Optimization of Large-scale Systems with Model Reduction

Presenter: Yao Yue

Promotor: Karl Meerbergen

October 21, 2009

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Acoustic Cavity Design

A cubic acoustic cavity:

- One side: Admittance boundary condition.
- Other sides: Dirichlet boundary condition.
- Design Parameter: Admittance ratio.
- Design Objective: Minimize the highest peak of FRF at a given point.

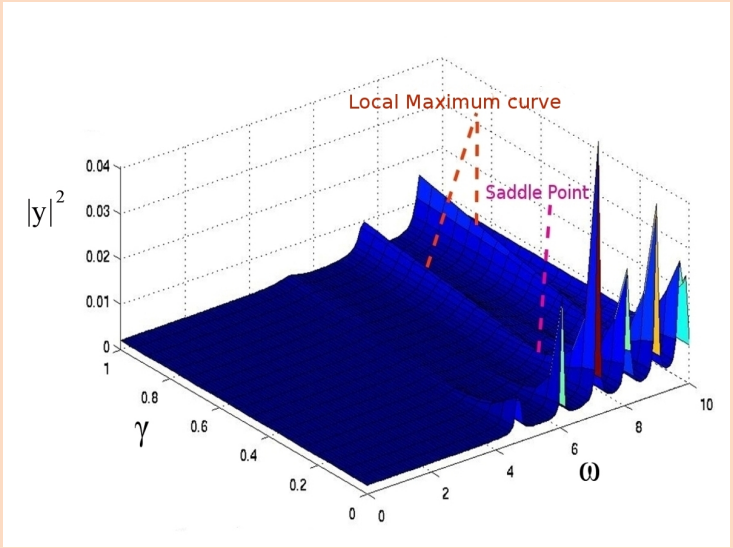
Mathematical Formulation

$$\min_{\gamma} \max_{\omega_L \le \omega \le \omega_H} |y(\omega,\gamma)|^2$$

$$\text{s.t.} \quad \begin{cases} y(\omega,\gamma) = \ell^* x(\omega,\gamma) \\ (K + i\omega\gamma C - \omega^2 M)x(\omega,\gamma) = f \end{cases}$$

Minimax Optimization

3D Figure



All local maximum curves are convex!

Optimizer

Local optimizer

$\Longleftrightarrow$ minimum of local maximum curve

$\Longleftrightarrow$ Saddle Point.

What about the global optimizer?

Global Optimizer

Project the local maximum curves to $|y|^2 - \gamma$ plane.



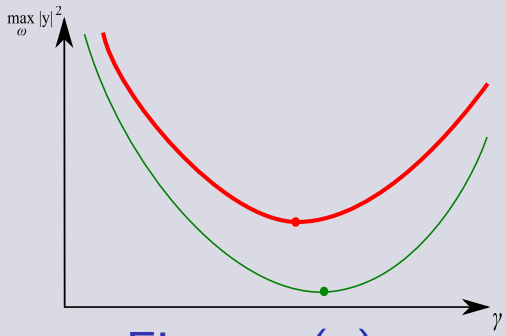


Figure: (a)

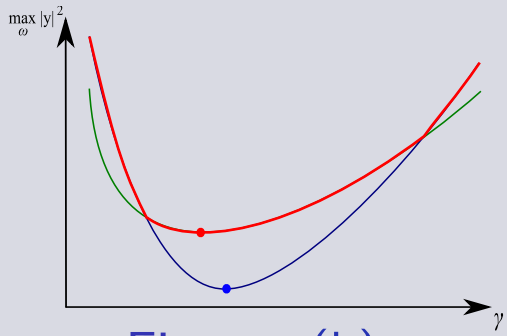


Figure: (b)

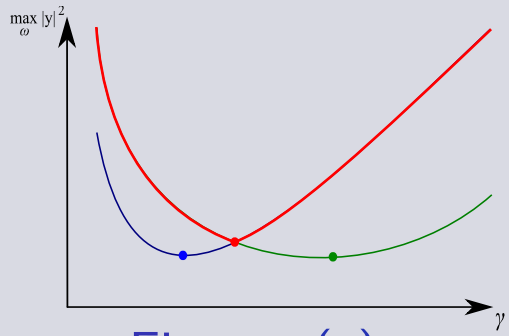


Figure: (c)

3 Possibilities:

(a) one dominant local maximum curve.

