Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning

Stuttgart, May 2011

a Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
{dihlmann, haasdonk}@mathematik.uni-stuttgart.de
www.ians.uni-stuttgart.de/agh

b Institute of Computational and Applied Mathematics, University of Münster, Einsteinstr. 62, 48149 Münster, Germany
martin.drohmann@math.uni-muenster.de

Abstract
Modern simulation scenarios require real-time or many query responses from a simulation model. This is the driving force for increased efforts in model order reduction for high dimensional dynamical systems or partial differential equations. This demand for fast simulation models is even more critical for parametrized problems. There exist several snapshot-based methods for model order reduction of parametrized problems, e.g. proper orthogonal decomposition (POD) or reduced basis (RB) methods. An often faced problem is that the produced reduced models for a given accuracy tolerance are still of too high dimension. This is especially the case for evolution problems where the model shows high variability during time evolution. We will present an approach to gain control over the online complexity of a reduced model by an adaptive time domain partitioning. Thereby we can prescribe simultaneously a desired error tolerance and a limiting size of the dimension of the reduced model. This leads to fast and accurate reduced models. The method will be applied to an advection problem.

Keywords
model order reduction; reduced basis method; evolution problem; parametrized partial differential equation; adaptive time partitioning

A-posteriori error estimation for DEIM reduced nonlinear dynamical systems

at the
PN7 Network Seminar
SRC SimTech, University of Stuttgart

Dipl. Math. Dipl. Inf. Daniel Wirtz
October 31, 2012

Dipl. Math. Dipl. Inf. Daniel Wirtz, IANS, University of Stuttgart
Prof. Dr. D. Sorensen, CAAM, Rice University
Jun.-Prof. Dr. Bernard Haasdonk, IANS, University of Stuttgart
daniel.wirtz@mathematik.uni-stuttgart.de
Introduction

The Menu today

- Introduction
  - DEIM
  - Dynamical Systems
  - Model Order Reduction (MOR)
  - The Experiment!
- DEIM approximation error estimation
- MOR error estimation
- Matrix DEIM and Partial Similarity Transformations
- Discussion
Notation

- Scalars $a, x, n$, vectors $y, z$ and matrices $A, V$

DEIM: Quick ’n Dirty

- Have nonlinear function/discrete Operator $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$
- $m$-th order DEIM approximation of $f$:

$$\tilde{f}_m(y) := U_m (P_m^T U_m)^{-1} P_m^T f(y)$$

- $U_m = [u_1, \ldots, u_m]$ $m$-basis
- $(P_m^T U_m)^{-1}$ combination weights
- $P_m = [e_{g1}, \ldots, e_{gm}]$ component selection
- $e_i \in \mathbb{R}^d$ denotes the $i$-th unit vector in $\mathbb{R}^d$

Details: [Chaturantabut & Sorensen(2010)]
Introduction

Dynamical Systems

Base considered dynamical system structure

\[ y'(t) = f(y(t)), \quad y(0) = y_0 \]  

- System state \( y(t) \in \mathbb{R}^d \) for \( t \in [0, T] \)
- Initial state \( y_0 \in \mathbb{R}^d \)

More complex settings

Concepts readily extendable by time-/parameter affine components

- Initial values \( y_0(\mu) \in \mathbb{R}^d \)
- Inputs \( u(t) \in \mathbb{R}^k, \quad B(t, \mu) = \sum_{i=1}^{Q_B} \theta^B_i(\mu) B_i, \quad B_i \in \mathbb{R}^{d \times k} \)
- Output \( C(t, \mu) = \sum_{i=1}^{Q_C} \theta^C_i(\mu) C_i, \quad C_i \in \mathbb{R}^{l \times d} \)
- See Experiment!
Petrov-Galerkin projection of the full system (1)

\[ z'(t) = W^T \tilde{f}_m(Vz(t)) = \tilde{U}_m P_m^T f(Vz(t)) \]

\[ z(0) = z_0 := W^T y_0, \]

