Projection method for approximating the minimal eigenvalue of parametrized symmetric matrices

Koen Ruymbeek
Supervisors: Karl Meerbergen, Wim Michiels

Notation

1. $\lambda_i, \mathbf{x}_i, i = 1, \ldots, n$ and $(\lambda_i, \mathbf{x}_i)$ are respectively **eigenvalue, eigenvector** and **eigenpair** of $(\mathbf{A}, \mathbf{B})$ if
$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{B}\mathbf{x}_i, \mathbf{x}_i \neq 0$$
2. Compact parameterspace $\Omega \subset \mathbb{R}^d$
3. $\mathbf{A}(\omega), \mathbf{B}(\omega) \in \mathbb{R}^{n \times n}$ matrices depending on $\omega \in \Omega$,
   - Non-singular
   - Elementswise differentiable
   - symmetric, $\mathbf{B}$ positive-definite
   - n is large

Restrictions on matrices

1. $\lambda_1, \lambda_2, \ldots, \lambda_n$
   - real
   - $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$
   - smooth (if eigenvalue is simple (= muliplicity is 1))
2. $x_1(\omega), \ldots, x_n(\omega)$
   - form an $\mathbf{B}$-orthonormal basis ($\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j = 1$ if $i = j, = 0$ if $i \neq j$
   - smooth (if they are simple)
3. $\left(\hat{\lambda}(\omega), \hat{\mathbf{x}}(\omega)\right)$ estimation of eigenpair, then

$$\min_{\lambda(\omega) \in \Lambda(\mathbf{A}(\omega), \mathbf{B}(\omega))} \mid \tilde{\lambda}(\omega) - \lambda(\omega) \mid \leq \frac{\|\mathbf{r}(\omega)\|_2}{\sqrt{\lambda_1(\mathbf{B}(\omega))}}.$$

references: Peter Lax - Linear Algebra and its applications, Yousef Saad - Numerical methods for large eigenvalue problems

Goal

Efficient global approximation of $\lambda_1(\omega)$ over Ω.

1. $\lambda_1(\omega)$ can be non-smooth, if multiplicity $\lambda_1 > 1 \rightarrow$ Polynomial approximation is not well-suited
2. Looking for a method
   - Can deal with non-smoothness
   - as less calculations of λ_1 as possible
3. We choose projection methods

Projection method

1. $\mathcal{V} = \text{span}\{\mathbf{v}_i | \mathbf{V}(:, i) = \mathbf{v}_i, i = 1, \ldots, m\} \subset \mathbb{R}^n$ with $\mathbf{V}$ orthonormal matrix
2. $(\lambda_1^{\mathcal{V}}, \mathbf{x}_1^{\mathcal{V}}), \ldots, (\lambda_m^{\mathcal{V}}, \mathbf{x}_m^{\mathcal{V}})$ eigenpair of $\left(\mathbf{V}^T\mathbf{A}(\omega)\mathbf{V}, \mathbf{V}^T\mathbf{B}(\omega)\mathbf{V}\right)$ (= projected eigenvalue problem)
3. deals with non-smoothness

Projection method

1. $\mathcal{V} = \text{span}\{\mathbf{v}_i | \mathbf{V}(:, i) = \mathbf{v}_i, i = 1, \ldots, m\} \subset \mathbb{R}^n$ with $\mathbf{V}$ orthonormal matrix
2. $(\lambda_1^{\mathcal{V}}, \mathbf{x}_1^{\mathcal{V}}), \ldots, (\lambda_m^{\mathcal{V}}, \mathbf{x}_m^{\mathcal{V}})$ eigenpair of $\left(\mathbf{V}^T\mathbf{A}(\omega)\mathbf{V}, \mathbf{V}^T\mathbf{B}(\omega)\mathbf{V}\right)$ (= projected eigenvalue problem)
3. deals with non-smoothness

Property

If $\mathcal{V}_1 \subset \mathcal{V}_2$ *then* $|\lambda_1(\omega) - \lambda_1^{\mathcal{V}_2}(\omega)| \leq |\lambda_1(\omega) - \lambda_1^{\mathcal{V}_1}(\omega)|$

Property

$\lambda_1^{\mathcal{V}}(\omega) \geq \lambda_1(\omega), \forall \omega \in \Omega$

