ISSUES IN THE COMPUTATIONAL STUDY OF THE MAGNETIC CONTROL OF FLUID FLOWS

Max Gunzburger

School of Computational Science and Information Technology

Florida State University

QUESTION

Can computational methodologies be used to design and control processes that involve MHD flows?

computational methodologies

=

sophisticated simulation algorithms for the flows

+

sophisticated optimization and control algorithms

ANSWER 1: YES

Such approaches have been successfully used in several academic and industrial settings involving MHD flows

- plasma wakes, fusion reactors, liquid metals, sea water, crystal growth, solidification processes, chemical vapor deposition, turbulence reduction or enhancement, heating and cooling, confinement, etc.

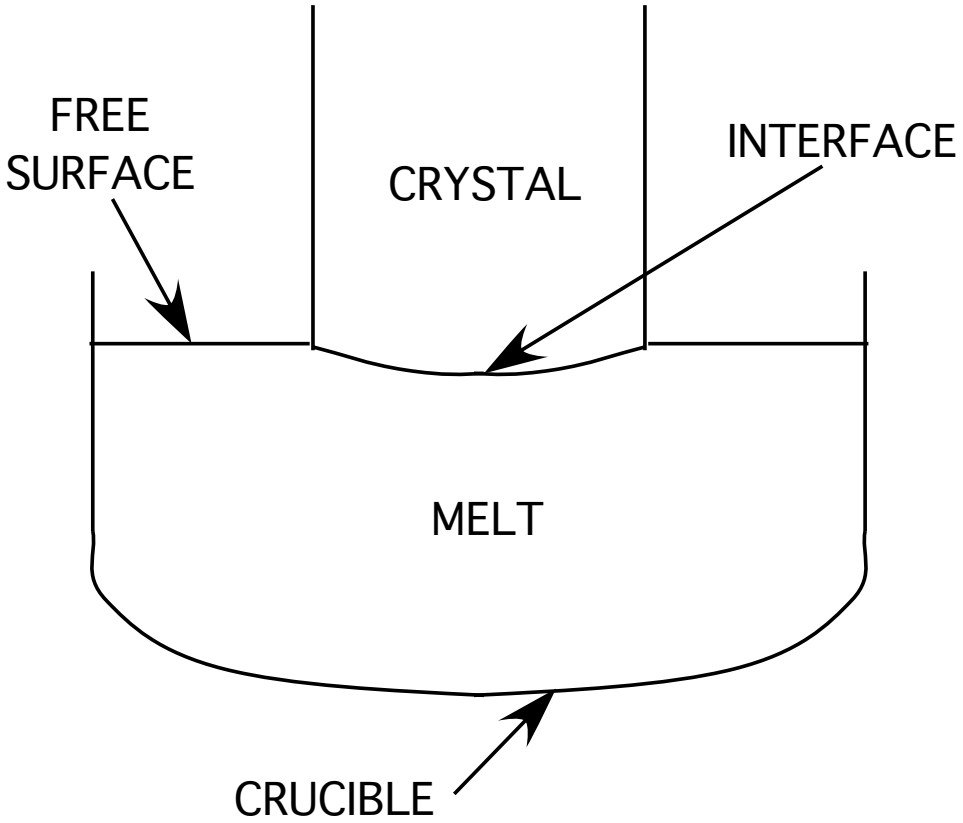
For some settings (e.g., incompressible MHD), rigorous mathematical analyses of some control and optimization algorithms have been performed

ANSWER 2: MAYBE

Progress still needs to be made in order to be able to routinely use computational approaches for the control or optimal design of processes that involve MHD flows

- Here, we discuss
  - what are some the impediments
  - possible ways to overcome some of them

MAGNETIC CONTROL IN THE CZOCHRALSKI CRYSTAL GROWTH PROCESS



Simplified schematic of process

- Goals: grow better (bigger, more uniform, cheaper, etc.) crystals
- Approach: control the growth process by adjusting the
  - heat input
  - pull rate
  - rotation rates
  - magnetic field

  so that the "goals" are met as well as possible
- Quantification of goals is a difficult problem
  - usually not done in terms of direct material (e.g., crystalline, electromagnetic) properties of the crystal
  - instead, it is usually done in terms of indirect properties, e.g., of the fluid melt, that hopefully positively impact the quality of the resulting crystal

- Quantified goal: minimize the magnitude of the vorticity (curl of the velocity) in the melt
  - reduce the number of eddies in the melt
  - believed to result in improvements in the shape of the melt/crystal interface and the properties of the crystal

- Controls available:
  - axial magnetic field
  - heating distribution along the crucible wall
  - pull rate of the crystal
  - rotation rate of the crucible
  - rotation rate of the crystal

- Governing equations:
  - MHD equations in the melt (coupled fluids/electromagnetics)
  - energy equation in the crystal
  - fluid equations in the encapsulating gas

- Computational approach: apply a sophisticated optimization algorithm
  - to determine the optimal values of the control parameters

  such that

  - the cost (the magnitude of the vorticity) is minimized

  and

  - the constraints (the governing equations) are satisfied

- Thus, optimal values of the control parameters are determined automatically and systematically

- This is a very large-scale computational problem
  - even though we left out lots from the model equations

- It is found that the most effective control is the magnetic field

ISSUES

STATE MODELING AND SIMULATION ISSUES

- How true-to-life to make the computational model?
  - computationally tractable models usually involve many simplifications
    - actuators are often not modeled realistically
    - geometry is often simplified
    - some system components are often ignored

- Despite the state modeling simplifications invoked
  - the state equations are a very complex system of partial differential eqations
    - so solving them is a computationally expensive task
  - useful and practical information can be determined by solving the state system

- For most processes, state simulation codes have already been developed
  - these are large-scale codes with respect to
    - complexity
    - cost to run
    - person-years of development effort

- As a result,
  - one often has to optimize or control under the constraint that one has to use existing state simulation codes
    - this may be either a good thing or a bad thing, depending on the quality of the code
  - it may be difficult to develop the adjoint or sensitivity codes needed for popular gradient-based optimization approaches

OBJECTIVE FUNCTIONAL MODELING ISSUES

- Quantifying objectives and goals is usually problematical
  - desired goals may not be easily described in terms of the state variables

- One often tries to achieve goals by indirect means
  - some functional of the state is minimized with the expectation (hope?) that, indirectly, the real goal is achieved (at least approximately)
  - it is often difficult to determine, even after the fact, if one has indeed been able to meet the real goal