- Reduced variable \( z(t) \in \mathbb{R}^r \) at times \( t \in [0, T] \)
- Biorthogonal matrices \( V, W \in \mathbb{R}^{r \times d}, V^T W = I_r \) with \( r \leq d \) (ideally \( r \ll d \))
- \( \tilde{U}_m := W^T U_m (P_m^T U_m)^{-1} \)
- Note: We have Galerkin case \( V = W \) in experiments.
Introduction

Experiment: 1D unsteady viscous Burgers’ equation

- Domain $\Omega := [0, 1]$ and time $t \in [0, T]$ with $T = 1$
- Governing system with parameter $\mu \in \mathcal{P} := [0.01, 0.06]$

$$\frac{\partial y}{\partial t}(x,t) = \mu \frac{\partial^2 y}{\partial x^2}(x,t) - \frac{\partial}{\partial x} \left( \frac{y(x,t)^2}{2} \right) + \langle \beta(x), u(t) \rangle,$$

- Boundary conditions

$$y(0,t) = y(1,t) = 0, \quad t \in [0, T], \quad y(x,0) = 0, \quad x \in \Omega.$$

- External forces $u(t)$ at locations $\beta(x)$:

$$u_1(t) = \sin(2\pi t), \quad \quad b_1(x) = \begin{cases} 4e^{-\left(\frac{x-0.2}{0.03}\right)^2} & x \in [0.1, 0.3], \\ 0 & \text{else}, \end{cases}$$

$$u_2(t) = \begin{cases} 1 & t \in [0.2, 0.4], \\ 0 & \text{else}, \end{cases} \quad \quad b_2(x) = \begin{cases} 4 & x \in [0.6, 0.7], \\ 0 & \text{else}. \end{cases}$$
Introduction
Experiment implementation

- Finite differences over a $n = 500$ grid gives dynamical system

$$\frac{d}{dt} y(t) = \mu A y(t) + f(y(t)) + B u(t)$$

- Discrete Laplacian $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times 2}$
- $f(y) = -y \cdot A x y$ with 1st-order discrete diff. operator $A x \in \mathbb{R}^{n \times n}$
- Time integration with semi-implicit Euler scheme

$$(I_n - \Delta t \mu A) y(t_{i+1}) = y(t_i) + \Delta t \left( f(y(t_i)) + B u(t_i) \right)$$

- $n_t = 100$ equidistant time-steps $t_i := (i - 1) \Delta t$, $\Delta t = \frac{T}{n_t - 1}, i = 1 \ldots n_t$
- Training for 100 log-spaced parameters
- $V = W$ by POD on trajectories, incl. $f(x_i)$ values and $\text{span} \{B\}$
Introduction

Experiment plots

Figure: Simulation results for both ends of the parameter range $\mathcal{P}$ and their difference

Figure: Full (left), reduced simulation (middle) and the absolute error (right) for $m = 12, \mu = 0.04$
DEIM approximation error estimation
First central question: How to measure $f - \tilde{f}$?

**Matrix decomposition lemma**

- $d, m, m' \in \mathbb{N}, m + m' \leq d$
- $U_m, P_m \in \mathbb{R}^{d \times m}, U_{m'}, P_{m'} \in \mathbb{R}^{d \times m'}$
- $U := [U_m \ U_{m'}], \ P := [P_m \ P_{m'}]$ lin. indep. col.
- $P_m^T U_m$ is nonsingular

