MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

A Low-rank Inexact Newton-Krylov Method for Stochastic Eigenvalue Problems

Peter Benner

Joint work with Martin Stoll (TU Chemnitz)
and Akwum Onwunta (MPI DCTS Magdeburg, moving to U Maryland)

39th Northern German Colloquium
on Applied Analysis and Numerical Mathematics
TU Braunschweig, June 1–2, 2018

Outline

1. Introduction
2. Discretization via Stochastic Galerkin Method
3. The Newton-Krylov Approach
4. Low-rank Inexact Newton-Krylov Method
5. Numerical Experiments

Outline

1. Introduction
2. Discretization via Stochastic Galerkin Method
3. The Newton-Krylov Approach
4. Low-rank Inexact Newton-Krylov Method
5. Numerical Experiments

Consider the real symmetric stochastic eigenvalue problem (SEVP)

Let Ω be a sample space with possible outcomes ω.

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega \\ \varphi(\omega)^T\varphi(\omega) &= 1, \end{aligned}$$

where $\lambda(\omega) \in \mathbb{R}, \quad \varphi(\omega) \in \mathbb{R}^{N_x}, \quad \mathcal{A}(\omega) \in \mathbb{R}^{N_x \times N_x}.$

Introduction

Consider the real symmetric stochastic eigenvalue problem (SEVP)

Let Ω be a sample space with possible outcomes ω.

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega \\ \varphi(\omega)^T\varphi(\omega) &= 1, \end{aligned}$$

$$\text{where} \quad \lambda(\omega) \in \mathbb{R}, \quad \varphi(\omega) \in \mathbb{R}^{N_x}, \quad \mathcal{A}(\omega) \in \mathbb{R}^{N_x \times N_x}.$$

Assumptions on $\mathcal{A}(\omega)$

- Real symmetric random matrix on probability space $(\Omega, \mathcal{F}, \mathbb{P})$.
- For example, stiffness matrix in structural mechanics problems.
- Inherits randomness, e.g., from uncertain elastic and dynamic parameters.

Introduction

Parametrization of uncertainty by random variables

- Recall ω is a random event in $(\Omega, \mathcal{F}, \mathbb{P})$.
- We set $\mathcal{A}(\omega) = \mathcal{A}(\xi(\omega))$, where $\xi = \{\xi_1, \ldots, \xi_m\}$.
- $\mathrm{supp}(\xi_i) = \Gamma_i \subseteq \mathbb{R}$, with measure $\rho_i(\xi_i) d\xi_i$.
- $\mathrm{supp}(\xi) = \Gamma_1 \times \ldots \times \Gamma_m$, with measure $\rho(\xi) = \prod_{i=1}^{m} \rho_i(\xi_i) d\xi_i$.

Introduction

Parametrization of uncertainty by random variables

- Recall ω is a random event in $(\Omega, \mathcal{F}, \mathbb{P})$.
- We set $\mathcal{A}(\omega) = \mathcal{A}(\xi(\omega))$, where $\xi = \{\xi_1, \ldots, \xi_m\}$.
- $\text{supp}(\xi_i) = \Gamma_i \subseteq \mathbb{R}$, with measure $\rho_i(\xi_i) d\xi_i$.
- $\text{supp}(\xi) = \Gamma_1 \times \ldots \times \Gamma_m$, with measure $\rho(\xi) = \prod_{i=1}^m \rho_i(\xi_i) d\xi_i$.

Random field representations

(Truncated) Karhunen-Lòeve expansion of $\mathcal{A}(\omega)$:

$$\mathcal{A}(\omega) \approx A_0 + \sum_{k=1}^{m} \xi_k(\omega) A_k, \quad A_k \in \mathbb{R}^{N_x \times N_x}, \quad k = 0, 1, \ldots, m.$$

Introduction

Parametrization of uncertainty by random variables

- Recall ω is a random event in $(\Omega, \mathcal{F}, \mathbb{P})$.
- We set $\mathcal{A}(\omega) = \mathcal{A}(\xi(\omega))$, where $\xi = \{\xi_1, \ldots, \xi_m\}$.
- $\mathrm{supp}(\xi_i) = \Gamma_i \subseteq \mathbb{R}$, with measure $\rho_i(\xi_i)d\xi_i$.
- $\mathrm{supp}(\xi) = \Gamma_1 \times \ldots \times \Gamma_m$, with measure $\rho(\xi) = \prod_{i=1}^m \rho_i(\xi_i)d\xi_i$.