Building $\mathcal{V}$

1. Goal: Make $\mathcal{V}$ such that
   - $\lambda_1^{\mathcal{V}}(\omega) \approx \lambda_1(\omega), \forall \omega \in \Omega$ (minimal eigenvalue of projected problem $\approx$ minimal eigenvalue of large problem)
   - $\dim(\mathcal{V})$ as low as possible
2. Idea is similar to Reduced Basis method

Building $\mathcal{V}$

1. Goal: Make $\mathcal{V}$ such that
   - $\lambda_1^{\mathcal{V}}(\omega) \approx \lambda_1(\omega), \forall \omega \in \Omega$ (minimal eigenvalue of projected problem $\approx$ minimal eigenvalue of large problem)
   - $\dim(\mathcal{V})$ as low as possible
2. Idea is similar to Reduced Basis method

3. Let $\Omega_{\text{init}} := \{\omega^1, \omega^2, \ldots, \omega^{n_0}\}$ be an initial set
4. $\mathcal{V}$ consists of
   - $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - Partial derivatives of $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - $\mathbf{x}_j(\omega^i), i = 1, \ldots, n_0, j = 2, \ldots, n_1$

Building $\mathcal{V}$

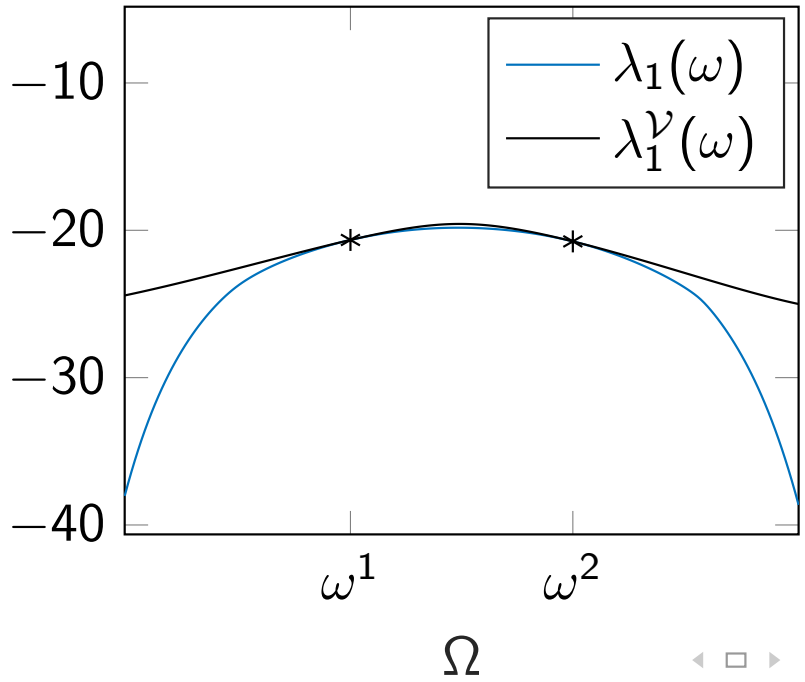
1. Let $\Omega_{\text{init}} := \{\omega^1, \omega^2, \ldots, \omega^{n_0}\}$ be an initial set
2. $\mathcal{V}$ consists of
   - $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - Partial derivatives of $\mathbf{x}_1(\omega^{\mathbf{i}}), i = 1, \ldots, n_0$
   - $\mathbf{x}_j(\omega^i), i = 1, \ldots, n_0, j = 2, \ldots, n_1$

Adding $\mathbf{x}_1(\omega)$

Property (Hermite Interpolation, degree 1)

Let $(\lambda(\omega), \mathbf{x}(\omega))$ *be an eigenpair of* $(\mathbf{A}(\omega), \mathbf{B}(\omega))$, $\lambda(\omega)$ *simple and let* $\mathbf{x} \in \mathcal{V}$, *then*

1. $\left(\lambda^{\mathcal{V}}(\omega), \mathbf{x}^{\mathcal{V}}(\omega)\right) = \left(\lambda(\omega), \mathbf{V}^T\mathbf{x}(\omega)\right)$ *(interpolation of eigenvalue).*
2. $\dfrac{\partial \lambda^{\mathcal{V}}(\omega)}{\partial \omega} = \dfrac{\partial \lambda(\omega)}{\partial \omega}$ *(interpolation of the derivative)*