Then

$$U(P^T U)^{-1} P^T = U_m (P_m^T U_m)^{-1} P_m^T$$

$$+ (U_m A^{-1} B - U_{m'}) F^{-1} (EP_m^T - P_m^T),$$

with suitable $A, B, C, D, E, F.$
DEIM approximation error estimation

Approximation error

**Theorem (DEIM error representation)**

- **DEIM basis** $\mathcal{U} := \{u_1, \ldots, u_M\}$, points $\mathcal{E} = \{\varrho_1, \ldots, \varrho_M\}$
- **Assume** $\tilde{f}_M \equiv f$ on $\Omega$. (exactness for $m = M$ !)
- $P_m := [e_{\varrho_1}, \ldots, e_{\varrho_m}]$, $P_{m'} := [e_{\varrho_m+1}, \ldots, e_{\varrho_{m+m'}}]$
- $U_m := [u_1, \ldots, u_m]$, $U_{m'} := [u_{m+1}, \ldots, u_{m+m'}]$

Then with $m \leq M - 1, m' := M - m$ the approximation error is

$$f(y) - \tilde{f}_m(y) = (U_m A^{-1} B - U_{m'}) F^{-1} (EP_m^T - P_{m'}^T) f(y),$$

$$= (M_1 P_m^T - M_2 P_{m'}^T) f(y),$$

with $M_1, M_2$ suitable.

- Already mentioned in [Barrault et al.(2004)Barrault, Maday, Nguyen & Patera], used with $m' > 1$ in [Drohmann et al.(2012)Drohmann, Haasdonk & Ohlberger]
DEIM approximation error estimation

Experiments

- Nonlinearity $f$ of Burgers equation
- Approximation errors over DEIM training data $Y$, $|Y| = 10100$
- Full dimension $d = 500$, max. DEIM order $M = 200$
- Using $L^2$ in state space and $L^\infty$ norm over $Y$

Figure: True (left) and estimated (right) absolute DEIM approximation errors for different orders/error orders
DEIM approximation error estimation

Experiments

<table>
<thead>
<tr>
<th>$m$</th>
<th>$0.1$</th>
<th>$0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$14/4$</td>
<td>$25/14$</td>
</tr>
<tr>
<td>8</td>
<td>$14/4$</td>
<td>$25/13$</td>
</tr>
<tr>
<td>16</td>
<td>$15/6$</td>
<td>$21/15$</td>
</tr>
<tr>
<td>24</td>
<td>$13/5$</td>
<td>$23/13$</td>
</tr>
<tr>
<td>32</td>
<td>$12/5$</td>
<td>$19/12$</td>
</tr>
<tr>
<td>40</td>
<td>$13/6$</td>
<td>$20/11$</td>
</tr>
<tr>
<td>48</td>
<td>$12/6$</td>
<td>$20/12$</td>
</tr>
<tr>
<td>56</td>
<td>$16/4$</td>
<td>$21/12$</td>
</tr>
<tr>
<td>64</td>
<td>$13/3$</td>
<td>$28/13$</td>
</tr>
<tr>
<td>72</td>
<td>$18/5$</td>
<td>$28/16$</td>
</tr>
<tr>
<td>80</td>
<td>$18/8$</td>
<td>$32/17$</td>
</tr>
<tr>
<td>88</td>
<td>$21/4$</td>
<td>$33/20$</td>
</tr>
<tr>
<td>96</td>
<td>$23/12$</td>
<td>$33/22$</td>
</tr>
<tr>
<td>104</td>
<td>$20/5$</td>
<td>$32/18$</td>
</tr>
<tr>
<td>112</td>
<td>$17/7$</td>
<td>$36/17$</td>
</tr>
<tr>
<td>120</td>
<td>$23/6$</td>
<td>$32/21$</td>
</tr>
</tbody>
</table>

Figure: Maximum relative error $||f - \tilde{f} - (M_1 P_{m}^T - M_2 P_{m'}^T) f|| ||f - \tilde{f}||^{-1}$ between true and estimated DEIM approximation error over $Y$. Transparent plane located at $10^{-2}$. Table: Req. min $m'$-values for max/avg rel. err smaller than column header value.
DEIM approximation error estimation

Experiments

Figure: Minimum required $m'$ values for relative error $< 0.1$ (left) and $< 0.01$ (right) for different discretizations.
MOR error estimation

MoRePaS

MOR error estimation
Let $G \in \mathbb{R}^{d \times d}$ denote a symmetric positive definite weighting matrix.

