The Reduced-Basis Method for RCS Scattering

Methodology Overview

Akil Narayan

Brown University

June 29, 2007

Outline

Expensive Problems

- Many-query systems: solve for u given many values of $\mu \in \mathcal{D}$.

$$\mathcal{L}(u; \mu) = 0$$

- Inverting $\mathcal{L}$ can be expensive – must do this for every parameter value μ.
- Take advantage of smooth dependence on μ.

Expensive Problems

- Many-query systems: solve for u given many values of $\mu \in \mathcal{D}$.

$$\mathcal{L}(u; \mu) = 0$$

- Inverting $\mathcal{L}$ can be expensive – must do this for every parameter value μ.
- Take advantage of smooth dependence on μ.

Expensive Problems

- Many-query systems: solve for u given many values of $\mu \in \mathcal{D}$.

$$\mathcal{L}(u; \mu) = 0$$

- Inverting $\mathcal{L}$ can be expensive – must do this for every parameter value μ.
- Take advantage of smooth dependence on μ.

Choose Basis Wisely

- Basic idea: solution u is a function of parameter $u = u(\mu)$.
  - Manifold $\mathcal{M} = \{u(\mu) : \mu \in \mathcal{D}\}$ is low-dimensional.
  - Dimension of all possible u is $\mathcal{N}_t$ (very large for large problems)
- Usually express solution as

$$u(\mu) = \sum_{i=1}^{\mathcal{N}_t} \tilde{u}_i(\mu)\psi_i$$

- A better idea: choose N representative basis functions $u(\mu_i)$, express

$$u^N(\mu) = \sum_{i=1}^{N} \tilde{u}_i(\mu)u(\mu_i)$$

- Clever choice of $\mu_i \Rightarrow N \ll \mathcal{N}_t$

Choose Basis Wisely

- Basic idea: solution u is a function of parameter $u = u(\mu)$.
  - Manifold $\mathcal{M} = \{u(\mu) : \mu \in \mathcal{D}\}$ is low-dimensional.
  - Dimension of all possible u is $\mathcal{N}_t$ (very large for large problems)
- Usually express solution as

$$u(\mu) = \sum_{i=1}^{\mathcal{N}_t} \tilde{u}_i(\mu)\psi_i$$

- A better idea: choose N representative basis functions $u(\mu_i)$, express

$$u^N(\mu) = \sum_{i=1}^{N} \tilde{u}_i(\mu)u(\mu_i)$$

- Clever choice of $\mu_i \Rightarrow N \ll \mathcal{N}_t$

Choose Basis Wisely

- Basic idea: solution u is a function of parameter $u = u(\mu)$.
  - Manifold $\mathcal{M} = \{u(\mu) : \mu \in \mathcal{D}\}$ is low-dimensional.
  - Dimension of all possible u is $\mathcal{N}_t$ (very large for large problems)
- Usually express solution as

$$u(\mu) = \sum_{i=1}^{\mathcal{N}_t} \tilde{u}_i(\mu)\psi_i$$

- A better idea: choose N representative basis functions $u(\mu_i)$, express

$$u^N(\mu) = \sum_{i=1}^{N} \tilde{u}_i(\mu)u(\mu_i)$$

- Clever choice of $\mu_i \Rightarrow N \ll \mathcal{N}_t$

Reduced-Basis Method: Ingredients

- 'Truth Approximation'
  - Once $u(\mu_i)$ are computed, reduced-basis algorithm is implementable.
  - How to compute $u(\mu_i)$? Require underlying numerical method
- Error estimator
  - $\Rightarrow$ need estimation of inverse operator norm ($\mathcal{L}^{-1}$)
  - Start with elliptic problems
- Online/offline decomposition can potentially eliminate orders-of-magnitude computational expense

Reduced-Basis Method: Ingredients

- 'Truth Approximation'
  - Once $u(\mu_i)$ are computed, reduced-basis algorithm is implementable.
  - How to compute $u(\mu_i)$? Require underlying numerical method
- Error estimator
  - $\Rightarrow$ need estimation of inverse operator norm ($\mathcal{L}^{-1}$)
  - Start with elliptic problems
- Online/offline decomposition can potentially eliminate orders-of-magnitude computational expense