Building $\mathcal{V}$

1. Let $\Omega_{\text{init}} := \{\omega^1, \omega^2, \ldots, \omega^{n_0}\}$ be an initial set
2. $\mathcal{V}$ consists of
   - $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - Partial derivatives of $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - $\mathbf{x}_j(\omega^i), i = 1, \ldots, n_0, j = 2, \ldots, n_1$

Adding partial derivatives

Property (Hermite Interpolation, degree 2)

Let $(\lambda(\omega), \mathbf{x}(\omega))$ *be eigenpair of* $(\mathbf{A}(\omega), \mathbf{B}(\omega))$ *and let* $\mathbf{x}, \dfrac{\partial \mathbf{x}}{\partial \omega_i} \in \mathcal{V}$ *and* $\lambda(\omega)$ *is simple, then*

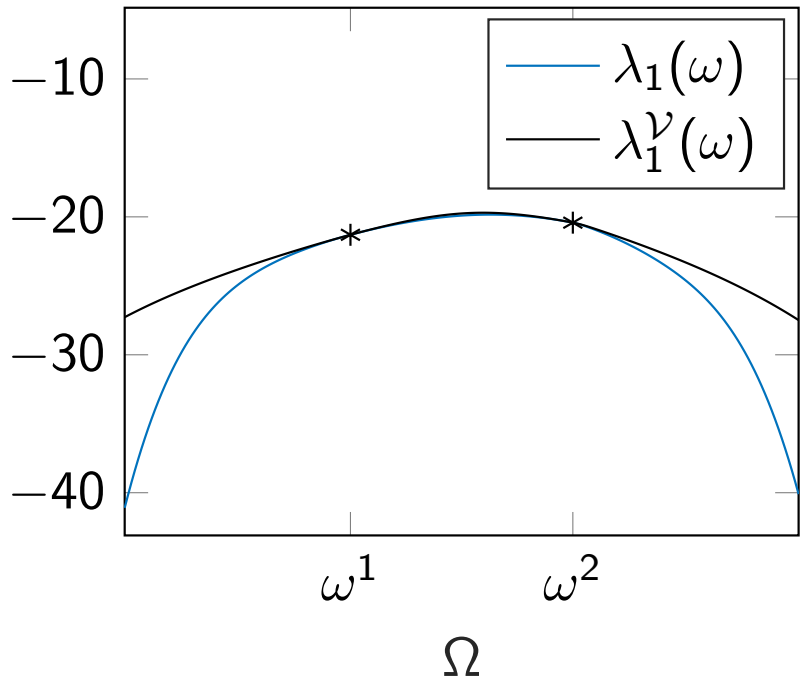
$$\frac{\partial^2 \lambda^{\mathcal{V}}(\omega)}{\partial \omega_i \partial \omega_j} = \frac{\partial^2 \lambda(\omega)}{\partial \omega_i \partial \omega_j}, \forall j = 1, \ldots, d$$

Adding partial derivatives

Property (Hermite Interpolation, degree 2)

Let $(\lambda(\omega), \mathbf{x}(\omega))$ *be eigenpair of* $(\mathbf{A}(\omega), \mathbf{B}(\omega))$ *and let* $\mathbf{x}, \dfrac{\partial \mathbf{x}}{\partial \omega_i} \in \mathcal{V}$ *and* $\lambda(\omega)$ *is simple, then*

$$\frac{\partial^2 \lambda^{\mathcal{V}}(\omega)}{\partial \omega_i \partial \omega_j} = \frac{\partial^2 \lambda(\omega)}{\partial \omega_i \partial \omega_j}, \forall j = 1, \ldots, d$$



Calculate $\frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_i}$

Property

Let $(\lambda(\omega), \mathbf{x}(\omega))$ *be eigenpair of* $(\mathbf{A}(\omega), \mathbf{B}(\omega))$ *and* $\lambda(\omega)$ *is simple, then*

$$\frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_j} = -\frac{\left(\mathbf{x}_1(\omega), \frac{\partial \mathbf{B}(\omega)}{\partial \omega_j}\mathbf{x}_1(\omega)\right)}{2}\mathbf{x}_1(\omega) + \sum_{i=2}^{n} \frac{\left(\mathbf{x}_i(\omega), \left(\frac{\partial \mathbf{A}(\omega)}{\partial \omega_j} - \lambda_1(\omega)\frac{\partial \mathbf{B}(\omega)}{\partial \omega_j}\right)\mathbf{x}_1(\omega)\right)}{\lambda_1(\omega) - \lambda_i(\omega)}\mathbf{x}_i(\omega)$$