CONTROL MODELING ISSUES

- Control and sensing mechanisms are also difficult to quantify and model
  - accurately including actuating and sensing devices is often not possible in computations
    - e.g., realistically depicting the coils that produce magnetic fields would greatly complicate the computational model

- As a result,
  - actuators and sensors are often modeled through boundary conditions or source terms appearing in the state system

COMPUTATIONAL ISSUES

- High-fidelity state simulations are very expensive
  - cannot use them for real-time feedback control applications
  - difficult to use them for optimal design applications

- Often, it is difficult or costly
  - to use adjoint or sensitivity-based gradient methods for effecting optimization
  - to develop optimal feedback laws

Despite all the issues mentioned,

there have been substantial successes

in the use of computations to determine

effective magnetic controls

for favorably affecting fluid flows

relevant to manufacturing processes

OTHER SETTINGS

- Many of the same issues arise in the computational determination of magnetic controls in other settings involving fluid flows
  - plasma wakes, fusion reactors, liquid metals, sea water, chemical vapor deposition, turbulence reduction or enhancement, heating and cooling, confinement, etc.
  - again, there have been great successes in the computational determination of magnetic controls for many of these settings

- We will now discuss approaches applicable to situations in which
  - existing state simulation codes have to be used as a black box
  - the high cost of the state simulations preclude their use in optimal design or feedback control settings

USING SURROGATES FOR CONTROL AND DESIGN PROBLEMS

MINIMIZATION USING SURROGATES

- Given the minimization problem:

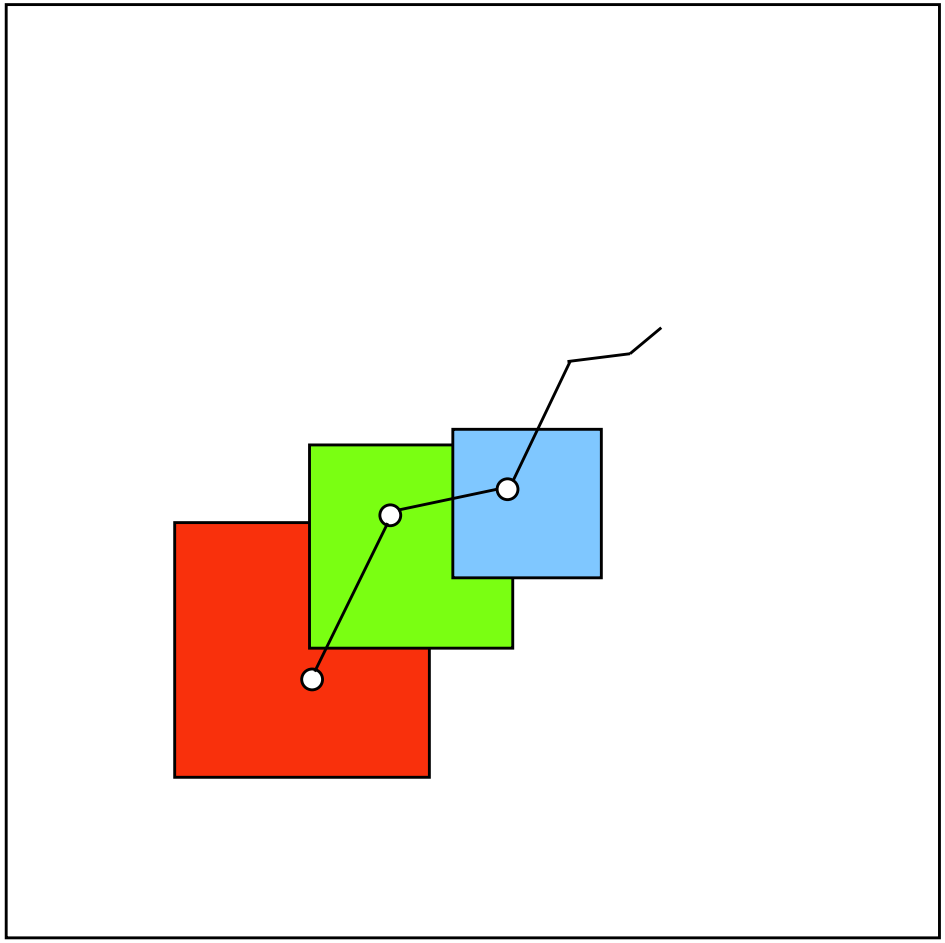
$$\text{minimize } f(\mathbf{x}) \qquad \text{subject to } \begin{cases} g_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m, \\ \mathbf{x}_\ell \leq \mathbf{x} \leq \mathbf{x}_u \end{cases}$$

we instead solve the surrogate-based optimization problem:

for $k = 0, 1, 2, \ldots, k_{max}$,

$$\text{minimize } \widetilde{f}^{(k)}(\mathbf{x}) \qquad \text{subject to } \begin{cases} \widetilde{g}_i^{(k)}(\mathbf{x}) \leq 0, \ i = 1, \ldots, m, \\ \mathbf{x}_\ell \leq \mathbf{x} \leq \mathbf{x}_u \\ \mathbf{x} \in \square_k \end{cases}$$

where $\widetilde{f}^{(k)}(\mathbf{x})$ and $\widetilde{g}_i^{(k)}(\mathbf{x})$ are surrogates for $f(\mathbf{x})$ and $g_i(\mathbf{x})$, respectively, defined within the trust region $\square_k$



Minimizer of surrogate in each trust region

Path to approximate optimum

- Note that for the problems we are interested in,
  - $\mathbf{x}$ may be a vector of parameters
  - $f(\mathbf{x}) = F(\phi(\mathbf{x}), \mathbf{x})$, where $\phi(\mathbf{x})$ is a solution of a (discretized) state system

  so that each evaluation $f(\mathbf{x})$ requires
  - the solution of a (discretized) state system

- The surrogate $\widetilde{f}^{(k)}(\mathbf{x})$ for $f(\mathbf{x})$ could be defined in terms of a surrogate $\widetilde{\phi}^{(k)}(\mathbf{x})$ for $\phi(\mathbf{x})$,
  - e.g., $\widetilde{f}^{(k)}(\mathbf{x}) = F(\widetilde{\phi}^{(k)}(\mathbf{x}), \mathbf{x})$

  so that now an evaluation of each surrogate functional $\widetilde{f}^{(k)}(\mathbf{x})$ requires
  - the solution of a system for the surrogate state $\widetilde{\phi}^{(k)}(\mathbf{x})$