Reduced-Basis Method: Ingredients

- 'Truth Approximation'
  - Once $u(\mu_i)$ are computed, reduced-basis algorithm is implementable.
  - How to compute $u(\mu_i)$? Require underlying numerical method
- Error estimator
  - $\Rightarrow$ need estimation of inverse operator norm ($\mathcal{L}^{-1}$)
  - Start with elliptic problems
- Online/offline decomposition can potentially eliminate orders-of-magnitude computational expense

Truth Approximation

$$\Delta u^e(\mu) = f(\mu) \qquad \text{in } \Omega$$
$$u^e(\mu) = 0 \qquad \text{on } \partial\Omega, \quad u^e \in X^e$$

- Discretize using numerical method (e.g. FD, FV, FEM/DG)

$$a(u, v; \mu) = f(v; \mu) \qquad \forall v \in X$$

- $dim(X) = \mathcal{N}_t$ can be very large for real applications.
- Application of RB method to this problem
  - Attempts to approximate **truth** soluiton u, not exact solution u^e
  - Error estimates compare RB solution u^N to truth approximation u.

Truth Approximation

$$\Delta u^e(\mu) = f(\mu) \qquad \text{in } \Omega$$
$$u^e(\mu) = 0 \qquad \text{on } \partial\Omega, \quad u^e \in X^e$$

- Discretize using numerical method (e.g. FD, FV, FEM/DG)

$$a(u, v; \mu) = f(v; \mu) \qquad \forall v \in X$$

- $dim(X) = \mathcal{N}_t$ can be very large for real applications.
- Application of RB method to this problem
  - Attempts to approximate **truth** soluiton u, not exact solution u^e
  - Error estimates compare RB solution u^N to truth approximation u.

Truth Approximation

$$\Delta u^e(\mu) = f(\mu) \qquad \text{in } \Omega$$
$$u^e(\mu) = 0 \qquad \text{on } \partial\Omega, \quad u^e \in X^e$$

- Discretize using numerical method (e.g. FD, FV, FEM/DG)

$$a(u, v; \mu) = f(v; \mu) \qquad \forall v \in X$$

- $dim(X) = \mathcal{N}_t$ can be very large for real applications.
- Application of RB method to this problem
  - Attempts to approximate **truth** soluiton u, not exact solution u^e
  - Error estimates compare RB solution u^N to truth approximation u.

Problem Restrictions

- Require well-posedness of problem
  - $a(\cdot, \cdot; \mu)$ should have specific properties
  - Provable RB stability, convergence, and error estimates
- Hope for extreme computational savings
  - 'Affine' decomposition of operators: $a(\cdot, \cdot; \mu) = \sum_{q=1}^{Q_a} \Theta_q^a(\mu) a_q(\cdot, \cdot)$, similarly for f
  - Allows greater RB computational efficiency (offline/online decomposition)

Problem Restrictions

- Require well-posedness of problem
  - $a(\cdot, \cdot; \mu)$ should have specific properties
  - Provable RB stability, convergence, and error estimates
- Hope for extreme computational savings
  - 'Affine' decomposition of operators:
    $a(\cdot, \cdot; \mu) = \sum_{q=1}^{Q_a} \Theta_q^a(\mu) a_q(\cdot, \cdot)$, similarly for f
  - Allows greater RB computational efficiency (offline/online decomposition)

Offline: Selection of Basis

- Select 'representative' $\{\mu_i\}_{i=1}^{N}$
  - At step i, search parameter space $\mathcal{D}$ for $\tilde{\mu}$, the parameter value for which the solution error is maximized given current reduced-basis functions.
  - Error estimators, well-behavedness of a
  - Set $\mu_i = \tilde{\mu}$, require truth solution $u(\mu_i)$ to add to basis functions
- Set $X^N = span\,\{u(\mu_i) : i = 1, 2, \ldots N\}$
  - Orthogonalize: $u(\mu_i) \to \zeta_i$
  - Error estimator: stop adding basis functions when specified error tolerance is reached
  - Typically $dim(X^N) = N \ll \mathcal{N}_t = dim(X)$