Random field representations

Expand ℓth eigenpair $(\lambda_\ell(\omega), \varphi_\ell(\omega))$ using polynomial chaos expansion (PCE):

$$\begin{aligned} \lambda_\ell(\omega) &\approx \sum_{k=0}^{N_\xi - 1} \lambda_k^{(\ell)} \psi_k(\xi(\omega)), \quad \lambda_k^{(\ell)} \in \mathbb{R}, \\ \varphi_\ell(\omega) &\approx \sum_{k=0}^{N_\xi - 1} \varphi_k^{(\ell)} \psi_k(\xi(\omega)), \quad \varphi_k^{(\ell)} \in \mathbb{R}^{N_x}. \end{aligned}$$

Outline

1. Introduction

2. Discretization via Stochastic Galerkin Method

3. The Newton-Krylov Approach

4. Low-rank Inexact Newton-Krylov Method

5. Numerical Experiments

Discretization via Stochastic Galerkin Method

Consider again the model SEVP

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), && (1) \\ \varphi(\omega)^T\varphi(\omega) &= 1. && (2) \end{aligned}$$

Discretization via Stochastic Galerkin Method

Consider again the model SEVP

$$
\begin{aligned}
\mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), && (1)\\
\varphi(\omega)^T\varphi(\omega) &= 1. && (2)
\end{aligned}
$$

Stochastic Galerkin Method with suitable orthogonal polynomials ψ_ℓ

$$
\sum_{i=0}^{m}\sum_{j=0}^{N_\xi-1}\mathbb{E}(\xi_i\psi_j\psi_k)A_i\varphi_j = \sum_{i=0}^{N_\xi-1}\sum_{j=0}^{N_\xi-1}\mathbb{E}(\psi_i\psi_j\psi_k)\lambda_i\varphi_j,\ k=0,\dots,N_\xi-1,
$$

$$
\Longrightarrow \quad \underbrace{\left[G_0\otimes A_0+\sum_{k=1}^{m}G_k\otimes A_k\right]}_{:=A}\Phi = \left[\sum_{k=0}^{N_\xi-1}\lambda_k\underbrace{(H_k\otimes I_{N_x})}_{:=B_k}\right]\Phi. \qquad (3)
$$

Discretization via Stochastic Galerkin Method

Matrices

$$\text{Stochastic}: \begin{cases} G_0 = \operatorname{diag}\left(\left\langle \psi_0^2 \right\rangle, \left\langle \psi_1^2 \right\rangle, \ldots, \left\langle \psi_{N_\xi - 1}^2 \right\rangle\right), \\ G_k(i,j) = \langle \psi_i \psi_j \xi_k \rangle, \quad k = 1, \ldots, m, \\ H_k(i,j) = \langle \psi_i \psi_j \psi_k \rangle, \quad k = 0, \ldots, N_\xi - 1. \end{cases}$$

- A_0 is symmetric, positive definite; captures mean info in the model.
- $A_k, \ k = 1, \ldots, m$, represent the fluctuations in the model

Discretization via Stochastic Galerkin Method

Matrices

$$\text{Stochastic}: \begin{cases} G_0 = \operatorname{diag}\left(\left\langle \psi_0^2 \right\rangle, \left\langle \psi_1^2 \right\rangle, \ldots, \left\langle \psi_{N_\xi - 1}^2 \right\rangle\right), \\ G_k(i,j) = \left\langle \psi_i \psi_j \xi_k \right\rangle, \quad k = 1, \ldots, m, \\ H_k(i,j) = \left\langle \psi_i \psi_j \psi_k \right\rangle, \quad k = 0, \ldots, N_\xi - 1. \end{cases}$$

- A_0 is symmetric, positive definite; captures mean info in the model.
- $A_k,\ k = 1, \ldots, m$, represent the fluctuations in the model

Multiparametric eigenvalue problem!

For each eigenpair (Λ, Φ):

$$\Lambda = \left[\lambda_0, \lambda_1, \ldots, \lambda_{N_\xi - 1}\right]^T \in \mathbb{R}^{N_\xi},$$
$$\mathrm{vec}^{-1}(\Phi) = \left[\varphi_0, \varphi_1, \ldots, \varphi_{N_\xi - 1}\right] \in \mathbb{R}^{N_x \times N_\xi}.$$

Matrices

$$\text{Stochastic}: \begin{cases} G_0 = \operatorname{diag}\left(\left\langle \psi_0^2 \right\rangle, \left\langle \psi_1^2 \right\rangle, \ldots, \left\langle \psi_{N_\xi - 1}^2 \right\rangle\right), \\ G_k(i,j) = \left\langle \psi_i \psi_j \xi_k \right\rangle, \quad k = 1, \ldots, m, \\ H_k(i,j) = \left\langle \psi_i \psi_j \psi_k \right\rangle, \quad k = 0, \ldots, N_\xi - 1. \end{cases}$$

- A_0 is symmetric, positive definite; captures mean info in the model.
- $A_k,\ k = 1, \ldots, m$, represent the fluctuations in the model

Multiparametric eigenvalue problem!