**Definition (Logarithmic Lipschitz constants)**

For a function $f : \mathbb{R}^d \to \mathbb{R}^d$ we define the logarithmic Lipschitz constant with respect to $G$ by

$$L_G[f] := \lim_{h \to 0^+} \frac{1}{h} \left( \sup_{x \neq y \in \mathbb{R}^d} \frac{\|x - y + h(f(x) - f(y))\|_G}{\|x - y\|_G} - 1 \right).$$

For Lipschitz-continuous functions we have the equivalent

$$L_G[f] = \sup_{x \neq y \in \mathbb{R}^d} \frac{\langle x - y, f(x) - f(y) \rangle_G}{\|x - y\|^2_G}.$$ 

Background: [Dahlquist(1959), Söderlind(2006)]
MOR error estimation

Main estimation result

- \( y^r(t) := V z(t), \ e(t) := y(t) - y^r(t), \ t \in [0, T] \)
- Obtained by estimation of error system [incl. Comp. Lemma]

\[
e'(t) = f(y(t)) - VW^T \tilde{f}_m(y^r(t)), \quad e(0) = y_0 - VW^T y_0.
\]
MOR error estimation
Main estimation result

- \( y^r(t) := V z(t), \ e(t) := y(t) - y^r(t), \ t \in [0, T] \)
- Obtained by estimation of error system [incl. Comp. Lemma]

\[
e'(t) = f(y(t)) - VW^T \tilde{f}_m(y^r(t)), \quad e(0) = y_0 - VW^T y_0.
\]

Theorem (A-post. error estimation for DEIM reduced systems)

The state space error is rigorously bounded via

\[
\|e(t)\|_G \leq \Delta(t) \quad \forall t \in [0, T],
\]

with

\[
\Delta(t) := \int_0^t \alpha(s)e^{LG[f]}(t-s) \, ds + e^{LG[f]}t \left\| y_0 - VW^T y_0 \right\|_G,
\]

\[
\alpha(t) := \left\| M_1 P_m^T f(y^r(t), t) - M_2 P_{m'}^T f(y^r(t), t) \right\|_G,
\]

- \( M_1, M_2 \) suitable, \( \alpha(t) \) offline/online decomposable
MOR error estimation

Estimation results

- All estimations use “true local log. Lip. const.”

\[ L_G[f] \Rightarrow \frac{\langle e(t), f(y(t)) - f(y^r(t)) \rangle_G}{\|e(t)\|_G^2} \]

- Reference estimate: Uses true DEIM approximation error

Figure: Absolute errors over time for different DEIM \( m' \) approximation error orders, \( \mu = 0.04, m = 12 \)
MOR error estimation
More workarounds!

- Always: We don’t have \( y(t) \) at reduced simulations
- We most likely also don’t have \( L_G[f] \)

### Key idea with approximation and localization

\[
\frac{\langle e(t), f(y(t)) - f(y^r(t))\rangle}{\|e(t)\|_G^2} = \frac{\langle e(t), J(y^r(t))e(t) + O\left(\|e(t)\|_G^2\right)\rangle}{\|e(t)\|_G^2} \\
= \frac{\langle e(t), J(y^r(t))e(t)\rangle}{\|e(t)\|_G^2} + O\left(\|e(t)\|_G\right) \\
\leq L_G[J(y^r(t))] + O\left(\|e(t)\|_G\right).
\]

- Only \( y^r(t) \) required & localized
- Need Jacobian \( J \) and its logarithmic norm \( L_G[J] \)
MOR error estimation

Estimation results

- Now replace

\[
\left\langle e(t), f(y(t)) - f(y^r(t)) \right\rangle_G \Rightarrow L_G[J(y^r(t))]
\]

\[
\frac{\|e(t)\|_G^2}{\|e(t)\|_G}
\]

- Logarithmic norm for matrices: 

\[
L_G[A] = \max \left\{ \sigma \left( \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right) \right\}
\]

with \( \tilde{A} := C^T AC^{-T} \) and \( G = CC^T \).