(b) no dominant curve, but global optimizer is a saddle point.

(c) global optimizer is a kink point.

Kink point: non-differentiable point that is the intersect of two curves.

Minimax Optimization

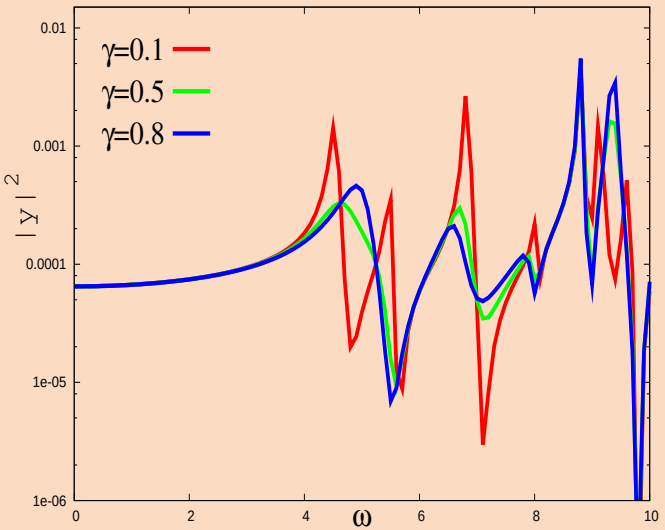
Two phases in Minimax Optimization

- **Min Phase:** Find the minimum of the global maximum curve: $\min_{\gamma} g(\gamma)$.
- **Max Phase:** Compute $g(\gamma)$ for a given γ_0: $g(\gamma_0) = \max_{\omega} |y(\omega, \gamma_0)|^2$, the global maximum for a given γ_0.

Algorithm

Max Phase: Highly nonconvex,
Many local optimizers.
Algorithm: Grid Search followed by
Quasi Newton refinement.
Locates only one global optimizer.

Min Phase: Convex.
Algorithm: Quasi Newton method.



Analysis of the Algorithm

- Jumping: "Jumping" between two local maximum curves is possible, but it is hidden to both the Min and the Max Phase.
- Kink Points: Kink point occurs when several local maximum curves are of the same height. Since we locate only one of them, we can continue with the derivative of curve found in the Min Phase without knowing the presence of kink points.
- Convergence: All Quasi Newton steps use backtracking strategy with Armijo condition. Convergence of the algorithm is guaranteed by the Armijo condition.
- Convergence rate:
  - No jumping occurs: Inherit superlinear convergence rate from Quasi Newton method.
  - Jumping occurs: The convergence rate of Quasi Newton method may degrade.
  - Kink Optimizer: Further degrade.

Algorithm: Stopping Conditions

Either of the following two conditions is satisfied.

1. $\left|\frac{\partial g}{\partial \gamma}\Big|_{\gamma=\gamma_k}\right| \leq \epsilon$: global optimizer is a saddle point.
2. $|\gamma_k - \gamma_{k-1}| \leq \tau$: global optimizer is a kink point.

Note: Near the kink optimizer, backtracking is used for many times to meet, or will never meet Armijo condition. Condition 2 pick out this possibility.

Computational Cost of the Algorithm

For each step in Inner Max Phase

$$y = \ell^*(K + i\omega\gamma C - \omega^2 M)^{-1} f$$

For each Quasi Newton step in Inner Max Phase

$$\frac{\partial y}{\partial \omega} = \ell^*(K + i\omega\gamma C - \omega^2 M)^{-1}(2\omega M - i\gamma C)(K + i\omega\gamma C - \omega^2 M)^{-1} f$$

For each step in the Outer Min Phase

$$\frac{\partial y}{\partial \gamma} = \ell^*(K + i\omega\gamma C - \omega^2 M)^{-1}(-i\omega C)(K + i\omega\gamma C - \omega^2 M)^{-1} f$$

Minimax Optimization

Observation

- When the system order is large, computational costs of y, $\frac{\partial y}{\partial \omega}$ and $\frac{\partial y}{\partial \gamma}$ are very expensive!
- Computations of derivatives are frequent.

Our Solution

Use Model Order Reduction (MOR) in optimization.

- It is well known that MOR can be used for function evaluations.
- We show MOR can also be used for derivative computations.

All data needed by the minimax algorithm can be computed with MOR, optimization time will be greatly reduced!

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Basic Idea

Project a large scale linear system onto a Krylov subspace to get a reduced order linear system.

