Online adaptive discrete empirical interpolation for nonlinear model reduction

Benjamin Peherstorfer (Courant Institute, NYU) and Karen Willcox (MIT)
Discretized equations stemming from (steady-state) nonlinear PDE

\[
A_{N \times N} y(\mu) + f(y(\mu)) = 0
\]

- Spatial domain \( \Omega \subset \mathbb{R}^{\{1,2,3\}} \)
- Linear term \( A \in \mathbb{R}^{N \times N} \)
- State variable \( y \in \mathbb{R}^N \)
- Parameter \( \mu \in \mathcal{D} \subset \mathbb{R}^d \)
- Nonlinear term \( f(y(\mu)) \in \mathbb{R}^N \)
Introduction: Discrete Empirical Interpolation Method

Derive POD-Galerkin reduced system

- Collect snapshots
  \[ y(\mu_1), \ldots, y(\mu_M) \in \mathbb{R}^N \]
- Construct POD basis \( V \in \mathbb{R}^{N \times n}, n \ll N \) of reduced space
- Project onto reduced space

\[
  V^T A V \tilde{y}(\mu) + V^T f(V \tilde{y}(\mu)) = 0
\]

DEIM interpolates \( f \) as linear combination of basis \( U \)

- Compute nonlinear snapshots \( f(y(\mu_1)), \ldots, f(y(\mu_M)) \in \mathbb{R}^N \)
- Compute DEIM basis \( U \in \mathbb{R}^{N \times m} \) of nonlinear snapshots
- Select interpolation points in \( P \in \mathbb{R}^{N \times m} \) at which to sample \( f \)

\[
  V^T A V \tilde{y}(\mu) + V^T U(P^T U)^{-1} P^T f(V \tilde{y}(\mu)) = 0
\]

[Barrault et al., 2004], [Grepl et al., 2007], [Astrid et al., 2008], [Chaturantabut et al., 2010], [Carlberg et al., 2011],

[Drohmann et al., 2012], [Drmac, Gugercin, 2016]
Nonlinear approximation

Have \( t \in [1, 3] \), grid \( \mathbf{x} = [x_1, \ldots, x_{64}]^T \) in \([0, 2\pi]\)

\[
f(t) = t (\sin(xt) + \sin(xt\pi/2) + \sin(xt\pi))
\]

Dimension of manifold induced by \( f \) is 1

\[
\mathcal{M} = \{f(t) | t \in [1, 3]\} \subset \mathbb{R}^{64}
\]

Classical reduced models approximate \( \mathcal{M} \) with space

\[
\mathcal{U} = \text{span}\{u_1, \ldots, u_m\}
\]

Here, one-dimensional space

\[
\mathcal{U} = \text{span}\{u_1\}
\]

fails to approximate \( \mathcal{M} \)

\( \Rightarrow \) **Localized and adaptive** reduced spaces
Nonlinear approximation

Have \( t \in [1, 3] \), grid \( \mathbf{x} = [x_1, \ldots, x_{64}]^T \) in \([0, 2\pi]\)

\[
f(t) = t (\sin(x t) + \sin(x t \pi/2) + \sin(x t \pi))
\]

Dimension of manifold induced by \( f \) is 1

\[
\mathcal{M} = \{ f(t) \mid t \in [1, 3] \} \subset \mathbb{R}^{64}
\]

Classical reduced models approximate \( \mathcal{M} \) with space

\[
\mathcal{U} = \text{span}\{u_1, \ldots, u_m\}
\]

Here, one-dimensional space

\[
\mathcal{U} = \text{span}\{u_1\}
\]

fails to approximate \( \mathcal{M} \)

\( \Rightarrow \) **Localized and adaptive reduced spaces**
Nonlinear approximation

Have \( t \in [1, 3] \), grid \( \mathbf{x} = [x_1, \ldots, x_{64}]^T \) in \([0, 2\pi]\)

\[
f(t) = t \left( \sin(x t) + \sin(x t \pi/2) + \sin(x t \pi) \right)
\]

Dimension of manifold induced by \( f \) is 1

\[
\mathcal{M} = \{f(t) \mid t \in [1, 3]\} \subset \mathbb{R}^{64}
\]

Classical reduced models approximate \( \mathcal{M} \) with space

\[
\mathcal{U} = \text{span}\{u_1, \ldots, u_m\}
\]

Here, one-dimensional space

\[
\mathcal{U} = \text{span}\{u_1\}
\]

fails to approximate \( \mathcal{M} \)