Figure: Absolute errors over time for different DEIM \( m' \) approximation error orders, \( \mu = 0.04, m = 12 \)
Partial Similarity Transformations and Matrix DEIM

MoRePaS

Partial Similarity Transformations and Matrix DEIM
Partial Similarity Transformations and Matrix DEIM

Now the crazy stuff.. partial similarity transformation

Lemma (Approx. of eigenvalues for a family of sym. matrices)

- \( H(t) \in \mathbb{R}^{d \times d} \) fam. of sym. matrices over \( t \in [a, b] \)
- \([\lambda(t), q(t)] := \lambda_{max}(H(t))\) largest e-val \( \lambda(t) \) / normed e-vec \( q(t) \)
- \( \sup_{t \in [a,b]} \|H(t)\| \leq C_H \) holds
- Define \( R = \int_a^b q(t)q(t)^T dt \) and let \( Q\Sigma^2Q^T = R \) with \( Q^TQ = I \)
- Let \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d) \) with \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > 0 \)
- For \( k \leq d \) let \( Q_k := Q(:,1:k) \) and \( \lambda_k(t) := \lambda_{max}(Q_k^T H(t) Q_k) \)

Then
\[
\int_a^b |\lambda(t) - \lambda_k(t)| dt \leq 4C_H \sum_{j > k} \sigma_j^2.
\]
Partial Similarity Transformations and Matrix DEIM

Application of transformation to Jacobian $J$

- Consider

$$H(t) := \frac{1}{2} \left( C^T J(y^r(t)) C^{-T} + (C^T J(y^r(t)) C^{-T})^T \right)$$

- Have the corresponding values $C_H > 0$, $\{\sigma_i\}_{i=1}^d$, $Q \in \mathbb{R}^{d \times d}$

- Identify

$$\lambda(t) = L_G[J(y^r(t))],$$
$$\lambda_k(t) = L_{I_k}\left[ Q_k^T C^T J(y^r(t)) C^{-T} Q_k \right],$$

Jacobian partial similarity transform

$$\int_0^T \left| L_G[J(y^r(t))] - L_{I_k}\left[ Q_k^T C^T J(y^r(t)) C^{-T} Q_k \right] \right| dt \leq C_H \sum_{j > k} \sigma_j^2.$$
What do we have?

- Already done: Replaced $L_G[f]$ by $L_G[J(y^r(t))]$
  - Localized
  - Likely available

Reduced cost for eigenvalue problem: Size $d$ to size $k \ll d$ But possibly high offline cost

Still unanswered: $Q^T_k C T J(y^r(t)) C - T Q_k \in \mathbb{R}^{k \times k}$, but $J(y^r(t)) \in \mathbb{R}^{d \times d}$!
What do we have?

- **Already done**: Replaced $L_G[f]$ by $L_G[J(y^r(t))]$
  - Localized
  - Likely available

- **New**: Replaced $L_G[J(y^r(t))]$ by $L_{I_k}[Q_k^T C^T J(y^r(t)) C^{-T} Q_k]$
  - Reduced cost for eigenvalue problem: Size $d$ to size $k \ll d$
  - But possibly high offline cost
Partial Similarity Transformations and Matrix DEIM
Take a breath..

What do we have?

- **Already done:** Replaced $L_G[f]$ by $L_G[J(y^r(t))]$
  - Localized
  - Likely available

- **New:** Replaced $L_G[J(y^r(t))]$ by $L_{I_k}[Q_k^T C^T J(y^r(t)) C^{-T} Q_k]$
  - Reduced cost for eigenvalue problem: Size $d$ to size $k \ll d$
  - But possibly high offline cost

- **Still unanswered:** $Q_k^T C^T J(y^r(t)) C^{-T} Q_k \in \mathbb{R}^{k \times k}$, but $J(y^r(t)) \in \mathbb{R}^{d \times d}$!

