Duality, Adjoint Operators, and Uncertainty in a Complex World

Donald Estep

Department of Mathematics

Department of Statistics

Colorado State University

Research supported by

Department of Energy: DE-FG02-04ER25620 and DE-FG02-05ER25699;

Sandia Corporation: PO299784;

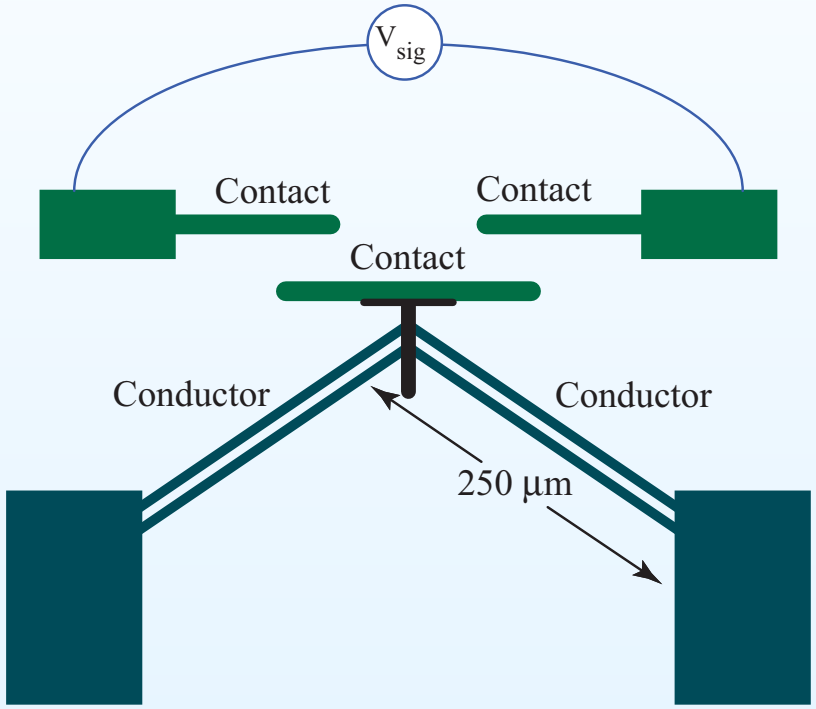
National Aeronautics and Space Administration: NNG04GH63G;

National Science Foundation: DMS-0107832, DGE-0221595003, MSPA-CSE-0434354

Motivation

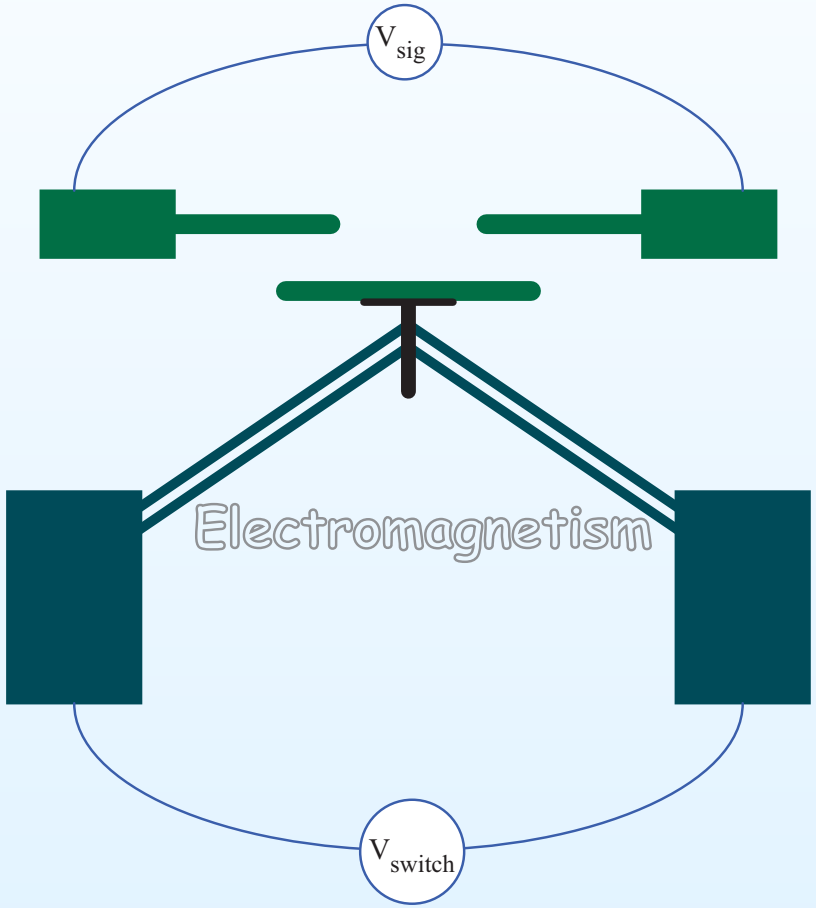
A Multiphysics Model of a Thermal Actuator

Example A thermal actuator is a MEMS scale electric switch



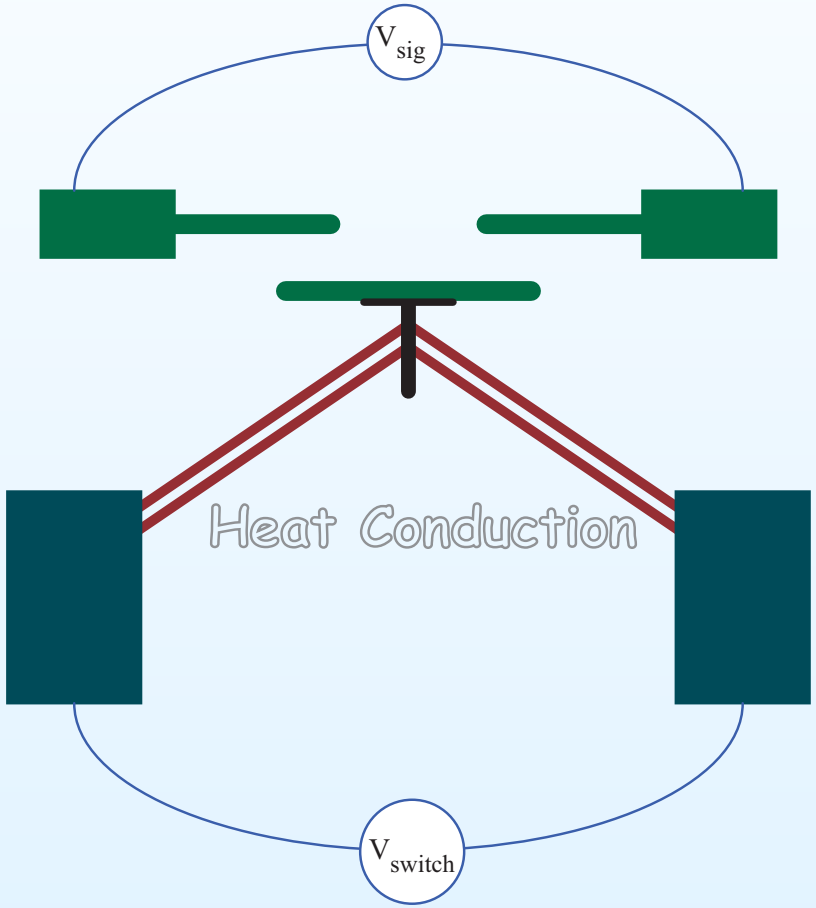
A Multiphysics Model of a Thermal Actuator

Example A thermal actuator is a MEMS scale electric switch



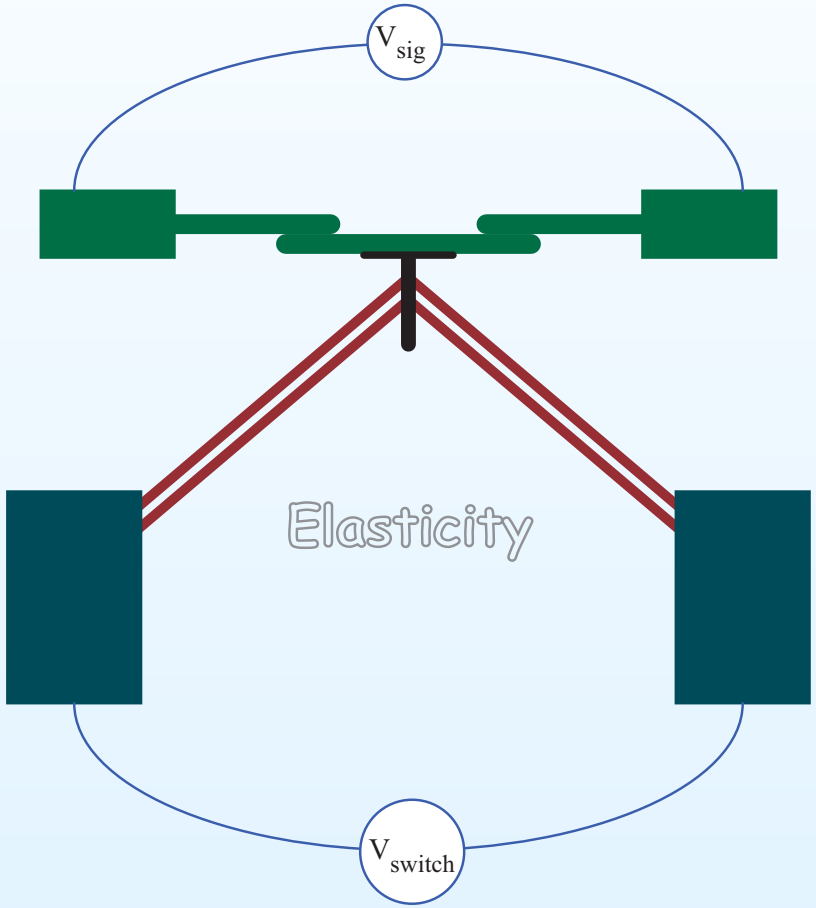
A Multiphysics Model of a Thermal Actuator

Example A thermal actuator is a MEMS scale electric switch



A Multiphysics Model of a Thermal Actuator

Example A thermal actuator is a MEMS scale electric switch



A Model for a Thermal Actuator

Electrostatic current equation ($J = -\sigma \nabla V$)

$$\nabla \cdot (\sigma(T)V) = 0$$

Steady-state energy equation

$$\nabla \cdot (\kappa(T)\nabla T) = \sigma(\nabla V \cdot \nabla V)$$

Steady-state displacement (linear elasticity)

$$\hat{\nabla} \cdot \left(\lambda \,\mathrm{tr}(E) I + 2\mu E - \beta(T - T_{ref})I\right) = 0$$

$$E = \left(\hat{\nabla} d + \hat{\nabla} d^{\top}\right)/2$$

Multiple physical components, multiple scales $\Longrightarrow$ complicated analytic and computational issues

Application Goals for Multiphysics Models

Typical applications of multiphysics models include

- Analyze the effects of uncertainties and variation in the physical properties of the model on its output
- Compute optimal parameter values with respect to producing a desired observation or consequence
- Determine allowable uncertainties for input parameters and data that yield acceptable uncertainty in output
- Predict the behavior of the system by matching model results to experimental observations

Applications requiring results for a range of data and parameter values raise a critical need for quantification and control of uncertainty

Computational Goals for Multiphysics Models

Application of multiphysics models invoke two computational goals

- Compute specific information from multiscale, multiphysics problems accurately and efficiently
- Accurately quantify the error and uncertainty in any computed information

The context is important:

It is often difficult or impossible to obtain uniformly accurate solutions of multiscale, multiphysics problems throughout space and time

Fundamental Tools

We employ two fundamental mathematical tools

- Duality and adjoint operators
- Variational analysis

These tools have a long history of application in analysis of model sensitivity and optimization

More recently, they have been applied to *a posteriori* error estimation for differential equations

Currently, they are being applied to the analysis and application of multiphysics problems

Outline of this course

The plan is roughly

1. Overview of duality and adjoints for linear operators
2. Uses of duality and adjoint operators
3. Adjoints for nonlinear operators
4. Application to computational science and engineering
   - Estimating the error of numerical solutions of differential equations
   - Adaptive mesh refinement
   - Investigations into stability properties of solutions
   - Kernel density estimation
   - Estimating the effects of operator decomposition
   - Adaptive error control for parameter optimization
   - Domain decomposition

Functionals and Dual Spaces

What Information is to be Computed?

The starting point is the computation of particular information obtained from a solution of a multiscale, multiphysics problem

Considering a particular quantity of interest is important because obtaining solutions that are accurate everywhere is often impossible

The application should begin by answering

> What do we want to compute from the model?

We use functionals and dual spaces to answer this

Definition of Linear Functionals

Let X be a vector space with norm $\| \ \|$

A bounded linear functional ℓ is a continuous linear map from X to the reals $\mathbb{R}$, $\ell \in \mathcal{L}(X, \mathbb{R})$

Example For v in $\mathbb{R}^n$ fixed, the map

$$\ell(x) = v \cdot x = (x, v)$$

is a linear functional on $\mathbb{R}^n$

Example For a continuous function f on $[a, b]$,

$$\ell(f) = \int_a^b f(x)\, dx \quad \text{and} \quad \ell(f) = f(y) \text{ for } a \leq y \leq b$$

are linear functionals

Sampling a Vector

A linear functional is a one dimensional "sample" of a vector

Example The linear functional on $\mathbb{R}^n$ given by the inner product with the basis vector e_i gives the i^{th} component of a vector

Example Statistical moments like the expected value $E(X)$ of a random variable X are linear functionals

Example The Fourier coefficients of a continuous function f on $[0, 2\pi]$,

$$c_j = \int_0^{2\pi} f(x)\, e^{-ijx}\, dx$$

are functionals of f

Sampling a Vector

Using linear functionals of a solution means settling for a set of samples rather than the entire solution

Presumably, it is easier to compute accurate samples than solutions that are accurate everywhere

In many situations, we settle for an "incomplete" set of samples

Example We are often happy with a small set of moments of a random variable

Example In practical applications of Fourier series, we truncate the infinite series to a finite number of terms,

$$\sum_{j=-\infty}^{\infty} c_j e^{ijx} \rightarrow \sum_{j=-J}^{J} c_j e^{ijx}$$

Linear functionals and dual spaces

We are interested in the set of reasonable samples

Definition
If X is a normed vector space, the vector space $\mathcal{L}(X, \mathbb{R})$ of continuous linear functionals on X is called the dual space of X, and is denoted by X^*

The dual space is a normed vector space under the dual norm defined for $y \in X^*$ as

$$\|y\|_{X^*} = \sup_{\substack{x \in X \\ \|x\|_X = 1}} |y(x)| = \sup_{\substack{x \in X \\ x \neq 0}} \frac{|y(x)|}{\|x\|}$$

size of a "sample" $=$ largest value of the sample on vectors of length 1

Linear functionals and dual spaces

Example Consider $X = \mathbb{R}^n$. Every vector v in $\mathbb{R}^n$ is associated with a linear functional $F_v(\cdot) = (\cdot, v)$. This functional is clear bounded since $|(x, v)| \leq \|v\| \, \|x\| = C\|x\|$

A classic result in linear algebra is that *all* linear functionals on $\mathbb{R}^n$ have this form, i.e., we can make the identification $(\mathbb{R}^n)^* \simeq \mathbb{R}^n$

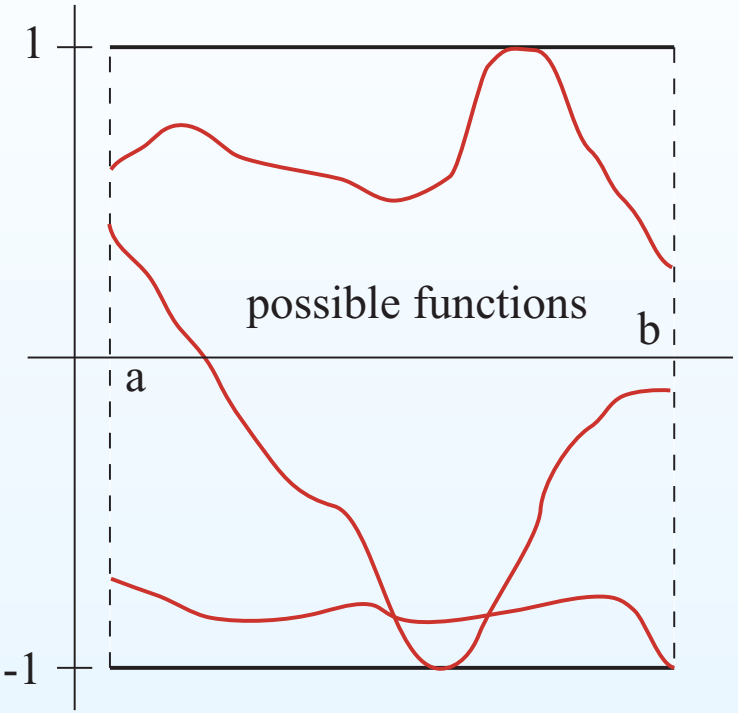
Linear functionals and dual spaces

Example For $C([a,b])$, consider $I(f) = \int_a^b f(x)\,dx$. It is easy to compute

$$\|I\|_{C([a,b])^*} = \sup_{\substack{f\in C([a,b]) \\ \max|f|=1}} \left| \int_a^b f(x)\,dx \right|$$

by looking at a picture.

Linear functionals and dual spaces



Computing the dual norm of the integration functional

The maximum value for $I(f)$ is clearly given by $f = 1$ or $f = -1$, and $\|I\|_{C([a,b])^*} = b - a$

Linear functionals and dual spaces

Recall Hölder's inequality: if $f \in L^p(\Omega)$ and $g \in L^q(\Omega)$ with $p^{-1} + q^{-1} = 1$ for $1 \leq p, q \leq \infty$, then

$$\|fg\|_{L^1(\Omega)} \leq \|f\|_{L^p(\Omega)} \|g\|_{L^q(\Omega)}$$

Example Each g in $L^q(\Omega)$ is associated with a bounded linear functional on $L^p(\Omega)$ when $p^{-1} + q^{-1} = 1$ and $1 \leq p, q \leq \infty$ by

$$F(f) = \int_\Omega g(x) f(x)\, dx$$

We can "identify" $(L^p)^*$ with L^q when $1 < p, q < \infty$

The cases $p = 1, q = \infty$ and $p = \infty, q = 1$ are trickier

Duality for Hilbert Spaces

Hilbert spaces are Banach spaces with an inner product $(\, , \,)$

Example $\mathbb{R}^n$ and L^2 are Hilbert spaces

If X is a Hilbert space, then $\psi \in X$ determines a bounded linear functional via the inner product

$$\ell_\psi(x) = (x, \psi), \quad x \in X$$

The Riesz Representation theorem says this is the only kind of linear functional on a Hilbert space

We can identify the dual space of a Hilbert space with itself

Linear functionals are commonly represented as inner products

Riesz Representors

Some useful choices of Riesz representors ψ for functions f in a Hilbert space include:

- $\psi = \chi_\omega / |\omega|$ gives the error in the average value of f over a subset $\omega \subset \Omega$, where χ_ω is the characteristic function of ω
- $\psi = \delta_c$ gives the average value $\oint_c f(s)\, ds$ of f on a curve c in $\mathbb{R}^n$, $n = 2, 3$, and $\psi = \delta_s$ gives the average value of f over a plane surface s in $\mathbb{R}^3$ (δ denotes the corresponding delta function)
- We can obtain average values of derivatives using dipoles similarly
- $\psi = f / \|f\|$ gives the L^2 norm of f

Only some of these ψ have spatially local support

The Bracket Notation

We “borrow” the Hilbert space notation for the general case:

Definition
If x is in X and y is in X^*, we denote the value

$$y(x) = \langle x, y \rangle$$

This is called the bracket notation

The generalized Cauchy inequality is

$$|\langle x, y \rangle| \leq \|x\|_X \, \|y\|_{X^*}, \quad x \in X, \, y \in X^*$$

Adjoint Operators

Motivation for the Adjoint Operator

Let X, Y be normed vector spaces

Assume that $L \in \mathcal{L}(X, Y)$ is a continuous linear map

The goal is to compute a sample or functional value of the output

$$\ell\big(L(x)\big), \quad \text{some } x \in X$$

Some important questions:

- Can we find a way to compute the sample value efficiently?
- What is the error in the sample value if approximations are involved?
- Given a sample value, what can we say about x?
- Given a collection of sample values, what can we say about L?

Definition of the Adjoint Operator

Let X, Y be normed vector spaces with dual spaces X^*, Y^*

Assume that $L \in \mathcal{L}(X,Y)$ is a continuous linear map

For each $y^* \in Y^*$ there is an $x^* \in X^*$ defined by

$$x^*(x) = y^*\big(L(x)\big)$$

$$sample\ of\ x\ in\ X = sample\ of\ image\ L(x)\ of\ x\ in\ Y$$

The adjoint map $L^* : Y^* \to X^*$ satisfies the bilinear identity

$$\langle L(x), y^* \rangle = \langle x, L^*(y^*) \rangle, \quad x \in X,\, y^* \in Y^*$$

Adjoint of a Matrix

Example Let $X = \mathbb{R}^m$ and $Y = \mathbb{R}^n$ with the standard inner product and norm

$L \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$ is associated with a $n \times m$ matrix A:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

and

$$y_i = \sum_{j=1}^{m} a_{ij} x_j, \quad 1 \leq i \leq n$$

Adjoint of a Matrix

The bilinear identity reads

$$(Lx, y) = (x, L^*y), \quad x \in \mathbb{R}^m,\, y \in \mathbb{R}^n.$$

For a linear functional $y^* = (y_1^*, \cdots, y_n^*)^\top \in Y^*$

$$L^*y^*(x) = y^*(L(x)) = \left((y_1^*, \cdots, y_n^*), \begin{pmatrix} \sum_{j=1}^m a_{1j}x_j \\ \vdots \\ \sum_{j=1}^m a_{nj}x_j \end{pmatrix} \right)$$

$$= \sum_{j=1}^m y_1^* a_{1j} x_j + \cdots \sum_{j=1}^m y_n^* a_{nj} x_j$$

$$= \sum_{j=1}^m (\sum_{i=1}^n y_i^* a_{ij}) x_j$$

Adjoint of a Matrix

$L^*(y^*)$ is given by the inner product with $\tilde{y} = (\tilde{y}_1, \cdots, \tilde{y}_m)^\top$ where

$$\tilde{y}_j = \sum_{i=1}^{n} y_i^* a_{ij}.$$

The matrix A^* of L^* is

$$A^* = \begin{pmatrix} a_{11}^* & \cdots & a_{1n}^* \\ \vdots & & \vdots \\ a_{m1}^* & \cdots & a_{mn}^* \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ \vdots & & & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix} = A^\top.$$

Properties of Adjoint Operators

Theorem

Let X, Y, and Z be normed linear spaces. For $L_1, L_2 \in \mathcal{L}(X,Y)$:

$$L_1^* \in \mathcal{L}(Y^*, X^*)$$

$$\|L_1^*\| = \|L_1\|$$

$$0^* = 0$$

$$(L_1 + L_2)^* = L_1^* + L_2^*$$

$$(\alpha L_1)^* = \alpha L_1^*, \quad \text{all } \alpha \in \mathbb{R}$$

If $L_2 \in \mathcal{L}(X,Y)$ and $L_1 \in \mathcal{L}(Y,Z)$, then $(L_1 L_2)^* \in \mathcal{L}(Z^*, X^*)$ and

$$(L_1 L_2)^* = L_2^* L_1^*.$$

Adjoints for Differential Operators

Computing adjoints for differential operators is more complicated

We need to make assumptions on the spaces, e.g., the domain of the operator and the corresponding dual space have to be sufficiently “big”

The Hahn-Banach theorem is often involved

We consider differential operators on Sobolev spaces using the L^2 inner product and ignore analytic technicalities

Adjoints for Differential Operators

The adjoint of the differential operator L

$$(Lu, v) \rightarrow (u, L^* v)$$

is obtained by a succession of integration by parts

Boundary terms involving functions and derivatives arise from each integration by parts

We use a two step process

1. We first compute the formal adjoint by assuming that all functions have compact support and ignoring boundary terms

2. We then compute the adjoint boundary and data conditions to make the bilinear identity hold

Formal Adjoints

Example Consider

$$Lu(x) = -\frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right) + \frac{d}{dx}(b(x)u(x))$$

on $[0,1]$. Integration by parts neglecting boundary terms gives

$$-\int_0^1 \frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right)v(x)\,dx$$

$$= \int_0^1 a(x)\frac{d}{dx}u(x)\frac{d}{dx}v(x)\,dx - a(x)\frac{d}{dx}u(x)v(x)\Big|_0^1$$

$$= -\int_0^1 u(x)\frac{d}{dx}\left(a(x)\frac{d}{dx}v(x)\right)\,dx + u(x)a(x)\frac{d}{dx}v(x)\Big|_0^1$$

Formal Adjoints

$$\int_0^1 \frac{d}{dx}(b(x)u(x))v(x)\,dx = -\int_0^1 u(x)b(x)\frac{d}{dx}v(x)\,dx + b(x)u(x)v(x)\Big|_0^1,$$

All of the boundary terms vanish

Therefore,

$$Lu(x) = -\frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right) + \frac{d}{dx}(b(x)u(x))$$

$$\Longrightarrow\ L^*v = -\frac{d}{dx}\left(a(x)\frac{d}{dx}v(x)\right) - b(x)\frac{d}{dx}(v(x))$$

Formal Adjoints

In higher space dimensions, we use the divergence theorem

Example A general linear second order differential operator L in $\mathbb{R}^n$ can be written

$$L(u) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} b_i \frac{\partial u}{\partial x_i} + cu,$$

where $\{a_{ij}\}$, $\{b_i\}$, and c are functions of $x_1, x_2, \cdots, x_n$. Then,

$$L^*(u) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 (a_{ij} v)}{\partial x_i \partial x_j} - \sum_{i=1}^{n} \frac{\partial (b_i v)}{\partial x_i} + cv$$

Adjoint Boundary Conditions

In the second stage, we deal with the boundary terms that arise during integration by parts

Definition
The adjoint boundary conditions are the minimal conditions required in order that the bilinear identity hold true

The form of the boundary conditions imposed on the differential operator is important, but not the values

We assume homogeneous boundary values for the differential operator when determining the adjoint conditions

Adjoint Boundary Conditions

Example Consider Newton's equation of motion $s''(x) = f(x)$ with $x =$ "time", normalized with mass 1

If $s(0) = s'(0) = 0$ and $0 < x < 1$,

$$\int_0^1 (s''v - sv'')\,dx = (vs' - sv')\Big|_0^1$$

The boundary conditions imply the contributions at $x = 0$ vanish, while at $x = 1$ we have

$$v(1)s'(1) - v'(1)s(1)$$

The adjoint boundary conditions are $v(1) = v'(1) = 0$

Adjoint Boundary Conditions

Example Since

$$\int_\Omega (u\Delta v - v\Delta u)\,dx = \int_{\partial\Omega} \left(u\frac{\partial v}{\partial n} - v\frac{\partial u}{\partial n}\right)\,ds,$$

the Dirichlet and Neumann boundary value problems for the Laplacian are their own adjoints

Adjoint Boundary Conditions

Example Let $\Omega \subset \mathbb{R}^2$ be bounded with a smooth boundary and let $s =$ arclength along the boundary

Consider

$$\begin{cases} -\Delta u = f, & x \in \Omega, \\ \frac{\partial u}{\partial n} + \frac{\partial u}{\partial s} = 0, & x \in \partial\Omega \end{cases}$$