- There many other ways to define surrogate functionals and constraints

- The idea, of course, is to arrange things so that

  solving for the surrogate state $\widetilde{\phi}^{(k)}(\mathbf{x})$

  is much cheaper than

  solving for the state $\phi(\mathbf{x})$

  so that

  the evaluation of the surrogate functional $\widetilde{f}^{(k)}(\mathbf{x})$

  is much cheaper than

  evaluating the functional $f(\mathbf{x})$

- For example, the surrogate state could be determined through the use of a reduced-order model

- Response surfaces are another class of surrogate models; for example:

- In each trust region, the surrogate functional $\widetilde{f}^{(k)}(\mathbf{x})$ is chosen to be a quadratic polynomial approximation (e.g., an interpolant or a least-squares approximation) of the functional $f(\mathbf{x})$

- Since in our applications we have that $f(\mathbf{x}) = F(\phi(\mathbf{x}), \mathbf{x})$, we have that for each $k = 0, 1, 2, \ldots, k_{max}$, building the response surface requires

  1. choosing a set of points $\{\mathbf{x}_j\}_{j=1}^{J}$ in the trust region $\square_k$

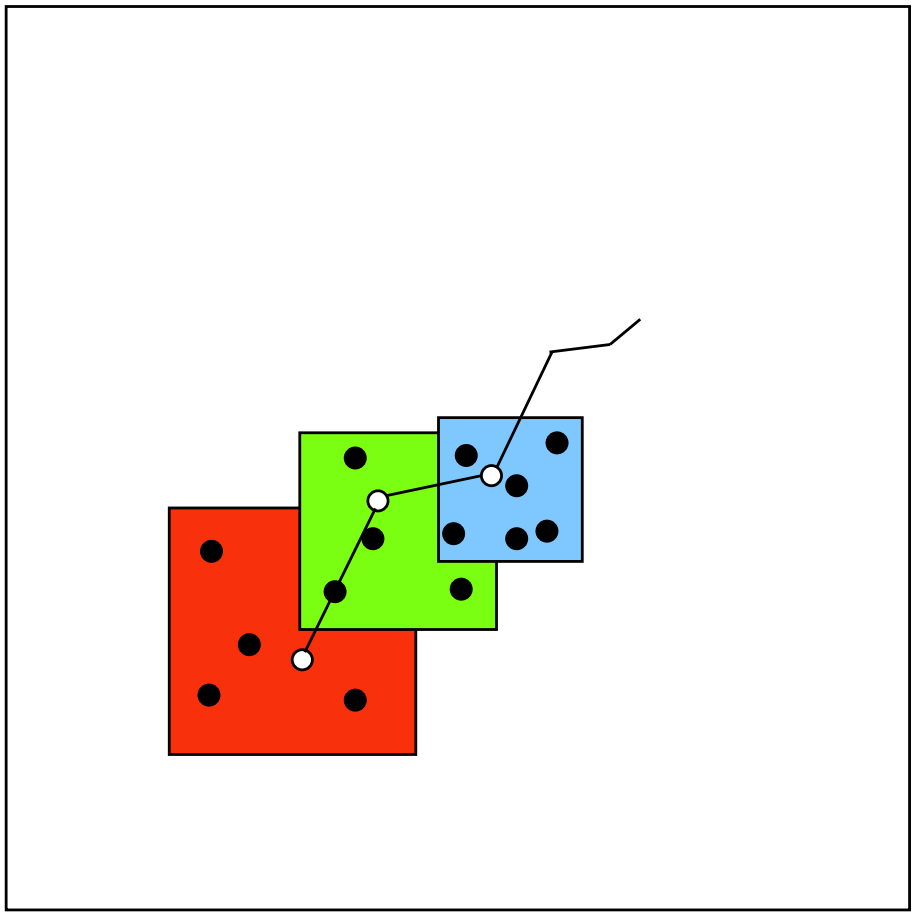
  2. determining the state $\phi(\mathbf{x})$ or a surrogate state $\widetilde{\phi}(\mathbf{x})$ at each of the points $\mathbf{x}_j$, $j = 1, \ldots, J$, chosen in step 1

  3. using the data from steps 1 and 2 to approximate the function $f(\mathbf{x})$ by a quadratic polynomial

- The quadratic polynomial built in step 3 is the surrogate model for the functional $f(\mathbf{x})$

- One then defines the next guess for the minimizer of $f(\mathbf{x})$ to be the minimizer the the quadratic polynomial within the trust region

- One then updates the trust region



● Points sampled in each trust region

○ Minimizer of surrogate in each trust region

Path to approximate optimum

- Thus, the sampling of points within the trust regions is a key element to this approach

- This approach is implemented in the DAKOTA suite of codes developed at Sandia National Laboratories (M. Eldred and A. Guinta); it has been found to be effective for problems with "badly behaved" functionals
  - Latin hypercube sampling is used for determining the points within each trust region
  - no surrogate for the state is used, i.e., the full (discrete) state system is used to evaluate the functional at each of the sample points

REDUCED-ORDER MODELING

THE NEED FOR REDUCING THE COST OF STATE SIMULATIONS

- Solutions of (nonlinear) complex systems are expensive with respect to both storage and CPU costs

- As a result, it is difficult if not impossible to deal with
  - optimization and control problems (multiple state solutions)
  - feedback control settings (real-time state solutions)

- Not surprinsingly, a lot of attention has been paid to reducing the costs of the nonlinear state solutions by using reduced-order models for the state
  - these are low-dimensional approximations to the state
  - reduced-order modeling approaches related to our work:
    proper orthogonal decomposition (POD)
    centroidal Voronoi tessellations (CVT)

REDUCED-ORDER MODELING

- For a state simulation, a reduced-order method would proceed as follows
  - one chooses a reduced basis $\mathbf{u}_i$, $i = 1, \ldots, n$

    n is hopefully very small compared to the usual number of functions used in a finite element approximation or the number of grid points used in a finite difference approximation

  - next, one seeks an approximation $\widetilde{\mathbf{u}}$ to the state of the form

$$\widetilde{\mathbf{u}} = \sum_{i=1}^{n} c_i \mathbf{u}_i \in V \equiv \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$$

  - then, one determines the coefficients c_i, $i = 1, \ldots, n$, by solving the state equations in the set V

    e.g., one could find a Galerkin solution of the state equations in a standard way, using V for the space of approximations

  - the cost of such a computation would be very small if n is small (ignoring the cost of the off-line determination of the reduced basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$)

Does reduced-order modeling work?