Krylov subspace

Krylov subspace:

$$\mathcal{K}_k\{A, b\} \triangleq \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}.$$

Generate it in a numerically stable way:

Arnoldi Process, Lanczos Method.

Arnoldi Process

$v_1 = b/\|b\|$
For $i = 2 \ldots k$
$w_i = Av_{i-1}$
Orthogonalize w_i to $v_1 \ldots v_{i-1}$, and normalize it to get v_i

Linear system with no design parameters

$$\begin{cases} (K - \alpha M)x = f \\ y = \ell^* x \end{cases} \qquad \textit{order } n \textit{ system}$$

MOR on this system

Let $V_k = [v_1, v_2, \ldots, v_k]$, $W_k = [w_1, w_2, \ldots, w_k] \in \mathbb{C}^{n \times k}$.
Approximate $x \approx V_k z \in \text{colspan}\{V_k\}$ and left multiply the first equation by W^*

$$\begin{cases} (\hat{K} - \alpha \hat{M})z = \hat{f} \\ \hat{y} = \hat{\ell}^* z \end{cases} \qquad \begin{array}{l} \text{order } k \text{ system} \\ k \ll n! \end{array}$$

where $\hat{K} = W^* K V$, $\hat{M} = W^* M V$, $\hat{f} = W^* f$, $\hat{\ell} = V^* \ell$.

Question: What about the accuracy?

KATHOLIEKE UNIVERSITEIT LEUVEN

Definition of Moment

Let $y = \sum_{i=0}^{\infty} m_i \alpha^i$, $\hat{y} = \sum_{i=0}^{\infty} \hat{m}_i \alpha^i$

m_i, $\hat{m}_i$ are called the i-th moment of y and $\hat{y}$ respectively.

Accuracy of Krylov Method: Moment Matching Property

Theorem:

1. If colspan$\{V_k\} \supseteq$ the k-th right Krylov subspace $\mathcal{K}_k(K^{-1}M, K^{-1}b)$, the first k moments of y and $\hat{y}$ match;
2. If colspan$\{W_k\} \supseteq$ the k-th left Krylov subspace $\mathcal{K}_k(K^{-*}M^*, K^{-*}\ell^*)$, the first k moments of y and $\hat{y}$ match;
3. If colspan$\{V_k\} \supseteq$ the k-th right Krylov subspace and colspan$\{W_k\}$ spans the k-th left Krylov subspace $\mathcal{K}_k(K^{-*}M^*, K^{-*}\ell^*)$, the first $2k$ moments of y and $\hat{y}$ match.

Case 1 and 2 are called one-sided methods.
Case 3 is called two-sided method.

Challenge of Optimization on MOR

Recall the minimax optimization problem:

$$\min_{\gamma} \max_{\omega_L \le \omega \le \omega_H} |y(\omega,\gamma)|^2$$
$$\text{s.t.} \quad \begin{cases} y(\omega,\gamma) = \ell^* x(\omega,\gamma) \\ (K + i\omega\gamma C - \omega^2 M)x(\omega,\gamma) = f \end{cases}$$

It contains two parameters: frequency ω and design parameter γ.

How to use MOR for this system with 2 parameters?

Two Approaches

1. For each Max Phase, fix $\gamma = \gamma_0$ and do MOR with ω.
   - Representative: SOAR.
2. Parameterized MOR: Do MOR on both γ and ω.
   - Representative: PIMTAP.

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Introduction

SOAR stands for Second Order ARnoldi Process [Bai & Su, 05].

Conventional Method: Arnoldi process on the linearized system.

Advantages of SOAR:

1. More accurate.
2. Structure preserving.

SOAR MOR

Project the system onto a "2nd Order Krylov Subspace".

$$\underbrace{\begin{cases} y = \ell^* x \\ (K + i\omega\gamma_0 C - \omega^2 M)x = f \end{cases}}_{\text{Order } n \text{ System}} \xrightarrow[n \gg k]{SOAR} \underbrace{\begin{cases} \hat{y} = \hat{\ell}^* z \\ (\hat{K} + i\omega\gamma_0 \hat{C} - \omega^2 \hat{M})z = \hat{f} \end{cases}}_{\text{Order } k \text{ System}}$$

$\hat{K} = W^* K V$, $\hat{C} = W^* C V$, $\hat{M} = W^* M V$, $\hat{f} = W^* f$ and $\hat{\ell} = V^* \ell$.