Since

$$\int_\Omega (u\Delta v - v\Delta u)\, dx = \int_{\partial\Omega} \left(u\left(\frac{\partial v}{\partial n} - \frac{\partial v}{\partial s}\right) - v\left(\frac{\partial u}{\partial n} + \frac{\partial u}{\partial s}\right)\right) ds,$$

the adjoint problem is

$$\begin{cases} -\Delta v = g, & x \in \Omega, \\ \frac{\partial v}{\partial n} - \frac{\partial v}{\partial s} = 0, & x \in \partial\Omega. \end{cases}$$

Adjoint for an Evolution Operator

For an initial value problem, we have $\frac{d}{dt}$ and an initial condition

Now

$$\int_0^T \frac{du}{dt} v \, dt = u(t)v(t)\Big|_0^T - \int_0^T u \frac{dv}{dt} \, dt$$

The boundary term at 0 vanishes because $u(0) = 0$

The adjoint is a final-value problem with "initial" condition $v(T) = 0$

The adjoint problem has $-\frac{dv}{dt}$ and time "runs backwards"

Adjoint for an Evolution Operator

Example

$$\begin{cases} Lu = \frac{du}{dt} - \Delta u = f, & x \in \Omega,\ 0 < t \leq T, \\ u = 0, & x \in \partial\Omega,\ 0 < t \leq T, \\ u = u_0, & x \in \Omega,\ t = 0 \end{cases}$$

$$\Longrightarrow \begin{cases} L^*v = -\frac{dv}{dt} - \Delta v = \psi, & x \in \Omega,\ T > t \geq 0, \\ v = 0, & x \in \partial\Omega,\ T > t \geq 0, \\ v = 0, & x \in \Omega,\ t = T \end{cases}$$

The Usefulness of Duality and Adjoints

The Dual Space is Nice

The dual space can be better behaved than the original normed vector space

Theorem
If X is a normed vector space over $\mathbb{R}$, then X^* is a Banach space (whether or not X is a Banach space)

Condition of an Operator

There is an intimate connection between the adjoint problem and the stability properties of the original problem

Theorem
The singular values of a matrix $\mathbf{L}$ are the square roots of the eigenvalues of the square, symmetric transformations $\mathbf{L}^*\mathbf{L}$ or $\mathbf{L}\mathbf{L}^*$

This connects the condition number of a matrix $\mathbf{L}$ to $\mathbf{L}^*$

Solving Linear Problems

Given normed vector spaces X and Y, an operator $\mathcal{L}(X,Y)$, and $b \in Y$, find $x \in X$ such that

$$Lx = b$$

Theorem
A necessary condition that b is in the range of L is $y^*(b) = 0$ for all y^* in the null space of L^*

This is a sufficient condition if the range of L is closed in Y

Example If A is an $n \times m$ matrix, a necessary and sufficient condition for the solvability of $Ax = b$ is b is orthogonal to all linearly independent solutions of $A^\top y = 0$

Solving Linear Problems

Example When X is a Hilbert space and $L \in \mathcal{L}(X, Y)$, then necessarily the range of L^* is a subset of the orthogonal complement of the null space of L

If the range of L^* is "large", then the orthogonal complement of the null space of L must be "large" and the null space of L must be "small"

The existence of sufficiently many solutions of the homogeneous adjoint equation $L^*\phi = 0$ implies there is at most one solution of $Lu = b$ for a given b

The Augmented System

Consider the potentially nonsquare system $Ax = b$, where A is a $n \times m$ matrix, $x \in \mathbb{R}^m$, and $b \in \mathbb{R}^n$

The augmented system is obtained by adding a problem for the adjoint, e.g. an $m \times n$ system $A^\top y = c$, where y and c are nominally independent of x and b

The new problem is an $(n+m) \times (n+m)$ symmetric problem

$$\begin{pmatrix} 0 & A \\ A^\top & 0 \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$$

The Augmented System

Consequences:

- The theorem on the adjoint condition for solvability falls out right away
- This yields a “natural” definition of a solution in the over-determined case
- This gives conditions for a solution to exist in the under-determined case

The more over-determined the original system, the more under-determined the adjoint system, and so forth

The Augmented System

Example Consider $2x_1 + x_2 = 4$, where $L : \mathbb{R}^2 \to \mathbb{R}$.

$L^* : \mathbb{R} \to \mathbb{R}^2$ is given by $L^* = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

The extended system is

$$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ c_1 \\ c_2 \end{pmatrix},$$

so $2c_1 = c_2$ is required in order to have a solution

The Augmented System

Example If the problem is

$$2x_1 + x_2 = 4$$
$$x_2 = 3,$$

with $L : \mathbb{R}^2 \to \mathbb{R}^2$, then there is a unique solution

The extended system is

$$\begin{pmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ 4 \\ 3 \end{pmatrix},$$

where we can specify any values for c_1, c_2

The Augmented System

In the under-determined case, we can eliminate the deficiency by posing the method of solution

$$\begin{aligned} AA^\top y &= b \\ x &= A^\top y \end{aligned} \qquad \text{or} \qquad \begin{aligned} L(L^*(y)) &= b \\ x &= L^*(y) \end{aligned}$$

The Augmented System

This works for differential operators

Example Consider the under-determined problem

$$\text{div}\, F = \rho$$

The adjoint to div is -grad modulo boundary conditions

If $F = \text{grad}\, u$, where u is subject to the boundary condition $u(\text{“}\infty\text{”}) = 0$, then we obtain the “square”, well-determined problem

$$\text{div grad}\, u = \Delta u = -\rho,$$

which has a unique solution because of the boundary condition

Greens Functions

Suppose we wish to compute a functional $\ell(x)$ of the solution $x \in \mathbb{R}^n$ of a $n \times n$ system

$$\mathbf{L}x = b$$

For a linear functional $\ell(\cdot) = (\cdot, \psi)$, we define the adjoint problem

$$\mathbf{L}^*\phi = \psi$$

Variational analysis yields the representation formula

$$\ell(x) = (x, \psi) = (x, \mathbf{L}^*\phi) = (\mathbf{L}x, \phi) = (b, \phi)$$

We can compute many solutions by computing one adjoint solution and taking inner products

Greens Functions

This is the method of Green's functions in differential equations

Example For

$$\begin{cases} -\Delta u = f, & x \in \Omega, \\ u = 0, & x \in \partial\Omega \end{cases}$$

the Green's function solves

$$-\Delta\phi = \delta_x \qquad (delta\ function\ at\ x)$$

This yields

$$u(x) = (u, \delta_x) = (f, \phi)$$

The generalized Green's function solves the adjoint problem with general functional data, rather than just δ_x

The imposition of the adjoint boundary conditions is crucial

Nonlinear Operators

Nonlinear Operators

There is no “natural” adjoint for a general nonlinear operator

We assume that the Banach spaces X and Y are Sobolev spaces and use $(\ ,\ )$ for the L^2 inner product, and so forth

We define the adjoint for a specific kind of nonlinear operator

We assume f is a nonlinear map from X into Y, where the domain of f is a convex set

A Perturbation Operator

We choose u in the domain of f and define

$$F(e) = f(u+e) - f(u),$$

where we think of e as representing an "error", i.e., $e = U - u$

The domain of F is

$$\{v \in X \,|\, v + u \in \text{domain of } f\}$$

We assume that the domain of F is independent of e and dense in X

Note that 0 is in the domain of F and $F(0) = 0$

A Perturbation Operator

Two reasons to work with functions of this form:

- This is the kind of nonlinearity that arises when estimating the error of a numerical solution or studying the effects of perturbations
- Nonlinear problems typically do not enjoy the global solvability that characterizes linear problems, only a local solvability

Definition 1

The first definition is based on the bilinear identity

Definition
An operator $A^*(e)$ is an adjoint operator corresponding to F if

$(F(e), w) = (e, A^*(e)w)$ for all $e \in$ domain of F, $w \in$ domain of A^*

This is an adjoint operator associated with F, not the adjoint operator to F

Definition 1

Example Suppose that F can be represented as $F(e) = A(e)e$, where $A(e)$ is a linear operator with the domain of F contained in the domain of A

For a fixed e in the domain of F, define the adjoint of A satisfying

$$(A(e)w, v) = (w, A^*(e)v)$$

for all $w \in$ domain of A, $v \in$ domain of A^*

Substituting $w = e$ shows this defines an adjoint of F as well

If there are several such linear operators A, then there will be several different possible adjoints.

An Adjoint for a Nonlinear Differential Equation

Example Let $(t,x) \in \Omega = (0,1) \times (0,1)$, with $X = X^* = Y = Y^* = L^2$ denoting the space of periodic functions in t and x, with period equal to 1

Consider a periodic problem

$$F(e) = \frac{\partial e}{\partial t} + e\frac{\partial e}{\partial x} + ae = f$$

where $a > 0$ is a constant and the domain of F is the set of continuously differentiable functions.

An Adjoint for a Nonlinear Differential Equation

We can write $F(e) = A_i(e)e$ where

$$A_1(e)v = \frac{\partial v}{\partial t} + e\frac{\partial v}{\partial x} + av \implies A_1^*(e)w = -\frac{\partial w}{\partial t} - \frac{\partial (ew)}{\partial x} + aw$$

$$A_2(e)v = \frac{\partial v}{\partial t} + \left(a + \frac{\partial e}{\partial x}\right)v \implies A_2^*(e)w = -\frac{\partial w}{\partial t} + \left(a + \frac{\partial e}{\partial x}\right)w$$

$$A_3(e)v = \frac{\partial v}{\partial t} + \frac{1}{2}\frac{\partial (ev)}{\partial x} + av \implies A_3^*(e)w = -\frac{\partial w}{\partial t} - \frac{e}{2}\frac{\partial w}{\partial x} + aw$$

Definition 2

If the nonlinearity is Frechet differentiable, we base the second definition of an adjoint on the integral mean value theorem

The integral mean value theorem states

$$f(U) = f(u) + \int_0^1 f'(u + se)\, ds\; e$$

where $e = U - u$ and f' is the Frechet derivative of f

Definition 2

We rewrite this as

$$F(e) = f(U) - f(u) = A(e)e$$

with

$$A(e) = \int_0^1 f'(u + se)\, ds.$$

Note that we can apply the integral mean value theorem to F:

$$A(e) = \int_0^1 F'(se)\, ds.$$

To be precise, we should discuss the smoothness of F.

Definition 2

Definition
For a fixed e, the adjoint operator $A^*(e)$, defined in the usual way for the linear operator $A(e)$, is said to be an adjoint for F

Example Continuing the previous example,

$$F'(e)v = \frac{\partial v}{\partial t} + e\frac{\partial v}{\partial x} + \left(a + \frac{\partial e}{\partial x}\right) v.$$

After some technical analysis of the domains of the operators involved,

$$A^*(e)w = -\frac{\partial w}{\partial t} - \frac{e}{2}\frac{\partial w}{\partial x} + aw.$$

This coincides with the third adjoint computed above

Application and Analysis

Solving the Adjoint Problem

In the first part of this course, I tried to hint at the theoretical importance of the adjoint with respect to the study of the properties of a given operator

In the second part of this course, I will try to hint at the practical importance of the adjoint problem

I hope to motivate going to the expense of actually computing solutions of adjoint problems numerically

Accurate Error Estimation

A Posteriori Error Analysis

Problem: Estimate the error in a quantity of interest computed using a numerical solution of a differential equation

We assume that the quantity of information can be represented as a linear functional of the solution

We use the adjoint problem associated with the linear functional

What About Convergence Analysis?

Recall the standard *a priori* convergence result for an initial value problem

$$\begin{cases} \dot{y} = f(y), \quad 0 < t, \\ y(0) = y_0 \end{cases}$$

Let $Y \approx y$ be an approximation associated with time step Δt

A typical *a priori* bound is

$$\|Y - y\|_{L^\infty(0,t)} \leq C\, \mathbf{e^{Lt}}\, \Delta t^p \left\| \frac{d^{p+1}y}{dt^{p+1}} \right\|_{L^\infty(0,t)}$$

L is often large in practice, e.g. $L \sim 100 - 10000$

It is typical for an *a priori* convergence bound to be orders of magnitude larger than the error

A Linear Algebra Problem

We compute a quantity of interest (u, ψ) from a solution of

$$\mathbf{A}u = b$$

If U is an approximate solution, we estimate the error

$$(e, \psi) = (u - U, \psi)$$

We can compute the residual

$$R = \mathbf{A}U - b$$

Using the adjoint problem $\mathbf{A}^\top \phi = \psi$, variational analysis gives

$$|(e, \psi)| = |(e, \mathbf{A}^\top \phi)| = |(Ae, \phi)| = |(R, \phi)|$$

We solve for ϕ numerically to compute the estimate

Discretization by the Finite Element Method

We first consider: approximate $u : \mathbb{R}^n \to \mathbb{R}$ solving

$$\begin{cases} Lu = f, & x \in \Omega, \\ u = 0, & x \in \partial\Omega, \end{cases}$$

where

$$L(D, x)u = -\nabla \cdot a(x)\nabla u + b(x) \cdot \nabla u + c(x)u(x),$$

- $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$, is a convex polygonal domain
- $a = (a_{ij})$, where $a_{i,j}$ are continuous and there is a $a_0 > 0$ such that $v^\top a v \geq a_0$ for all $v \in \mathbb{R}^n \setminus \{0\}$ and $x \in \Omega$
- $b = (b_i)$ where b_i is continuous
- c and f are continuous

Discretization by the Finite Element Method

The variational formulation reads

Find $u \in H_0^1(\Omega)$ such that

$$A(u,v) = (a\nabla u, \nabla v) + (b \cdot \nabla u, v) + (cu, v) = (f, v)$$

for all $v \in H_0^1(\Omega)$

$H_0^1(\Omega)$ is the space of $L^2(\Omega)$ functions whose first derivatives are in $L^2(\Omega)$

This says that the solution solves the "average" form of the problem for a large number of weights v

Discretization by the Finite Element Method

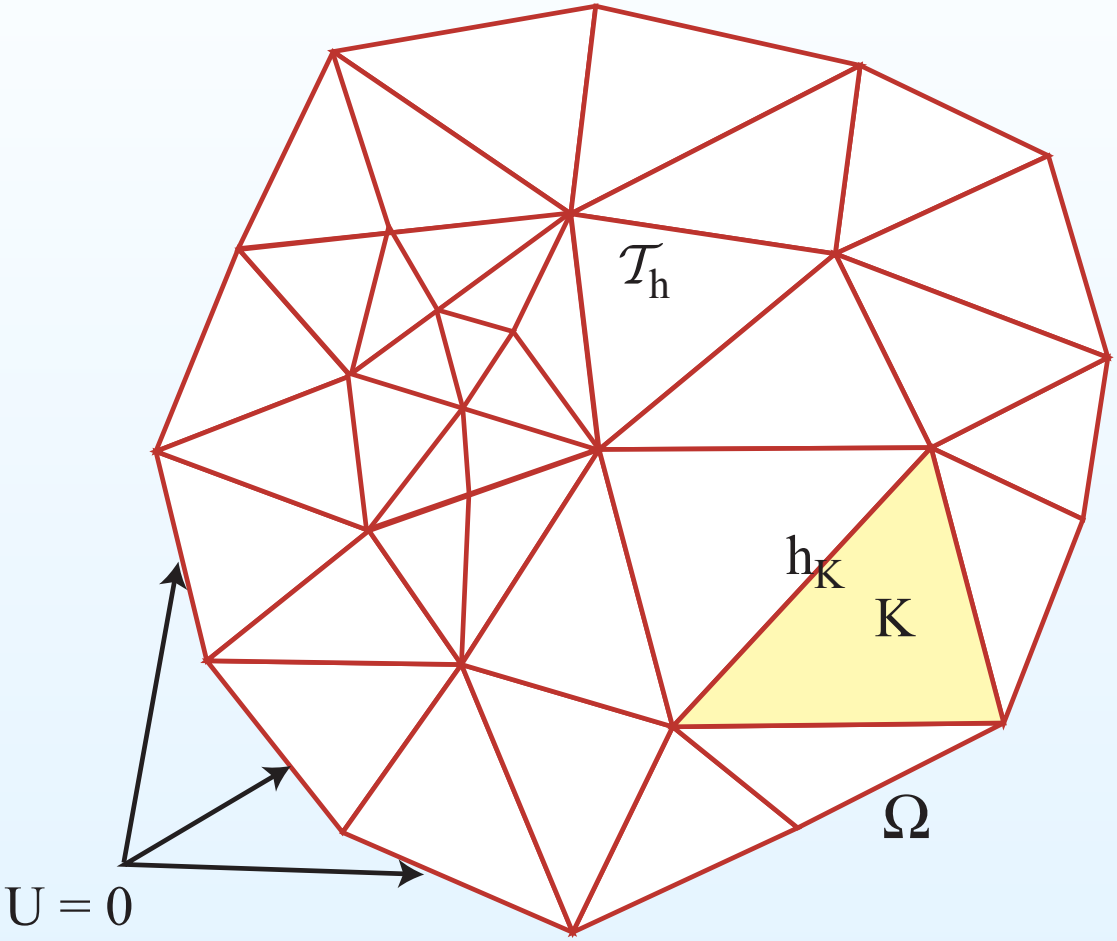
We construct a triangulation of Ω into simplices, or elements, such that boundary nodes of the triangulation lie on $\partial\Omega$

$\mathcal{T}_h$ denotes a simplex triangulation of Ω that is locally quasi-uniform, i.e. no arbitrarily long, skinny triangles

h_K denotes the length of the longest edge of $K \in \mathcal{T}_h$ and h is the mesh function with $h(x) = h_K$ for $x \in K$

We also use h to denote $\max_K h_K$

Discretization by the Finite Element Method



Triangulation of the domain Ω

Discretization by the Finite Element Method

V_h denotes the space of functions that are

- continuous on Ω
- piecewise linear with respect to $\mathcal{T}_h$
- zero on the boundary

$V_h \subset H_0^1(\Omega)$, and for smooth functions, the error of interpolation into V_h is $\mathbf{O}(h^2)$ in $\| \ \|$

Definition
The finite element method is:

Compute $U \in V_h$ such that $A(U, v) = (f, v)$ for all $v \in V_h$

This says that the finite element approximation solves the "average" form of the problem for a finite number of weights v

An *A Posteriori* Analysis for a Finite Element Method

We assume that quantity of interest is the functional (u, ψ)

Definition
The generalized Green's function ϕ solves the weak adjoint problem : Find $\phi \in H_0^1(\Omega)$ such that

$$A^*(v, \phi) = (\nabla v, a\nabla\phi) - (v, \text{div}\,(b\phi)) + (v, c\phi) = (v, \psi) \text{ for all } v \in H_0^1(\Omega),$$

corresponding to the adjoint problem $L^*(D, x)\phi = \psi$

An *a posteriori* analysis for a finite element method

We now estimate the error $e = U - u$:

$$
\begin{aligned}
(e,\psi) &= (\nabla e, a\nabla\varphi) - (e, \mathrm{div}(b\varphi)) + (e, c\varphi) \\
&= (a\nabla e, \nabla\varphi) + (b\bullet\nabla e, \varphi) + (ce, \varphi) \quad \text{undo adjoint} \\
&= \cancel{(a\nabla u, \nabla\varphi) + (b\bullet\nabla u, \varphi) + (cu, \varphi)} \to (f,\varphi) \\
&\quad - (a\nabla U, \nabla\varphi) - (b\bullet\nabla U, \varphi) - (cU, \varphi) \\
&= (f,\varphi) - (a\nabla U, \nabla\varphi) - (b\bullet\nabla U, \varphi) - (cU, \varphi)
\end{aligned}
$$

Definition The weak residual of U is

$$R(U,v) = (f,v) - (a\nabla U, \nabla v) - (b\cdot\nabla U, v) - (cU, v), \quad v \in H_0^1(\Omega)$$

$R(U,v) = 0$ for $v \in V_h$ but not for general $v \in H_0^1(\Omega)$

An *A Posteriori* Analysis for a Finite Element Method

Definition

$\pi_h\phi$ denotes an approximation of ϕ in V_h

Theorem

The error in the quantity of interest computed from the finite element solution satisfies the error representation,

$$(e,\psi) = (f,\phi-\pi_h\phi) - (a\nabla U,\nabla(\phi-\pi_h\phi)) \\ - (b\cdot\nabla U,\phi-\pi_h\phi) - (cU,\phi-\pi_h\phi),$$

where ϕ is the generalized Green's function corresponding to data ψ.

An *A Posteriori* Analysis for a Finite Element Method

We use the error representation by approximating ϕ using a relatively high order finite element method

For a second order elliptic problem, good results are obtained using the space V_h^2

Definition The approximate generalized Green's function $\Phi \in V_h^2$ solves

$$A^*(v,\Phi) = (\nabla v, a\nabla\Phi) - (v, \text{div}\,(b\Phi)) + (v, c\Phi) = (v,\psi) \text{ for all } v \in V_h^2$$

The approximate error representation is

$$(e,\psi) \approx (f, \Phi - \pi_h\Phi) - (a\nabla U, \nabla(\Phi - \pi_h\Phi)) - (b\cdot\nabla U, \Phi - \pi_h\Phi) - (cU, \Phi - \pi_h\Phi)$$

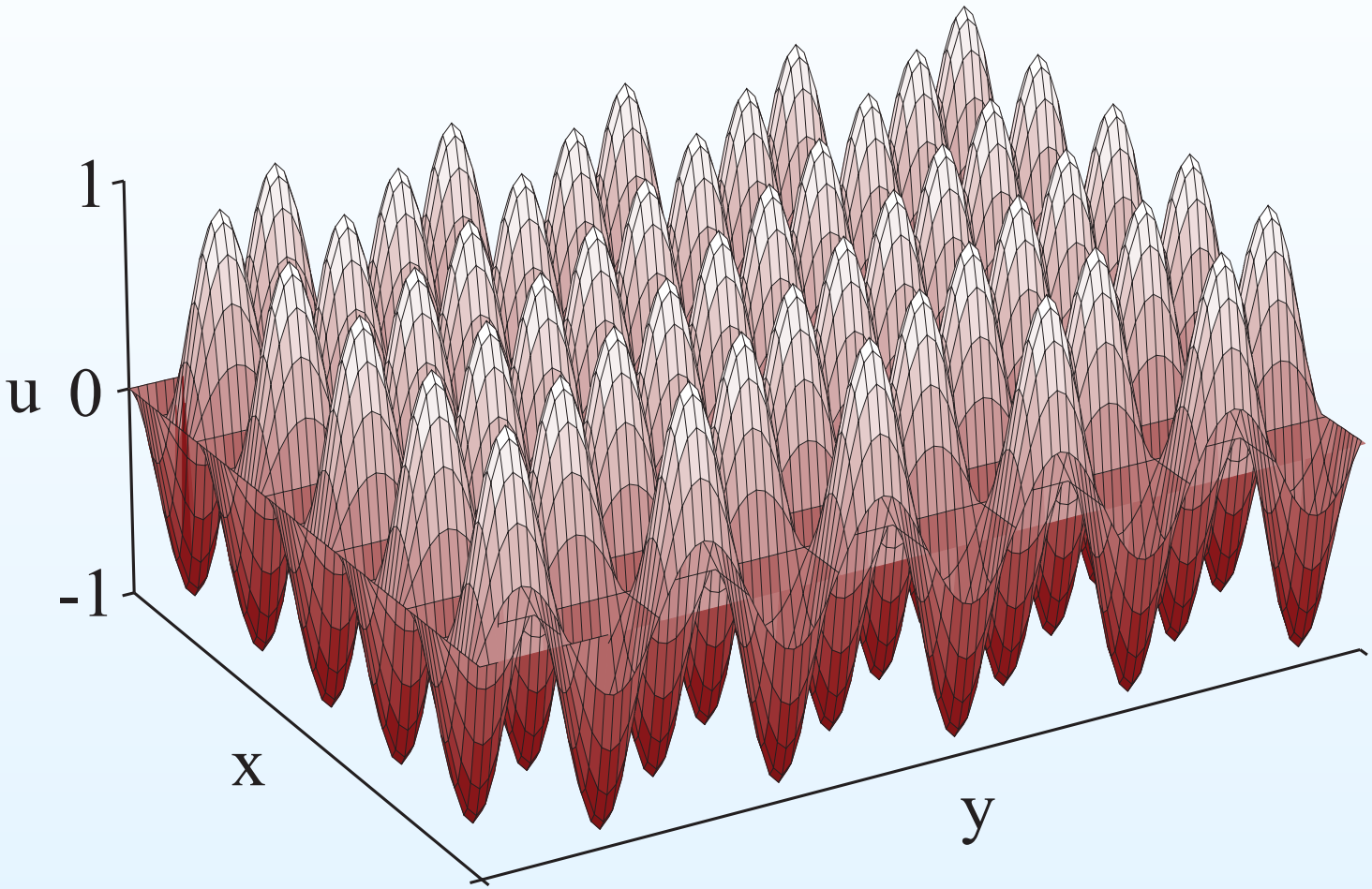
An Estimate for an Oscillatory Elliptic Problem

Example

$$\begin{cases} -\Delta u = 200\sin(10\pi x)\sin(10\pi y), & (x,y) \in \Omega = [0,1]\times[0,1], \\ u = 0, & (x,y) \in \partial\Omega \end{cases}$$

The solution is $u = \sin(10\pi x)\sin(10\pi y)$

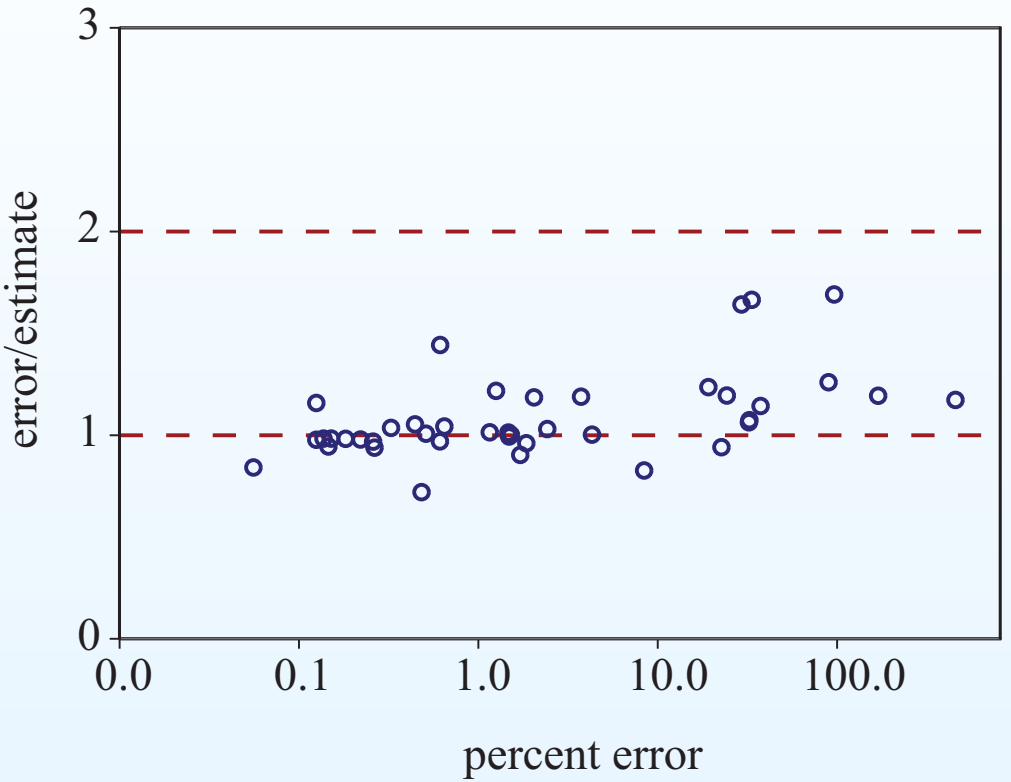
An Estimate for an Oscillatory Elliptic Problem