Calculate $\frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_i}$

Property

Let $(\lambda(\omega), \mathbf{x}(\omega))$ *be eigenpair of* $(\mathbf{A}(\omega), \mathbf{B}(\omega))$ *and* $\lambda(\omega)$ *is simple, then*

$$\frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_j} = -\frac{\left(\mathbf{x}_1(\omega), \frac{\partial \mathbf{B}(\omega)}{\partial \omega_j}\mathbf{x}_1(\omega)\right)}{2}\mathbf{x}_1(\omega) + \sum_{i=2}^{n} \frac{\left(\mathbf{x}_i(\omega), \left(\frac{\partial \mathbf{A}(\omega)}{\partial \omega_j} - \lambda_1(\omega)\frac{\partial \mathbf{B}(\omega)}{\partial \omega_j}\right)\mathbf{x}_1(\omega)\right)}{\lambda_1(\omega) - \lambda_i(\omega)}\mathbf{x}_i(\omega)$$

1. If $\mathbf{B}(\omega)$ independent of ω_j : $\mathbf{x}_1 \mathbf{B} \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_j} = 0$
2. Depends most on eigenpairs close to $(\lambda_1(\omega), \mathbf{x}_1(\omega))$
3. Not practical to use, calculate derivative by solving a system

Building $\mathcal{V}$

1. Let $\Omega_{\text{init}} := \{\omega^1, \omega^2, \ldots, \omega^{n_0}\}$ be an initial set
2. $\mathcal{V}$ consists of
   - $\mathbf{x}_1(\omega^i), i = 1, \ldots, n_0$
   - Partial derivatives of $\mathbf{x}_1(\omega^{\mathbf{i}}), i = 1, \ldots, n_0$
   - $\mathbf{x}_j(\omega^i), i = 1, \ldots, n_0, j = 2, \ldots, n_1$

Adding $\mathbf{x}_j(\omega), j = 2, \ldots, n_1$: idea

- $\mathbf{x}_1(\omega^2) \approx \mathbf{x}_2(\omega^1)$

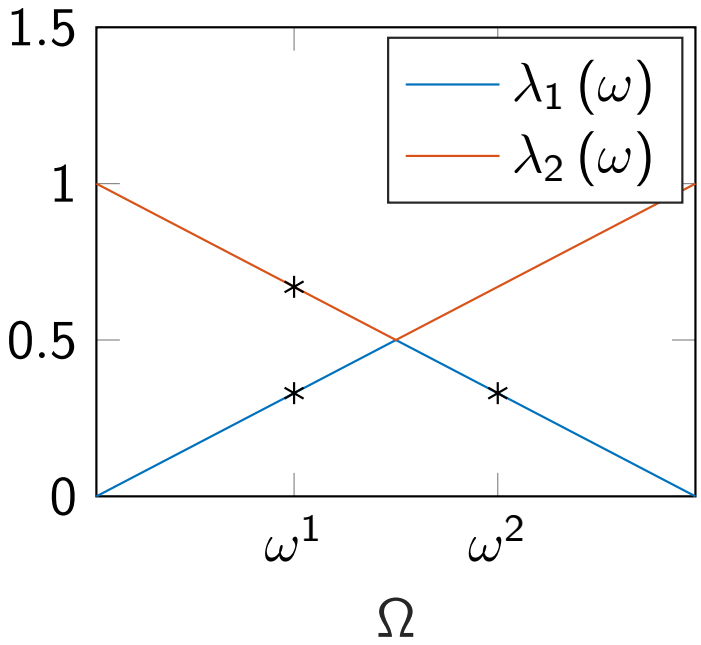


Figure: $n_1 = 2$

Adding $\mathbf{x}_j(\omega), j = 2, \ldots, n_1$: idea

- $\mathbf{x}_1(\omega^2) \approx \mathbf{x}_2(\omega^1)$
- $\mathcal{V}_1 := \text{span}\{\mathbf{x}_1(\omega^1), \mathbf{x}_1(\omega^2)\} \approx \text{span}\{\mathbf{x}_1(\omega^1), \mathbf{x}_2(\omega^1)\} =: \mathcal{V}_2$