KATHOLIEKE UNIVERSITEIT LEUVEN

The idea of SOAR

The Linearized System:

$$\begin{cases} \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -iC & M \\ I & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ \omega x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \\ y = [l^*, 0] \begin{bmatrix} x \\ \omega x \end{bmatrix} \end{cases}$$

Krylov Vectors:

$$\begin{bmatrix} v_{i+1}^{(1)} \\ v_{i+1}^{(2)} \end{bmatrix} = \begin{bmatrix} K^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} -iC & M \\ I & 0 \end{bmatrix} \begin{bmatrix} v_i^{(1)} \\ v_i^{(2)} \end{bmatrix} = \begin{bmatrix} -iK^{-1}Cv_i^{(1)} + K^{-1}Mv_i^{(2)} \\ v_i^{(1)} \end{bmatrix}$$

$$v_0 = \begin{bmatrix} K^{-1}f \\ 0 \end{bmatrix} \qquad \text{Note: } v_{i+1}^{(2)} = v_i^{(1)}$$

Arnoldi Projector: $\begin{bmatrix} Q_k \\ P_k \end{bmatrix}$, satisfying $\text{colspan}\{P_k\} \subseteq \text{colspan}\{Q_k\}$

SOAR Projector:

$\begin{bmatrix} Q_k & 0 \\ 0 & Q_k \end{bmatrix}$, satisfying $\text{colspan}\left\{ \begin{bmatrix} Q_k & 0 \\ 0 & Q_k \end{bmatrix} \right\} \supseteq \text{colspan}\left\{ \begin{bmatrix} Q_k \\ P_k \end{bmatrix} \right\}$,

where $\text{colspan}\{Q_k\}$ is called the k-th second order Krylov subspace.

Advantage of SOAR

- More Accurate:

  Project the system onto a subspace bigger than the one generated by Arnoldi process.

- Structure Preserving:

$$\begin{cases} \begin{bmatrix} W_k^* & 0 \\ 0 & W_k^* \end{bmatrix} \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -iC & M \\ I & 0 \end{bmatrix} \right) \begin{bmatrix} Q_k & 0 \\ 0 & Q_k \end{bmatrix} \begin{bmatrix} z \\ \omega z \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \\ y = [I^*, 0] \begin{bmatrix} x \\ \omega x \end{bmatrix} \end{cases}$$

  where W_k is the left 2nd-order Krylov subspace projector.

  Zero blocks are preserved.

  Identity blocks are both changed to the $W_k^* Q_k$.

Moment Matching Property

Theorem:

1. If colspan$\{V_k\} \supseteq$ the k-th right 2nd order Krylov subspace, the first k moments of y and $\hat{y}$ match;
2. If colspan$\{W_k\} \supseteq$ the k-th left 2nd order Krylov subspace, the first k moments of y and $\hat{y}$ match;
3. If colspan$\{V_k\} \supseteq$ the k-th right 2nd order Krylov subspace and colspan$\{W_k\} \supseteq$ the k-th left 2nd order Krylov subspace, the first $2k$ moments of y and $\hat{y}$ match.

Case 1 and 2 are called one-sided SOAR [Bai & Su, 05].
Case 3 is called two-sided SOAR [Salimbahrami & Lohmann, 06]

SOAR generates the 2-nd order Krylov subspaces in a numerically stable way.

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

PIMTAP MOR

PIMTAP: Parameterized Interconnect Macromodeling via a Two-directional Arnoldi Process [Li, Bai, Su & Zeng, 08].

- Could do MOR on more than one parameters.
- Structure preserving; relatively numerically stable.

Prototype:

$$\begin{cases} \big(G_0 + \gamma G_1 + s(C_0 + \gamma C_1)\big)x = b \\ y = \ell^* x \end{cases}$$

Order n System

$$\Downarrow$$

$$n \gg k$$

$$\begin{cases} \big(\hat{G}_0 + \gamma \hat{G}_1 + s(\hat{C}_0 + \gamma \hat{C}_1)\big)z = \hat{b} \\ \hat{y} = \hat{\ell}^* z \end{cases}$$

Order k System

$$[\hat{G}_0, \hat{G}_1, \hat{C}_0, \hat{C}_1] = V^*[G_0, G_1, C_0, C_1]V, \quad [\hat{b}, \hat{\ell}] = V^*[b, \ell].$$

V: generated by PIMTAP algorithm.