\( \Rightarrow \) **Localized and adaptive** reduced spaces
from global reduced spaces to localized subspaces

\[ \mathcal{U} \Rightarrow \mathcal{U}_1, \ldots, \mathcal{U}_k \]
reduced models with online adaptive (DEIM) subspaces

\[ \mathcal{U} \Rightarrow \mathcal{U}_1, \ldots, \mathcal{U}_k \Rightarrow \mathcal{U}(t) \]

- Amsallem, Zahr, Washabaugh; 2015
- Carlberg; 2015
- Kramer, P., Willcox; 2017
- Lass; 2014
- P., Willcox; 2015a, 2015b, 2016
- Schindler, Ohlberger; 2015, 2017
- Zahr, Farhat; 2015
- ...
reduced models with online adaptive (DEIM) subspaces

\[ \mathcal{U} \Rightarrow \mathcal{U}_1, \ldots, \mathcal{U}_k \Rightarrow \mathcal{U}(t) \]

- Amsallem, Zahr, Washabaugh; 2015
- Carlberg; 2015
- Kramer, P., Willcox; 2017
- Lass; 2014
- P., Willcox; 2015a, 2015b, 2016
- Schindler, Ohlberger; 2015, 2017
- Zahr, Farhat; 2015
- ...


- Sparse residual from full model \(\Rightarrow\) avoid pre-computed quantities
- Low-rank updates \(\Rightarrow\) online efficiency
- Additive updates \(\Rightarrow\) arbitrary changes to reduced basis
Nonlinear approximation through online adaptivity

Reduced state vector \( \tilde{y}(\mu_{M+i}) \)

Sample \( f \)

Samples of nonlinear function

Process

DEIM interpolant

Approximate

Nonlinear function approximation
Nonlinear approximation through online adaptivity

outer loop iteration

- reduced state vector $\tilde{y}(\mu_{M+1})$
- sample $f$
- samples of nonlinear function
- process
- DEIM interpolant
- approximate
- nonlinear function approximation
Nonlinear approximation through online adaptivity

outer loop iteration

1. Reduced state vector $\tilde{y}(\mu_{M+1})$
   - Sample $f$
   - Samples of nonlinear function
     - DEIM interpolant
     - Approximate nonlinear function
     - Nonlinear function approximation

2. Reduced state vector $\tilde{y}(\mu_{M+2})$
   - Sample $f$
   - Samples of nonlinear function
     - Approximate nonlinear function
     - Nonlinear function approximation
Nonlinear approximation through online adaptivity

outer loop iteration

- reduced state vector $\tilde{y}(\mu_{M+1})$
  - sample $f$
  - samples of nonlinear function
  - process
  - DEIM interpolant
  - approximate nonlinear function approximation

- reduced state vector $\tilde{y}(\mu_{M+2})$
  - sample $f$
  - samples of nonlinear function
  - process
  - adapted DEIM interpolant
  - approximate nonlinear function approximation
Nonlinear approximation through online adaptivity

outer loop iteration

reduced state vector $\tilde{y}(\mu_{M+1})$

sample $f$
samples of nonlinear function

DEIM interpolant

approximate nonlinear function approximation

adapt

reduced state vector $\tilde{y}(\mu_{M+2})$

sample $f$
samples of nonlinear function

adapted DEIM interpolant

approximate nonlinear function approximation

adapt

reduced state vector $\tilde{y}(\mu_{M+3})$

sample $f$
samples of nonlinear function

adapted DEIM interpolant

approximate nonlinear function approximation

adapt...
Nonlinear approximation through online adaptivity

Key ingredients of online adaptation

- Sparse residual from full model ⇒ avoid pre-computed quantities
- Low-rank updates ⇒ online efficiency
- Additive updates ⇒ arbitrary changes to DEIM space
Adaptive DEIM: Problem setup

Outer loop

- Online phase consists of $k = 1, \ldots, M'$ outer loop iterations
- Reduced solution $\tilde{y}(\mu_{M+k})$ requested at each iteration
- DEIM interpolant used to approximate $f(V\tilde{y}(\mu_{M+k}))$

Adaptive interpolant

- DEIM interpolant $(U_0, P_0)$ built offline from snapshots
- Adaptation initialized with $(U_0, P_0)$
- In each step $k = 1, \ldots, M'$
  - Adapt DEIM basis $U_{k-1}$ to obtain $U_k$
  - Adapt DEIM interpolation points matrix $P_{k-1}$ to obtain $P_k$
  - Derive adapted interpolant $(U_k, P_k)$
  - Use $(U_k, P_k)$ to approximate nonlinear term
Adaptive DEIM: Problem setup

Outer loop
- Online phase consists of $k = 1, \ldots, M'$ outer loop iterations
- Reduced solution $\tilde{y}(\mu_{M+k})$ requested at each iteration
- DEIM interpolant used to approximate $f(V\tilde{y}(\mu_{M+k}))$