- It is clear that reduced-order methods should work in an interpolatory setting

- What happens in an extrapolatory setting is not so clear
  - it is obvious that if the reduced set V does not contain a good approximation to the solution one is trying to obtain, then one cannot hope to successfully determine that solution

SNAPSHOT SETS

- The state of a complex system is determined by parameters that appear in the specification of a mathematical model for the system

- Of course, the state of a complex system also depends on the independent variables appearing in the model

- Snapshot sets consist of state solutions corresponding to several parameter values and/or evaluated at several values of one or more of the dependent variables
  - steady-state solutions corresponding to several sets of design parameters
  - a time-dependent state solution for a fixed set of design parameter values evaluated at several time instants during the evolution process
  - several state solutions corresponding to different sets of parameter values evaluated at several time instants during the evolution process

- Snapshot sets are often determined by solving the full, very large-dimensional discretized system obtained by, e.g., a finite volume or finite element discretization

- Experimental data have also been used to determine a snapshot set

- Snapshots sets often contain "redundant" information
  - therefore, snapshot sets must usually be post-processed to remove as much of the redundancy as possible before they can be used for reduced-order modeling
  - POD and CVT may be viewed as simply different ways to post-process snapshot sets

- Since snapshot sets are the underpinning for most usages of POD (and also for the envisioned usages of CVT), we briefly discuss how they are generated in practice

Generating snapshot sets

- At this time, the generation of snapshot sets is an art and not a science
  - in fact, it is a rather primitive art

- The generation of snapshot sets is an exercise in the design of experiments
  - e.g., for stationary systems, how does one choose the sets of parameters at which the state (and sensitivities) are to be calculated (using expensive, high-fidelity computations) in order to generate the snapshot set?
  - clearly, some a priori knowledge about the types of states to be simulated or optimized using the reduced-order model is very useful in this regard
  - the large body of statistics literature on the design of experiments has not been used in a systematic manner

- For time-dependent systems, many (ad hoc) measures have been invoked in the hope that they will lead to good snapshot sets
  - time-dependent parameters (e.g., in boundary conditions) are used to generate states that are "rich" in transients
    - even if the state of interest depends only on time-independent parameters
  - in order to generate even "richer" dynamics, impulsive forcing is commonly used, e.g.,
    - starting the evolution impulsively with different strength impulses
    - introducing impulses in the middle of a simulation

- In the future, a great deal of effort needs to be directed towards developing and justifying methodologies for generating good snapshot sets
  - after all, a POD or CVT or CVOD basis is only as good as the snapshot set used to generate it

    if it ain't in the snapshot set, it ain't in the POD basis

PROPER ORTHOGONAL DECOMPOSITION (POD)

- Given n snapshots $\widetilde{\mathbf{x}}_j \in \mathbb{R}^N$, $j = 1, \ldots, n$, set

$$\mathbf{x}_j = \widetilde{\mathbf{x}}_j - \widetilde{\boldsymbol{\mu}}\,, \qquad j = 1 \ldots, n, \qquad \text{where} \qquad \widetilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{j=1}^{n} \widetilde{\mathbf{x}}_j$$

  the set $\{\mathbf{x}_j\}_{j=1}^n$ are the modified snapshots

- Let $d \leq n$

- Then, the POD basis $\{\boldsymbol{\phi}_i\}_{i=1}^d$ of cardinality d is found by successively solving, for $i = 1, \ldots, d$, the problem

$$\lambda_i = \max_{|\boldsymbol{\phi}_i|=1} \frac{1}{n}\sum_{j=1}^{n} |\boldsymbol{\phi}_i^T \mathbf{x}_j|^2 \qquad \text{and} \qquad \boldsymbol{\phi}_i^T \boldsymbol{\phi}_\ell = 0 \quad \text{for } \ell \leq i-1$$

- Let A denote the snapshot matrix $N \times n$ matrix whose columns are the modified snapshots $\mathbf{x}_j$, i.e.,

$$A = \left(\mathbf{x}_1 \ , \ \mathbf{x}_2 \ , \ \ldots \ , \ \mathbf{x}_n\right) = \left(\widetilde{\mathbf{x}}_1 - \widetilde{\boldsymbol{\mu}} \ , \ \widetilde{\mathbf{x}}_2 - \widetilde{\boldsymbol{\mu}} \ , \ \ldots \ , \ \widetilde{\mathbf{x}}_n - \widetilde{\boldsymbol{\mu}}\right)$$

- Let K denote the the $n \times n$ (normalized) correlation matrix for the modified snapshots, i.e.,

$$K_{j\ell} = \frac{1}{n}\mathbf{x}_j^T \mathbf{x}_\ell \qquad \text{or} \qquad K = \frac{1}{n}A^T A$$

- Let $\boldsymbol{\chi}_i$ with $|\boldsymbol{\chi}_i| = 1$ denotes the eigenvector corresponding to the i-th largest eigenvalue λ_i of K

- Then, the POD basis is given by $\qquad \boldsymbol{\phi}_i = \dfrac{1}{\sqrt{n\lambda_i}} A\boldsymbol{\chi}_i$

- The POD basis is orthonormal, i.e.,

$$\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j = 0 \quad \text{for } i \neq j \qquad \text{and} \qquad \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i = 1$$

- POD is closely related to the statistical method known as Karhunen-Loève analysis or the method of empirical orthogonal eigenfunctions or principal component analysis

- POD is also closely related to the singular value decomposition (SVD) of the modified snapshot matrix A

  – let $A = U\Sigma V^T$ denote the SVD of A

  – then,

  $$\sigma_i^2 = n\lambda_i \qquad \text{for } i = 1, \ldots, n$$

  where $\sigma_i =$ the i-th singular value of A

  $$\lambda_i = \text{the } i\text{-th largest eigenvalue of } K = \frac{1}{n}A^T A$$

  – the POD basis vectors are the first n left singular vectors of the snapshot matrix A

  $$\boldsymbol{\phi}_i = \mathbf{u}_i \qquad \text{for } i = 1, \ldots, n$$

- The POD basis is optimal in the following sense
  - let $\{\boldsymbol{\psi}_i\}_{i=1}^n$ denote an arbitrary orthonormal basis for the span of the modified snapshot set $\{\mathbf{x}_j\}_{j=1}^n$
  - let $P_{\psi,d}\mathbf{x}_j$ denote the projection of the modified snapshot $\mathbf{x}_j$ onto the d-dimensional subspace spanned by $\{\boldsymbol{\psi}_i\}_{i=1}^d$
  - clearly we have, for each $j = 1, \ldots, n$,

$$P_{\psi,d}\mathbf{x}_j = \sum_{i=1}^{d} c_{ji}\boldsymbol{\psi}_i \qquad \text{where} \qquad c_{ji} = \boldsymbol{\psi}_i^T \mathbf{x}_j \ \text{ for } i = 1, \ldots, d$$