Multiparameter moments

Let $x = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} r_i^j s^i \gamma^j$,

r_i^j is called the (i, j)-th 2-parameter moment w.r.t x.

Moment Matching Pattern

PIMTAP generates the projection matrix V according to Moment Matching Pattern.

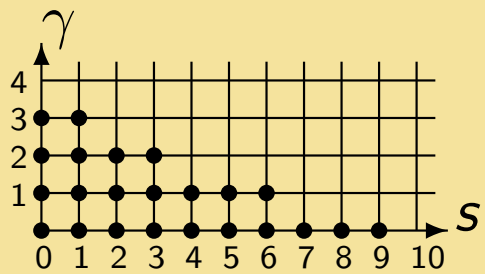


Figure: Example of Moment Matching Pattern.

All moments specified by the moment matching pattern will be matched for x and z (y and $\hat{y}$)!

Case of Square Pattern

Define

$$r_{[i]}^{[j]} = \begin{bmatrix} r_{i-1}^{0} \\ r_{i-1}^{1} \\ \vdots \\ r_{i-1}^{j-1} \end{bmatrix}, \quad G_{[j]} = \underbrace{\begin{bmatrix} G_0 & & & & \\ G_1 & G_0 & & & \\ & G_1 & G_0 & & \\ & & \ddots & \ddots & \\ & & & G_1 & G_0 \end{bmatrix}}_{j \text{ blocks}}, \quad C_{[j]} = \underbrace{\begin{bmatrix} C_0 & & & & \\ C_1 & C_0 & & & \\ & C_1 & C_0 & & \\ & & \ddots & \ddots & \\ & & & C_1 & C_0 \end{bmatrix}}_{j \text{ blocks}}$$

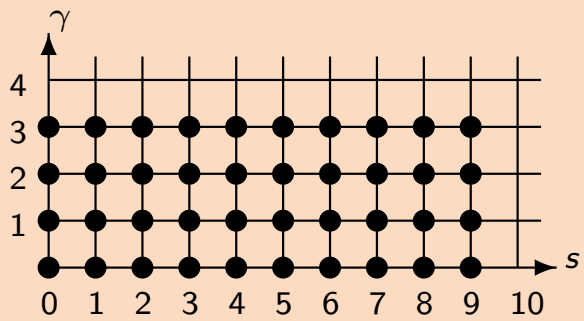


Figure: Square Pattern

We have

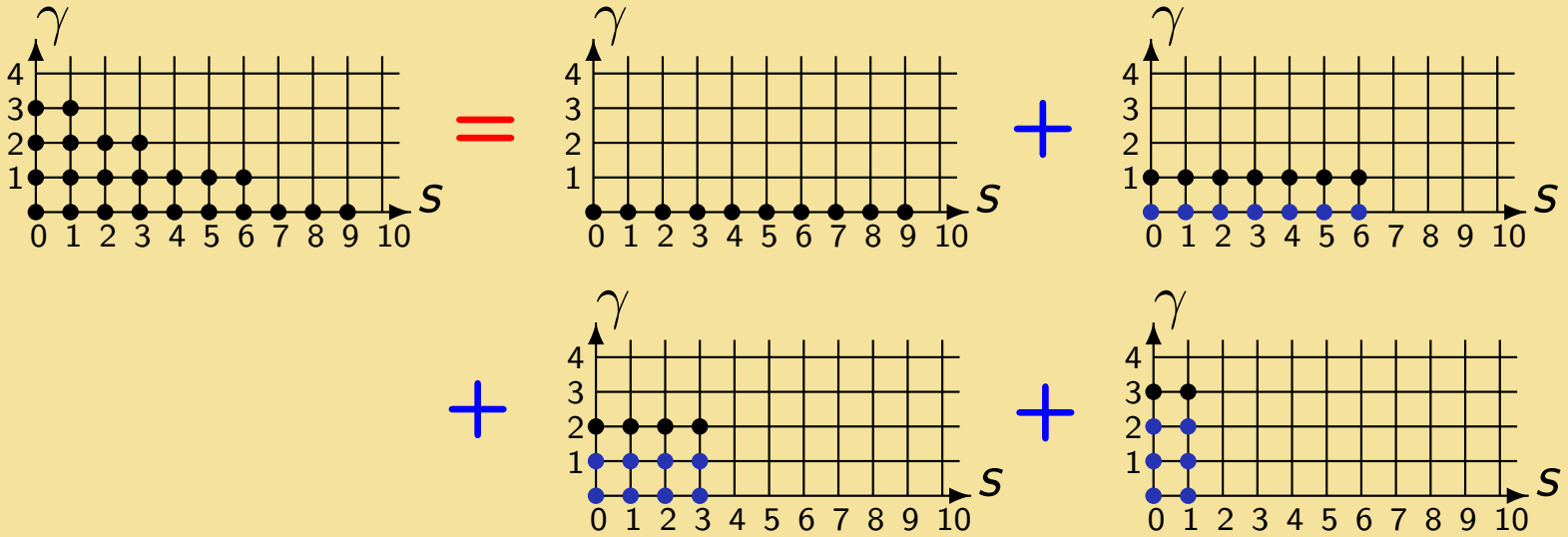
$$r_{[i]}^{[j]} = -G_{[j]}^{-1} C_{[j]} r_{[i-1]}^{[j]}, \quad \text{for all } i > 1.$$