The Solution $u = \sin(10\pi x)\sin(10\pi y)$

The solution is highly oscillatory

An Estimate for an Oscillatory Elliptic Problem



Error/Estimate Ratios versus Accuracy

We hope for an error/estimate ratio of 1. Plotted are the ratios for finite element approximations of different error. At the 100% side, we are using 5×5 - 9×9 *meshes! Generally, we want accurate error estimates on bad meshes.*

A Posteriori Analysis for Evolution Problems

We write the numerical methods as space-time finite element methods solving a variational form of the problem

We define the weak residual as for the elliptic example above

The estimate has the form

$$\int_0^T (e,\psi)\,dt = \int_0^T (space\ residual, space\ adjoint\ weight)\,dt + \int_0^T (time\ residual, time\ adjoint\ weight)\,dt$$

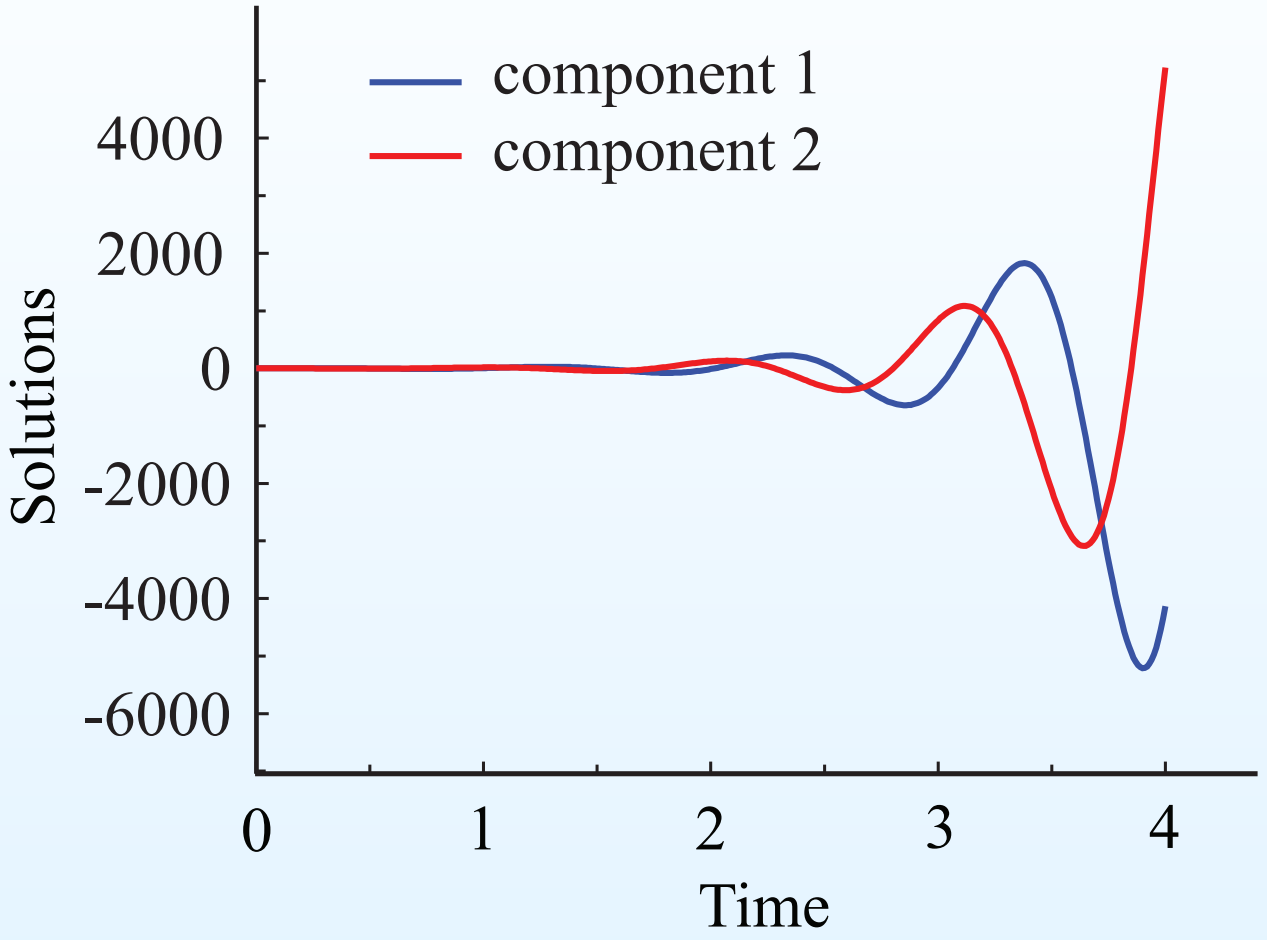
An Estimate for Vinograd's Problem

Example

$$\begin{cases} \dot{u} = \begin{pmatrix} 1 + 9\cos^2(6t) - 6\sin(12t) & -12\cos^2(6t) - 4.5\sin(12t) \\ 12\sin^2(6t) - 4.5\sin(12t) & 1 + 9\sin^2(6t) + 6\sin(12t) \end{pmatrix} u, \\ u(0) = u_0 \end{cases}$$

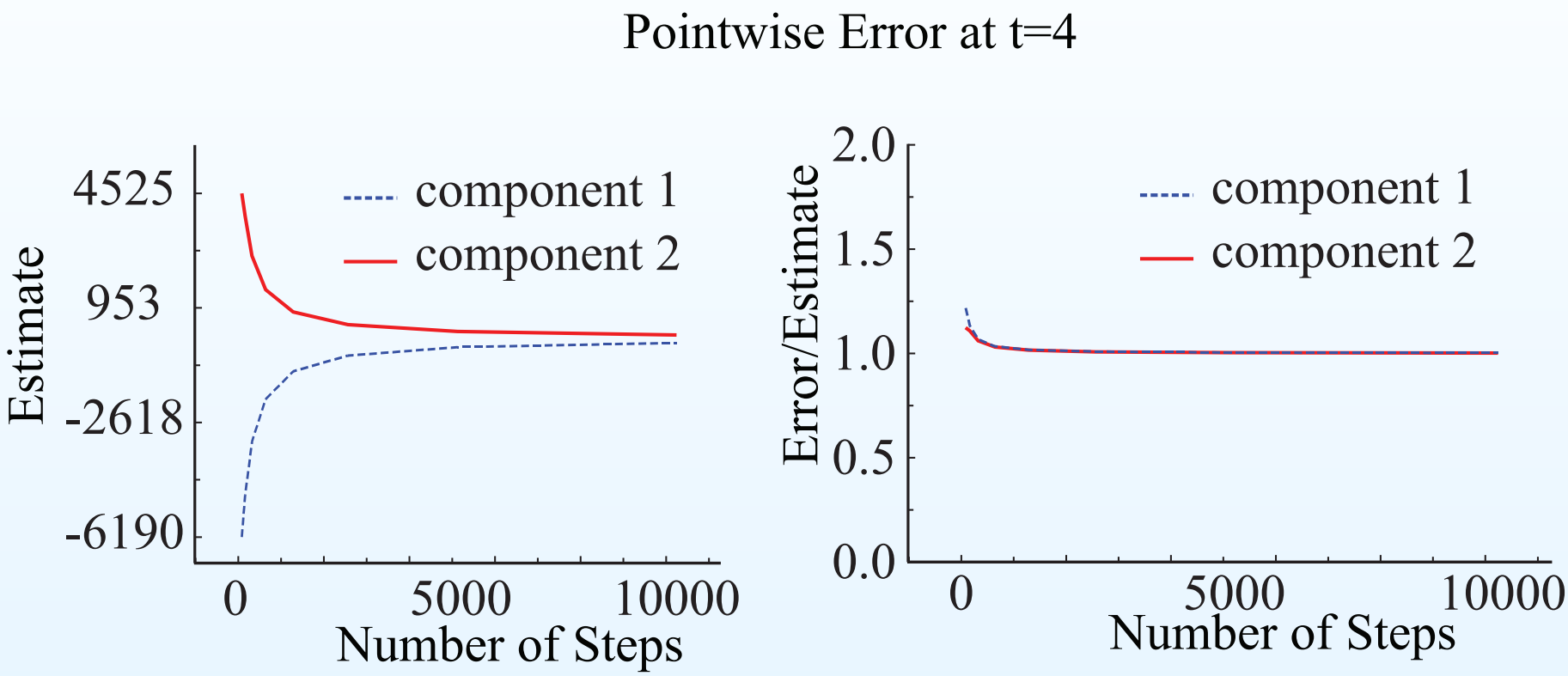
The solution is known

An Estimate for Vinograd's Problem



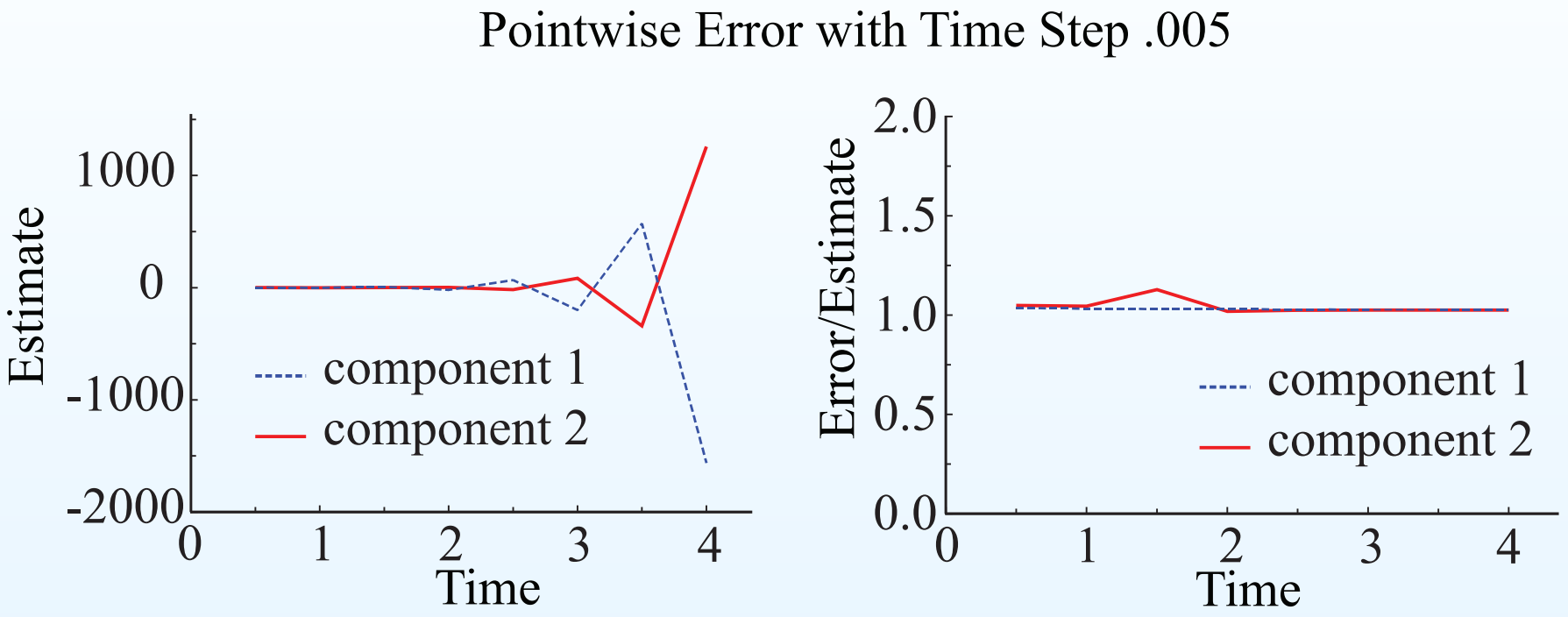
The Solution of Vinograd's Problem

An Estimate for Vinograd's Problem



Results for Decreasing Step Size

An Estimate for Vinograd's Problem



Results for Increasing Time

A Posteriori Analysis for Nonlinear Problems

Recall that we linearize the equation for the error operator to define an adjoint operator

Nominally, we need to know the true solution and the approximation for the linearization

What is the effect of linearizing around the wrong trajectory?

This is a subtle issue of structural stability: do nearby solutions have similar stability properties?

This depends on the information being computed and the properties of the problem

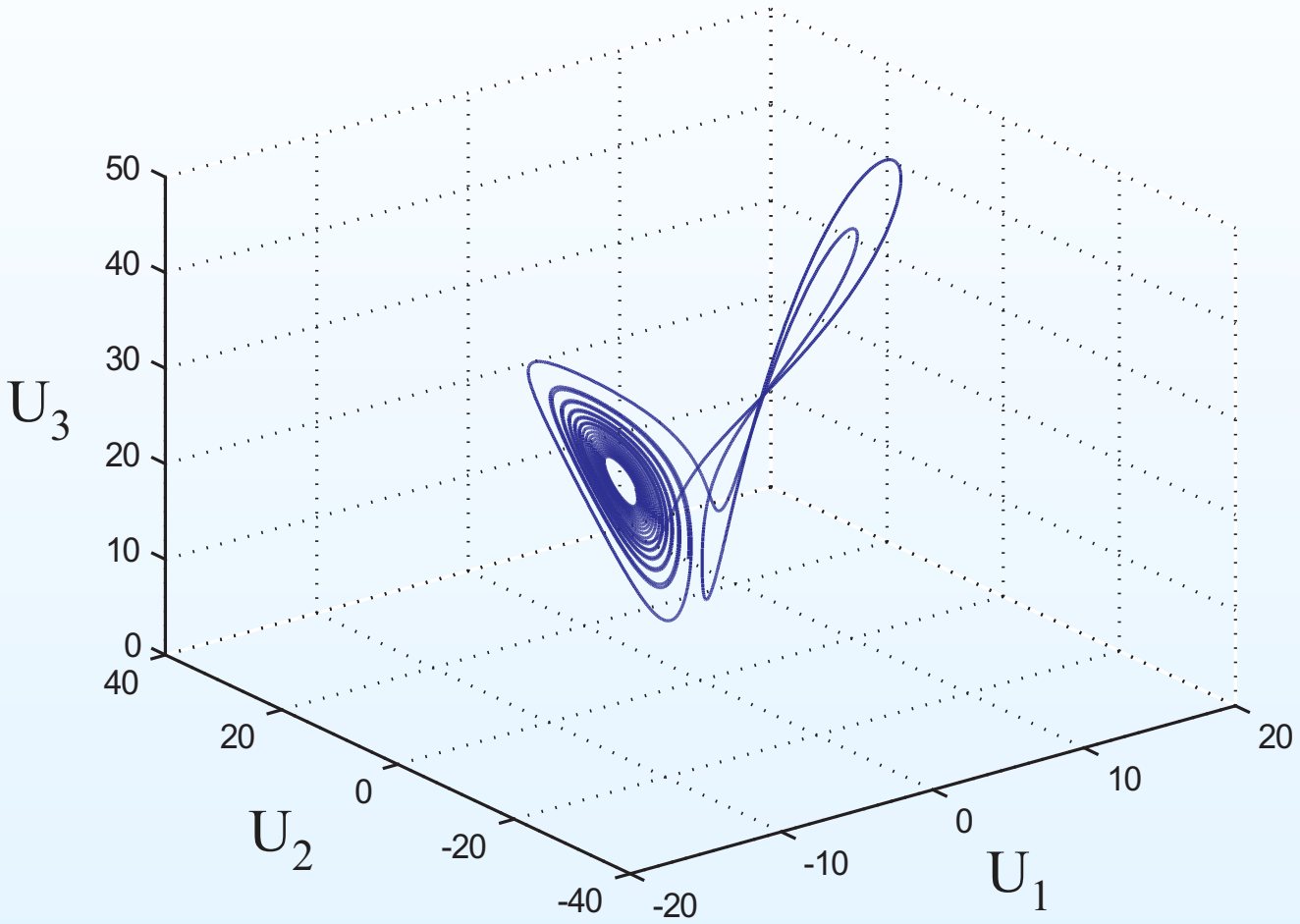
We commonly expect this to be true: if it does not hold, there are serious problems in defining approximations!

Estimates for the Lorenz Problem

Example We consider the chaotic Lorenz problem

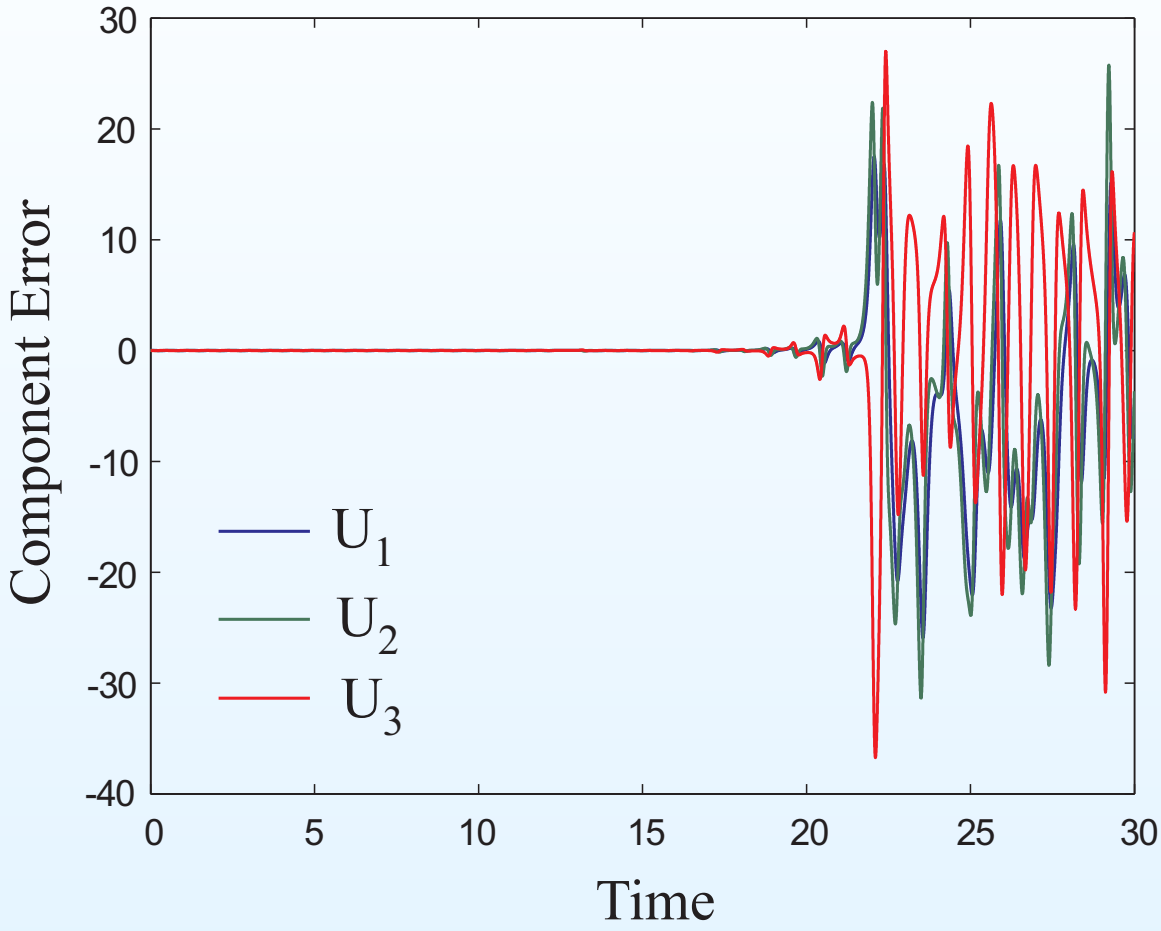
$$\begin{cases} \dot{u}_1 = -10u_1 + 10u_2, \\ \dot{u}_2 = 28u_1 - u_2 - u_1u_3, & 0 < t, \\ \dot{u}_3 = -\frac{8}{3}u_3 + u_1u_2, \\ u_1(0) = -6.9742, u_2(0) = -7.008, u_3(0) = 25.1377 \end{cases}$$

Estimates for the Lorenz Problem



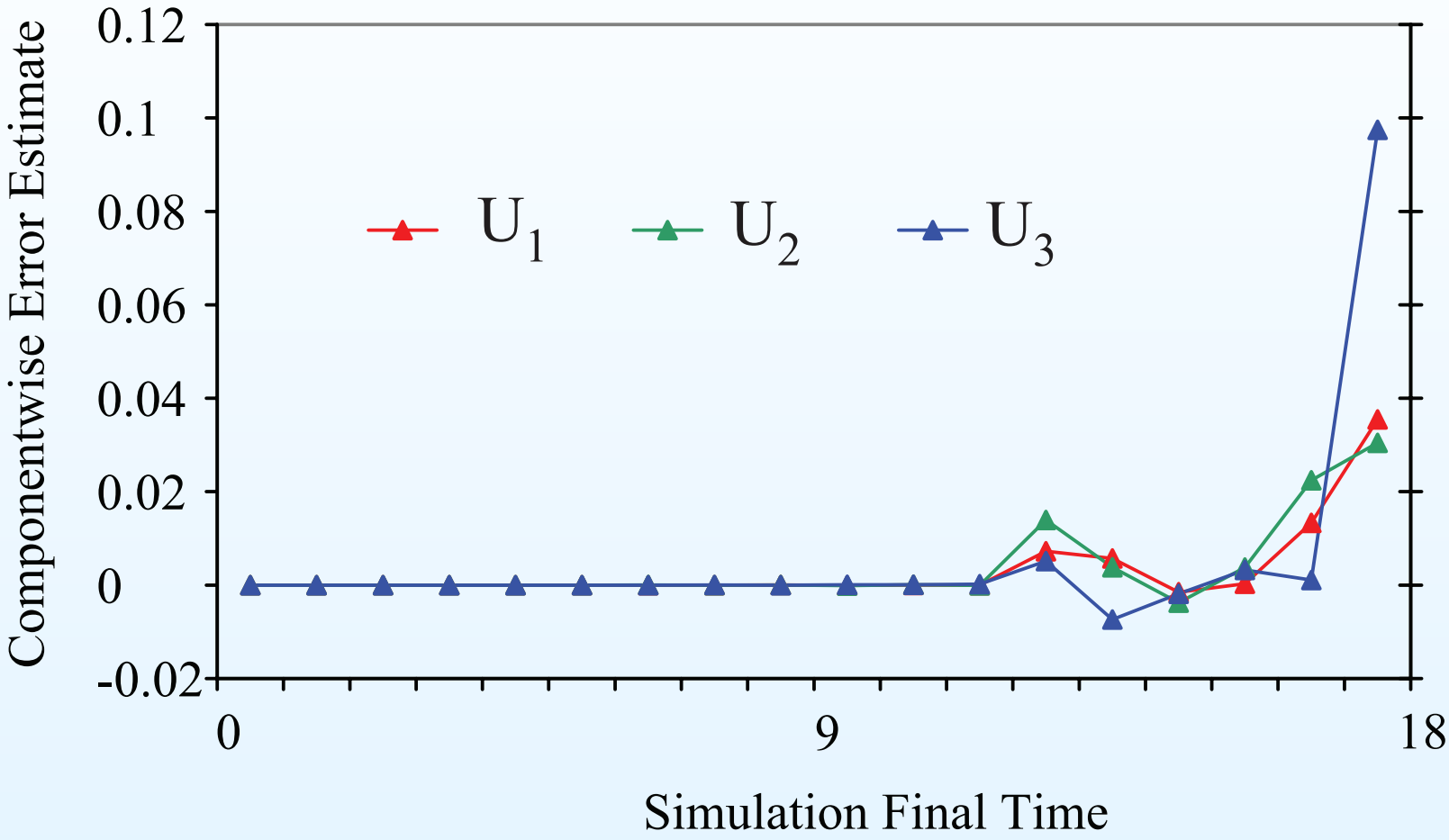
The Numerical Solution for Tolerance .001

Estimates for the Lorenz Problem



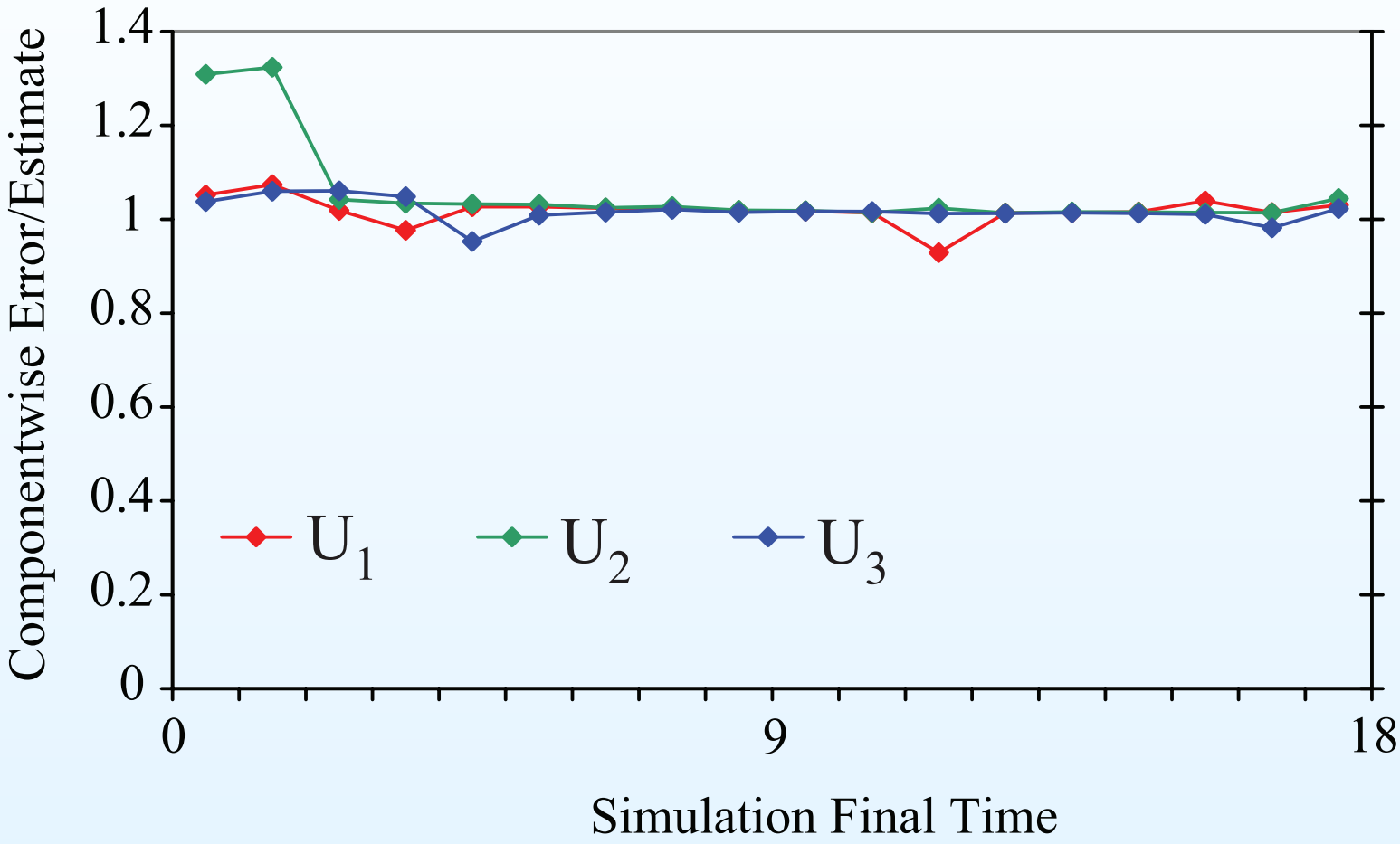
Componentwise “Errors” Computed using Solutions with Tolerances .001 and .0001