  - let the error be defined by

$$\mathcal{E} = \sum_{j=1}^{n} |\mathbf{x}_j - P_{\psi,d}\mathbf{x}_j|^2$$

  - then, the minimum error is obtained when $\boldsymbol{\psi}_i = \boldsymbol{\phi}_i$ for $i = 1, \ldots, d$, i.e., when the $\boldsymbol{\psi}_i$'s are the POD basis vectors

- The connection between POD and SVD makes it is easy to show that the error of the d-dimensional POD subspace is given by

$$\mathcal{E}_{\text{pod}} = \sum_{j=d+1}^{n} \sigma_j^2 = n \sum_{j=d+1}^{n} \lambda_j$$

n = number of snapshots
d = dimension of the POD subspace

- If one wishes for the relative error to be less than a prescribed tolerance δ, i.e., if one wants

$$\mathcal{E}_{\text{pod}} \leq \delta \sum_{j=1}^{n} |\mathbf{x}_j|^2$$

one should

choose d to be the smallest integer such that

$$\frac{\sum_{j=1}^{d} \sigma_j^2}{\sum_{j=1}^{n} \sigma_j^2} = \frac{\sum_{j=1}^{d} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \geq \gamma = 1 - \delta$$

Variations on POD

- There have been several variations introduced in attempts to "improve" POD

- Weighted POD – gives more weight to some members of the snapshot set
  - can be accomplished, e.g., by inculding multiple copies of an "important" snapshot in the snapshot set

- POD with derivatives – get more information into the snapshot set in order to get a "better" POD basis
  - add derivatives or numerical approximations to derivatives (especially time derivatives) of simulated states to the snapshot set

- H^1 POD – change the error measure for POD in order to get a "better" POD basis
  - use H^1 norms and inner products (instead of L^2) in the definition and construction of POD bases

- Constrained POD – impose a constraint (e.g., symmetry) on the POD basis
- Adaptive POD – change the POD basis when it no longer seems to be working
  - requires detection of failure of the POD basis
  - requires the determination of new snapshot vectors
  - requires the solution of the eigenvalue problem for the new correlation matrix determined from the new snapshot vectors

CENTROIDAL VORONOI TESSELLATIONS (CVT)

Voronoi tessellations

- Given a set of modified snapshots $W = \{\mathbf{x}_j\}_{j=1}^n$ belonging to $\mathbb{R}^N$, a set $\{V_i\}_{i=1}^k$ is a tessellation of W if

$$\left.\begin{array}{l} V_i \subset W \quad \text{for } i = 1, \ldots, k \\ V_i \cap V_j = \emptyset \quad \text{for } i \neq j \\ \cup_{i=1}^k V_i = W \end{array}\right\} \quad \{V_i\}_{i=1}^k \text{ is a subdivision of } W \text{ into disjoint, covering subsets}$$

- Given a set of points $\{\mathbf{z}_i\}_{i=1}^k$ belonging to $\mathbb{R}^N$ (but not necessarily to W) the Voronoi region corresponding to the point $\mathbf{z}_i$ is defined by

$$\widehat{V}_i = \left\{ \mathbf{x} \in W \quad : \quad |\mathbf{x} - \mathbf{z}_i| \leq |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k, \ j \neq i, \right.$$
$$\left. \text{where equality holds only for } i < j \right\}$$

- The set $\{\widehat{V}_i\}_{i=1}^k$ is called a Voronoi tessellation or Voronoi diagram of W corresponding to the set of points $\{\mathbf{z}_i\}_{i=1}^k$

- The points in the set $\{\mathbf{z}_i\}_{i=1}^k$ are called the generators of the Voronoi diagram $\{\widehat{V}_i\}_{i=1}^k$ of W

Centroids

- Given a density function $\rho(\mathbf{y}) \geq 0$, defined for $\mathbf{y} \in W$, the mass centroid $\mathbf{z}^*$ of any subset $V \subset W$ is defined by

$$\sum_{\mathbf{y} \in V} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}^*|^2 = \inf_{\mathbf{z} \in V^*} \sum_{\mathbf{y} \in V} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}|^2 \,,$$

  where

  – the sums extend over the points belonging to V

  – the set V^* can be taken to be V or it can be an even larger set such as all of $\mathbb{R}^N$

- in case $V^* = \mathbb{R}^N$, $\mathbf{z}^*$ is the ordinary mean

$$\mathbf{z}^* = \frac{\sum_{\mathbf{y} \in V} \rho(\mathbf{y}) \mathbf{y}}{\sum_{\mathbf{y} \in V} \rho(\mathbf{y})}$$

  – in this case, $\mathbf{z}^* \notin W$ in general

Centroidal Voronoi tessellations

- If

$$\mathbf{z}_i = \mathbf{z}_i^* \qquad \text{for } i = 1, \ldots, k$$

where

$\{\mathbf{z}_i\}_{i=1}^k$ is the set of generating points of the Voronoi tessellation $\{\widehat{V}_i\}_{i=1}^k$

$\{\mathbf{z}_i^*\}_{i=1}^k$ is the set of mass centroids of the Voronoi regions $\{\widehat{V}_i\}_{i=1}^k$

we refer to the Voronoi tessellation as being a

centroidal Voronoi tessellation

or CVT for short

- The concept of CVT's can be extended to
  - more general sets, including regions in Euclidean spaces
  - more general metrics

- CVT's are optimal in the following sense
  - given the discrete set of points $W = \{\mathbf{x}_j\}_{j=1}^n$ belonging to $\mathbb{R}^N$, we define the error of a tessellation $\{V_i\}_{i=1}^k$ of W and a set of points $\{\mathbf{z}_i\}_{i=1}^k$ belonging to $\mathbb{R}^N$ by

$$\mathcal{F}\big((\mathbf{z}_i, V_i), i = 1, \ldots, k\big) = \sum_{i=1}^{k} \sum_{\mathbf{y} \in V_i} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2$$

  - then, it can be shown that a necessary condition for the error $\mathcal{F}$ to be minimized is that the pair $\{\mathbf{z}_i, V_i\}_{i=1}^k$ form a CVT of W