So, we want to compute the tuple:

$$x := \{\lambda_0, \lambda_1, \ldots, \lambda_{N_\xi - 1}, \varphi_0, \varphi_1, \ldots, \varphi_{N_\xi - 1}\} \in \mathbb{R}^{(N_x + 1)N_\xi}.$$

Discretization via Stochastic Galerkin Method

Matrices

$$\text{Stochastic}: \begin{cases} G_0 = \operatorname{diag}\left(\left\langle\psi_0^2\right\rangle, \left\langle\psi_1^2\right\rangle, \ldots, \left\langle\psi_{N_\xi-1}^2\right\rangle\right), \\ G_k(i,j) = \langle\psi_i\psi_j\xi_k\rangle, \quad k = 1, \ldots, m, \\ H_k(i,j) = \langle\psi_i\psi_j\psi_k\rangle, \quad k = 0, \ldots, N_\xi - 1. \end{cases}$$

- A_0 is symmetric, positive definite; captures mean info in the model.
- $A_k, \ k = 1, \ldots, m$, represent the fluctuations in the model

Multiparametric eigenvalue problem!

Since

$$N_\xi = (m + r)!/m!r!, \quad m = \#\text{random variables and } r = \deg(\psi_i),$$

the dimension of the eignproblem can become quite huge.
(E.g., $m = 10,\ r = 4 \rightsquigarrow \ \sim 1000 \cdot N_x$ unknowns!)

Multi-parametric Eigenproblem

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

  $\Rightarrow$ standard eigenproblem for fixed λ_1.

Multi-parametric Eigenproblem

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

  $\Rightarrow$ continuum of real solutions $\lambda_0(\lambda_1)$ parameterized by λ_1.

Multi-parametric Eigenproblem

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

  $\Rightarrow$ $2N_x + 2 = 2(N_x + 1)$ unknowns in only $2N_x$ equations.

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

  We append normalization condition (2) to (3) to avoid this.

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

That is, $\quad \Phi^T (H_k \otimes I_{N_x})\Phi = \delta_{k0}, \quad k = 0, 1, \ldots, N_\xi - 1.$

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

  Thus, SEVP (1) and (2) now posed as $N_x N_\xi + N_\xi = (N_x + 1) N_\xi$ non-linear deterministic equations.

Multi-parametric Eigenproblem

Eigentuple-eigenvector problem

- We can re-write (3) as

$$A\Phi = \sum_{k=0}^{N_\xi - 1} \lambda_k B_k \Phi.$$

- $A := \sum_{k=0}^{m} G_k \otimes A_k, \quad B_k := H_k \otimes I_{N_x}$ with $B_0 := I_{N_\xi} \otimes I_{N_x}$.
- $N_\xi = 1$ yields standard eigenproblem $A\Phi = \lambda_0 \Phi$.
- For $N_\xi = 2$, we obtain $(A - \lambda_1 B_1)\Phi = \lambda_0 B_0 \Phi$:

We use low rank solvers to approach the dimensionality problem.

Outline

1. Introduction

2. Discretization via Stochastic Galerkin Method

3. The Newton-Krylov Approach

4. Low-rank Inexact Newton-Krylov Method

5. Numerical Experiments

The Newton system for stochastic eigenvalue problem

- Set $F(x) = 0$, where $x := (\Lambda, \Phi) \in \mathbb{R}^{(N_x+1)N_\xi}$ and

$$F(x) = \begin{bmatrix} \sum_{k=0}^{N_\xi-1} \left[(G_k \otimes A_k) - \lambda_k (H_k \otimes I_{N_x})\right] \Phi \\ \Phi^T (H_0 \otimes I_{N_x}) \Phi - 1 \\ \Phi^T (H_1 \otimes I_{N_x}) \Phi \\ \vdots \\ \Phi^T (H_{N_\xi-1} \otimes I_{N_x}) \Phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (4)$$