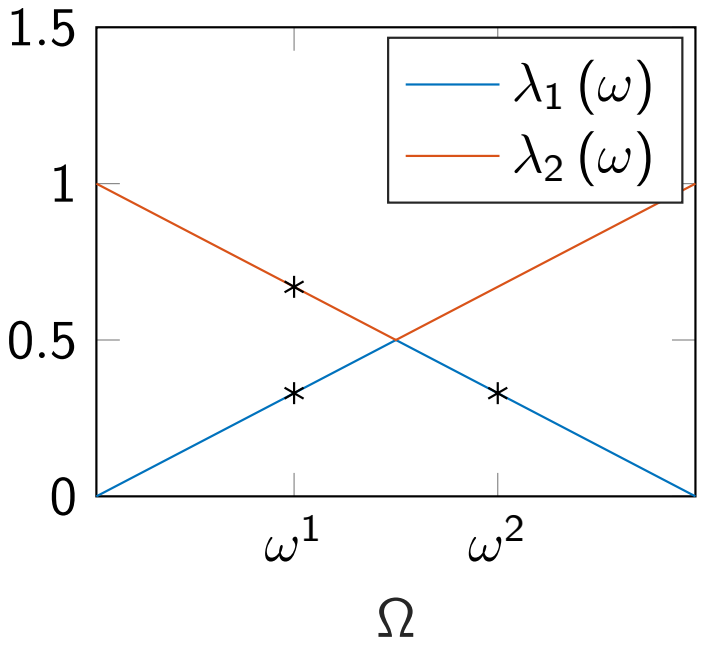


Figure: $n_1 = 2$

Adding $\mathbf{x}_j(\omega), j = 2, \ldots, n_1$: idea

- $\mathbf{x}_1(\omega^2) \approx \mathbf{x}_2(\omega^1)$
- $\mathcal{V}_1 := \text{span}\{\mathbf{x}_1(\omega^1), \mathbf{x}_1(\omega^2)\} \approx \text{span}\{\mathbf{x}_1(\omega^1), \mathbf{x}_2(\omega^1)\} =: \mathcal{V}_2$
- Difference:
  - $\mathcal{V}_1$ solving 2 large eigenvalue problems
  - $\mathcal{V}_2$ solving 1 large eigenvalue problem

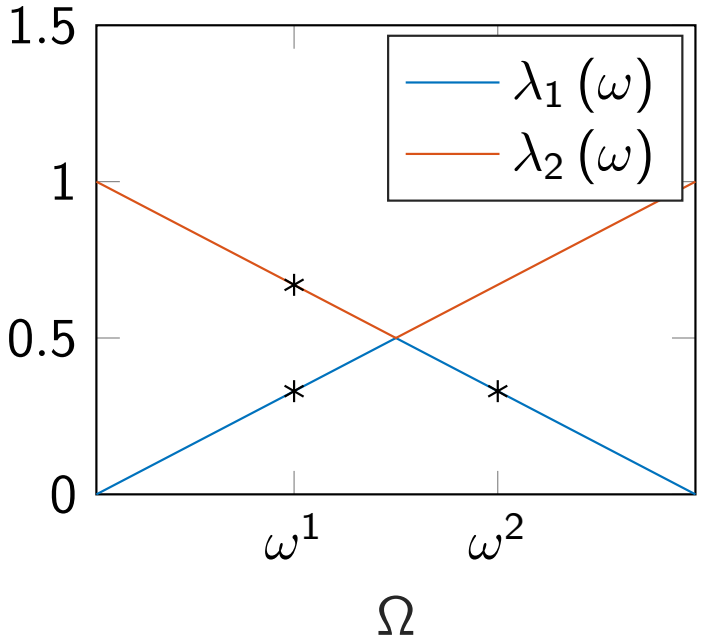


Figure: $n_1 = 2$

Adding $\mathbf{x}_j(\omega), j = 2, \ldots, n_1$: advantages

1. Good strategy if eigenvectors don't change much over the parameter
2. We can make a better upperbound for the error since

$$| \lambda_1^{\mathcal{V}}(\omega) - \lambda_1(\omega) | \leq \frac{\|\mathbf{r}(\omega)\|_2^2}{\lambda_1(\mathbf{B}(\omega))(\lambda_1^{\mathcal{V}}(\omega) - \lambda_2(\omega))} =: \mathbf{u}(\omega).$$