Offline: Selection of Basis

- Select 'representative' $\{\mu_i\}_{i=1}^N$
  - At step i, search parameter space $\mathcal{D}$ for $\tilde{\mu}$, the parameter value for which the solution error is maximized given current reduced-basis functions.
  - Error estimators, well-behavedness of a
  - Set $\mu_i = \tilde{\mu}$, require truth solution $u(\mu_i)$ to add to basis functions
- Set $X^N = span\{u(\mu_i) : i = 1, 2, \ldots N\}$
  - Orthogonalize: $u(\mu_i) \to \zeta_i$
  - Error estimator: stop adding basis functions when specified error tolerance is reached
  - Typically $dim(X^N) = N \ll \mathcal{N}_t = dim(X)$

Offline: Compute Global Operators

$$a(u^N, v; \mu) = f(v; \mu) \quad \forall \ v \in X^N$$

- Goal: make all online calculations $\mathcal{N}_t$-independent
- a,f affine in μ
  - Can compute $a_q(\cdot,\cdot)$ offline for every q
  - Online computation of $a(\mu)$ is $\mathcal{N}_t$-independent

Offline: Compute Global Operators

$$a(u^N, v; \mu) = f(v; \mu) \quad \forall \; v \in X^N$$

- Goal: make all online calculations $\mathcal{N}_t$-independent
- a,f affine in μ
  - Can compute $a_q(\cdot, \cdot)$ offline for every q
  - Online computation of $a(\mu)$ is $\mathcal{N}_t$-independent

Online: Solve Reduced-Basis System

- Given a parameter μ, construct $(\mathbb{A})_{ij} = a(\zeta_i, \zeta_j; \mu)$ and $(\mathbf{f})_j = f(\zeta_j, \mu)$
- Solve $\mathbb{A}\hat{\mathbf{u}} = \mathbf{f}$ for $\hat{\mathbf{u}}$
- $N \times N$ solve instead of $\mathcal{N}_t \times \mathcal{N}_t$ truth-approximation solve
- Solution $u^N(\mu) = \sum_{i=1}^{N} \hat{u}_i \zeta_i$
- Online computation is *very* fast.
- Sharp *a posteriori* error estimates (produced as a by-product of offline computations)

Online: Solve Reduced-Basis System

- Given a parameter μ, construct $(\mathbb{A})_{ij} = a(\zeta_i, \zeta_j; \mu)$ and $(\mathbf{f})_j = f(\zeta_j, \mu)$
- Solve $\mathbb{A}\hat{\mathbf{u}} = \mathbf{f}$ for $\hat{\mathbf{u}}$
- $N \times N$ solve instead of $\mathcal{N}_t \times \mathcal{N}_t$ truth-approximation solve
- Solution $u^N(\mu) = \sum_{i=1}^{N} \hat{u}_i \zeta_i$
- Online computation is *very* fast.
- Sharp *a posteriori* error estimates (produced as a by-product of offline computations)

Complications

- Coercivity: $a(u,u) > C||u||^2$ for some $C > 0$ for all $u \in X$
  - Laplacian satisfies this property
  - Coercivity allows for easily-computable error estimate
  - Noncoercive problems (e.g. Helmholtz) require more sophisticated computations
- Output of interest not $u(\mu)$, but some other linear functional $s(u;\mu)$
  - Error estimates and search for μ_i is tailored to minimize error in $u(\mu_i)$
  - This does not determine error bound on $s(u,\mu)$.
  - To attain error bound on $s(u,\mu)$, need to define an 'adjoint' problem $\Rightarrow$ increased computational expense

Complications

- Coercivity: $a(u,u) > C||u||^2$ for some $C > 0$ for all $u \in X$
  - Laplacian satisfies this property
  - Coercivity allows for easily-computable error estimate
  - Noncoercive problems (e.g. Helmholtz) require more sophisticated computations
- Output of interest not $u(\mu)$, but some other linear functional $s(u;\mu)$
  - Error estimates and search for μ_i is tailored to minimize error in $u(\mu_i)$
  - This does not determine error bound on $s(u,\mu)$.
  - To attain error bound on $s(u,\mu)$, need to define an 'adjoint' problem $\Rightarrow$ increased computational expense