The Newton system for stochastic eigenvalue problem cont'd

- Given x_0, we obtain iterations via the Newton equations:

$$F(x_k) + F'(x_k)s_k = 0. \tag{5}$$

- Newton step: $s_k = \delta x_k = x_{k+1} - x_k$.
- When Newton step is obtained, next iterate is given by $x_{k+1} = x_k + s_k$.
- Set $(v, \lambda) := (v_0, v_1, \ldots, v_{N_\xi - 1}, \lambda_0, \lambda_1, \ldots, \lambda_{N_\xi - 1}) \approx (\Phi, \Lambda)$,

$$\begin{bmatrix} v^+ \\ \lambda^+ \end{bmatrix} = \begin{bmatrix} v \\ \lambda \end{bmatrix} - \underbrace{\begin{bmatrix} T(\lambda) & T'(\lambda)v \\ Q'(v) & 0 \end{bmatrix}^{-1}}_{\mathcal{J} := F'} \underbrace{\begin{bmatrix} T(\lambda)v \\ Q(v) \end{bmatrix}}_{F}.$$

Ingredients of the Newton equations

$$T(\lambda) = \sum_{k=0}^{N_\xi - 1} \left[(G_k \otimes A_k) - \lambda_k (H_k \otimes I_{N_x}) \right] \in \mathbb{R}^{N_x N_\xi \times N_x N_\xi},$$

$$T(\lambda)v = \sum_{k=0}^{N_\xi - 1} \left[(G_k \otimes A_k) - \lambda_k (H_k \otimes I_{N_x}) \right] v \in \mathbb{R}^{N_x N_\xi},$$

$$T'(\lambda)v = - \sum_{k=0}^{N_\xi - 1} (H_k \otimes v_k) \in \mathbb{R}^{N_x N_\xi \times N_\xi},$$

$$Q(v) = \mathbf{d} := \left[v^T (H_0 \otimes I_{N_x}) v - 1, \cdots, v^T (H_{N_\xi - 1} \otimes I_{N_x}) v \right]^T \in \mathbb{R}^{N_\xi}.$$

Algorithm 1 Inexact Newton Method (INM)

1: Given $x_0 \in \mathbb{R}^{(N_x+1)N_\xi}$
2: **for** $k = 0, 1, \ldots$ (until $\{x_k\}$ convergence) **do**
3: Choose some $\eta_k \in [0, 1)$.
4: Solve the Newton equations (5) approximately to obtain a step s_k such that
5: $||F(x_k) + F'(x_k)s_k|| \leq \eta_k ||F(x_k)||$.
6: Set $x_{k+1} = x_k + s_k$.
7: **end for**

The Inexact Newton's Method

- Solving the Newton equation $F(x_k) + F'(x_k)s_k = 0$ is expensive due to large-scale setting.
- It can be solved approximately with Krylov solvers.
- This leads to an inexact Newton-Krylov method.
- Since the Jacobian F' is not symmetric, we use `BiCGStab`.
- However, the choice of forcing terms η_k is very important:

$$\eta_1 := \eta_k = \min\{1/(k+2), ||F(x_k)||\}.$$

The Inexact Newton's Method

- Solving the Newton equation $F(x_k) + F'(x_k)s_k = 0$ is expensive due to large-scale setting.
- It can be solved approximately with Krylov solvers.
- This leads to an inexact Newton-Krylov method.
- Since the Jacobian F' is not symmetric, we use `BiCGStab`.
- However, the choice of forcing terms η_k is very important:

$$\eta_2 := \eta_k = 10^{-4}.$$

The Inexact Newton's Method

- Solving the Newton equation $F(x_k) + F'(x_k)s_k = 0$ is expensive due to large-scale setting.
- It can be solved approximately with Krylov solvers.
- This leads to an inexact Newton-Krylov method.
- Since the Jacobian F' is not symmetric, we use `BiCGStab`.
- However, the choice of forcing terms η_k is very important:

$$\eta_3 := \eta_k = \begin{cases} \zeta_k, & \eta_{k-1}^{(1+\sqrt{5})/2} \leq 0.1, \\ \max\left\{\zeta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\right\}, & \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1, \end{cases}$$

$$\text{where} \quad \zeta_k = \frac{||F(x_k) - F(x_{k-1}) - F'(x_{k-1})s_{k-1}||}{||F(x_{k-1})||}, \; k = 1, 2 \ldots$$

The Inexact Newton's Method

- Solving the Newton equation $F(x_k) + F'(x_k)s_k = 0$ is expensive due to large-scale setting.
- It can be solved approximately with Krylov solvers.
- This leads to an inexact Newton-Krylov method.
- Since the Jacobian F' is not symmetric, we use `BiCGStab`.
- However, the choice of forcing terms η_k is very important:

$$\eta_3 := \eta_k = \begin{cases} \zeta_k, & \eta_{k-1}^{(1+\sqrt{5})/2} \leq 0.1, \\ \max\left\{\zeta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\right\}, & \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1, \end{cases}$$

$$\text{where} \quad \zeta_k = \frac{||F(x_k) - F(x_{k-1}) - F'(x_{k-1})s_{k-1}||}{||F(x_{k-1})||},\ k = 1, 2 \ldots$$

- We use low rank solvers to approach the dimensionality problem.