Estimates for the Lorenz Problem



Componentwise Point Error Estimates

Estimates for the Lorenz Problem



Componentwise Error/Estimate Ratios

General Comments on *A Posteriori* Analysis

In general, deriving a useful *a posteriori* error estimate is a four step process

1. identify or approximate functionals that yield the quantities of interest and write down an appropriate adjoint problem
2. understand the sources of error
3. derive computable residuals (or approximations) to measure those sources
4. derive an error representation using a suitable adjoint weights for each residual

General Comments on *A Posteriori* Analysis

Typical sources of error include

- space and time discretization (approximation of the solution space)
- use of quadrature to compute integrals in a variational formulation (approximation of the differential operator)
- solution error in solving any linear and nonlinear systems of equations
- model error
- data and parameter error
- operator decomposition

Different sources of error typically accumulate and propagate at different rates

Investigating Stability Properties of Solutions

The Adjoint and Stability

The solution of the adjoint problem scales local perturbations to global effects on a solution

The adjoint problem carries stability information about the quantity of interest computed from the solution

We can use the adjoint problem to investigate stability

Condition Numbers and Stability Factors

The classic error bound for an approximate solution U of $\mathbf{A}u = b$ is

$$\|e\| \leq C\kappa(\mathbf{A})\|R\|, \quad R = \mathbf{A}U - b$$

The condition number $\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$ measures stability

$$\kappa(\mathbf{A}) = \frac{1}{distance\ from\ \mathbf{A}\ to\ \{singular\ matrices\}}$$

The *a posteriori* estimate $|(e,\psi)| = |(R,\phi)|$ yields

$$|(e,\psi)| \leq \|\phi\|\,\|R\|$$

The stability factor $\|\phi\|$ is a weak condition number for the quantity of interest

We can obtain κ from $\|\phi\|$ by taking the $\sup$ over all $\|\psi\| = 1$

Condition Numbers and Stability Factors

Example We consider the problem of computing (u, e_1) from the solution of

$$\mathbf{A}u = b$$

where $\mathbf{A}$ is a random 800×800 matrix

The condition number of $\mathbf{A}$ is 6.7×10^4

estimate of the error in the quantity of interest $\approx 1.0 \times 10^{-15}$

a posteriori error bound for the quantity of interest $\approx 5.4 \times 10^{-14}$

The traditional error bound for the error $\approx 3.5 \times 10^{-5}$

The Condition of the Lorenz Problem

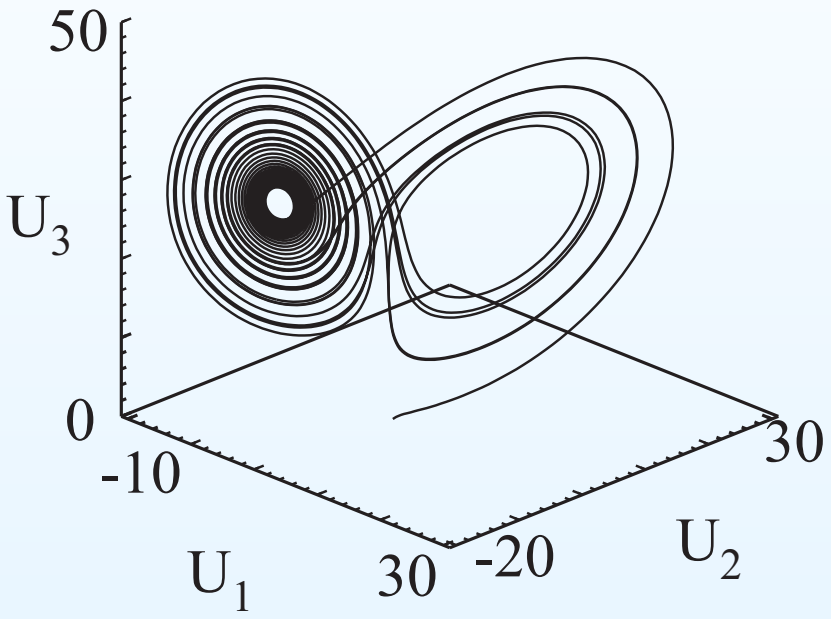
Example We consider the chaotic Lorenz problem

$$\begin{cases} \dot{u}_1 = -10u_1 + 10u_2, \\ \dot{u}_2 = 28u_1 - u_2 - u_1u_3, & 0 < t, \\ \dot{u}_3 = -\frac{8}{3}u_3 + u_1u_2, \\ u_1(0) = 1, u_2(0) = 0, u_3(0) = 0 \end{cases}$$

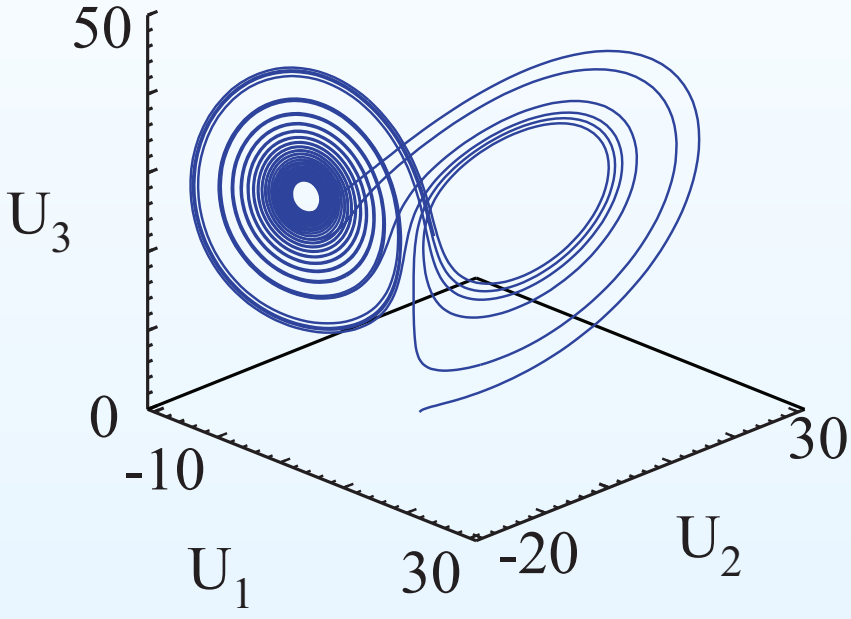
Numerical solutions always become inaccurate pointwise after some time

The Condition of the Lorenz Problem

2% error on [0,30]

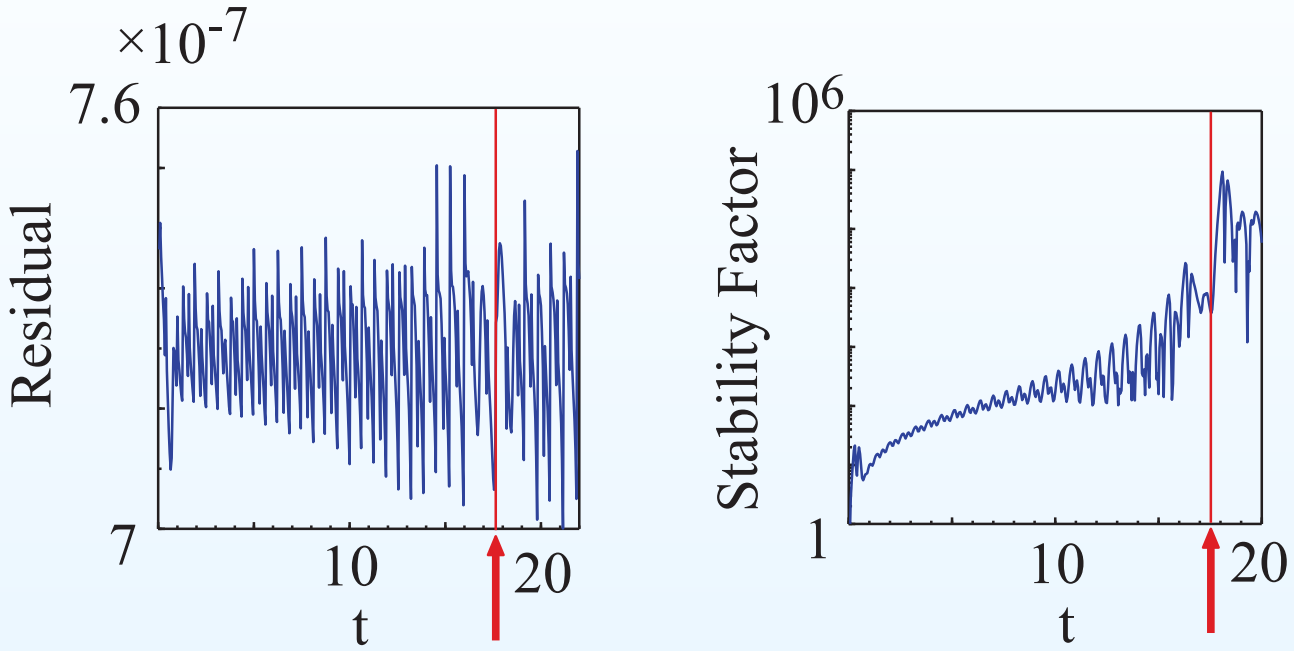


100% error at t=18



Accurate and Inaccurate Numerical Solutions

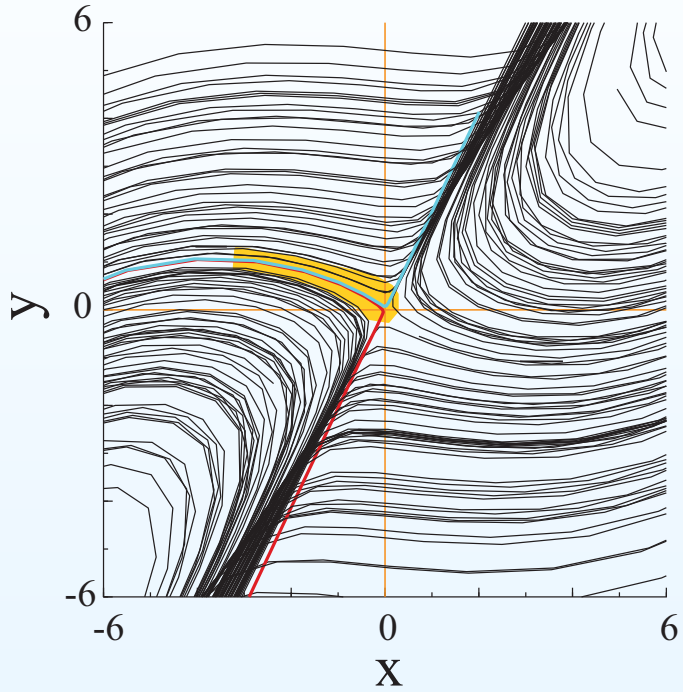
The Condition of the Lorenz Problem



The Residual and Stability Factor for the Inaccurate Solution

The residual is small even when the error is large. In fact, this is a theorem: the residual of a consistent discretization for a wide class of problems is small regardless of the size of the error! This indicates the problems in trying to use the residual or "local error" for adaptive error control. On the other hand, the size of the adjoint grows at an exponential rate during a brief period at the time when the error becomes 100%.

The Condition of the Lorenz Problem



Looking Down at Many Solutions

We look straight down at many solutions. Solutions in the lower left are circulating around one steady state, while solutions show in the upper right are circulating around another steady state. Solutions going towards both steady states are close together in the yellow region. The adjoint solution grows rapidly when a trajectory passes through there.

Adaptive Computation

Adaptive Computation

The possibility of accurate error estimation suggests the possibility of optimizing discretizations

Unfortunately, cancellation of errors significantly complicates the optimization problem

In fact, there is no good theory for adaptive control of error

There is good theory for adaptive control of error bounds

The standard approach is based on optimal control theory

The stability information in adjoint-based *a posteriori* error estimates is useful for this

Optimization Approach to Adaptivity

An abstract *a posteriori* error estimate has the form

$$|(e,\psi)| = \left|\left(Residual, Adjoint\ Weight\right)\right|$$

Given a tolerance TOL, a given discretization is refined if

$$\left|\left(Residual, Adjoint\ Weight\right)\right| \geq \text{TOL}$$

Refinement decisions are based on a bound consisting of a sum of element contributions

$$|(e,\psi)| \leq \sum_{elements\ K} \left|\left(Residual, Adjoint\ Weight\right)_K\right|$$

where $(\ ,\ )_K$ is the inner product on K

The element contributions in the bound do not cancel

Optimization Approach to Adaptivity

There is no cancellation of errors across elements in the bound, so optimization theory yields

Principle of Equidistribution: The optimal discretization is one in which the element contributions are equal

The adaptive strategy is to refine some of the elements with the largest element contributions

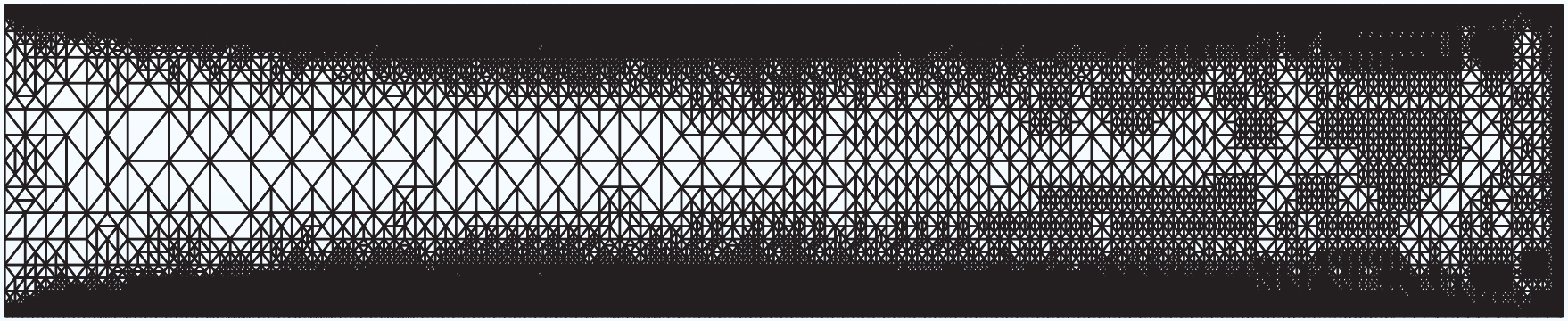
The adjoint weighted residual approach is different than traditional approaches because the element residuals are scaled by an adjoint weight, which measures how much error in that element affects the solution on other elements

A Singularly Perturbed Convection Problem

Example

$$\begin{cases} -\nabla \cdot \left((.05 + \tanh(10(x-5)^2 + 10(y-1)^2))\nabla u \right) \\ \qquad\qquad + \begin{pmatrix} -100 \\ 0 \end{pmatrix} \cdot \nabla u = 1, \quad (x,y) \in \Omega = [0,10] \times [0,2], \\ u = 0, \quad (x,y) \in \partial\Omega \end{cases}$$

A Singularly Perturbed Convection Problem

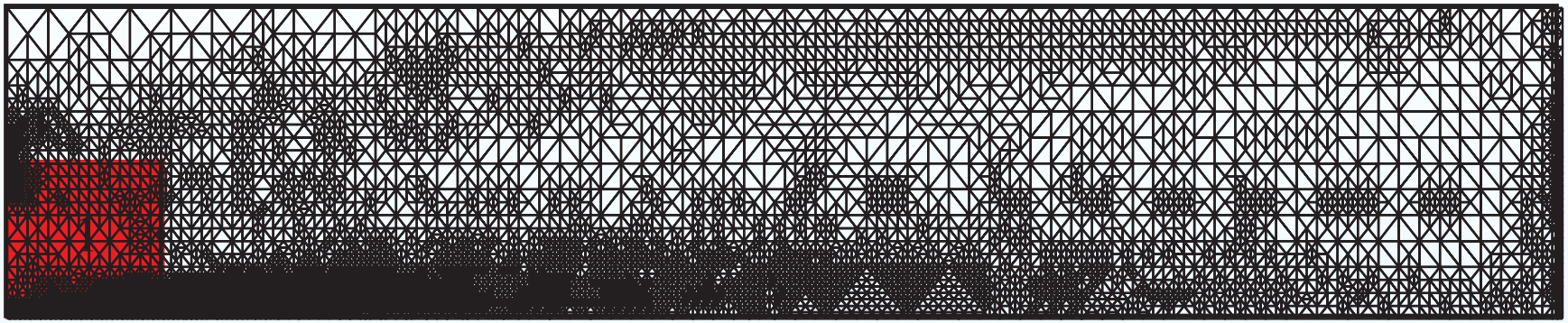


Final Mesh for an Error of 4% in the Average Value (24,000 Elements)

A Singularly Perturbed Convection Problem

Quantity of Interest is Average Error in a Patch

A Singularly Perturbed Convection Problem

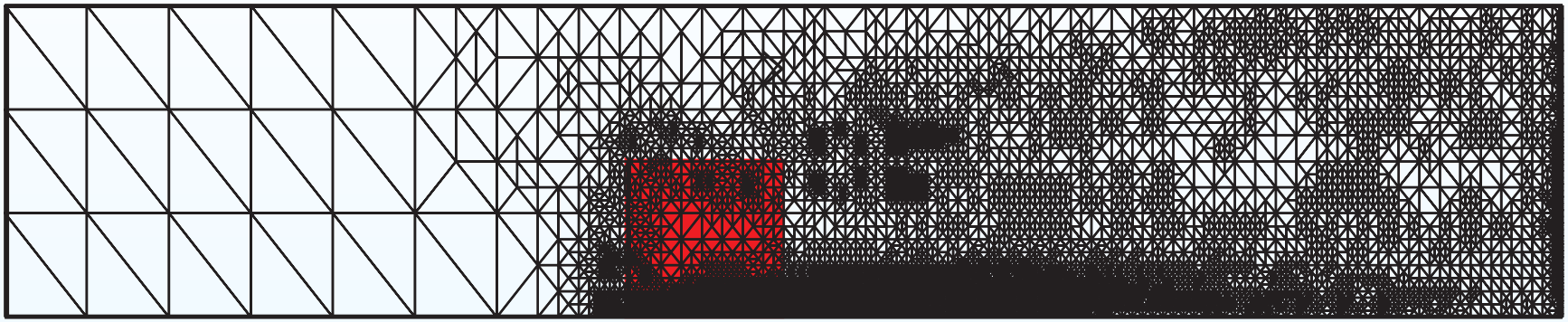


Final Mesh for an Average Error in a Patch (7,300 Elements)

A Singularly Perturbed Convection Problem

Quantity of Interest is Average Error in a Patch

A Singularly Perturbed Convection Problem



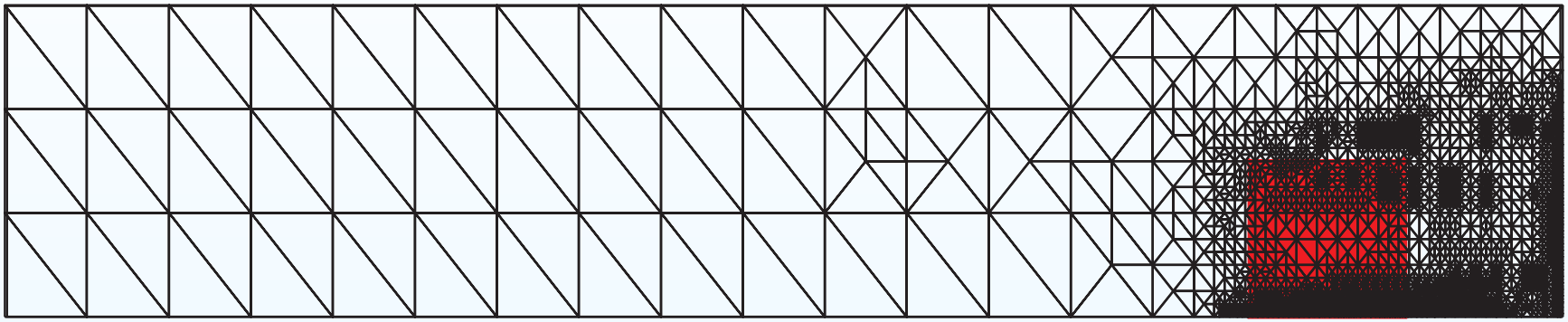
Final Mesh for an Average Error in a Patch (7,300 Elements)

The residuals of the approximation are large in the coarsely discretized region to the right - but the adjoint weights are very small, so this region does not contribute significantly to the error.

A Singularly Perturbed Convection Problem

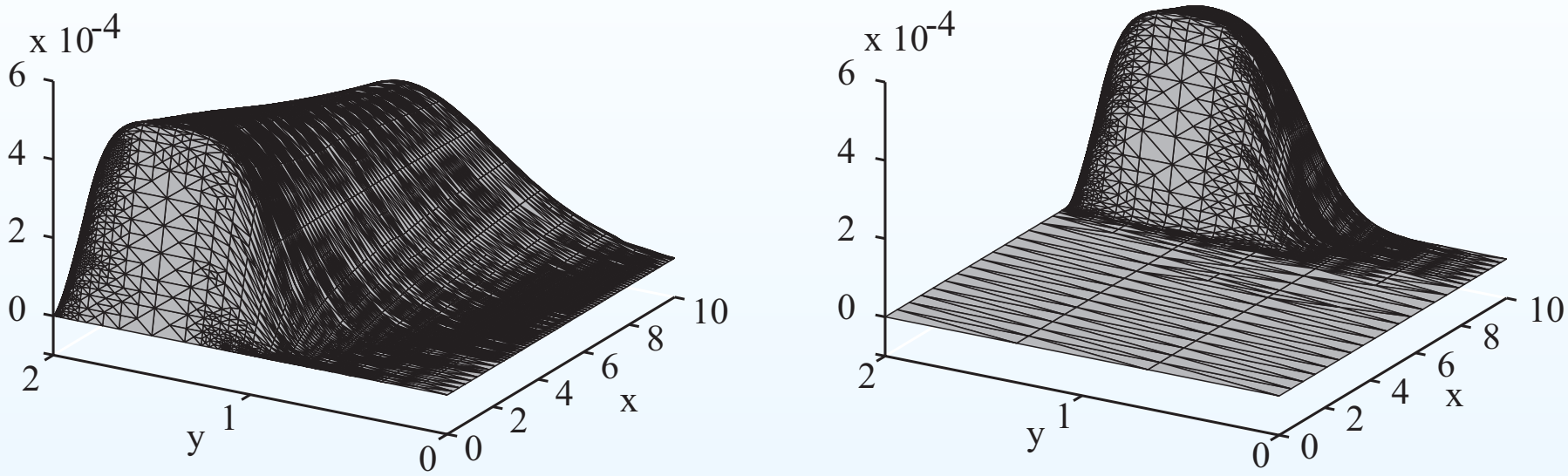
Quantity of Interest is Average Error in a Patch

A Singularly Perturbed Convection Problem



Final Mesh for an Average Error in a Patch (3,500 Elements)

A Singularly Perturbed Convection Problem



Adjoint Solutions for the First and Last Patches

Probability Approach to Adaptivity

We use a new approach to adaptivity that is probabilistic in nature

To mark elements for refinement, we first decompose

$$E = (e, \psi) \approx \sum_{elements} \text{elt. contrib.}$$

$$= \sum_{positive\ contrib.'s} \text{elt. contrib.} + \sum_{negative\ contrib.'s} \text{elt. contrib.}$$

$$= E^+ - E^-$$

We apportion the number of elements N to be refined between the positive and negative contributions as

$$N^+ = N\frac{E^+}{E^+ + E^-}, \quad N^- = N\frac{E^-}{E^+ + E^-}$$

Probability Approach to Adaptivity

The goal is to balance the positive element contributions so they cancel to reach the tolerance

To select elements for refinement, we create a probability density function using the absolute element contributions and the current steps sizes and then select randomly according to this distribution

We may also sample so as to reduce the variance of the element contributions

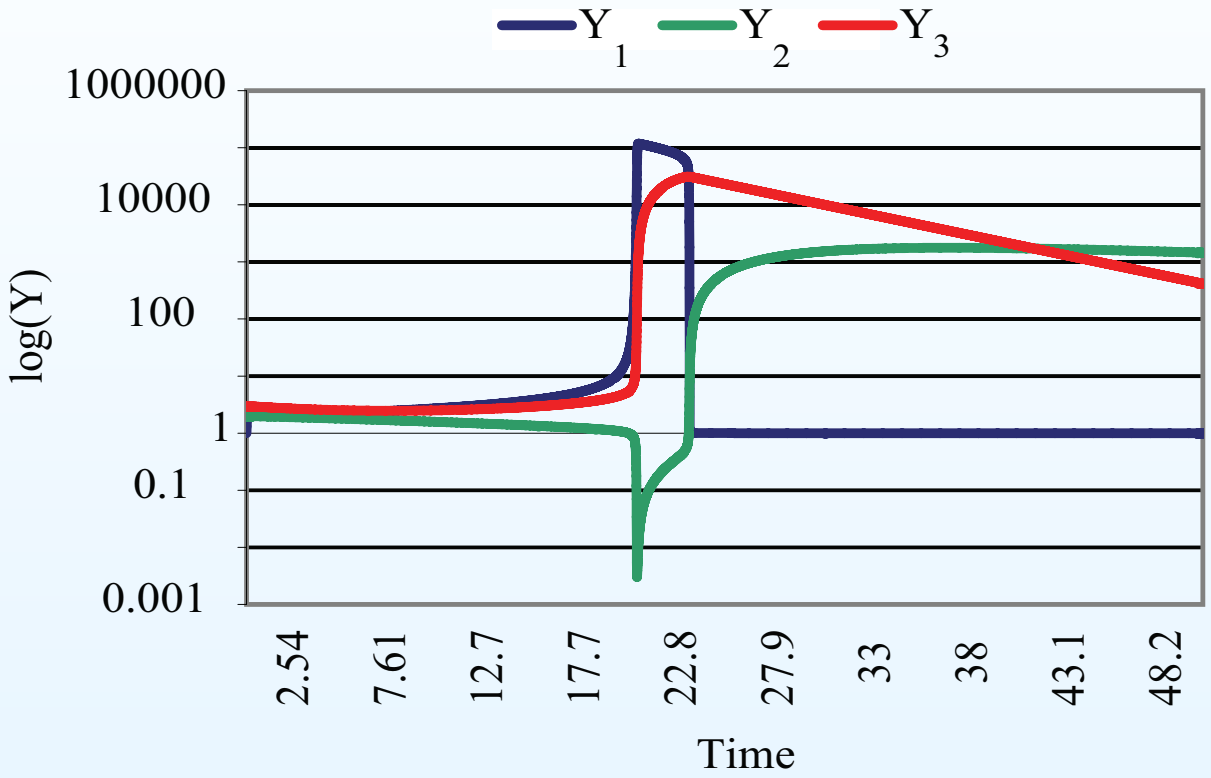
Probabilistic Adaptivity for the Oregonator

Example We consider the Oregonator problem

$$\begin{cases} \dot{y}_1 = 2(y_2 - y_1 y_2 + y_1 - q y_1^2) & y_1(0) = 1 \\ \dot{y}_2 = \frac{1}{s}(-y_2 - y_1 y_2 + y_3), & y_2(0) = 0, \\ \dot{y}_3 = w(y_1 - y_3), & y_3(0) = 0, \\ s = 77.27,\ w = .161,\ q = 8.375 \times 10^{-6} \end{cases}$$

We compute with $TOL = 10^{-8}$ over time $T = 50$

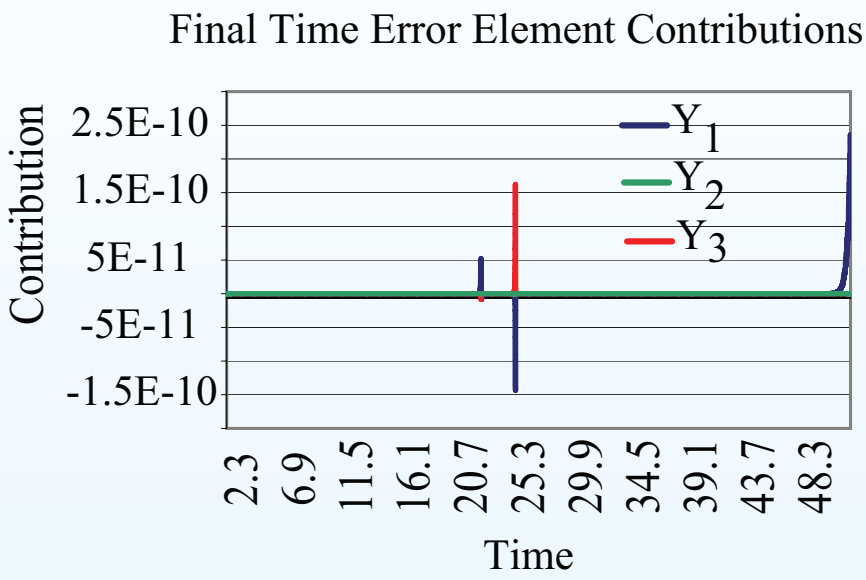
Probabilistic Adaptivity for the Oregonator

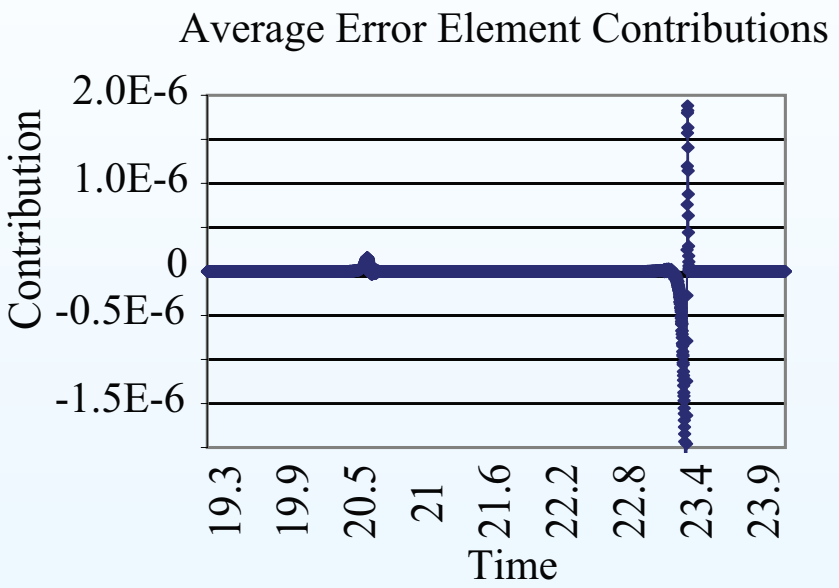


Solution of the Oregonator problem

This problem is difficult because the solution has long periods of time on which little happens punctuated by very rapid transients where the solution changes dramatically. We plot the components in the region around one of the transient periods.