- **We need some offline/online separability of $J$!**
For $A \in \mathbb{R}^{d \times d}$ define the transformation (vec-operation)

$$
\mathcal{V} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{d^2}
A \mapsto \mathcal{V}[A] := (A_1^T, A_2^T, \ldots, A_d^T)^T,
$$

Matrix DEIM

- Choose $M_J \leq d$
- Let $U_{M_J}, P_{M_J} \in \mathbb{R}^{d^2 \times M_J}$ be basis/point matrices for $\mathcal{V}[J(y)]$

Then, for $m_J \leq M_J$, the $m_J$-th order MDEIM approximation of $J$ is given via

$$
\tilde{J}_{m_J}(y) := \mathcal{V}^{-1} \left[ (U_{m_J}(P_{m_J} U_{m_J})^{-1} P_{m_J}^T \mathcal{V}[J(y)]) \right],
$$

where $U_{m_J} := U_{M_J}(; 1:m_J)$ and $P_{m_J} := P_{M_J}(; 1:m_J)$. 
Partial Similarity Transformations and Matrix DEIM

Matrix DEIM: Offline/online decomposition

off1 Compute $U_{M,J}$, $P_{M,J}$, $Q$ once, then for each new $m,J, k$ do
Partial Similarity Transformations and Matrix DEIM

Matrix DEIM: Offline/online decomposition

off1 Compute $U_{M_J}, P_{M_J}, Q$ once, then for each new $m_J, k$ do

off2 Compute offline vectors for Matrix DEIM of Jacobian via

$$
\hat{U}_{m_J} := U_{M_J}(\cdot, 1:m_J), \quad \hat{P}_{m_J} := P_{M_J}(\cdot, 1:m_J),
$$

$$
\hat{U} := \hat{U}_{m_J}(\hat{P}_{m_J}^T \hat{U}_{m_J})^{-1}.
$$
Partial Similarity Transformations and Matrix DEIM

Matrix DEIM: Offline/online decomposition

**off1** Compute $U_{M_J}, P_{M_J}, Q$ once, then for each new $m_J, k$ do

**off2** Compute offline vectors for Matrix DEIM of Jacobian via

\[
\hat{U}_{m_J} := U_{M_J}(:, 1:m_J), \quad \hat{P}_{m_J} := P_{M_J}(:, 1:m_J),
\]
\[
\hat{U} := \hat{U}_{m_J}(\hat{P}_{m_J}^T \hat{U}_{m_J})^{-1}.
\]

**off2** Select partial similarity transform matrix of size $k$ as

$Q_k := Q(:, 1:k)$ and compute

\[
\tilde{U}(\cdot, j) := \mathcal{V}_k [Q_k^T C^T \mathcal{V}_k^{-1} [\hat{U}(\cdot, j)] C^{-T} Q_k] \in \mathbb{R}^{k^2}, j = 1 \ldots m_J
\]

where $\mathcal{V}_k$ denotes the same transformation as $\mathcal{V}$ but for $k \times k$ matrices.
Partial Similarity Transformations and Matrix DEIM

Matrix DEIM: Offline/online decomposition

off1  Compute $U_{M_J}, P_{M_J}, Q$ once, then for each new $m_J, k$ do

off2  Compute offline vectors for Matrix DEIM of Jacobian via

\[
\hat{U}_{m_J} := U_{M_J}(:, 1:m_J), \quad \hat{P}_{m_J} := P_{M_J}( :, 1:m_J),
\]
\[
\hat{U} := \hat{U}_{m_J}( \hat{P}^T_{m_J} \hat{U}_{m_J} )^{-1}.
\]

off2  Select partial similarity transform matrix of size $k$ as $Q_k := Q(:, 1:k)$ and compute

\[
\hat{U}( :, j ) := \mathcal{V}_k [ Q_k^T C^T \mathcal{V}^{-1}_k [ \hat{U}( :, j ) ] C^{-T} Q_k ] \in \mathbb{R}^{k^2}, j = 1 \ldots m_J,
\]

where $\mathcal{V}_k$ denotes the same transformation as $\mathcal{V}$ but for $k \times k$ matrices.