The Inexact Newton's Method — Globalization

Inexact Newton method with backtracking

- Globalization is required to enhance the robustness of INM.
- It means augmenting Newton's method with auxiliary procedures that increase the likelihood of convergence.
- It adjusts step lengths (usually shortened) to obtain satisfactory steps.
- It can be backtracking or trust-region.
- In Algorithm 2, backtracking globalization resides in the while-loop

$$||F(x_k + s_k)|| \leq [1 - t(1 - \eta_k)]||F(x_k)||.$$

The Inexact Newton's Method — Globalization

Algorithm 2 Inexact Newton Backtracking Method (INBM)

Require: $x_0 \in \mathbb{R}^{(N_x+1)N_\xi}$, $\eta_{\max} \in [0,1)$, $t \in (0,1)$, and $0 < \theta_{\min} < \theta_{\max} < 1$.

```
for k = 0, 1, ... (until {x_k} converges) do
    Choose initial η_k ∈ [0, η_max) and solve (5) approximately to obtain s_k such
    that ||F(x_k) + F'(x_k)s_k|| ≤ η_k||F(x_k)||.
    while ||F(x_k + s_k)|| > [1 − t(1 − η_k)]||F(x_k)|| do
        Choose θ ∈ [θ_min, θ_max].
        Update s_k ← θs_k and η_k ← 1 − θ(1 − η_k).
    end while
    Set x_{k+1} = x_k + s_k.
end for
```

The Inexact Newton's Method — Globalization

Inexact Newton method with backtracking

- Globalization is required to enhance the robustness of INM.
- It means augmenting Newton's method with auxiliary procedures that increase the likelihood of convergence.
- It adjusts step lengths (usually shortened) to obtain satisfactory steps.
- It can be backtracking or trust-region.
- In Algorithm 2, backtracking globalization resides in the while-loop

$$||F(x_k + s_k)|| \leq [1 - t(1 - \eta_k)]||F(x_k)||.$$

We combine the backtracking with low-rank techniques to tackle the high-dimensional SEVP.

Outline

1. Introduction

2. Discretization via Stochastic Galerkin Method

3. The Newton-Krylov Approach

4. Low-rank Inexact Newton-Krylov Method

5. Numerical Experiments

Low-rank Inexact Newton-Krylov Method

Low-rank inexact Newton-BiCGStab

We solve the Newton equations with low-rank BiCGStab.

Since $\mathrm{vec}(WXV) = (V^T \otimes W)\mathrm{vec}(X)$, we have $\mathcal{J}\mathcal{X} = \mathcal{R}$, where

$$\mathcal{J} := F' = \begin{bmatrix} \sum_{i=0}^{N_\xi-1} [(G_i \otimes A_i) - \lambda_i (H_i \otimes I_{N_x})] & -\sum\limits_{i=0}^{N_\xi-1} H_i \otimes v_i \\ 2\sum\limits_{i=0}^{N_\xi-1} H_i \otimes v_i^T & 0 \end{bmatrix},$$

$$\mathcal{X} := s = \begin{bmatrix} \delta v_i \\ \delta \lambda_i \end{bmatrix} =: \begin{bmatrix} \mathrm{vec}(Y) \\ \mathrm{vec}(Z) \end{bmatrix}, \quad \mathcal{R} := -F = \begin{bmatrix} \mathrm{vec}(R_1) \\ \mathrm{vec}(R_2) \end{bmatrix},$$

and

$$R_1 = \mathrm{vec}^{-1}\left(\sum_{i=0}^{N_\xi-1} [(G_i \otimes A_i) - \lambda_i (H_i \otimes I_{N_x})]\, v \right), \quad R_2 = \mathrm{vec}^{-1}(\mathbf{d}).$$

Rewriting the linear system as matrix equation

Hence, we have

$$\begin{aligned}\mathcal{J}\mathcal{X} &= \operatorname{vec}\left(\sum_{i=0}^{N_\xi-1}\begin{bmatrix} \left(A_i Y G_i^T - \lambda_i I_{N_x} Y H_i^T\right) & -v_i Z H_i^T \\ 2v_i^T Y H_i^T & \end{bmatrix}\right)\\ &= \operatorname{vec}\left(\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}\right).\end{aligned}$$