Probabilistic Adaptivity for the Oregonator





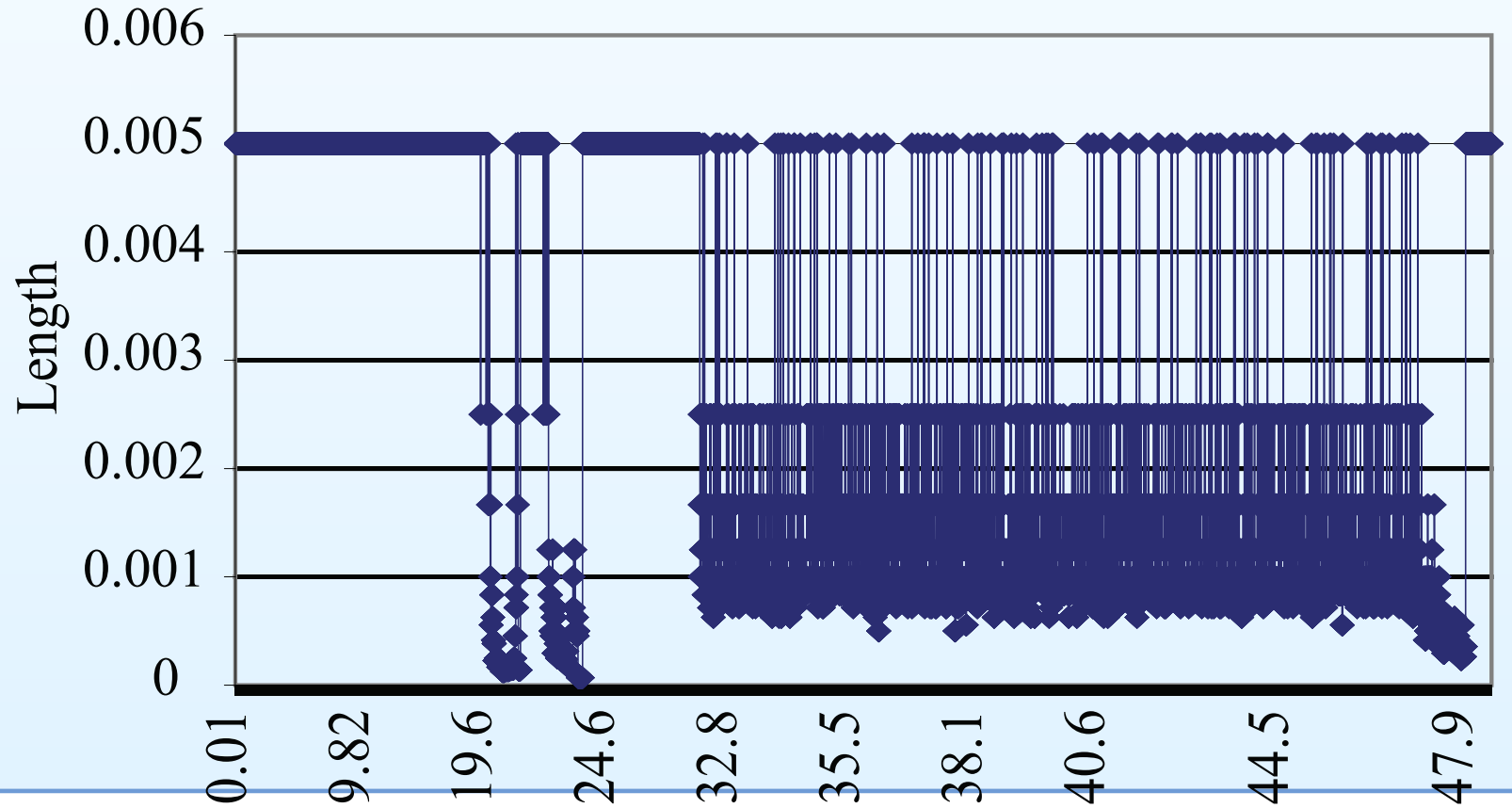
Element contributions to the error: final time (left) and average error (right)

We see that the element contributions are largest in the brief transient periods.

Probabilistic Adaptivity for the Oregonator

Using the classical dual-weighted optimal control approach to adaptivity requires 188,279 time steps

Using the probability approach requires 20108 time steps



Operator Decomposition for Multiscale, Multiphysics Problems

Analyzing the Effects of Operator Decomposition

In operator decomposition, the instantaneous interaction between different physics is discretized

This results in new sources of instability and error

We use duality, adjoints, and variational analysis in new ways to analyze operator decomposition

- We estimate the error in the specific information passed between components
- We account for the fact that the adjoints to the original problem and an operator decomposition discretization are not the same

Additional work is required to obtain computable estimates

Operator Decomposition for Elliptic Systems

$$\begin{cases} -\Delta u_1 = \sin(4\pi x)\sin(\pi y), & x \in \Omega \\ -\Delta u_2 = b \cdot \nabla u_1 = 0, & x \in \Omega, \\ u_1 = u_2 = 0, & x \in \partial\Omega, \end{cases} \qquad b = \frac{2}{\pi}\begin{pmatrix} 25\sin(4\pi x) \\ \sin(\pi x) \end{pmatrix}$$

The quantities of interest are

$$u_2(.25, .25) \approx \langle \delta_{\text{reg}}(.25, .25), u_2 \rangle$$

and the average value

Estimating the error requires auxiliary estimates of the error in the information that is passed between components

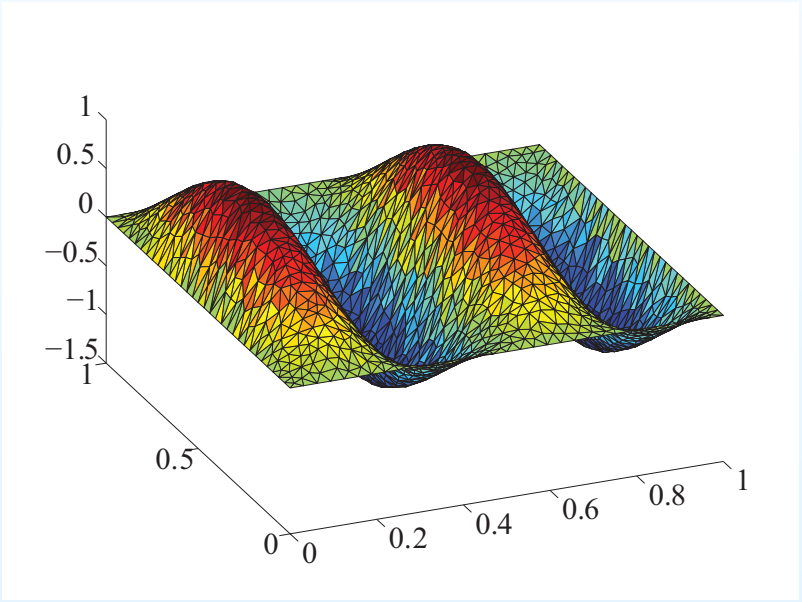
Operator Decomposition for Elliptic Systems

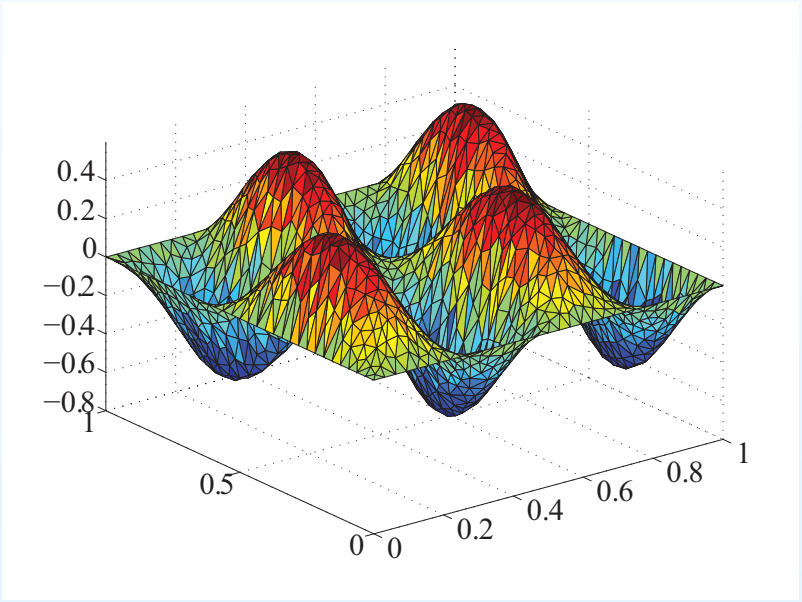
We use uniformly fine meshes for both components

For the error in $u(.25, .25)$

discretization contribution $\approx .0042$

decomposition contribution $\approx .0006$



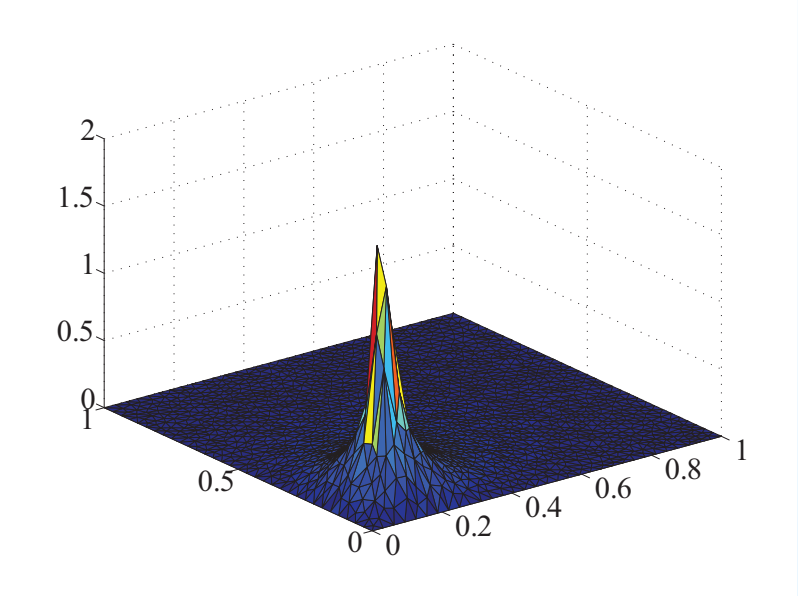


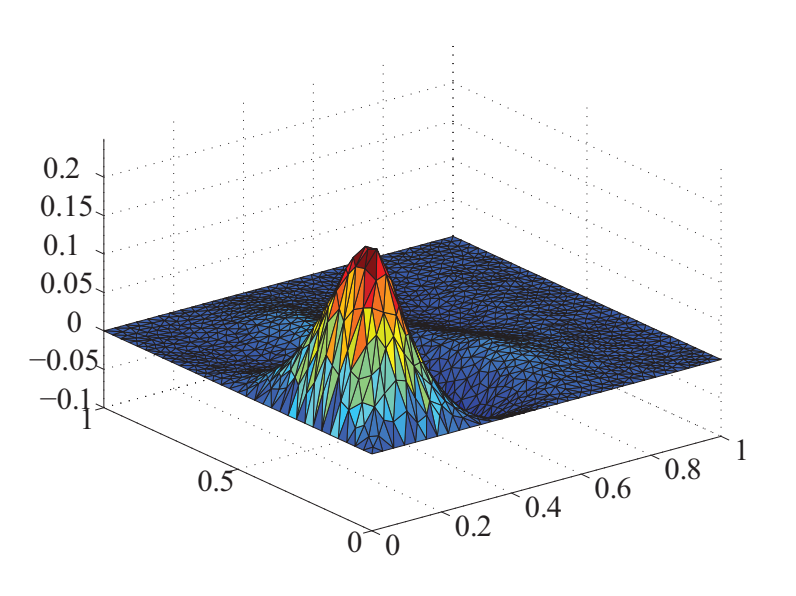
Solutions of components 1 and 2

We see that the error in the transferred information is small but not significant.

Operator Decomposition for Elliptic Systems

We use uniformly fine meshes for both components





Primary and decomposition-contribution adjoint solutions

The adjoint solution for the functional on u_2 is large due to the adjoint data (an approximate delta function). The adjoint associated with the information transferred between the components is large in the same region. This indicates that the accuracy of u_1 in this region impacts the accuracy of u_2.

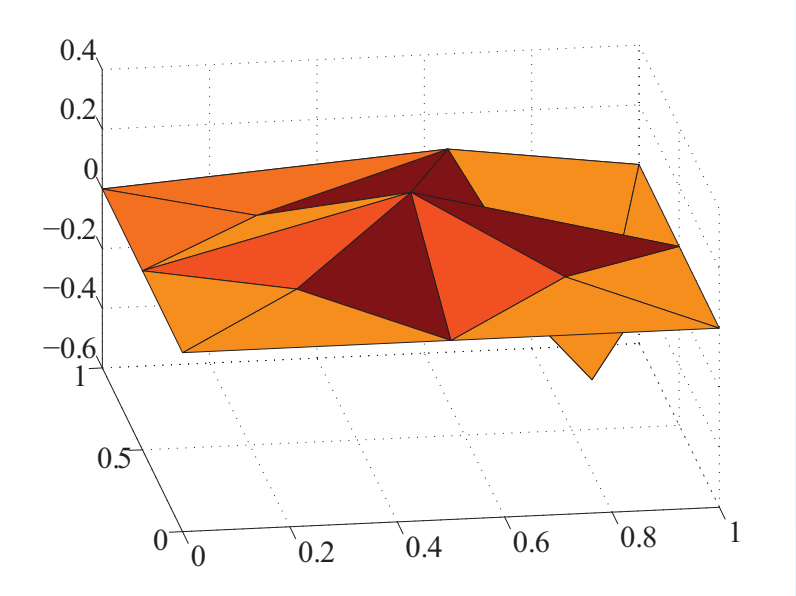
Operator Decomposition for Elliptic Systems

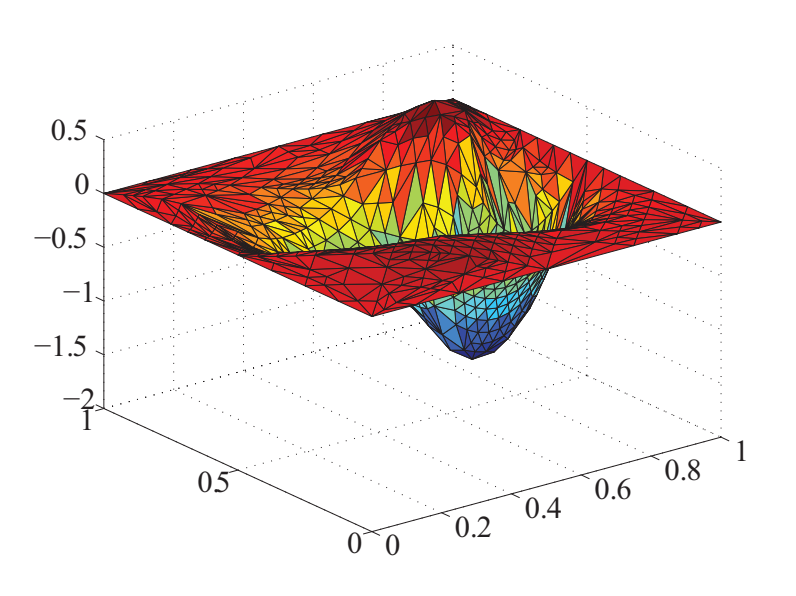
We adapt the mesh while ignoring the contributions to the error from operator decomposition

For the average error

$$\text{discretization error} \approx .0001$$

$$\text{decomposition contribution} \approx .2244$$





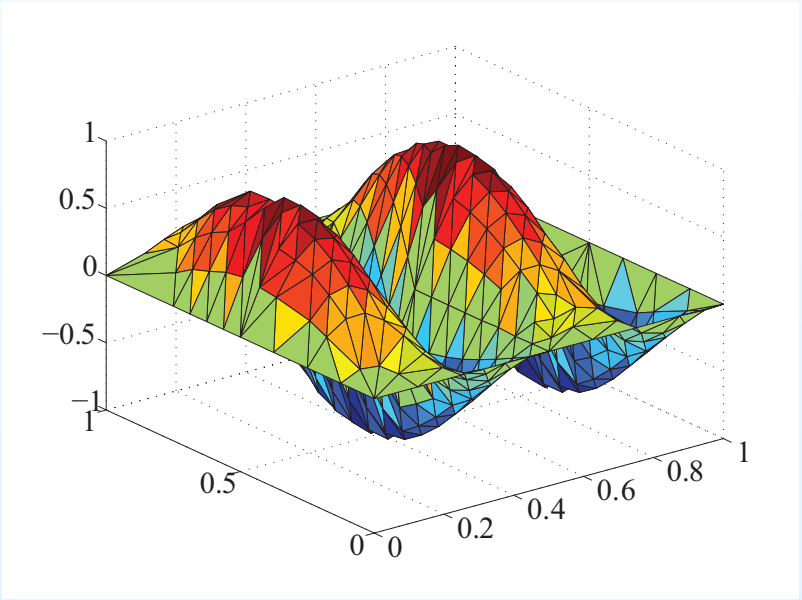
U_1 on a coarse mesh and the total error of U_2

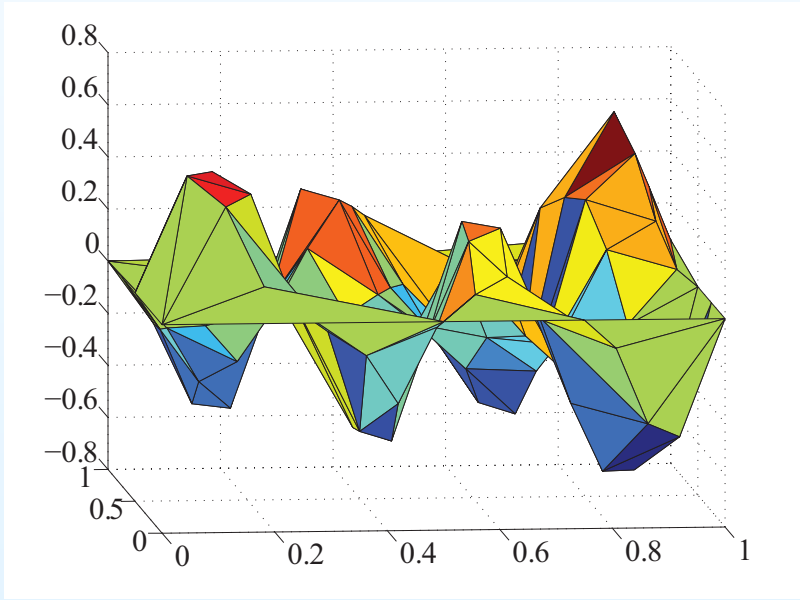
Independently adapting each component mesh makes the error worse!

Operator Decomposition for Elliptic Systems

We adapt the meshes for both components using the primary error representation for U_2 and the secondary representation for U_1 respectively

Transferring the gradient ∇U_1 leads to increased sensitivity to errors





Final refined solutions for components 1 and 2

We actually refine the mesh for u_1 more than the mesh for u_2.

Operator Splitting for Reaction-Diffusion Equations

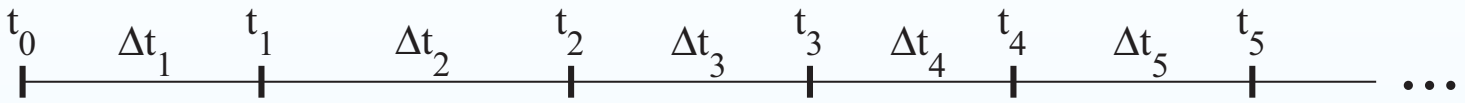
We consider the reaction-diffusion problem

$$\begin{cases} \frac{du}{dt} = \Delta u + F(u), \quad 0 < t, \\ u(0) = u_0 \end{cases}$$

The diffusion component Δu induces stability and change over long time scales

The reaction component F induces instability and change over short time scales

Operator Splitting for Reaction-Diffusion Equations



On $(t_{n-1}, t_n]$, we numerically solve

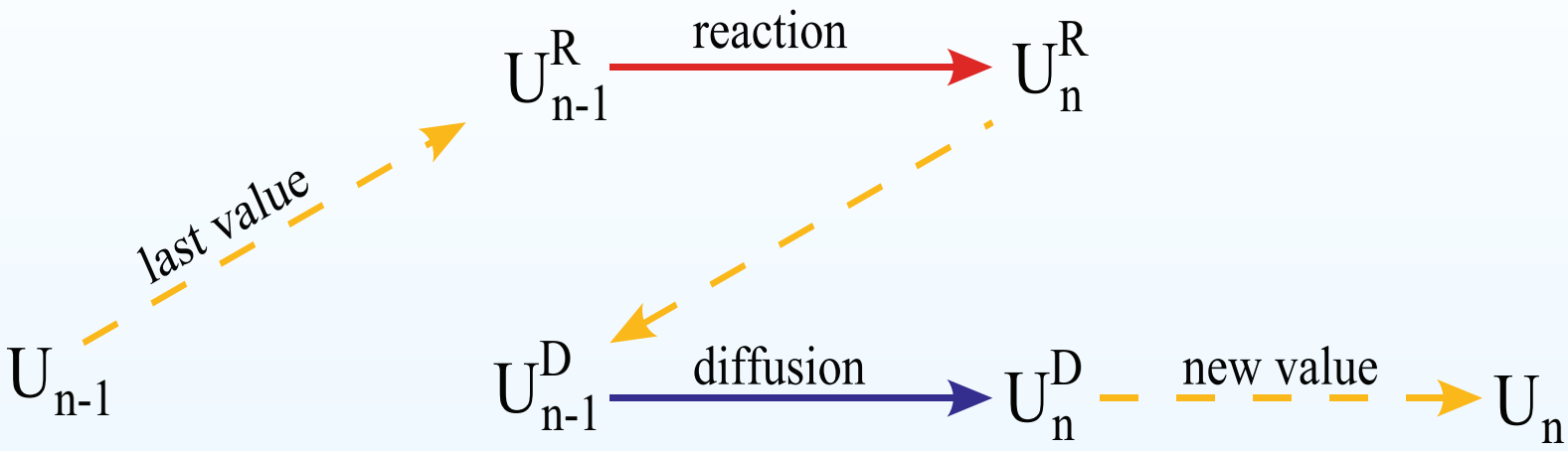
$$\begin{cases} \frac{du^R}{dt} = F(u^R), & t_{n-1} < t \leq t_n, \\ u^R(t_{n-1}) = u^D(t_{n-1}) \end{cases}$$

Then on $(t_{n-1}, t_n]$, we numerically solve

$$\begin{cases} \frac{du^D}{dt} - \Delta(u^D), & t_{n-1} < t \leq t_n, \\ u^D(t_{n-1}) = u^R(t_n) \end{cases}$$

