. URL https://doi.org/10.1137/060663660
28. Nobile, F., Vidličková, E.: A posteriori error estimation for the stochastic collocation finite element approximation of the heat equation with random coefficients. In: H.J. Bungartz, J. Garcke, D. Pflüger (eds.) Sparse Grids and Applications - Munich 2018, pp. 127–159. Springer International Publishing, Cham (2021). URL https://doi.org/978-3-030-81362-8
29. Novak, E., Ritter, K.: Simple cubature formulas with high polynomial exactness. Constructive Approximation **15**(4), 499–522 (1999). DOI 10.1007/s003659900119. URL https://doi.org/10.1007/s003659900119
30. Phillips, E.G., Elman, H.C.: A stochastic approach to uncertainty in the equations of MHD kinematics. Journal of Computational Physics **284**, 334–350 (2015). DOI 10.1016/j.jcp.2014.12.002. URL https://doi.org/10.1016/j.jcp.2014.12.002
31. Piazzola, C., Tamellini, L.: The sparse grids MATLAB kit - a MATLAB implementation of sparse grids for high-dimensional function approximation and uncertainty quantification. ArXiv (2203.09314) (2022). URL https://arxiv.org/abs/2203.09314
32. Shampine, L.: Error estimation and control for ODEs. Journal of Scientific Computing **25**, 3–16 (2005). DOI 10.1007/BF02728979. URL https://doi.org/10.1007/BF02728979
33. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite. SIAM Journal on Scientific Computing **18**(1), 1–22 (1997). DOI 10.1137/S1064827594276424. URL https://doi.org/10.1137/S1064827594276424
34. Shampine, L.F., Watts, H.: Algorithm 504: GERK: Global error estimation for ordinary differential equations [d]. ACM Trans. Math. Softw. **2**(2), 200–203 (1976). DOI 10.1145/355681.355690. URL https://doi.org/10.1145/355681.355690
35. Shampine, L.F., Watts, H.: Global error estimates for ordinary differential equations. ACM Transactions on Mathematical Software (TOMS) **2**(2), 172–186 (1976). DOI 10.1145/355681.355687. URL https://doi.org/10.1145/355681.355687

36. Skeel, R.D.: Thirteen ways to estimate global error. Numerische Mathematik **48**(1), 1–20 (1986). DOI 10.1007/BF01389440. URL https://doi.org/10.1007/BF01389440
37. Smolyak, S.A.: Quadrature and interpolation formulae on tensor products of certain function classes. Soviet Math. Dokl. **4**(5), 240–243 (1963)
38. Wan, X., Karniadakis, G.E.: Long-term behavior of polynomial chaos in stochastic flow simulations. Computer Methods in Applied Mechanics and Engineering **195**(41), 5582–5596 (2006). DOI 10.1016/j.cma.2005.10.016. URL https://doi.org/10.1016/j.cma.2005.10.016
39. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. SIAM Journal on Scientific Computing **27**(3), 1118–1139 (2005). DOI 10.1137/040615201. URL https://doi.org/10.1137/040615201
40. Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing **24**(2), 619–644 (2002). DOI 10.1137/S1064827501387826. URL https://doi.org/10.1137/S1064827501387826

Statements & Declarations

Funding

This work was partially supported by the EPSRC grants EP/V048376/1 and EP/W033801/1.

Conflict of interest

The authors declare that they have no conflict of interest.

Data Availability

All data used in the manuscript is numerically generated using open source software packages SPARSE GRIDS MATLAB KIT (V22.02 CALIFORNIA) and IFISS 3.6. The MATLAB source code used to generate the numerical results will be made available on GitHub upon acceptance of the manuscript.