Low-rank Inexact Newton-Krylov Method

Low-rank inexact Newton-BiCGStab

Rewriting the linear system as matrix equation

Hence, we have

$$
\begin{aligned}
\mathcal{J}\mathcal{X} &= \operatorname{vec}\left(\sum_{i=0}^{N_\xi - 1} \begin{bmatrix} \left(A_i Y G_i^T - \lambda_i I_{N_x} Y H_i^T \right) & - v_i Z H_i^T \\ 2 v_i^T Y H_i^T & \end{bmatrix} \right) \\
&= \operatorname{vec}\left(\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \right).
\end{aligned}
$$

Low-rank format for solution

$$
\begin{cases}
Y = W_Y V_Y^T, \quad \text{with} \quad W_Y \in \mathbb{R}^{(N_x+1)\times r_1}, \quad V_Y \in \mathbb{R}^{N_\xi \times r_1}, \\
Z = W_Z V_Z^T, \quad \text{with} \quad W_Z \in \mathbb{R}^{(N_x+1)\times r_2}, \quad V_Z \in \mathbb{R}^{N_\xi \times r_2},
\end{cases}
$$

where $r_{1,2} \ll N_\xi, N_x$.

Low-rank inexact Newton-BiCGStab

Rewriting the linear system as matrix equation

Hence, we have

$$\begin{aligned}
\mathcal{J}\mathcal{X} &= \mathrm{vec}\left(\sum_{i=0}^{N_\xi-1}\begin{bmatrix} \left(A_i Y G_i^T - \lambda_i I_{N_x} Y H_i^T\right) & -v_i Z H_i^T \\ 2v_i^T Y H_i^T & \end{bmatrix}\right) \\
&= \mathrm{vec}\left(\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}\right).
\end{aligned}$$

Left hand side of linear system in low-rank format

$$\begin{cases}
\text{(first row)} \displaystyle\sum_{i=0}^{N_\xi-1} \left[\, (A_i W_Y \quad -I_{N_x} W_Y) \quad -v_i W_Z \,\right] \begin{bmatrix} V_Y^T G_i & -\lambda_i V_Y^T H_i \\ V_Z^T H_i & \end{bmatrix}, \\
\text{(second row)} \displaystyle\sum_{i=0}^{N_\xi-1} \left[\, 2v_i W_Y \,\right] \left[\, V_Y^T H_i \,\right].
\end{cases}$$

The low-rank format

The low-rank nature of the factors

- guarantees fewer multiplications with the submatrices;
- maintains smaller storage requirements;
- can be obtained via MATLAB® functions `svd` or `svds`.

The iterates of Y and Z in low-rank BiCGStab are truncated using a prescribed tolerance.

Low-rank truncation

Suppose $X = WV^T \approx \mathcal{T}_\varepsilon(X) = \tilde{W}\tilde{V}^T$ s.t.

$$||X - \mathcal{T}_\varepsilon(X)|| \leq \varepsilon ||X||_F,$$

- skinny QR factorization: $X = Q_w R_w R_v^T Q_v^T$.
- SVD: $R_w R_v^T = B \Sigma C^T$.
- new factorization: $\tilde{W} = Q_w B(:, 1:r)$ and $\tilde{V} = Q_v C(:, 1:r)\Sigma(1:r, 1:r)$.
- singular values: $\sqrt{s_{r'+1}^2 + \ldots + s_r^2} \leq \varepsilon \sqrt{s_1^2 + \ldots + s_{r'}^2}$, $r' \leq r$.

Low-rank Inexact Newton-Krylov Method

Low-rank truncation

Suppose $X = WV^T \approx \mathcal{T}_\varepsilon(X) = \tilde{W}\tilde{V}^T$ s.t.

$$||X - \mathcal{T}_\varepsilon(X)|| \leq \varepsilon ||X||_F,$$

or: use truncated SVD of $WV^T \approx B\Sigma C^T$ computed via `svds` without ever forming the matrix.