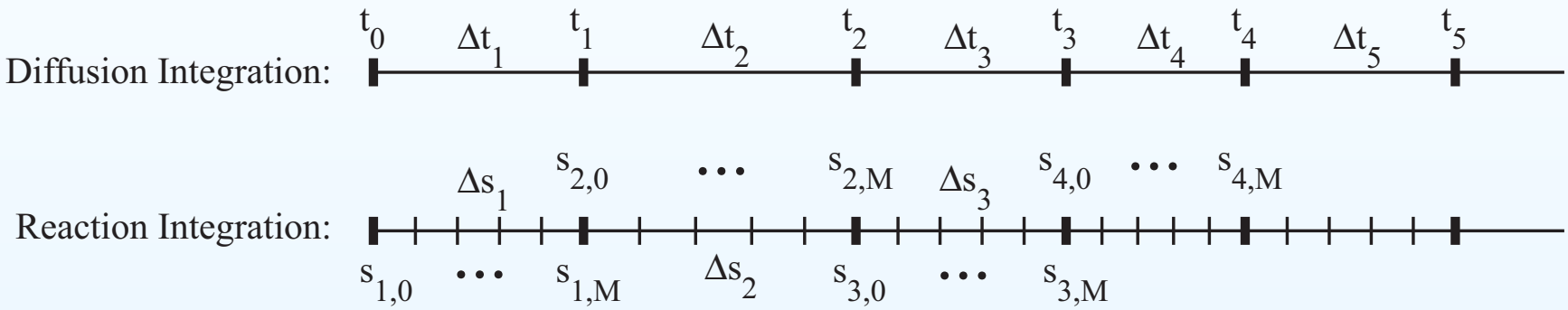
The operator split approximation is $u(t_n) \approx u^D(t_n)$

Operator Splitting for Reaction-Diffusion Equations



Operator Splitting for Reaction-Diffusion Equations

To account for the fast reaction, we approximate u^r using many time steps inside each diffusion step



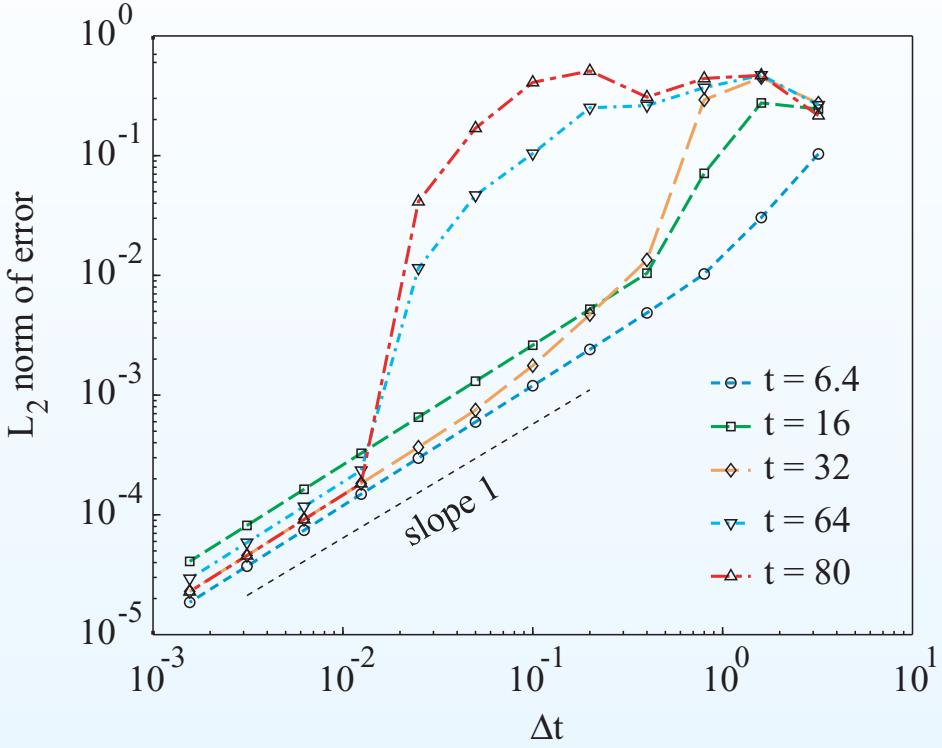
Operator Splitting for Reaction-Diffusion Equations

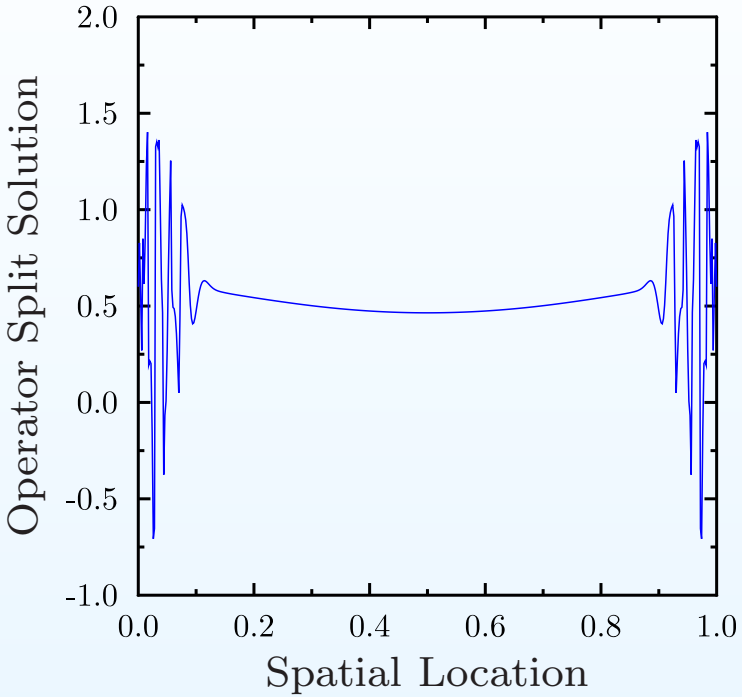
The Brusselator problem

$$\begin{cases} \frac{\partial u_i}{\partial t} - 0.025\,\frac{\partial^2 u_i}{\partial x^2} = f_i(u_1,u_2) \;\; i=1,2 \\ f_1(u_1,u_2) = 0.6 - 2u_1 + u_1^2 u_2 \\ f_2(u_1,u_2) = 2u_1 - u_1^2 u_2 \end{cases}$$

- Use a linear finite element method in space with 500 elements
- Use a standard first order splitting scheme
- Use Trapezoidal Rule with time step of $.2$ for the diffusion and Backward Euler with time step of $.004$ for the reaction

Operator Splitting for Reaction-Diffusion Equations





Instability in the Brusselator Operator Splitting

On the left we plot the error versus time step at different times. For large times, there is a critical step size above which there is no convergence. On the right, we plot one of the inaccurate solutions. The instability is a direct consequence of the discretization of the operator splitting in time

Operator Splitting for Reaction-Diffusion Equations

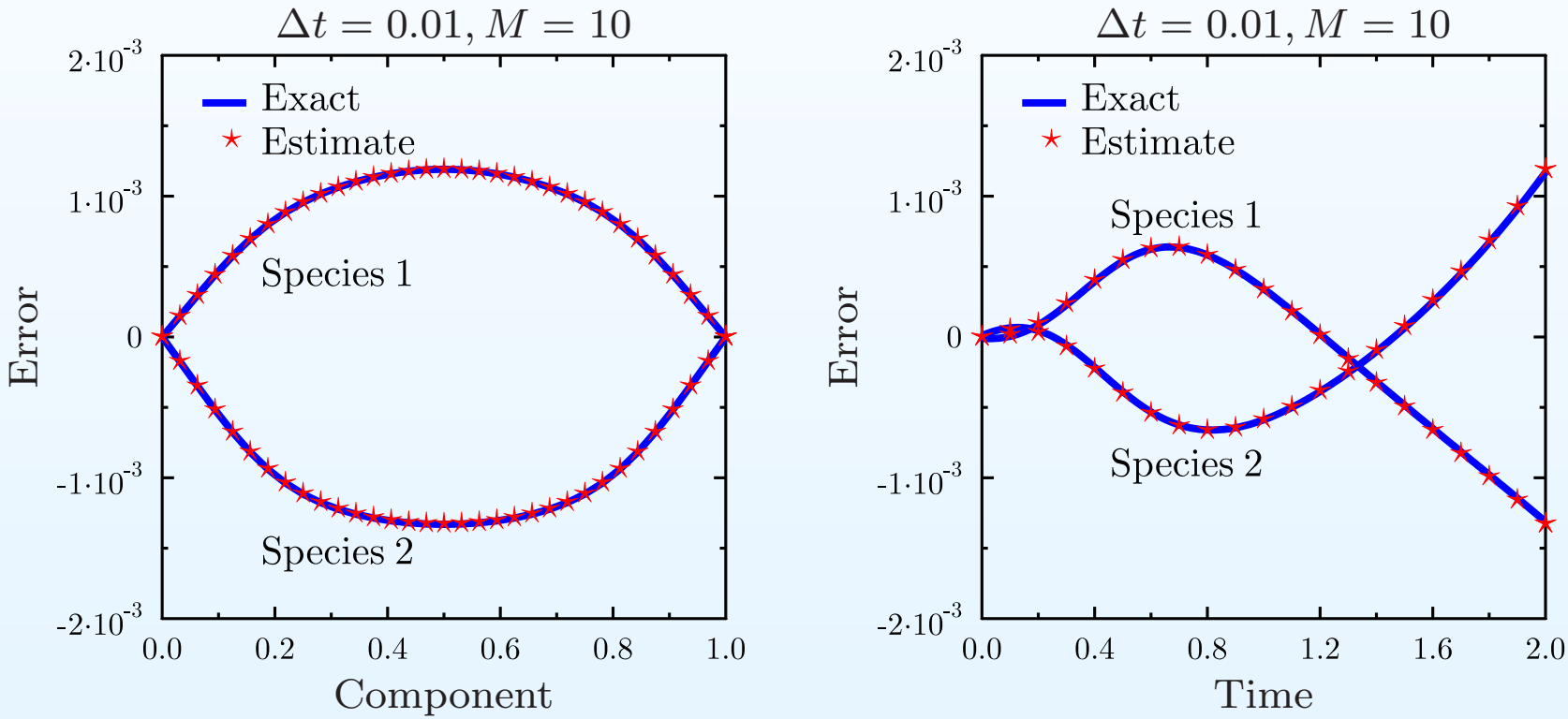
We derive a new type of hybrid *a priori* - *a posteriori* estimate

$$(e(t_N), \psi) = \mathcal{Q}_1 + \mathcal{Q}_2 + \mathcal{Q}_3$$

- $\mathcal{Q}_1$ estimates the error of the numerical solution of each component
- $\mathcal{Q}_2 \approx \sum_{n=1}^{N} (U_{n-1}, E_{n-1}), \quad E \approx$ a computable estimate for the error in the adjoint arising from operator splitting
- $\mathcal{Q}_3 = \mathbf{O}(\Delta t^2)$ is an *a priori* expression that is provably higher order

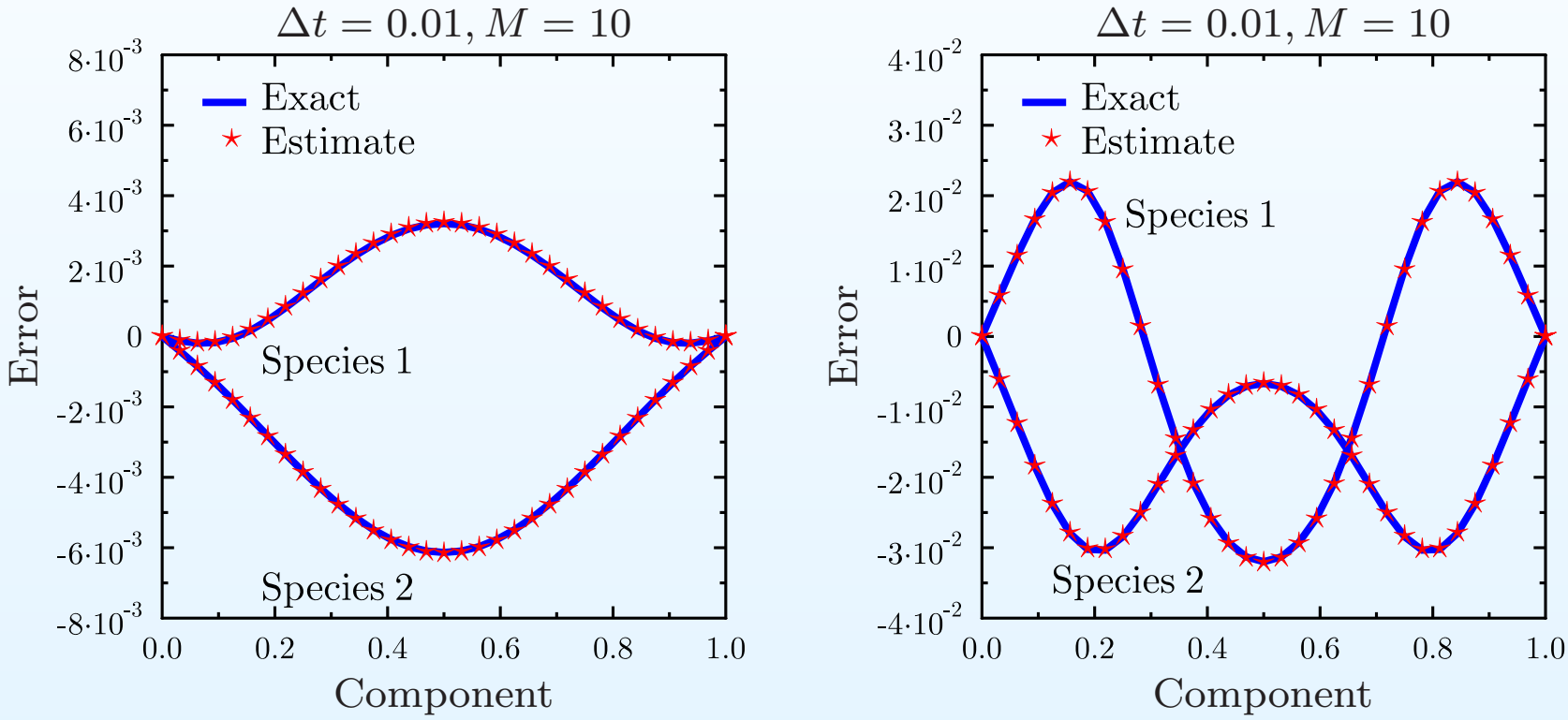
Operator Splitting for Reaction-Diffusion Equations

Accuracy of the error estimate for the Brusselator example at $T = 2$



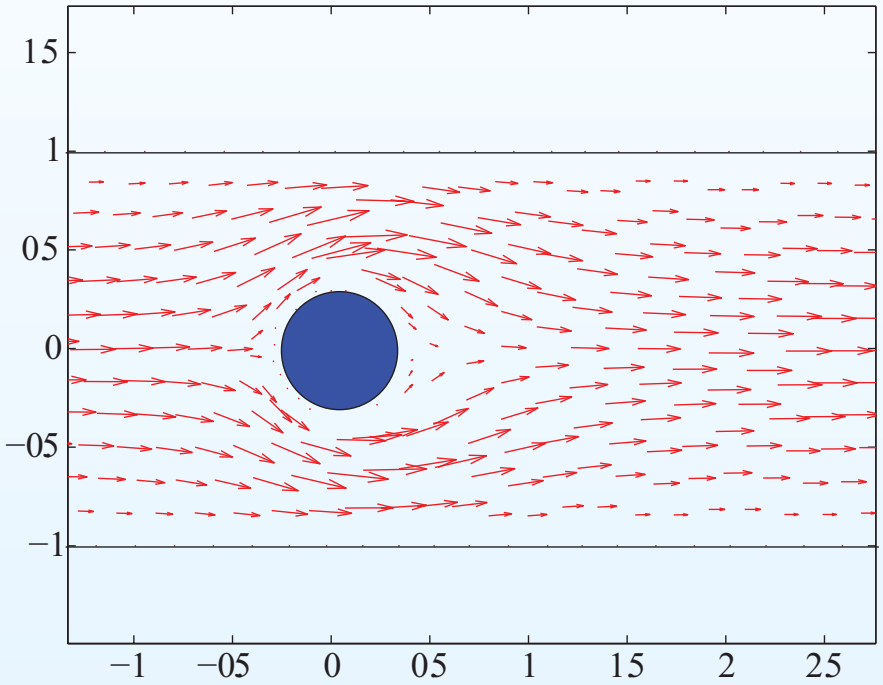
Operator Splitting for Reaction-Diffusion Equations

Accuracy of the error estimate for the Brusselator example at $T = 8$ (left) and $T = 40$ (right)



Operator Decomposition for Conjugate Heat Transfer

Example A relatively cool solid object is immersed in the flow of a hot fluid



The goal is to describe the temperature of the solid object

Operator Decomposition for Conjugate Heat Transfer

We use the Boussinesq equations for the fluid, coupled to the heat equation for the solid

We use application codes optimized for single physics

The solution of each component is sought independently using data obtained from the solution of the other component

On the boundary of the object, Neumann conditions are passed from the solid to the fluid, and Dirichlet conditions are passed from the fluid to the solid

Operator Decomposition for Conjugate Heat Transfer

Passing derivative information in the Neumann boundary condition causes a loss of order

This error can be estimated accurately using an adjoint computation

We devise an inexpensive postprocessing technique for computing the transferred information accurately

We recover the full order of convergence

Analysis of Model Sensitivity

Kernel Density Estimation

$\mathcal{F}$ is a nonlinear operator:

$$\text{Space of data and parameters} \quad \xrightarrow[\mathcal{F}]{} \quad \text{Space of outputs}$$

Assume that the data and/or the parameters are unknown within a given range and/or subject to random or unknown variation

Problem: determine the effect of the uncertainty or variation on the output of the operator

We consider the input to be a random vector associated with a probability distribution

The output of the model is random vector associated with a new distribution

The Monte-Carlo Method

The Monte-Carlo method is the standard technique for solving this problem

The model is solved for many samples drawn from the input space according to its distribution

The Monte-Carlo method has robust convergence properties and is easy to implement on scalar and parallel computers

However, the Monte-Carlo method may be expensive because it converges very slowly

A Finite-Dimensional Problem

We determine the distribution of a linear functional (x, ψ) given the random vector $\lambda \in \mathbb{R}^d$, where x satisfies

$$f(x; \lambda) = b$$

Assuming λ is distributed near a sample value μ, we solve

$$f(y; \mu) = b$$

With $A = D_x f(y; \mu)$, the adjoint problem is

$$A^T \phi = \psi,$$

Applying Taylor's theorem to the representation formula yields

$$\langle x, \psi \rangle \approx \langle y, \psi \rangle - \left\langle D_\lambda f(y; \mu)(\lambda - \mu), \phi \right\rangle$$

Using the Adjoint Problem

If $\lambda - \mu$ is a random vector then $\left\langle D_\lambda f(y;\mu)(\lambda - \mu), \phi \right\rangle$ is a new random variable

We can use this information to speed up random sampling in several ways

For example, we compute an error estimate for the constant approximation generated from the sample point and adaptively sample (Fast Adaptive Parameter Sampling (FAPS)) Method

Note that we adapt the sample according to the output distribution rather than the input distribution!

A Predator Prey Example

Example We model a prey u with a logistic birth/death process consumed by predator v

$$\begin{cases} \partial_t v - \delta \Delta v = \lambda_1 v\, h(u;\lambda_2) - \lambda_3 v, & \Omega \times (0,T], \\ \partial_t u - \delta \Delta u = \lambda_4 u(1 - \frac{u}{\lambda_5}) - \lambda_6 v\, h(u;\lambda_2), & \\ \partial_n v = \partial_n u = 0, & \partial\Omega \times (0,T], \\ v = v_0, u = u_0, & \Omega \times \{0\} \end{cases}$$

The (Holling II) functional response $h(u) = h(u;\lambda_2)$ satisfies

- $h(0) = 0$
- $\lim_{x\to\infty} h(x) = 1$
- h is strictly increasing

Predator Prey Reference Parameter Values

We assume a (truncated) normal distribution in the region

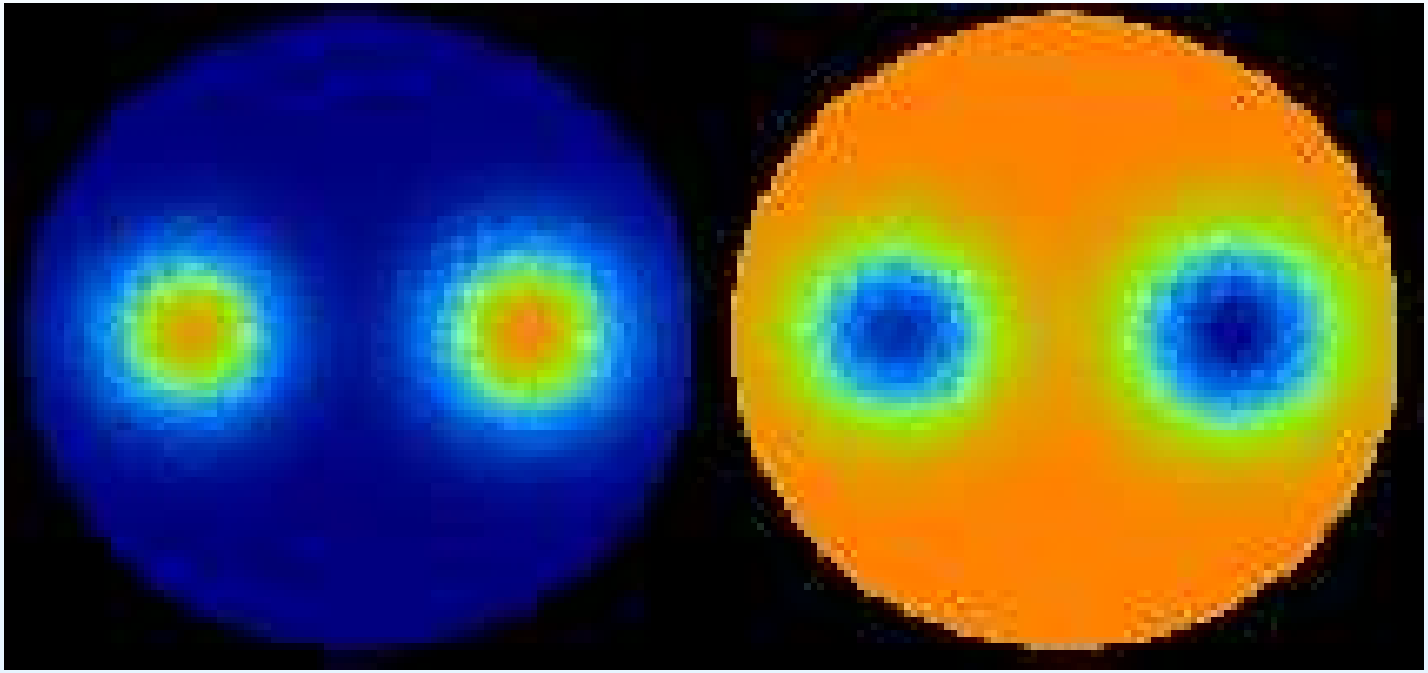
Description	Name	Mean	Perturbation
encounter gain	μ_1	1	$\pm 50\%$
response gain	μ_2	10.1	$\pm 50\%$
predator death rate	μ_3	1	$\pm 50\%$
prey growth rate	μ_4	5	$\pm 50\%$
prey carrying capacity	μ_5	1	$\pm 50\%$
encounter loss	μ_6	1	$\pm 50\%$

We use the L^1 norm of the prey population at $t = 10$ as the quantity of interest $(\psi = \delta(t - 10)(0, 1)^\top)$

We use a 12400 point Monte-Carlo computation as a reference

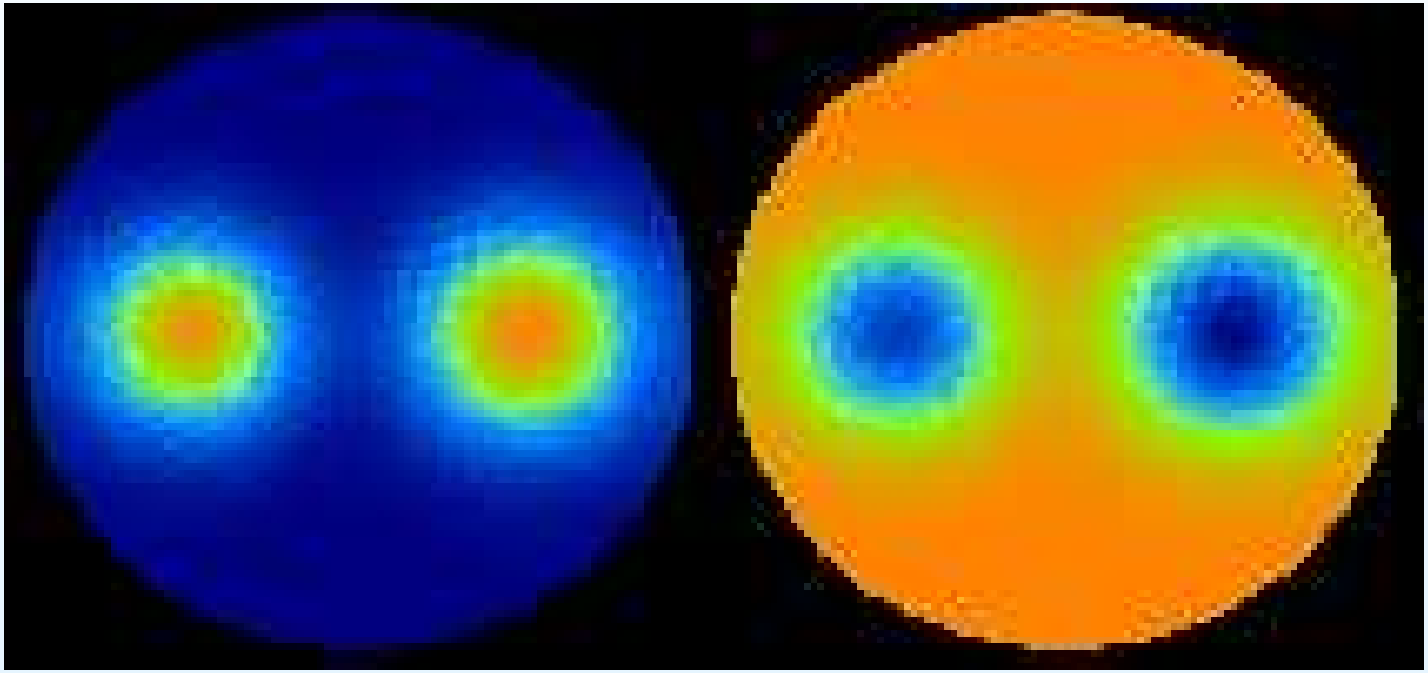
Evolution of the Solution

We show a few snapshots in time of the predator (left) and prey (right). Warmer colors mean higher density.