(= Kato-Temple theorem) where we take $\lambda_2(\omega) \approx \lambda_2^{\mathcal{V}}(\omega)$

Algorithm: idea

Algorithm used in Reduced basis method for solving PDE's

1. Ω_{init} and Ω_{train}
2. Make initial subspace

$$\mathcal{V} := \text{span}\{\mathbf{x}_1(\omega), \ldots, \mathbf{x}_m(\omega), \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_1}, \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_2}, \ldots, \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_d} | \omega \in \Omega_{\text{init}}\}$$

3. update $\mathcal{V}$ such that

$$\max_{\omega \in \Omega_{\text{train}}} |\lambda_1^{\mathcal{V}}(\omega) - \lambda_1(\omega)| < \text{tol}$$

$$\mathcal{V} := \text{span}\{\mathbf{x}_1(\omega), \ldots, \mathbf{x}_m(\omega), \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_1}, \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_2}, \ldots, \frac{\partial \mathbf{x}_1(\omega)}{\partial \omega_d} | \omega \in \Omega_{\text{init}}\}$$

for $i = 1 : n_{max}$ **do**

Update the trainingset

Calculate upperbound $\mathbf{u}(\omega)$ for all $\omega \in \Omega_{\text{train}}$

$\Omega_{\text{valid}} = \{\omega | \omega \in \Omega_{\text{train}} \text{ and } \mathbf{u}(\omega) < \text{tol}\}$

$\Omega_{\text{train}} = \Omega_{\text{train}} \setminus \Omega_{\text{valid}}$

if Ω_{train} *empty* **then**

break

else

Adding vectors to the subspace

$\omega^i = \arg\max_{\omega \in \Omega_{\text{train}}} u(\omega)$

if λ_1 *is simple* **then**

$$\mathcal{V} = \mathcal{V} \bigcup \text{span}\{\mathbf{x}_1(\omega^i), \ldots, \mathbf{x}_m(\omega^i), \frac{\partial \mathbf{x}_1(\omega)^i}{\partial \omega_1}, \frac{\partial \mathbf{x}_1(\omega^i)}{\partial \omega_2}, \ldots, \frac{\partial \mathbf{x}_1(\omega^{\mathbf{i}})}{\partial \omega_d}\}$$

else

$$\mathcal{V} = \mathcal{V} \bigcup \text{span}\{\mathbf{x}_1(\omega^i), \ldots, \mathbf{x}_m(\omega^i)\}$$

end

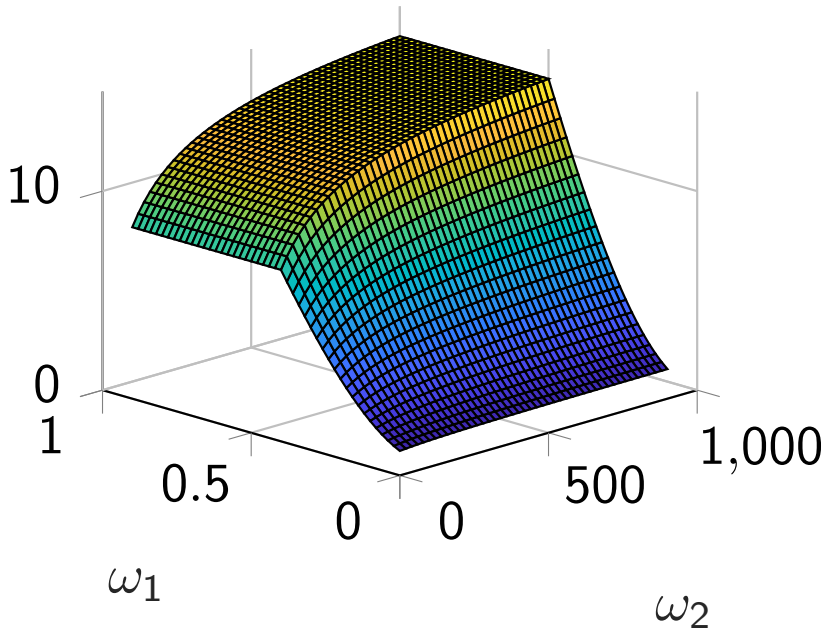
Make basis for $\mathcal{V}$

end

end

Example Structural Mechanics

1. Comes from a finite element discretisation
2. $\Omega = [0.1, 1] \times [100, 1000]$
3. Matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{1020 \times 1020}$, $|\Omega_{\text{init}}| = 9$, $|\Omega_{\text{train}}| = 1000$
4. tolerance: 10^{-5}