online  For new $y^r(t)$ compute $L_{I_k} [ \mathcal{V}^{-1}_k [ \hat{U} \hat{P}^T_{m_J} \mathcal{V} [ J(y^r(t)) ] ] ]$ in

$\mathcal{O}(k^3 + g_f(m_J))$ (g_f “internal f complexity”)
Some more experiments!

MoRePaS

Some more experiments!
Some more experiments!
Error estimation results for Burger’s equation

**Figure**: Absolute errors over time for different matrix DEIM orders and partial similarity transformation sizes
Some more experiments!

Error estimation results for Burger’s equation: Zoom

Figure: Absolute errors over time for different matrix DEIM orders and partial similarity transformation sizes
Some more experiments!

PCD Model

The apoptosis model

\[
\frac{\partial x_a}{\partial t} = k_{c1} x_i y_a - k_{d1} x_a + D_1 \Delta x_a,
\]
\[
\frac{\partial y_a}{\partial t} = k_{c2} y_i x_a^2 - k_{d2} y_a + D_2 \Delta y_a,
\]
\[
\frac{\partial x_i}{\partial t} = -k_{c1} x_i y_a - k_{d3} x_i + k_{p1} + D_3 \Delta x_i,
\]
\[
\frac{\partial y_i}{\partial t} = -k_{c2} y_i x_a^2 - k_{d4} y_i + k_{p2} + D_4 \Delta y_i,
\]

- “Caspase cascade” \( x_i, y_i, x_a, y_a \): caspase-8, caspase-3, procaspase-8 and procaspase-3.
- \( k_* \) and \( D_i \): scaled constants for creation, interaction and diffusion
- [Daub et al.(2010)Daub, Waldherr, Allgöwer, Scheurich & Schneider]
Some more experiments!

Boundary conditions

- Procaspass-8 ($x_a$) gets activated by receptors on cellular membrane
- Gives geometrically parameter-dependent boundary conditions!
- For $\mu_1 \in [0, 1]$ we define a part

$$\Gamma_{\mu_1} := \{x \in \partial\Omega \mid |x_1 - 0.5| \leq 0.5\mu_1 \lor |x_2 - 0.75| \leq 0.75\mu_1\} \subseteq \partial\Omega$$

- On $\Gamma_{\mu_1}$ we impose Neumann conditions

$$\left. \frac{\partial x_a}{\partial n} \right|_{\Gamma_{\mu_1}} = - \left. \frac{\partial x_i}{\partial n} \right|_{\Gamma_1} = \mu_2 x_i,$$

- Reaction rate $\mu_2 \in [10^{-5}, 10^{-2}]$ of procaspase-8 activation.
- Homogeneous Neumann on rest of $\partial\Omega$, initial conditions

$$x_a = y_a = 10^{-9}, \quad x_i = y_i = 0.01.$$
Some more experiments!

**Eye candy**

- Discretization on a $100 \times 150$ grid
- 60000-dimensional state space $\mathbf{y}(t) = (x_a, y_a, x_i, y_i)^T$.

**Figure:** Upper row: Caspase concentrations of full model for $\mathbf{\mu} = (0.777, 0.00132)^T$. Lower row: Absolute errors
Some more experiments!

Reduction settings & estimates

- 200 random parameters $\Xi \subset \mathcal{P} = [0, 1] \times [10^{-5}, 10^{-2}]$ for training
- DEIM procedure on $f$ with $M = 200$
- Tolerance $10^{-6}$: $V \in \mathbb{R}^{60000 \times 282}$ (reduction by $\approx 212$).
- Matrix DEIM / Sim. trans.: 20200 sample points, $M_J = 200$, $k_{max} = 50$.