Matrix inner products

- Suppose

$$Y = W_Y V_Y^T, \quad W_Y \in \mathbb{R}^{N_x \times r_Y}, \; V_Y \in \mathbb{R}^{N_\xi \times r_Y},$$

$$Z = W_Z V_Z^T, \quad W_Z \in \mathbb{R}^{N_x \times r_Z}, \; V_Z \in \mathbb{R}^{N_\xi \times r_Z}.$$

Matrix inner products

- Suppose

$$Y = W_Y V_Y^T, \quad W_Y \in \mathbb{R}^{N_x \times r_Y},\ V_Y \in \mathbb{R}^{N_\xi \times r_Y},$$

$$Z = W_Z V_Z^T, \quad W_Z \in \mathbb{R}^{N_x \times r_Z},\ V_Z \in \mathbb{R}^{N_\xi \times r_Z}.$$

- Then,

$$\mathrm{trace}\big(\underbrace{(W_Y V_Y^T)^T}_{\text{Large}}\underbrace{(W_Z V_Z^T)}_{\text{Large}}\big) = \mathrm{trace}\big(\underbrace{V_Z^T V_Y}_{\text{Small}}\,\underbrace{W_Y^T W_Z}_{\text{Small}}\big)$$

Low-rank Inexact Newton-Krylov Method

Matrix inner products

- Suppose

$$Y = W_Y V_Y^T, \quad W_Y \in \mathbb{R}^{N_x \times r_Y}, \; V_Y \in \mathbb{R}^{N_\xi \times r_Y},$$

$$Z = W_Z V_Z^T, \quad W_Z \in \mathbb{R}^{N_x \times r_Z}, \; V_Z \in \mathbb{R}^{N_\xi \times r_Z}.$$

- Then,

$$\mathrm{trace}\Big(\underbrace{\left(W_Y V_Y^T\right)^T}_{\text{Large}} \underbrace{\left(W_Z V_Z^T\right)}_{\text{Large}}\Big) = \mathrm{trace}\Big(\underbrace{V_Z^T V_Y}_{\text{Small}} \underbrace{W_Y^T W_Z}_{\text{Small}}\Big)$$

- Hence, $\langle Y, Z \rangle = \mathrm{vec}(Y)^T \mathrm{vec}(Z) = \mathrm{trace}(Y^T Z)$
$\Rightarrow \; 2(N_x + N_\xi + 1) r_Y r_Z$ flops.

Low-rank Inexact Newton-Krylov Method

Mean-based Schur complement preconditioner

We precondition the Jacobian $\mathcal{J}$ with

$$\mathcal{P} := \begin{bmatrix} T(\Lambda) & 0 \\ 0 & S \end{bmatrix}, \quad S = Q'(v)\,(T(\Lambda))^{-1}\,T'(\Lambda)v$$

where we use the following approximations:

$$\begin{aligned} T(\Lambda) &\approx I_{N_\xi} \otimes (A_0 - \lambda_0 I_{N_x}), \\ S &\approx 2I_{N_\xi} \otimes \left[v_0^T (A_0 - \lambda_0 I_{N_x})^{-1} v_0\right]. \end{aligned}$$

Outline

1. Introduction

2. Discretization via Stochastic Galerkin Method

3. The Newton-Krylov Approach

4. Low-rank Inexact Newton-Krylov Method

5. Numerical Experiments

Numerical Experiments

Consider the SEVP

$$\begin{cases} -\nabla \cdot (a(\cdot,\omega)\nabla\varphi(\cdot,\omega)) = \lambda(\omega)\varphi(\cdot,\omega), \;\; \text{in} \;\; \mathcal{D}\times\Omega, \\ \varphi(\cdot,\omega) = 0, \;\; \text{on} \;\; \partial\mathcal{D}\times\Omega, \end{cases} \tag{5}$$

where

- $a : \mathcal{D}\times\Omega \to \mathbb{R}$ is a random coefficient field, characterized by $\mathbb{E}(a) = 1$ and

$$Cov(x,y) \;=\; \sigma^2 \exp\left(-\frac{|x_1-y_1|}{\ell_1} - \frac{|x_2-y_2|}{\ell_2}\right), \; \forall (x,y)\in\mathcal{D}.$$

- KLE: $\xi = \{\xi_1,\ldots,\xi_m\}$ with $\xi_j \sim \mathcal{U}(-1,1)$, and $\ell_1 = \ell_2 = 1$.
- PCE: m-dim. Legendre polynomials with support in $(-1,1)^m$.

Small-scale problem, $\lambda_2(\omega)$

Performance of the INBM solver for $m = 6$, $r = 4$, $N_x = 49$ ($\rightsquigarrow$ $(N_\xi + 1)N_x = 10,500$) and standard deviation $\sigma_a = 0.01$. Here, I, II, and III represent the different forcing parameter choices η_1, η_2, and η_3.