Adaptive interpolant
- DEIM interpolant $(U_0, P_0)$ built offline from snapshots
- Adaptation initialized with $(U_0, P_0)$
- In each step $k = 1, \ldots, M'$
  - Adapt DEIM basis $U_{k-1}$ to obtain $U_k$
  - Adapt DEIM interpolation points matrix $P_{k-1}$ to obtain $P_k$
  - Derive adapted interpolant $(U_k, P_k)$
  - Use $(U_k, P_k)$ to approximate nonlinear term
Adaptive DEIM: Oversampling

DEIM interpolates $f$ at interpolation points $\{p_1, \ldots, p_m\}$, i.e.,

$$\left\| P^T \left( U(P^T U)^{-1} P^T f(y(\mu)) - f(y(\mu)) \right) \right\|_2 = 0$$

Oversample nonlinear function

- Draw $m_s \in \mathbb{N}$ points uniformly from $\{1, \ldots, N\} \setminus \{p_1, \ldots, p_m\}$
- First $m$ points equal DEIM interpolation points
- Create sampling points matrix $S \in \mathbb{R}^{N \times m + m_s}$
- Solve regression problem to obtain coefficient

$$c(y(\mu)) = (S^T U)^+ S^T f(y(\mu))$$

- Residual

$$r(y(\mu)) = Uc(y(\mu)) - f(y(\mu))$$

In general, non-zero residual at sampling points,

$$\| S^T r(y(\mu)) \|_2 > 0$$
Adaptive DEIM: Low-rank basis updates

Window contains previous $w \in \mathbb{N}$ reduced solutions $\tilde{y}(\mu_{k_1}), \ldots, \tilde{y}(\mu_{k_w})$

Adapt space $U_{k-1} \in \mathbb{R}^{N \times m}$ with rank-one update $\alpha_k \beta_k^T \in \mathbb{R}^{N \times m}$

$$U_k = U_{k-1} + \alpha_k \beta_k^T$$

Update $\alpha_k \beta_k^T$ minimizes

$$\| S_k^T ((U_{k-1} + \alpha_k \beta_k^T) C_k - F_k) \|_F^2$$

- $S_k \in \mathbb{R}^{N \times m + m_s}$ is sampling points matrix at step $k$
- $C_k = [c(V \tilde{y}(\mu_{k_1})), \ldots, c(V \tilde{y}(\mu_{k_w}))] \in \mathbb{R}^{m \times w}$ coefficient matrix
- $F_k = [f(V \tilde{y}(\mu_{k_1})), \ldots, f(V \tilde{y}(\mu_{k_w}))] \in \mathbb{R}^{N \times w}$ RHS matrix

With $R_k = U_{k-1} C_k - F_k$ obtain minimization problem

$$\arg \min_{\alpha_k \in \mathbb{R}^N, \beta_k \in \mathbb{R}^m} \| \left( \underbrace{S_k^T R_k}_{m+m_s \times w} \right) + \left( \underbrace{S_k^T \alpha_k \beta_k^T}_{m+m_s} \right) C_k \|_F^2$$
Adaptive DEIM: Construction of basis update

**Optimization problem** With $R_k = U_{k-1} C_k - F_k$ obtain

$$\arg\min_{\alpha_k \in \mathbb{R}^N, \beta_k \in \mathbb{R}^m} \left\| \begin{pmatrix} S_k^T R_k + S_k^T \alpha_k \beta_k^T C_k \end{pmatrix} \right\|^2_F$$

Update $\alpha_k \beta_k^T$ derived from $m \times m$ generalized eigenproblem

- Size $m \times m$ scales with dimension $m$ of DEIM space
- Symmetric positive definite
- Optimal update that minimizes residual in $\| \cdot \|_F$

**Costly steps of optimization**

- Sampling nonlinear function at $m + m_s \times w$ components
- QR transformation (necessary if $C_k \in \mathbb{R}^{m \times w}$ rank deficient)
- Generalized eigenproblem of size $m \times m$
Adaptive DEIM: Interpolation points update

**Point adaptation**
- Offline point selection too expensive online
- Often not necessary to recompute all points
- Heuristic selects point to be replaced

**Online interpolation points update**
- Find basis vector that was rotated most

\[ \text{diag}(U_k^T U_{k-1}) \]

- Replace the corresponding interpolation point
- Rerun offline point selection for this point only
- Costs in \( \mathcal{O}(Nm + m^3) \)