Helmholtz Problem: 1D

$$\Delta \tilde{\mathbf{E}}^e + \omega^2 \tilde{\mathbf{E}}^e = 0$$

$$\downarrow$$

$$a(u, v) = f(v) \quad \forall \quad v \in X$$

- Parameter is $\omega \in [20, 40]$, spatial variable $x \in [0, 1]$
- Split real/imaginary parts of $\tilde{\mathbf{E}}$.
- Inflow boundary conditions on left, outflow on right

$$u'' + \omega^2 u = 0 \qquad v'' + \omega^2 v = 0$$

$$u(0) = 1 \qquad v(0) = 1$$

$$u(1) + \omega v'(1) = 0 \qquad v(1) - \omega u'(1) = 0$$

Helmholtz Problem: 1D

$$\Delta \tilde{\mathbf{E}}^e + \omega^2 \tilde{\mathbf{E}}^e = 0$$
$$\downarrow$$
$$a(u, v) = f(v) \quad \forall \quad v \in X$$

- Parameter is $\omega \in [20, 40]$, spatial variable $x \in [0, 1]$
- Split real/imaginary parts of $\tilde{\mathbf{E}}$.
- Inflow boundary conditions on left, outflow on right

$$\begin{aligned} u'' + \omega^2 u &= 0 \\ u(0) &= 1 \\ u(1) + \omega v'(1) &= 0 \end{aligned} \qquad \begin{aligned} v'' + \omega^2 v &= 0 \\ v(0) &= 1 \\ v(1) - \omega u'(1) &= 0 \end{aligned}$$

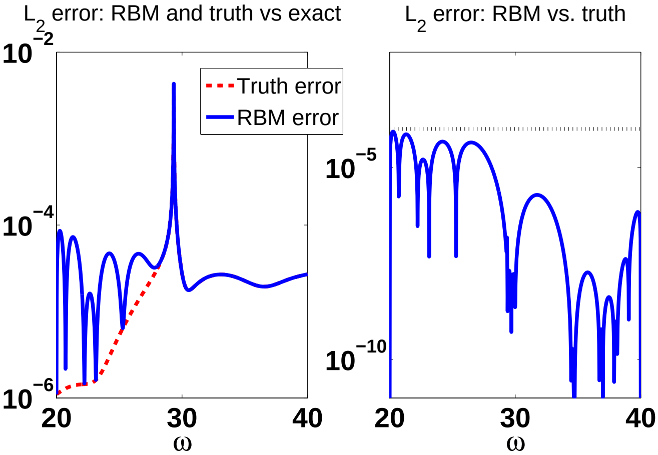
Apply RBM Method

- Discretize resulting system using the LDG method
- Desire to compute $u(\omega)$ for 10^4 different values of ω
- RBM overhead includes expensive computation of estimate for $||a^{-1}(\cdot, \cdot; \mu)||$
- Specify error tolerance of 10^{-4}, i.e. $||u^{DG} - u^N|| \leq 10^{-4}$
- RBM algorithm chose 18 basis functions before certifying desired error tolerance
- RBM 'offline' time: 98.9 seconds
- RBM 'online' time: 5.4 seconds
- DG time: 1130 seconds

Apply RBM Method

- Discretize resulting system using the LDG method
- Desire to compute $u(\omega)$ for 10^4 different values of ω
- RBM overhead includes expensive computation of estimate for $||a^{-1}(\cdot,\cdot;\mu)||$
- Specify error tolerance of 10^{-4}, i.e. $||u^{DG} - u^N|| \leq 10^{-4}$
- RBM algorithm chose 18 basis functions before certifying desired error tolerance
- RBM 'offline' time: 98.9 seconds
- RBM 'online' time: 5.4 seconds
- DG time: 1130 seconds