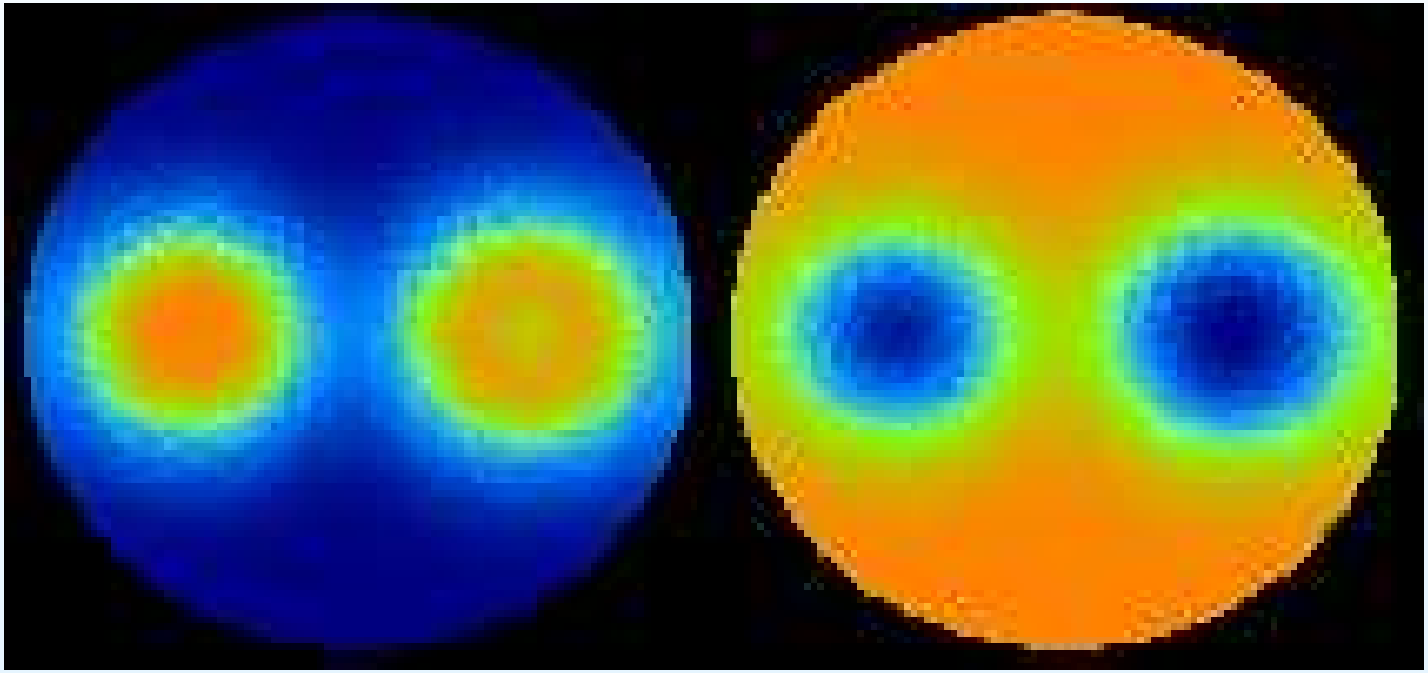
Evolution of the Solution

We show a few snapshots in time of the predator (left) and prey (right). Warmer colors mean higher density.



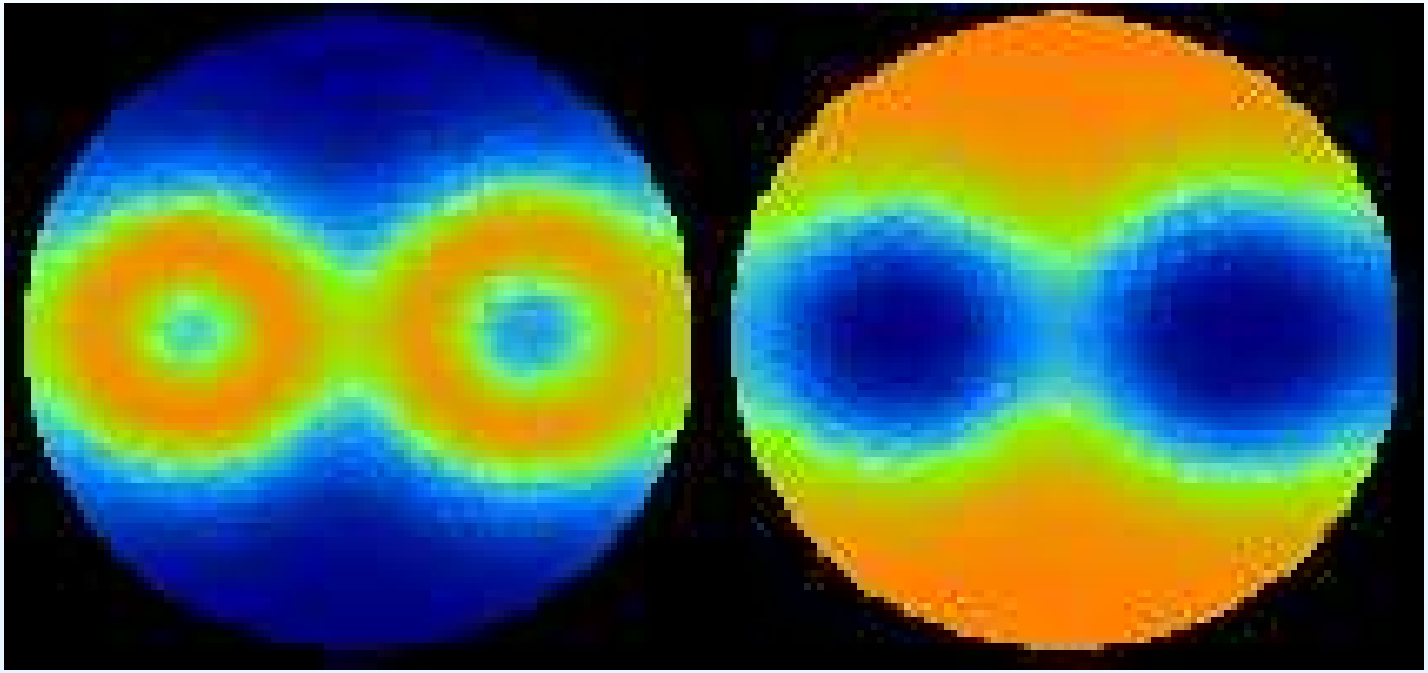
Evolution of the Solution

We show a few snapshots in time of the predator (left) and prey (right). Warmer colors mean higher density.



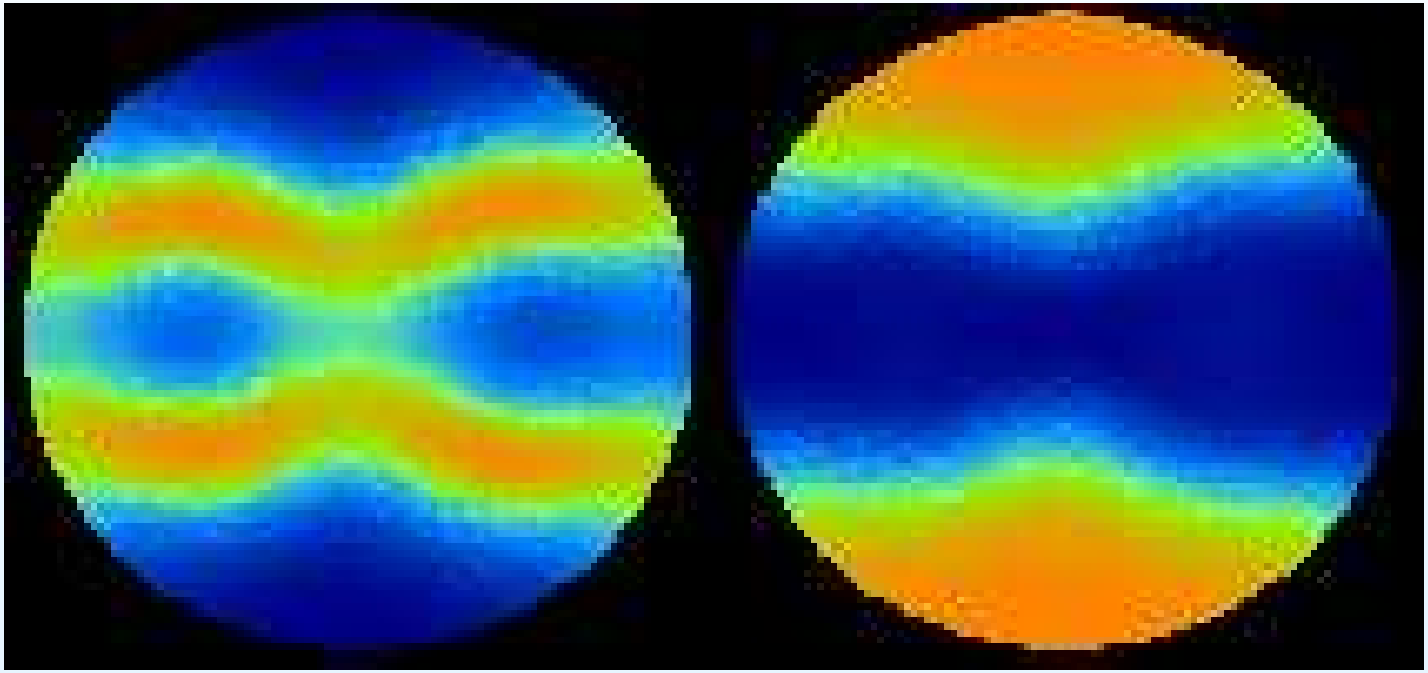
Evolution of the Solution

We show a few snapshots in time of the predator (left) and prey (right). Warmer colors mean higher density.

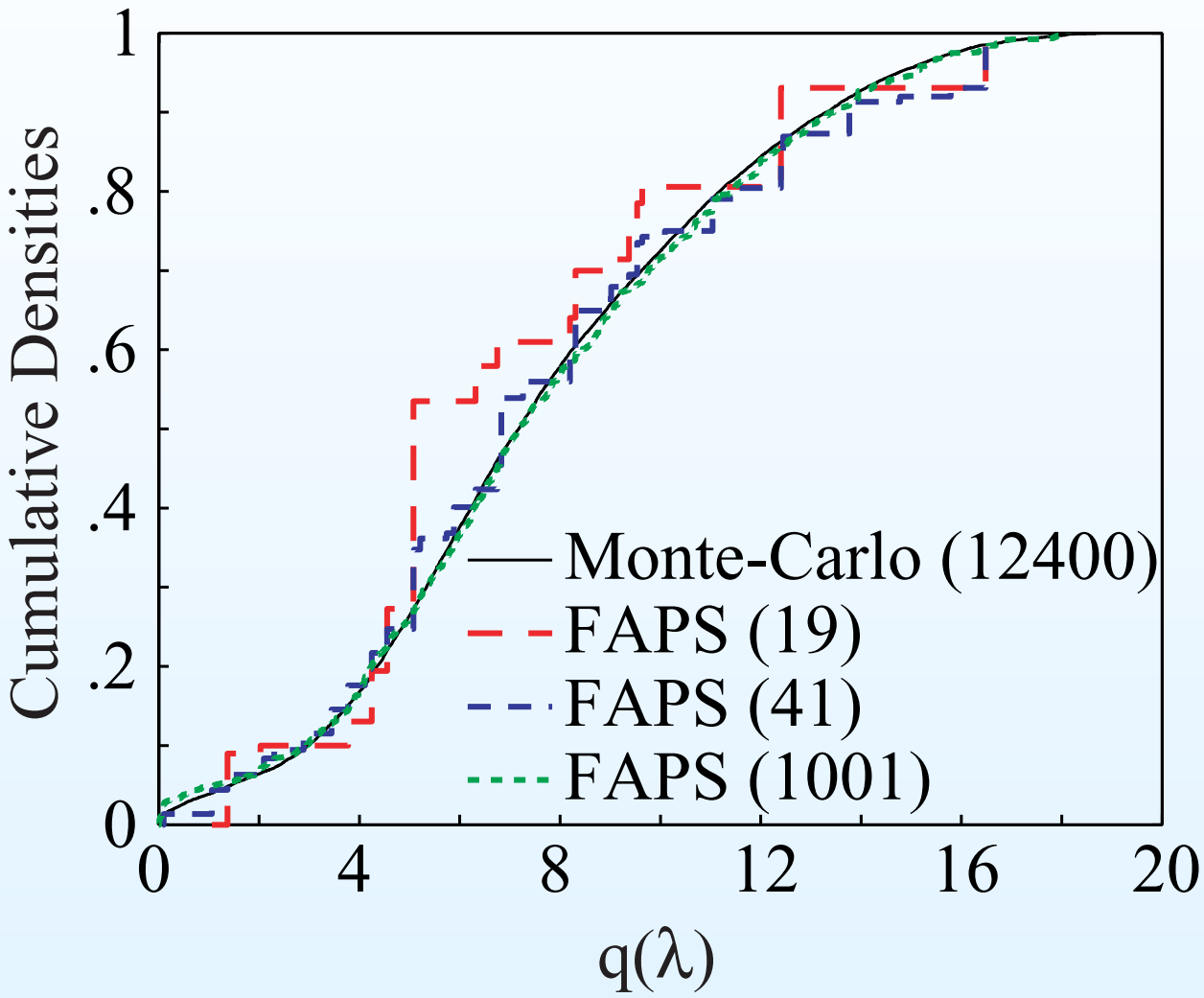


Evolution of the Solution

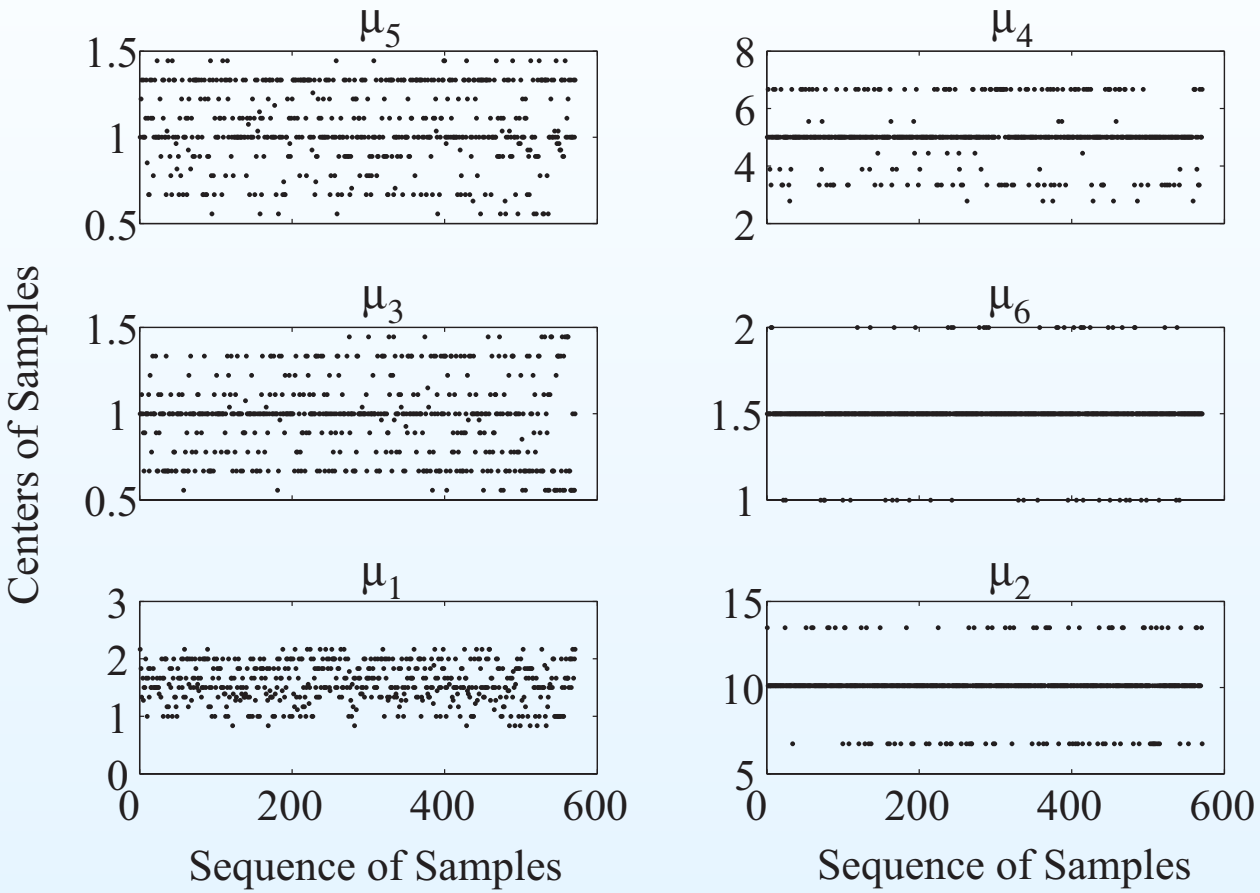
We show a few snapshots in time of the predator (left) and prey (right). Warmer colors mean higher density.



Predator Prey Results (Cumulative Density)



Predator Prey Results (Dimension Reduction)



Samples used for each parameter

The gradient can be used to determine which parameters do not contribute significantly, leading to dimension reduction.

Adaptive Error Control for Kernel Density Estimation

Solving the kernel density estimation problem requires solving the problem for a variety of data and parameter values

The corresponding solutions can exhibit a variety of behaviors

It is important to control the numerical errors so that they do not bias the analysis results

Adaptive Error Control for Kernel Density Estimation

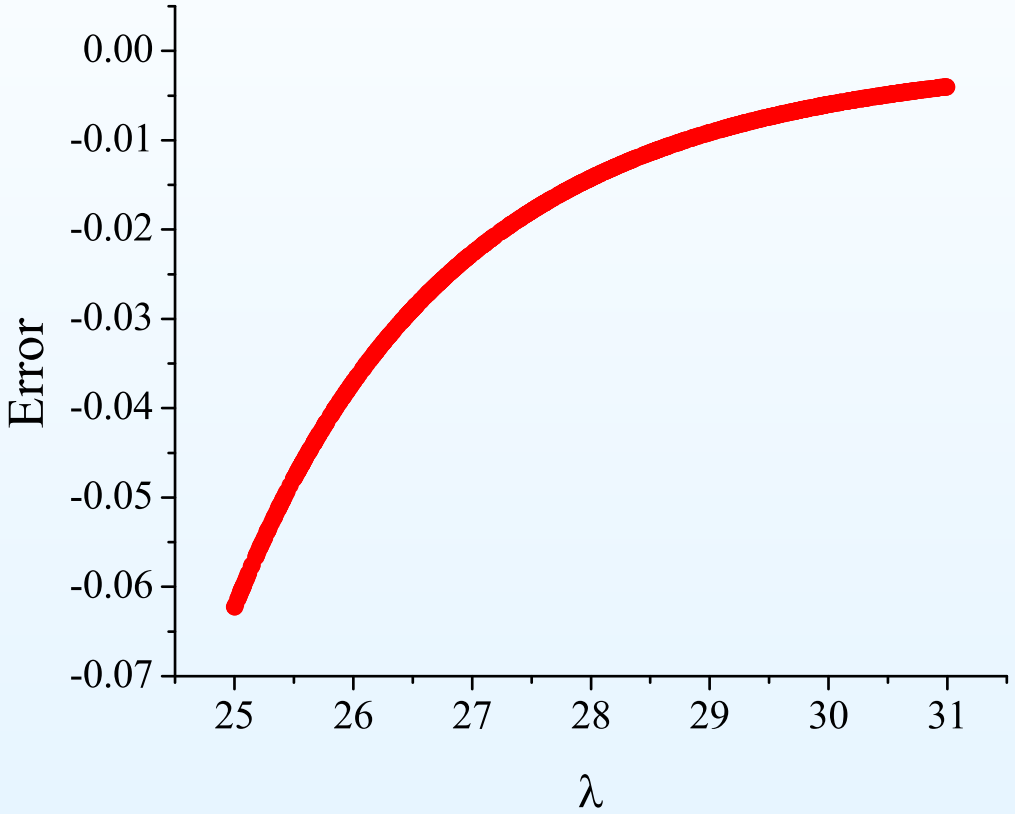
Example We consider the chaotic Lorenz problem

$$\begin{cases} \dot{u}_1 = -10u_1 + 10u_2, \\ \dot{u}_2 = \lambda u_1 - u_2 - u_1 u_3, & 0 < t, \\ \dot{u}_3 = -\frac{8}{3}u_3 + u_1 u_2, \\ u_1(0) = -6.9742, u_2(0) = -7.008, u_3(0) = 25.1377 \end{cases}$$

We vary $\lambda \sim$ Unif $[25, 31]$

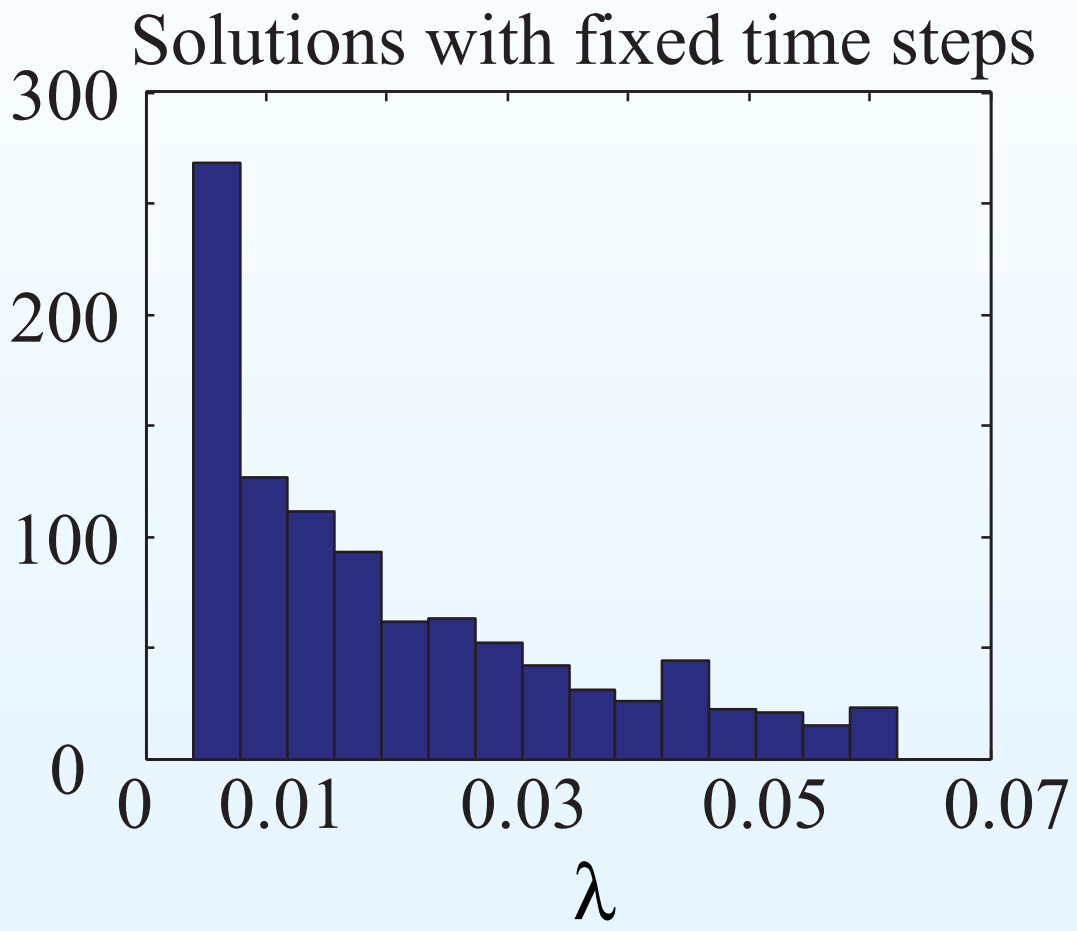
We use 1000 point Monte-Carlo sampling with both a fixed time step computation and an adaptive computation with error smaller than 10^{-5}

Adaptive Error Control for Kernel Density Estimation



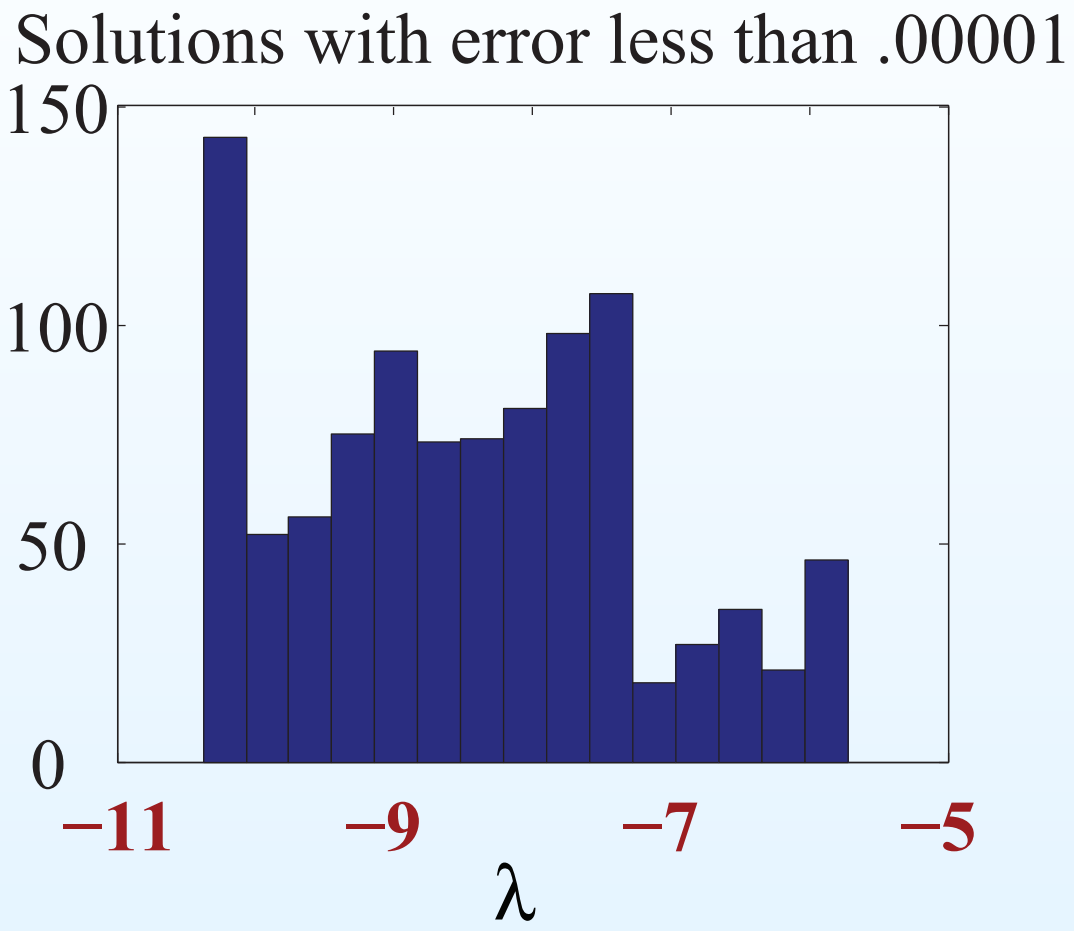
Numerical error versus parameter for a fixed time step