η_k	# NS	∅ BS	# KS	t[sec]	rank	mem(LR)	mem(FM)
I	22	1.5	22	16.5	9	18.7	82.03
II	22	1.5	22	17.5	10	20.8	82.03
III	22	1.5	23	17.2	10	20.8	82.03

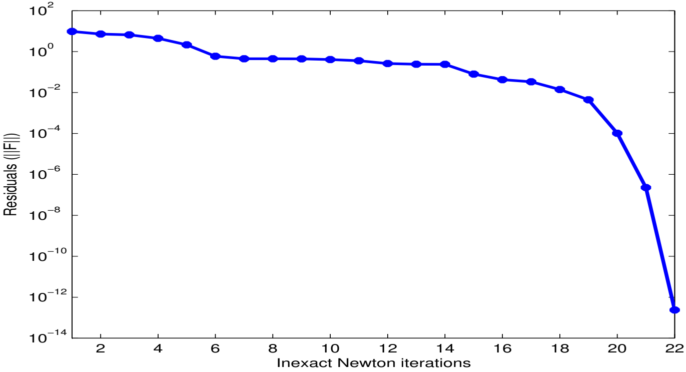
Legend:
NS — Newton steps, BS — backtracking steps, KS — Krylov steps,
rank — tensor rank of solution,
mem(LR) — memory low-rank solver in kb,

mem(FM) — memory full-rank solver in kb.

Numerical Experiments

Small-scale problem, $\lambda_2(\omega)$

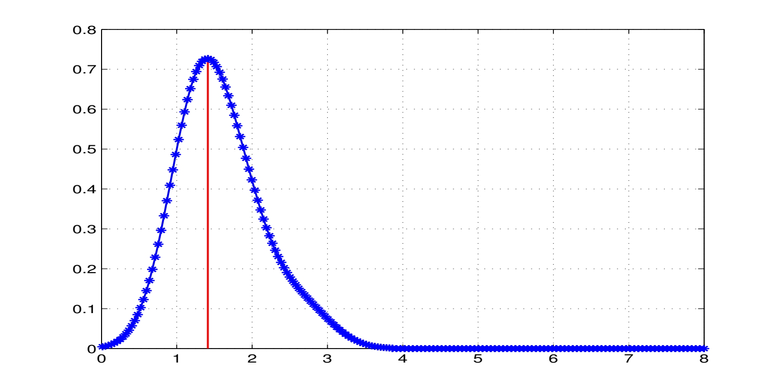
Convergence of low-rank INBM



Numerical Experiments

Small-scale problem, $\mathbb{E}(\lambda_2(\omega)) = 1.412 \approx \lambda_2(A_0) = 1.4166$

Probability density function (pdf) for $\sigma_a = 0.1$



Numerical Experiments

Large-scale problem, $\lambda_2(\omega)$

Performance of the INBM solver for $m = 6$, $r = 4$, $N_x = 392,704$ $(\rightsquigarrow\ (N_\xi + 1)N_x = 82,468,050)$ and standard deviation $\sigma_a = 0.1$. Here, I, II, and III represent the different forcing term choices η_1, η_2, and η_3.

η_k	# NS	∅ BS	# KS	t[sec]	rank	mem(LR)	mem(FM)
I	34	3.6	39	12,123	51	156,552	OoM (644,282)
II	32	3.5	43	12,113	51	156,552	OoM (644,282)
III	33	3.5	42	12,200	51	156,552	OoM (644,282)

Legend:
NS — Newton steps, BS — backtracking steps, KS — Krylov steps,
rank — tensor rank of solution,
mem(LR) — memory low-rank solver in kb,
mem(FM) — memory full-rank solver in kb.

Conclusions & Outlook

- We have presented a low-rank Newton-type algorithm for approximating selected eigenpairs of SEVPs.
- The low-rank approach provides significant storage savings.
- For details, see:
  Peter Benner, Akwum Onwunta, and Martin Stoll:
  An Inexact Newton-Krylov Method for Stochastic Eigenvalue Problems.
  Computational Methods in Applied Mathematics (CMAM), to appear.

  Preliminary version: arXiv:1710.09470, October 2017.

Conclusions & Outlook

- We have presented a low-rank Newton-type algorithm for approximating selected eigenpairs of SEVPs.
- The low-rank approach provides significant storage savings.
- For details, see:
  Peter Benner, Akwum Onwunta, and Martin Stoll:
  An Inexact Newton-Krylov Method for Stochastic Eigenvalue Problems.
  Computational Methods in Applied Mathematics (CMAM), to appear.

  Preliminary version: arXiv:1710.09470, October 2017.
- **Next steps:**
  - Block-variants to approximate several eigenpairs at once.
  - Non-symmetric stochastic eigenvalue problems.

This is the end.