RBM Results


L_2 error: RBM and truth vs exact
L_2 error: RBM vs. truth
Truth error
RBM error
10^{-2}
10^{-4}
10^{-6}
10^{-5}
10^{-10}
20
30
40
ω

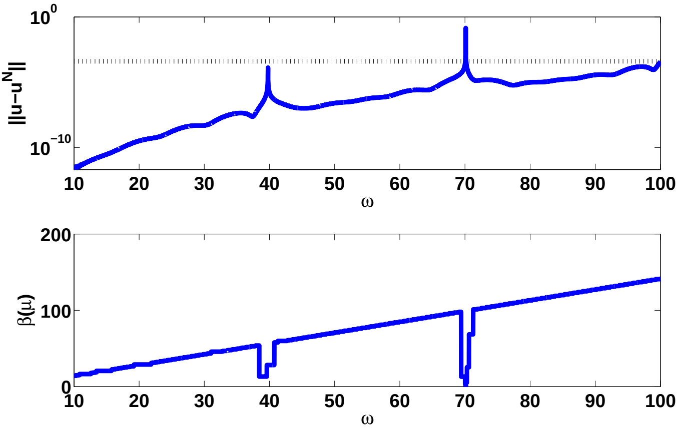
One More Time....

- Enlarge parameter space: $\omega \in [10, 100]$
- Again require $||u - u^N|| \leq 10^{-4}$
- $||a^{-1}(\cdot, \cdot; \mu)||$ is ill-behaved: can't find rigorous bound for it in certain regions of $\mathcal{D} = [10, 100]$
- RBM algorithm chooses 45 basis functions
- RBM 'offline' time: 477 seconds
- RBM 'online' time: 8.6 seconds
- DG time: 2419 seconds

One More Time....

- Enlarge parameter space: $\omega \in [10, 100]$
- Again require $||u - u^N|| \leq 10^{-4}$
- $||a^{-1}(\cdot, \cdot; \mu)||$ is ill-behaved: can't find rigorous bound for it in certain regions of $\mathcal{D} = [10, 100]$
- RBM algorithm chooses 45 basis functions
- RBM 'offline' time: 477 seconds
- RBM 'online' time: 8.6 seconds
- DG time: 2419 seconds

Is the bound for $||a^{-1}(\cdot,\cdot;\mu)||$ a big deal?



RCS: Problem Setup

- Boundary data and ω are parameters
- Within limits of elliptic Helmholtz/curl-curl formulation $\Rightarrow$ RB method is directly applicable
- Existing elliptic solver not present, requires significant additional coding to implement one
- Use existing time-domain solver to compute truth approximation to e.g. Helmholtz equation.
  - Must Fourier-transform from time to frequency domain
  - Additional error is introduced: error between time-domain solution and truth solution of elliptic problem

RCS: Problem Setup

- Boundary data and ω are parameters
- Within limits of elliptic Helmholtz/curl-curl formulation $\Rightarrow$ RB method is directly applicable
- Existing elliptic solver not present, requires significant additional coding to implement one
- Use existing time-domain solver to compute truth approximation to e.g. Helmholtz equation.
  - Must Fourier-transform from time to frequency domain
  - Additional error is introduced: error between time-domain solution and truth solution of elliptic problem

RCS: Problem Setup

- Boundary data and ω are parameters
- Within limits of elliptic Helmholtz/curl-curl formulation $\Rightarrow$ RB method is directly applicable
- Existing elliptic solver not present, requires significant additional coding to implement one
- Use existing time-domain solver to compute truth approximation to e.g. Helmholtz equation.
  - Must Fourier-transform from time to frequency domain
  - Additional error is introduced: error between time-domain solution and truth solution of elliptic problem

Every Path Hath a Puddle

- Avoid explicit construction of elliptic bilinear form $a(u, v)$ (i.e the matrix)
  - Current RBM research only supports elliptic problems
  - Still need bound on $||a^{-1}(\cdot, \cdot; \mu)|| \Rightarrow$ need to create routine to evaluate $a_q(u, \cdot)$ (i.e. the matrix-vector product Au)
- How to guarantee that $\mathcal{F}(\mathbf{E}) = \tilde{\mathbf{E}}$?
  - RBM can only be as strong as the truth approximation.
  - $u_0 = \tilde{\mathbf{E}}$: does $a(u_0, v; \mu) = f(v; \mu)$?
  - To ensure this, can run iterative method on $Au = f$ with initial guess $u_0 \Rightarrow$ again requires (matrix-free) evaluation of Au