Adaptive Error Control for Kernel Density Estimation



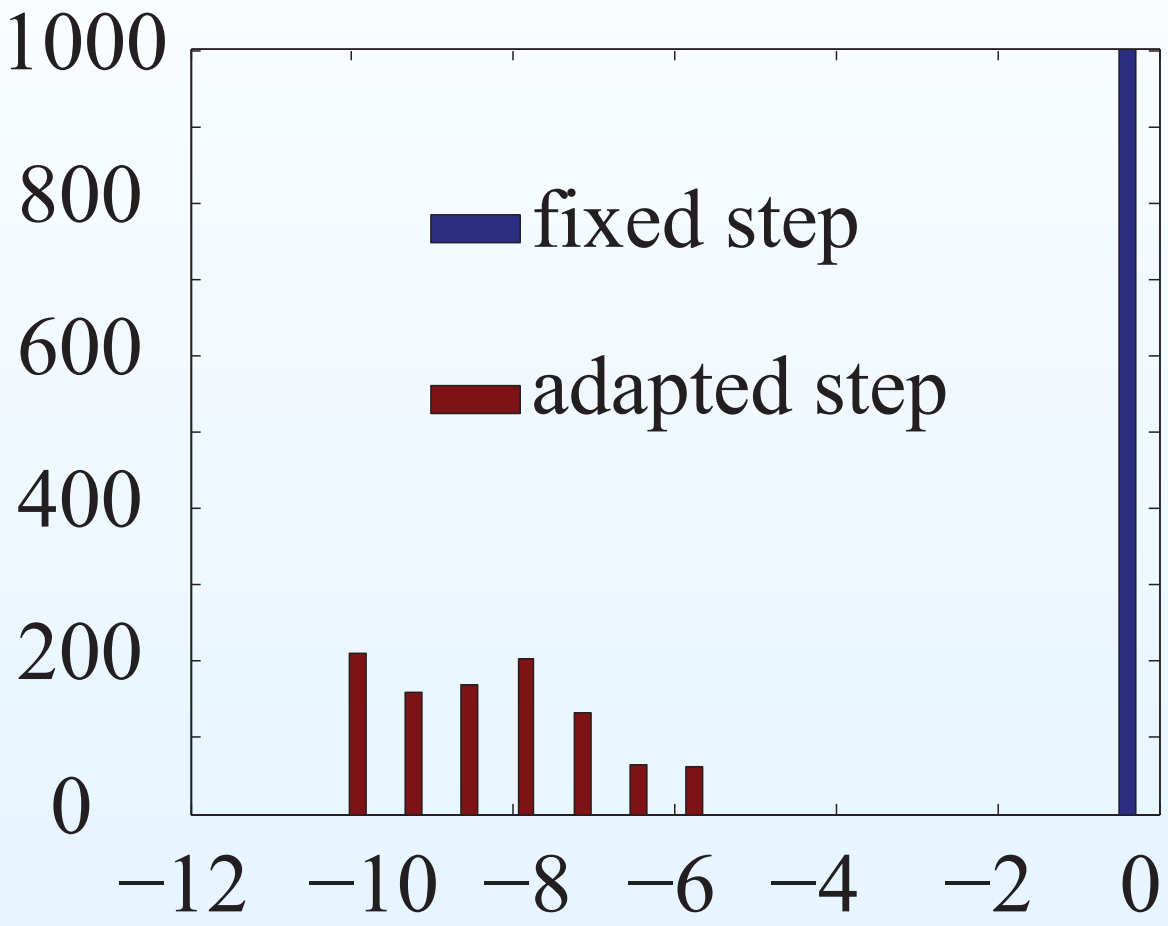
The distribution for the quantity of interest using a fixed time step

Adaptive Error Control for Kernel Density Estimation



The distribution for the quantity of interest using adapted time steps

Adaptive Error Control for Kernel Density Estimation



Both distributions

Comments on Model Sensitivity Analysis and Adjoints

In general, the adjoint solution provides an efficient way to approximate

$$\nabla_{\text{parameter}} \text{ quantity of interest}$$

This is one reason that adjoints are natural for analysis of model sensitivity and in optimization problems

A posteriori analysis represents the effects of random perturbation in terms of a convolution with the adjoint solution

This provides a way to include both deterministic and probabilistic representations of uncertainty in the same analysis framework

Parameter Optimization for an Elliptic Problem

An Elliptic Problem with Parameter

We solve the elliptic problem

$$\begin{cases} -\nabla \cdot (a(x)\nabla u) = f(u, x; \lambda), & x \in \Omega,\ \lambda \in \Lambda, \\ u = 0, & x \in \partial\Omega. \end{cases}$$

Search for $\lambda \in \Lambda$ that optimizes a linear functional

$$q(u(\lambda)) = (u, \psi) = \int_\Omega u(x; \lambda)\, \psi(x)\, dx$$

We use a conjugate gradient method using the Hestenes-Stiefel formula and the secant method for the line search

$$\tilde{\lambda} \xrightarrow[\nabla_\lambda q(u(\tilde{\lambda}))]{q(u(\tilde{\lambda}))} \lambda$$

old *new*

The Role of the Adjoint Problem

We optimize

$$q(u(\lambda)) = (u(\lambda), \psi)$$

ϕ solves the linearized adjoint problem

$$\begin{cases} -\nabla \cdot (a(x)\nabla\phi) - D_u^* f(u;\lambda)\phi = \psi, & x \in \Omega, \\ u = 0, & x \in \partial\Omega. \end{cases}$$

$D_u^* f$ is the adjoint of the Jacobian of f

The gradient formula at $\tilde{\lambda} \in \Lambda$

$$\nabla_\lambda q(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) \approx \left(\nabla_\lambda f(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}\right)$$

$\tilde{u}$ and $\tilde{\phi}$ solutions for parameter value $\tilde{\lambda}$

Numerical Approximations

Standard finite element approximation: compute $U \in V_h$

$$(a\nabla U, \nabla v) = (f(U), v) \quad \text{for all } v \in V_h$$

V_h is the standard space associated with a triangulation of Ω

Using U affects both

$$q(u(\lambda)) \to q(U(\lambda))$$
$$\nabla_\lambda q(u(\lambda)) \to \nabla_\lambda q(U(\lambda))$$

We need to control the errors in the value and the gradient used for the search

A Posteriori Estimate of Numerical Error

The *a posteriori* estimate is

$$\text{error in } q(U) \approx (a\nabla U, \nabla(\phi - \pi_h\phi)) - (f(U), \phi - \pi_h\phi)$$

The true value of the gradient can be estimated as

$$\nabla_\lambda q(\tilde{u};\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}) \approx \left(\nabla_\lambda f(\tilde{U};\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi}\right) + \mathcal{R}(U,\phi) - \mathcal{R}(U,\tilde{\phi})$$

This is the computable correction for the effects of numerical approximation

This expression reflects the change in stability arising from the change in parameter value

If we control this error, we can prove that the search leads to a minimum

Adaptive Error Control

The *a posteriori* estimate has the form

$$|\text{error in } q(U)| \approx \left| \sum_{\text{elements}} \text{element contribution} \right|$$

The corresponding bound on the error in the gradient has the form

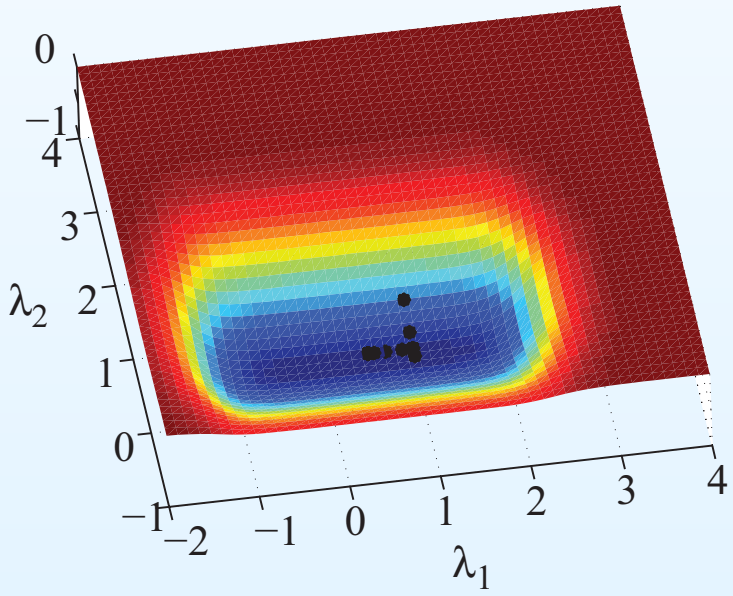
$$|\text{error in } \nabla_\lambda q(U)| \leq \sum_{\text{elements}} |\text{change in element contribution}|$$

We alter the adaptive strategy accordingly

Optimization Example

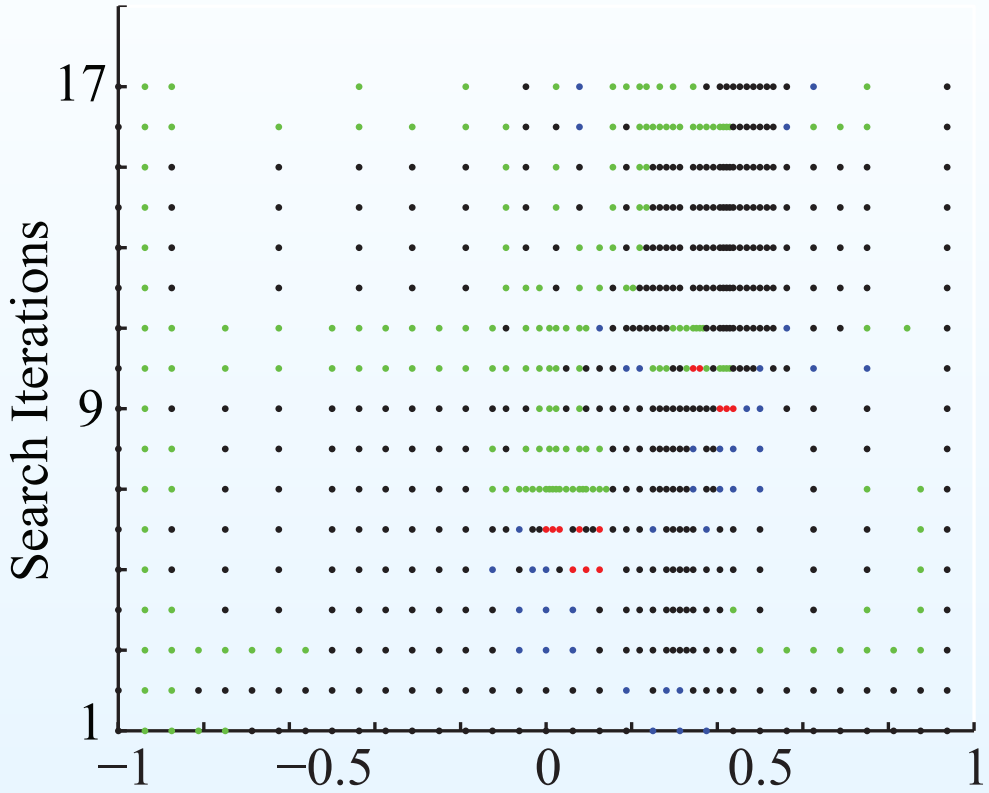
We optimize $(u,1)$ where u solves

$$\begin{cases} -u'' = u^2 + \tanh^2\left(20e^{\lambda_1(1-\lambda_1)}(x - e^{\lambda_2(1-\lambda_2)-1})\right)\,\cos^2(\frac{\pi}{2}x), \quad [-1,1], \\ u(-1) = u(1) = 0 \end{cases}$$



The quantity of interest

Optimization Example: Meshes



The sequence of meshes

The meshes for each search step are plotted vertically. The meshes are refined to control the error in the gradients. The refinement typically affects a small number of elements in each step.

Domain Decomposition

Stability in Elliptic Problems

A characteristic of elliptic problems is a global domain of influence

A local perturbation of data near one point affects a solution u throughout the domain of the problem

However in many cases, the strength of the effect of a perturbation on a point value of a solution decays significantly with the distance to the support of the perturbation

The effective domain of influence for a functional of the solution is reflected in the graph of the adjoint solution

A Decomposition of the Solution

The effective domain of influence of a particular functional will not be local unless the data for the adjoint problem has local support

We use a partition of unity to "localize" a problem in which supp (ψ) does not have local support

Corresponding to a partition of unity $\{p_i, \Omega_i\}$, $\psi \equiv \sum_{i=1}^{N} \psi p_i$,

Definition
The quantities $\{(U, \psi p_i)\}$ corresponding to the data $\{\psi_i = \psi p_i\}$ are called the localized information corresponding to the partition of unity

We consider the problem of estimating the error in the localized information for $1 \leq i \leq N$

A Decomposition of the Solution

We obtain a finite element solution via:

Compute $\hat{U}_i \in \hat{V}_i$ such that $A(\hat{U}_i, v) = (f, v)$ for all $v \in \hat{V}_i$,

where $\hat{V}_i$ is a space of continuous, piecewise linear functions on a locally quasi-uniform simplex triangulation $\mathcal{T}_i$ of Ω obtained by local refinement of an initial coarse triangulation $\mathcal{T}_0$ of Ω

$\{\hat{V}_i\}$ is globally defined and the "localized" problem is solved over the entire domain

We hope that this will require a locally refined mesh because the corresponding data has localized support

A Decomposition of the Solution

A partition of unity approximation in the sense of Babuška and Melenk uses $U_i = \chi_i \hat{U}_i$, $1 \leq i \leq N$, where χ_i is the characteristic function of Ω_i

The local approximation U_i is in the local finite element space $V_i = \chi_i \hat{V}_i$

Definition

The {partition of unity approximation is defined by $U_p = \sum_{i=1}^{N} U_i p_i$, which is in the fpartition of unity finite element space

$$V_p = \sum_{i=1}^{N} V_i p_i = \left\{ \sum_{i=1}^{N} v_i p_i : v_i \in V_i \right\}$$

A Decomposition of the Solution

We use the generalized Green's function satisfying the adjoint problem:

Find $\phi_i \in H_0^1(\Omega)$ such that $A^*(v,\phi_i) = (v,\psi_i)$ for all $v \in H_0^1(\Omega)$.

Letting $\pi_i\phi_i$ denote an approximation of ϕ_i in $\hat{V}_i$,

Theorem The error of the partition of unity finite element solution satisfies

$$(u - U_p, \psi) = \sum_{i=1}^{N} \big((f, \phi_i - \pi_i\phi_i) - (a\nabla\hat{U}_i, \nabla(\phi_i - \pi_i\phi_i)) - (b\cdot\nabla\hat{U}_i, \phi_i - \pi_i\phi_i) - (c\hat{U}_i, \phi_i - \pi_i\phi_i) \big)$$

Computation of Multiple Quantities of Interest

We present an algorithm for computing multiple quantities of interest efficiently using knowledge of the effective domains of influence of the corresponding Green's functions

We assume that the information is specified as $\{(U, \psi_i)\}_{i=1}^N$ for a set of N functions $\{\psi_i\}_{i=1}^N$

Two approaches:

Approach 1: **A Global Computation**
Find one triangulation such that the corresponding finite element solution satisfies $|(e, \psi_i)| \leq \text{TOL}_i$, for $1 \leq i \leq N$

Approach 2: **A Decomposed Computation**
Find N independent triangulations and finite element solutions U_i so that the errors satisfy $|(e_i, \psi_i)| \leq \text{TOL}_i$, for $1 \leq i \leq N$

Computation of Multiple Quantities of Interest

If the correlation between the effective domains of influence associated to the N data $\{\psi_i\}$ is relatively small, then each individual solution in the Decomposed Computation will require significantly fewer elements than the solution in the Global Computation to achieve the desired accuracy

This can yield significant computational advantage in terms of lowering the maximum memory requirement to solve the problem

We optimize resources by combining localized computations whose domains of influence are significantly correlated

Domain Decomposition Example

Consider once again

$$\begin{cases} -\nabla \cdot \big(\big(.05 + \tanh\big(10(x-5)^2 + 10(y-1)^2\big)\big) \nabla u \big) \\ \qquad\qquad + \begin{pmatrix} -100 \\ 0 \end{pmatrix} \cdot \nabla u = 1, & (x,y) \in \Omega, \\ u(x,y) = 0, & (x,y) \in \partial\Omega, \end{cases}$$

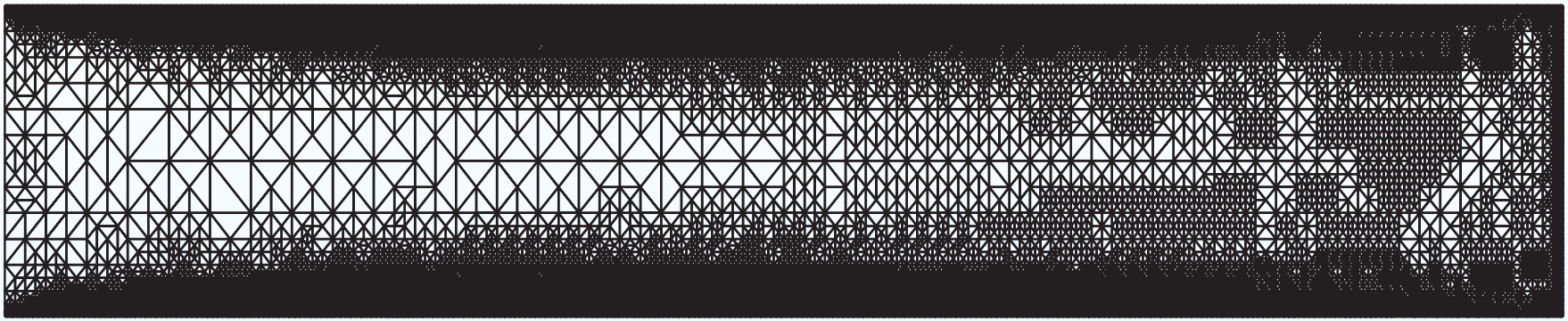
where $\Omega = [0,10] \times [0,2]$

Domain Decomposition Example

We use an initial mesh of 80 elements and an error tolerance of $TOL = .04\%$ for the average error over Ω

Level	Elements	Estimate
1	80	$-.0005919$
2	193	$-.001595$
3	394	$-.0009039$
4	828	$-.0003820$
5	1809	$-.0001070$
6	3849	$-.00004073$
7	9380	$-.00001715$
8	23989	$-.000007553$

Domain Decomposition Example



Final Mesh for an Error of 4% in the Average Value (24,000 Elements)

Domain Decomposition Example

11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

Domains for the partition of unity

Domain Decomposition Example

Significant Correlations:

Ω_3 with Ω_4 $\quad$ Ω_6 with Ω_7 $\quad$ Ω_7 with Ω_6 $\quad$ Ω_9 with Ω_8 $\quad$ Ω_{10} with Ω_8, Ω_9

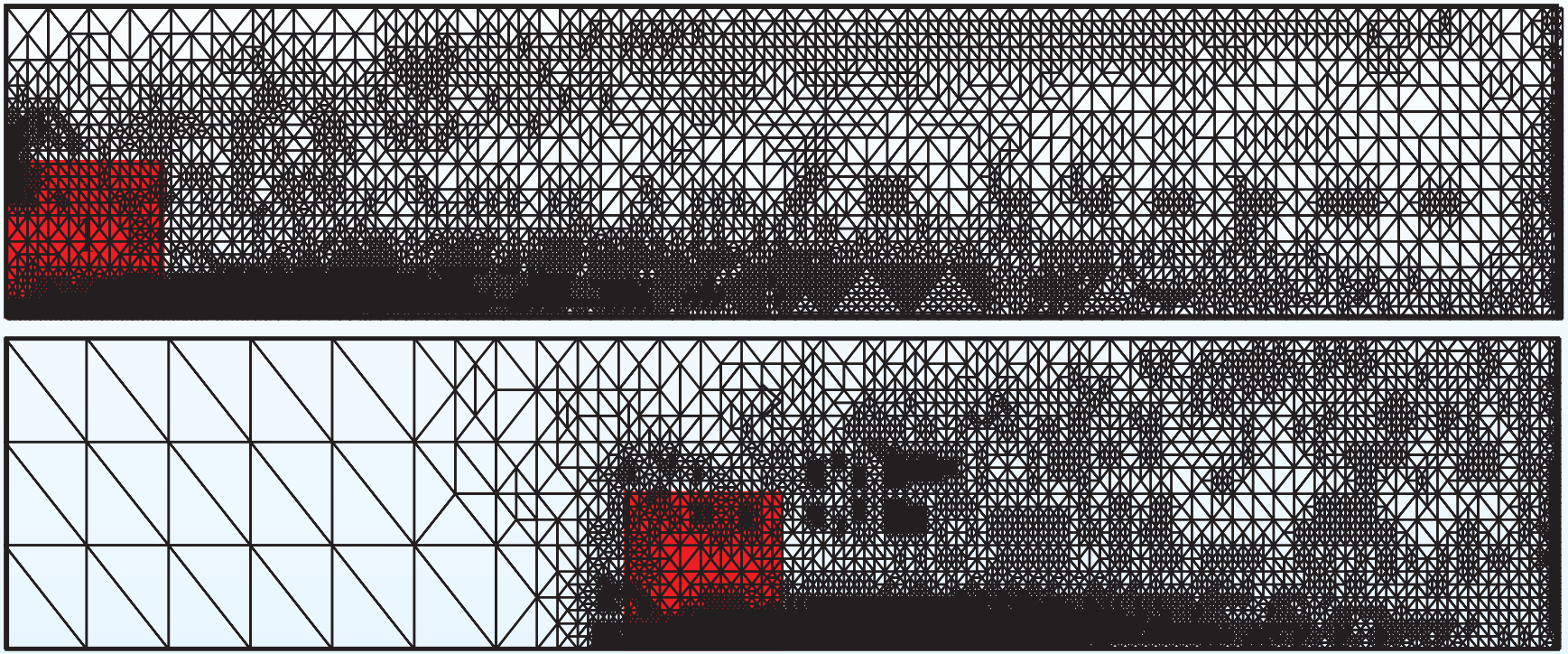
Ω_{13} with Ω_{14} $\quad$ Ω_{16} with Ω_{17} $\quad$ Ω_{17} with Ω_{16} $\quad$ Ω_{19} with Ω_{18} $\quad$ Ω_{20} with Ω_{18}, Ω_{19}

There are no significant correlations in the “cross-wind” direction

Domain Decomposition Example

Data	TOL	Level	Elements	Estimate
ψ_1	.04%	7	7334	-6.927×10^{-7}
ψ_2	.04%	7	8409	-5.986×10^{-7}
ψ_3	.04%	7	7839	-5.189×10^{-7}
ψ_4	.04%	7	7177	-5.306×10^{-7}
ψ_5	.04%	7	7301	-4.008×10^{-7}
ψ_6	.02%	7	6613	-2.471×10^{-7}
ψ_7	.02%	7	4396	-2.938×10^{-7}
ψ_8	.02%	7	4248	-1.656×10^{-7}
ψ_9	.02%	7	3506	-1.221×10^{-7}
ψ_{10}	.02%	7	1963	-5.550×10^{-8}

Domain Decomposition Example



Final Mesh for an Average Error in a two Patches

Domain Decomposition Example

The global computation uses roughly 3 times the number of elements of the largest individual computation in the decomposed computation

In a high performance computing environment, the cost of solution typically scales superlinearly with memory usage

There is a much greater effect of decay of influence on complex geometry, e.g. with “holes”, interior corners, and so on

Work in Progress

Applications Not Discussed

- Analysis of operator decomposition for coupling stochastic models, e.g. molecular dynamics, to continuum models
- Determining the range of acceptable error on parameters and data in order to compute a quantity of interest to an acceptable accuracy
- Error estimates for operator decomposition for multiphysics problems with “black box” components
- Data assimilation and model calibration under uncertainty
- Parallel space-time adaptive integration and compensated domain decomposition

References

References

These references are a starting point for further investigation

- Ainsworth, M. and J. T. Oden, A Posteriori Error Estimation In Finite Element Analysis, Pure and Applied Mathematics Series, 2000, Wiley-Interscience, 2000.
- Babuska, I. and T. Strouboulis, The Finite Element Method and its Reliability, 2001, Oxford University Press: New York.
- W. Bangerth and R. Rannacher, Adaptive Finite Element Methods for Differential Equations, 2003. Birkhauser Verlag: New York.
- Becker, R. and R. Rannacher, An Optimal Control Approach to A Posteriori Error Estimation in Finite Element Methods, Acta Numerica, 2001. p. 1-102.
- Braack, M. and A. Ern, A Posteriori Control Of Modeling Errors And Discretization Errors, SIAM J. Multiscale Modeling and Simulation, 2003. 1, p. 221-238.
- Cacuci, D.G., Sensitivity and Uncertainty Analysis I. Theory, 2003, Chapman and Hall: Boca Raton, Florida.
- Cacuci, D.G., M. Ionescu-Bujor, and I. M. Navon, Sensitivity and Uncertainty Analysis. II. Applications to Large Scale Systems, 2005, Chapman and Hall: Boca Raton, Florida.

References

- Carey, V., D. Estep, and S. Tavener, A Posteriori Analysis and Adaptive Error Control for Operator Decomposition Methods for Elliptic Systems I: One Way Coupled Systems, submitted to SIAM Journal on Numerical Analysis, 2007.
- Carey, V., D. Estep, and S. Tavener, A Posteriori Analysis and Adaptive Error Control for Operator Decomposition Methods for Elliptic Systems II: Fully Coupled Systems, in preparation.
- Eriksson, K., D. Estep, P. Hansbo and C. Johnson, Introduction To Adaptive Methods For Differential Equations, Acta Numerica, 1995. p. 105-158.
- Eriksson, K., D. Estep, P. Hansbo and C. Johnson, Computational Differential Equations, 1996, Cambridge University Press: New York.
- Estep, D., A Short Course on Duality, Adjoint Operators, Green's Functions, and A Posteriori Error Analysis, Sandia National Laboratories, Albuquerque, New Mexico, 2004. Notes can be downloaded from `http://math.colostate.edu/~estep`
- Estep, D., M. G. Larson, and R. D. Williams, Estimating The Error Of Numerical Solutions Of Systems Of Reaction-Diffusion Equations, Memoirs of the American Mathematical Society, 2000. 146, p. 1-109.

References

- Estep, D. and D. Neckels, Fast And Reliable Methods For Determining The Evolution Of Uncertain Parameters In Differential Equations, Journal on Computational Physics, 2006. 213, p. 530-556.
- Estep, D. and D. Neckels, Fast Methods For Determining The Evolution Of Uncertain Parameters In Reaction-Diffusion Equations, to appear in Computer Methods in Applied Mechanics and Engineering, 2006.
- Estep, D., V. Ginting, D. Ropp, J. Shadid and S. Tavener, An A Posteriori Analysis of Operator Splitting, submitted to SIAM Journal on Numerical Analysis, 2007.
- Estep, D., M. Holst and M. Larson, Generalized Green's Functions And The Effective Domain Of Influence, SIAM Journal on Scientific Computing, 2005. 26, p. 1314-1339.
- Estep, D., A. Malqvist, and S. Tavener, A Posteriori Analysis For Elliptic Problems With Randomly Perturbed Data And Coefficients, in preparation
- Estep, D., A. Malqvist, and S. Tavener, Adaptive Computation For Elliptic Problems With Randomly Perturbed Data And Coefficients, in preparation

References

- Estep, D., S. Tavener and T. Wildey, A Posteriori Analysis and Improved Accuracy for an Operator Decomposition Solution of a Conjugate Heat Transfer Problem, submitted to SIAM Journal on Numerical Analysis, 2006.
- Giles, M. and E. Suli, Adjoint Methods for PDEs: A Posteriori Error Analysis And Postprocessing By Duality, 2002. Acta Numerica, p. 145-236.
- Hundsdorfer, W. and Verwer, J., Numerical solution of time-dependent advection-diffusion-reaction equations, Springer Series in Computational Mathematics, 2003. Springer-Verlag: New York.
- Lanczos, C., Linear Differential Operators, 1996. SIAM: Philadelphia.
- Marchuk, G.I., Splitting and Alternating Direction Methods, in Handbook of Numerical Analysis, 1990, P. G. Ciarlet and J. L. Lions, editors. North-Holland: New York, p. 197-462.
- Marchuk, G.I., Adjoint Equations and Analysis of Complex Systems, 1995. Kluwer: Boston.
- Marchuk, G.I., V. I. Agoshkov, and V. Shutyaev, Adjoint Equations And Perturbation Algorithms In Nonlinear Problems, 1996. CRC Press: Boca Raton, Florida.

References

- Oden, J.T. and Prudhomee, S., Estimation of Modeling Error in Computational Mechanics, Journal on Computational Physics, 2002. 182, p. 496-515.