- CVT's are optimal in the following sense
  - given the discrete set of points $W = \{\mathbf{x}_j\}_{j=1}^{n}$ belonging to $\mathbb{R}^N$, we define the error of a tessellation $\{V_i\}_{i=1}^{k}$ of W and a set of points $\{\mathbf{z}_i\}_{i=1}^{k}$ belonging to $\mathbb{R}^N$ by

$$\mathcal{F}\big((\mathbf{z}_i, V_i), i = 1, \ldots, k\big) = \sum_{i=1}^{k} \sum_{\mathbf{y} \in V_i} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2$$

  - then, it can be shown that a necessary condition for the error $\mathcal{F}$ to be minimized is that the pair $\{\mathbf{z}_i, V_i\}_{i=1}^{k}$ form a CVT of W

- CVT's of discrete sets are closely related to optimal k-means clusters so that Voronoi regions and centroids can be referred to as clusters and cluster centers, respectively
  - the error $\mathcal{F}$ is also often referred to as the variance, cost, distortion error, or mean square error

Algorithms for constructing CVT's

- There are several algorithms known for constructing centroidal Voronoi tessellations of a given set
  - Lloyd's method is a deterministic algorithm which is the obvious iteration between computing Voronoi diagrams and mass centroids, i.e., a given set of generators is replaced in an iterative process by the mass centroids of the Voronoi regions corresponding to those generators
  - MacQueen's method, a very elegant probabilistic algorithm which divides sampling points into k sets or clusters by taking means of clusters
  - Various other methods based on the minimization properties of CVT's
  - We have developed a new probabilistic method which may be viewed as a generalization of both the MacQueen and Lloyd methods and is amenable to efficient parallelization.

CVT's and model reduction

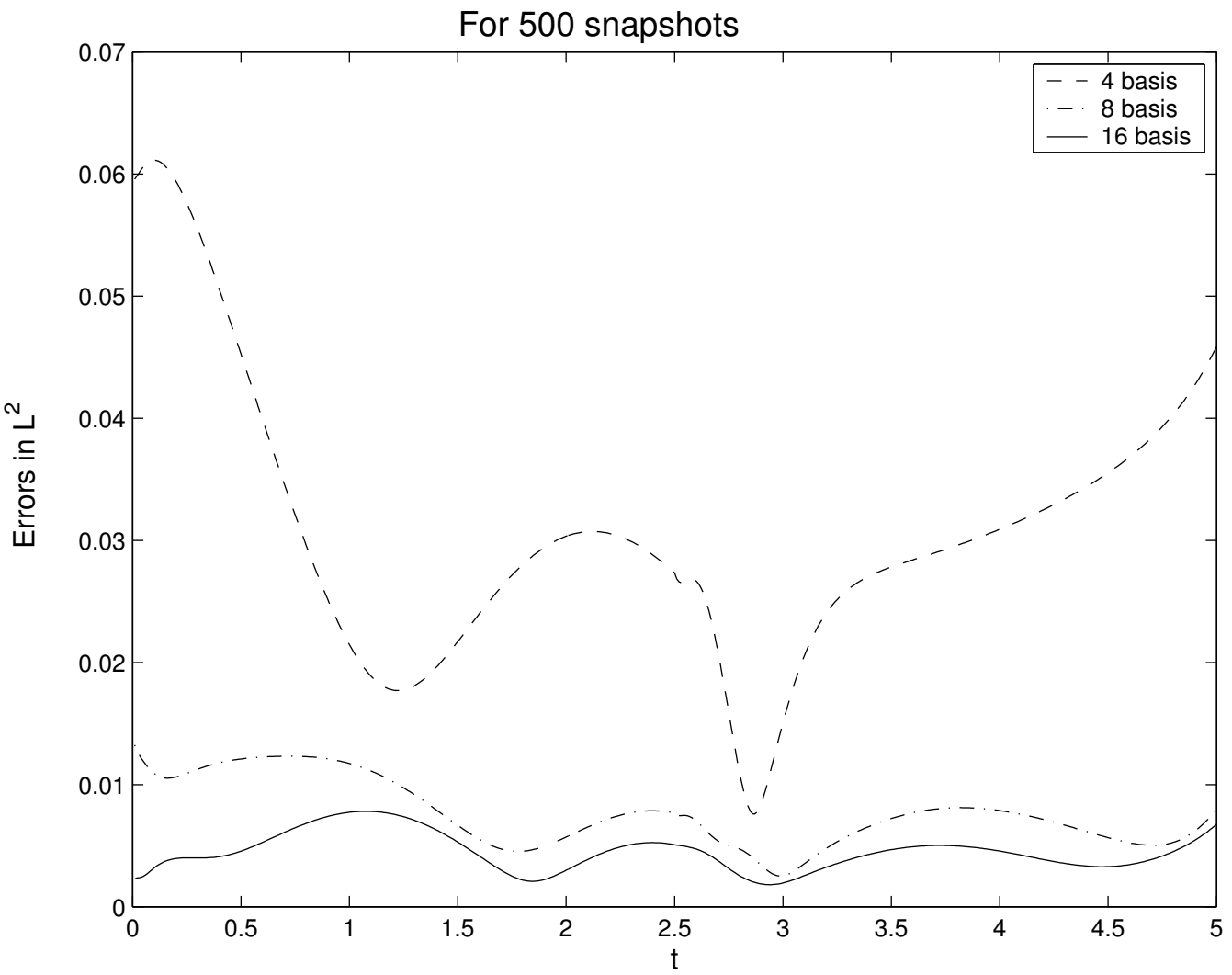
- CVT's have been successfully used in data compression; one particular application was to image reconstruction
  - therefore, it is natural to examine CVT's in another data compression setting, namely reduced-order modeling