Every Path Hath a Puddle

- Avoid explicit construction of elliptic bilinear form $a(u, v)$ (i.e the matrix)
  - Current RBM research only supports elliptic problems
  - Still need bound on $||a^{-1}(\cdot, \cdot; \mu)|| \Rightarrow$ need to create routine to evaluate $a_q(u, \cdot)$ (i.e. the matrix-vector product Au)
- How to guarantee that $\mathcal{F}(\mathbf{E}) = \tilde{\mathbf{E}}$?
  - RBM can only be as strong as the truth approximation.
  - $u_0 = \tilde{\mathbf{E}}$: does $a(u_0, v; \mu) = f(v; \mu)$?
  - To ensure this, can run iterative method on $Au = f$ with initial guess $u_0 \Rightarrow$ again requires (matrix-free) evaluation of Au

This Path Hath More Puddles

- Want proven error bounds in RCS
  - Have proven error bounds in solution $u(\mu)$, not in RCS output
  - Rigorous error bound for RCS $\Rightarrow$ need adjoint problem
  - Effectively doubles computational burden
  - Additional nontrivial, unphysical reformulation of Maxwell's equations
- Storage requirements in time-domain
  - RB discretization requires evaluation of e.g. $a(\zeta_i, \zeta_j; \mu)$
  - No obvious way to do this in without global solution $\Rightarrow$ storage of field variables over all space for some time interval $\Rightarrow$ memory burden.
  - Really only need $u(\omega)$ for one value of ω. Take advantage of Fourier Transform properties: can compute this by doing an $O(\mathcal{N}_t)$ operation at each timestep with minimal $O(\mathcal{N}_t)$ additional memory storage.

This Path Hath More Puddles

- Want proven error bounds in RCS
  - Have proven error bounds in solution $u(\mu)$, not in RCS output
  - Rigorous error bound for RCS $\Rightarrow$ need adjoint problem
  - Effectively doubles computational burden
  - Additional nontrivial, unphysical reformulation of Maxwell's equations
- Storage requirements in time-domain
  - RB discretization requires evaluation of e.g. $a(\zeta_i, \zeta_j; \mu)$
  - No obvious way to do this in without global solution$\Rightarrow$ storage of field variables over all space for some time interval$\Rightarrow$ memory burden.
  - Really only need $u(\omega)$ for one value of ω. Take advantage of Fourier Transform properties: can compute this by doing an $O(\mathcal{N}_t)$ operation at each timestep with minimal $O(\mathcal{N}_t)$ additional memory storage.

In theory there is no difference between theory and practice...

1. Compute estimation of $||a^{-1}(\cdot, \cdot; \mu)||$.
2. For $i = 1 \ldots N_{max}$
   1. Search over parameter space $\mathcal{D}$ for value μ_i to add to basis set
   2. Use time-domain solver to compute field values associated with μ_i
   3. During time evolution, compute $u(\mu_i)$, the Fourier-transformed fields
   4. Add $u(\mu_i)$ to basis set
   5. Stop if maximum error tolerance reached
3. Online part: simple

...In practice there is

- Verify that Fourier transform of time-dependent fields yields accurate enough ‘truth’ approximation
- Calculation of $||a^{-1}(\cdot, \cdot; \mu)||$
- Rigorously bound error in RCS

Summary

- RBM Methods effective for large problems
  - Large computational savings.
  - Well-developed theory for special problems
- RCS Problem not a direct application of existing theory
  - Sharp error bounds
  - Computationally efficient methods

Summary

- RBM Methods effective for large problems
  - Large computational savings.
  - Well-developed theory for special problems
- RCS Problem not a direct application of existing theory
  - Sharp error bounds
  - Computationally efficient methods