\( \text{select} \quad \text{new} \)
Adaptive DEIM: Toy example

![Graph showing the target solution, static basis vector, adapted basis vector, static DEIM, and adaptive DEIM in the spatial domain at time t = 1.](image-url)
Adaptive DEIM: Toy example
Define \( \Omega = D = [0, 1]^2 \) and \( g : \Omega \times D \rightarrow \mathbb{R} \)

\[
g(x, \mu) = \frac{\mu_1 \mu_2 \exp(x_1 x_2)}{\exp(20 \|x - \mu\|^2)}
\]

- Find parameter \( \mu \in D \) that maximizes

\[
\int_{\Omega} g(x, \mu) \, dx
\]

- Discretize in \( \Omega \) on \( 40 \times 40 \) equidistant grid
- Approximate nonlinear function with DEIM
- Search for optimum with DEIM interpolant
Numerical results: Optimization problem

- Search for maximum with Nelder-Mead algorithm
- DEIM interpolant adapted when Nelder-Mead evaluates function
- Adaptive space with 5 dim achieves accuracy of 100-dim static space
- Adaptive interpolant reduces online runtime
Numerical results: FitzHugh-Nagumo system

Models electrical activity in a neuron

\[ \epsilon \partial_t y^\nu = \epsilon^2 \partial_x^2 y^\nu + f(y^\nu) - y^w + c, \]
\[ \partial_t y^w = by^\nu - \gamma y^w + c, \]

- Spatial domain \( x \in \Omega = [0, 1] \), time domain \( t \in \mathcal{T} = [0, 1] \)
- Voltage \( y^\nu : \Omega \times \mathcal{T} \rightarrow \mathbb{R} \) and recovery of voltage \( y^w : \Omega \times \mathcal{T} \rightarrow \mathbb{R} \)
- Nonlinear function \( f(y^\nu) = y^\nu(y^\nu - 0.1)(1 - y^\nu) \)
- Equidistant grid in \( \Omega \), \( N = 2048 \) DoFs
- Forward Euler in time domain, \( 10^6 \) time steps

[Chaturantabut et al., 2010]
Numerical results: FitzHugh-Nagumo system

Adapt DEIM interpolant every 200-th time step
Accuray improvement of up to two orders of magnitude
Overall runtime not dominated by evaluation of nonlinear term
Therefore minor speedup compared to static DEIM interpolant
Numerical results: Combustor model

Nonlinear advection-diffusion-reaction

\[ \kappa \Delta y - \nu \nabla y + f(y, \mu) = 0 \quad \text{in } \Omega \]

- \[ f(\mu, t) = [y_{H_2}, y_{O_2}, y_{H_2O}, T]^T \] with
  - mass fractions \( y_{H_2}, y_{O_2}, y_{H_2O} \)
  - temperature \( T \)

- Nonlinear term \( f \) with two parameters
  - pre-exponential factor \( \mu_1 = A \)
  - activation energy \( \mu_2 = E \)

[Buffoni & Willcox, 2010]
Numerical results: Region of interest

![Graphs showing relative $L^2$ error in region of interest vs. adaptivity step and online time.](image)

- Adapt interpolant to region of interest $D_{RoI} \subset D$
- Accuracy improvement of up to three orders of magnitude
- Significant online runtime improvement through adaptation
Numerical results: Extended/shifted parameter domain

- Adapted interpolant to extended/shifted parameter domain
- Static DEIM interpolant fails to approximate nonlinear term
- Accuracy improvement of up to three orders of magnitude
- Similar accuracy as rebuilding interpolant from scratch
Numerical results: Extended/shifted parameter domain

- Adapted interpolant to extended/shifted parameter domain
- Static DEIM interpolant fails to approximate nonlinear term
- Accuracy improvement of up to three orders of magnitude
- Similar accuracy as rebuilding interpolant from scratch
Numerical results: Extended/shifted parameter domain

- Adapted interpolant to extended/shifted parameter domain
- Static DEIM interpolant fails to approximate nonlinear term
- Accuracy improvement of up to three orders of magnitude
- Similar accuracy as rebuilding interpolant from scratch
Numerical results: Expected failure rate

- Computed expected failure rate of combustor (temp above 2240K)
- Adapted interpolant while Monte Carlo sampling proceeds
- RMSE about an order of magnitude lower

(a) four DEIM basis vectors

(b) six DEIM basis vectors
Conclusions

Adaptive discrete empirical interpolation method
- Adaptivity targets nonlinear term approximation
- Avoids pre-computed quantities that restrict update (cf. localization)
- Adaptation is computationally cheap (sparse samples, low-rank updates)

Online adaptive model reduction
- Nonlinear approximation of solution manifold
- More robust with respect to initial reduced model construction
- Identifies problem structure that is amenable to low-rank approximation

References