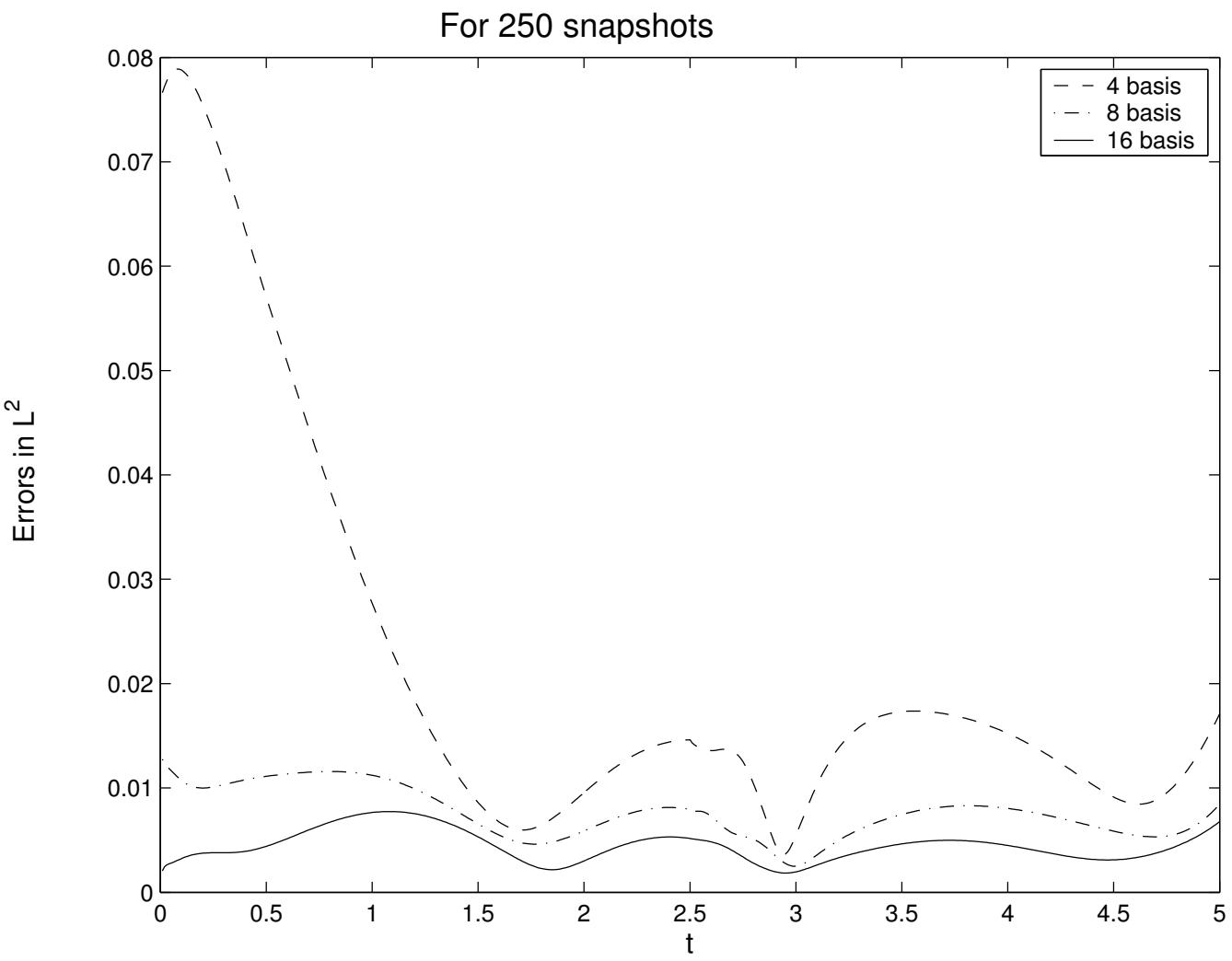
- The idea, just as it is in the POD setting, is to extract, from a given set of (modified) snapshots $\{\mathbf{x}_j\}_{j=1}^n$ of vectors in $\mathbb{R}^N$, a smaller set of vectors also belonging to $\mathbb{R}^N$
  - in the POD setting, the reduced set of vectors was the d-dimensional set of POD vectors $\{\boldsymbol{\phi}_j\}_{i=1}^d$
  - in the CVT setting, the reduced set of vectors is the k-dimensional set of vectors $\{\mathbf{z}_k\}_{i=1}^k$ that are the generators of a centroidal Voronoi tessellation of the set of modified snapshots

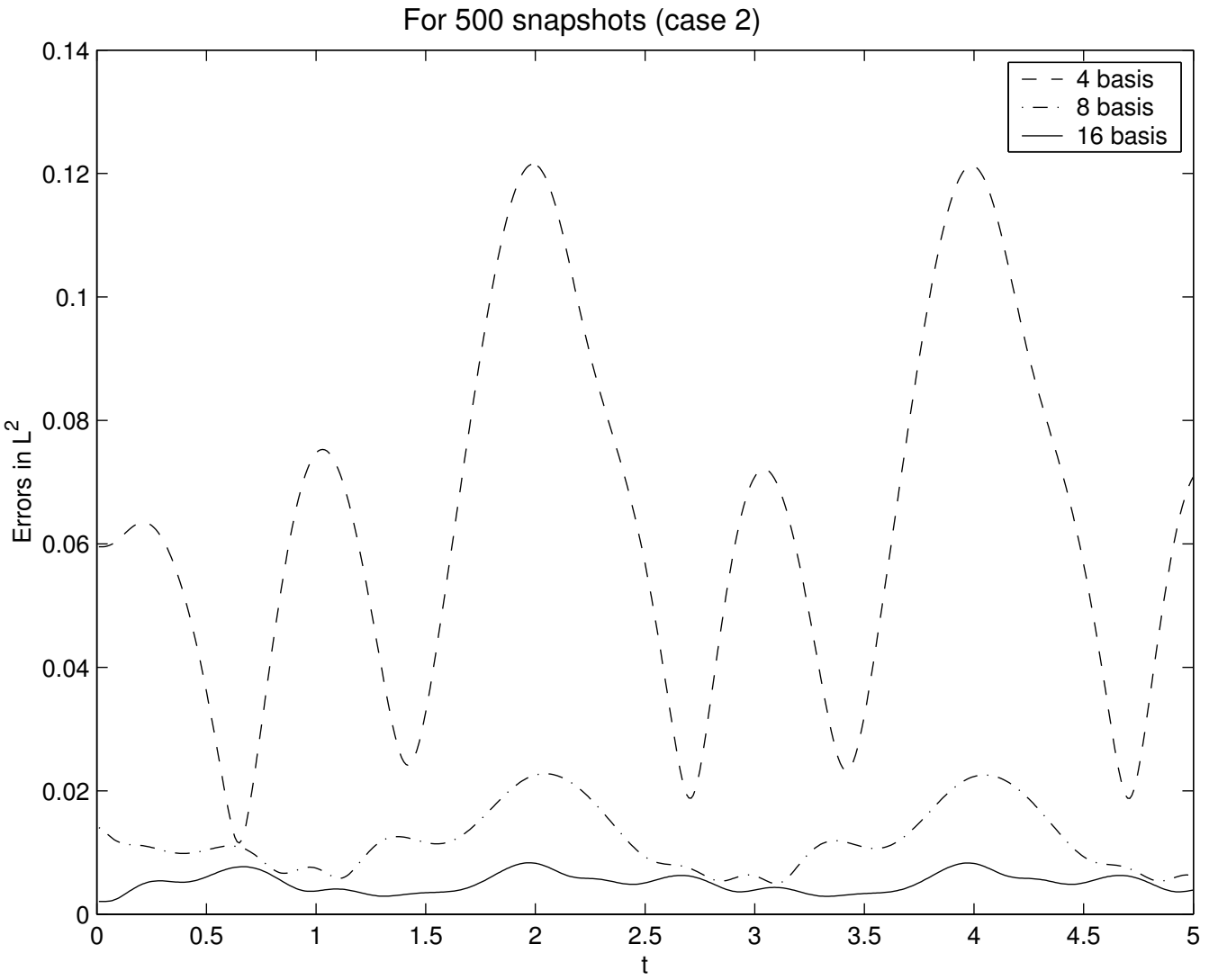
- Just as POD produced an optimal reduced basis in the sense that the error $\mathcal{E}$ is minimized, CVT produces an optimal reduced basis in the sense that the error $\mathcal{F}$ is minimized