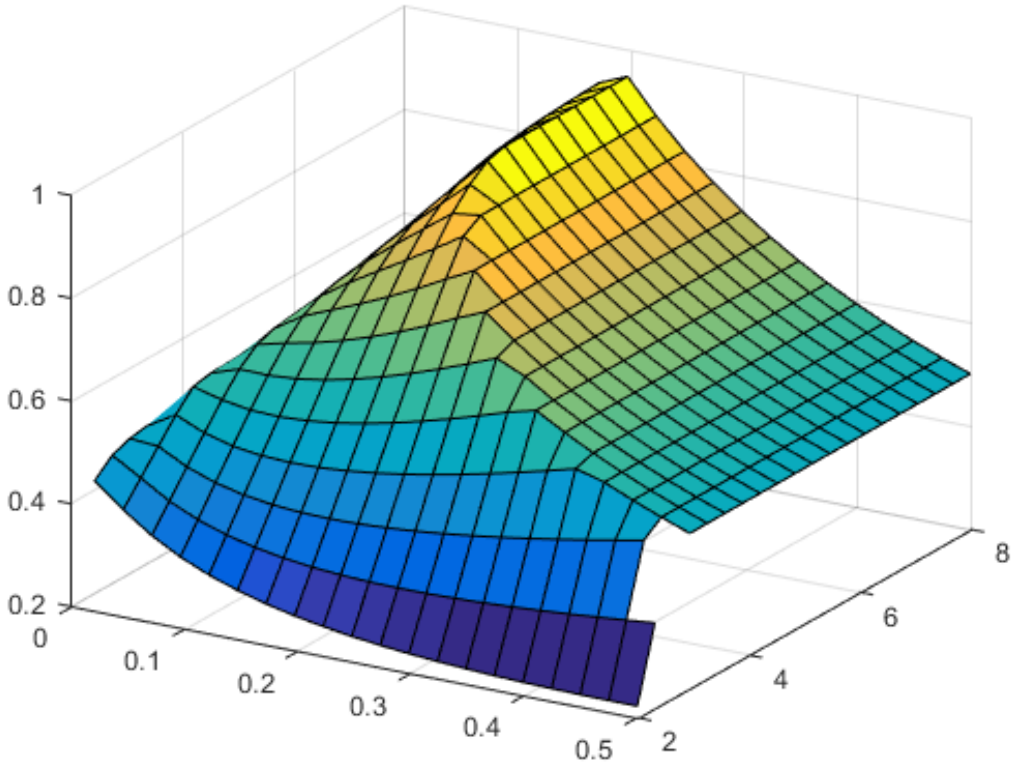
	2 eigv	2 eigv + part deriv
dimension $\mathcal{V}$	32	40
nbr points	16	10
total time	23.783	11.541
time spent on $\mathcal{V}$	0.572	0.413
vectors per second	55.950	96.808

Table: Results: In the first column we only added 2 eigenvectors per point, in the second column we added 2 eigenvectors per point and all partial derivatives

1. Dimension subspace is small
2. Adding derivatives reduces
   1. the number of points where we calculate the large eigenvalueproblem
   2. computational time

Example coercivity constant

1. $\Omega = [0.02, 0.5] \times [2, 8]$
2. Matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{1311 \times 1311}$, $|\Omega_{\text{init}}| = 16$, $|\Omega_{\text{train}}| = 1000$
3. tolerance: 10^{-5}



	2 eigv	2 eigv + part deriv
dimension $\mathcal{V}$	108	132
nbr points	54	33
total time (in s)	190.193	65.578
time spent on (in s) $\mathcal{V}$	4.176	2.935
vectors per second	25.859	44.967

Table: Results: In the first column we only added 2 eigenvectors per point, in the second column we added 2 eigenvectors per point and all partial derivatives

1. Dimension subspace is quite large
2. Adding derivatives reduces
   1. the number of points where we calculate the large eigenvalueproblem
   2. computational time

Conclusion

1. Projection using multiple minimal eigenvalues works
2. Adding derivative accelerate convergence
3. -: difficult to estimate how large needed subspace is