Figure: Absolute errors over time for different DEIM $m'$ approximation error orders
Some more experiments!
Relative errors & computation times

Figure: Relative errors (left) and computation times against estimates at $T = 500$ (right) for different estimator configurations.
### Estimation result table

<table>
<thead>
<tr>
<th>Version</th>
<th>$\Delta(500)$</th>
<th>Time</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True error</strong></td>
<td>$3.908 \times 10^{-4}$</td>
<td>11.31s</td>
<td>1.000</td>
</tr>
<tr>
<td>$\langle y-y^r, f(y) - f(y^r) \rangle$</td>
<td>$9.476 \times 10^{-4}$</td>
<td>21.44s</td>
<td>2.425</td>
</tr>
<tr>
<td>$\frac{</td>
<td>y-y^r</td>
<td>^2}{</td>
<td>y-y^r</td>
</tr>
<tr>
<td>$\langle y-y^r, J<a href="y-y%5Er">y^r(t)</a> \rangle$</td>
<td>$9.476 \times 10^{-4}$</td>
<td>23.63s</td>
<td>2.425</td>
</tr>
<tr>
<td>$m, J : 20, k : 5$</td>
<td>$2.989 \times 10^{-3}$</td>
<td>0.92s</td>
<td>7.649</td>
</tr>
<tr>
<td>$m, J : 20, k : 15$</td>
<td>$2.989 \times 10^{-3}$</td>
<td>0.94s</td>
<td>7.649</td>
</tr>
<tr>
<td>$m, J : 20, k : 50$</td>
<td>$2.989 \times 10^{-3}$</td>
<td>0.97s</td>
<td>7.649</td>
</tr>
<tr>
<td>$m, J : 100, k : 5$</td>
<td>$2.580 \times 10^{-3}$</td>
<td>1.08s</td>
<td>6.602</td>
</tr>
<tr>
<td>$m, J : 100, k : 15$</td>
<td>$2.588 \times 10^{-3}$</td>
<td>1.09s</td>
<td>6.622</td>
</tr>
<tr>
<td>$m, J : 100, k : 50$</td>
<td>$2.588 \times 10^{-3}$</td>
<td>1.15s</td>
<td>6.622</td>
</tr>
<tr>
<td>$m, J : 170, k : 5$</td>
<td>$3.869 \times 10^{-3}$</td>
<td>1.22s</td>
<td>9.900</td>
</tr>
<tr>
<td>$m, J : 170, k : 15$</td>
<td>$4.392 \times 10^{-3}$</td>
<td>1.20s</td>
<td>$1.124 \times 10^1$</td>
</tr>
<tr>
<td>$m, J : 170, k : 50$</td>
<td>$4.392 \times 10^{-3}$</td>
<td>1.27s</td>
<td>$1.124 \times 10^1$</td>
</tr>
</tbody>
</table>

Table: Overview for all used estimator configurations
Summary

...any questions?

Short summary

- DEIM error estimation using next basis functions
- Local approximation of Lipschitz constants via Jacobian logarithmic norms
- Efficient computation of Jacobian logarithmic norm via
  - Partial similarity transformation
  - Matrix DEIM
- Bottom line: Non-rigorous but applicable a-posteriori error estimator!
Summary

... any questions?

Short summary

- DEIM error estimation using next basis functions
- Local approximation of Lipschitz constants via Jacobian logarithmic norms
- Efficient computation of Jacobian logarithmic norm via
  - Partial similarity transformation
  - Matrix DEIM
- Bottom line: Non-rigorous but applicable a-posteriori error estimator!

Thank you for your attention! Any questions/remarks?


[Daub et al. (2010) Daub, Waldherr, Allgöwer, Scheurich & Schneider]


[Drohmann et al. (2012) Drohmann, Haasdonk & Ohlberger]

Image sources

- http://www.bildungsstiftung.org/05_aktionen.htm