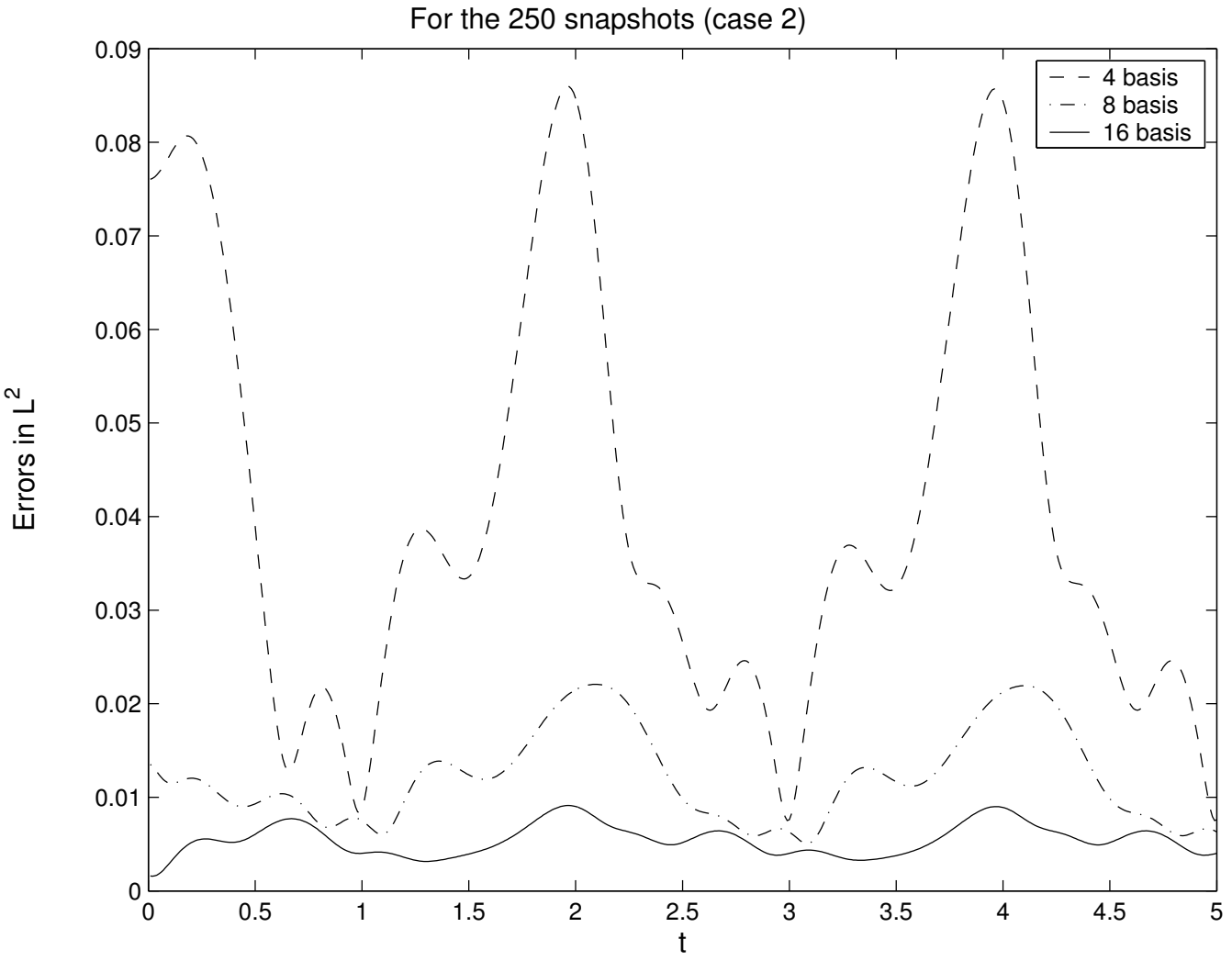
- One can, in principle, determine the dimension d of an effective POD basis, e.g., using the eigenvalues of the correlation matrix
  - similarly, one can, in principle, determine the dimension k of an effective CVT basis by examining the (computable) error $\mathcal{F}(\cdot)$

COMPUTATIONAL RESULTS FOR CVT MODEL REDUCTION




For 250 snapshots
4 basis
8 basis
16 basis
Errors in L2
t
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


For 500 snapshots (case 2)
Errors in L2
t
4 basis
8 basis
16 basis
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


For the 250 snapshots (case 2)
Errors in L2
t
4 basis
8 basis
16 basis
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

CVT VS. POD

- Question: why should one use CVT instead of POD?
  - although justifications have to be substantiated through analyses and extensive further numerical experiments, one can make some arguments

- CVT naturally introduces the concept of clustering into the construction of the reduced basis

- CVT is "cheaper" than POD
  - POD involves the solution of an $n \times n$ eigenproblem, where n is the number of snapshots
  - CVT requires no eigenproblem solution
    - CVT can handle many more snapshots
    - adaptively changing the reduced basis should be less expensive with CVT

CVT COMBINED WITH POD (CVOD)

- We have already mentioned that the concept of centroidal Voronoi tessellations can be extended to more general notions of distance
  - this allows us to combine POD and CVT to (hopefully) take advantage of the best features of both approaches.

- Why should one use CVOD instead of POD or CVT?
  - CVOD offers the possibility of taking advantage of the best features of both POD and CVT
  - CVOD is cheaper than POD since it requires the solution of several smaller eigenproblems instead of one large one