Can be generated by Arnoldi Process!

For the case in the figure: $\mathcal{K}_{10}\left(-G_{[4]}^{-1} C_{[4]}, r_{[1]}^{[4]}\right)$

KATHOLIEKE UNIVERSITEIT LEUVEN

More General Pattern



More general pattern $\longrightarrow$ Union of square patterns.

Problem: The blue moments are recomputed.

Solution: TAP [Li, Bai & Su, 09].
Recycle the already computed moments.

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Using MOR in Minimax Optimization

Importance of Derivative Computations

- In the minimax optimization algorithm, derivative computations cost a lot of time.
- If we can compute derivatives via the reduced model, the optimization time can further be saved.

SOAR case

SOAR prototype

$$\begin{cases} y = \ell^* x \\ (K + i\omega\gamma_0 C - \omega^2 M)x = f \end{cases}$$

can be directly used for the Max Phase.

Derivative Computation via Two-sided SOAR

Theorem

If two-sided SOAR is used and the order of the reduced model is k:

1. The first $2k-1$ moments of $\frac{\partial y}{\partial \omega}$ and $\frac{\partial \hat{y}}{\partial \omega}$ match.
2. The first k moments of $\frac{\partial y}{\partial \gamma}$ and $\frac{\partial \hat{y}}{\partial \gamma}$ match.

Reason

1. $$y = \sum_{i=0}^{\infty} m_i \omega^i \quad \longrightarrow \quad \frac{\partial y}{\partial \omega} = \sum_{i=0}^{\infty} (i+1) m_{i+1} \omega^i$$

2. $$\left.\frac{\partial y}{\partial \gamma}\right|_{\gamma=\gamma_0} = \underbrace{\ell^*(K + i\omega\gamma_0 C - \omega^2 M)^{-1}}_{\text{Matched by Left Krylov Subspace}}(-i\omega C)\underbrace{(K + i\omega\gamma_0 C - \omega^2 M)^{-1} f}_{\text{Matched by Right Krylov Subspace}}$$

$$\frac{\partial y}{\partial \omega} = [I, 0]^* \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -i\gamma C & M \\ I & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} -i\gamma C & M \\ I & 0 \end{bmatrix} \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -i\gamma C & M \\ I & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} f \\ 0 \end{bmatrix}$$

$$\frac{\partial y}{\partial \gamma} = [I, 0]^* \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -i\gamma C & M \\ I & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} -i\omega C & 0 \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -i\gamma C & M \\ I & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} f \\ 0 \end{bmatrix}$$

No moment matching property for $\frac{\partial y}{\partial \gamma}$ and $\frac{\partial \hat{y}}{\partial \gamma}$ in one-sided SOAR!

KATHOLIEKE UNIVERSITEIT LEUVEN

PIMTAP case

PIMTAP can be used in the Minimax Algorithm by $G_0 \leftarrow K$, $G_1 \leftarrow C$, $C_0 \leftarrow M$, $C_1 \leftarrow 0$, $\gamma \leftarrow i\omega\gamma$, $s \leftarrow -\omega^2$ and $b \leftarrow f$.

$$\begin{cases} (G_0 + \gamma G_1 + s(C_0 + \gamma C_1))x = b \\ y = \ell^* x \end{cases} \longrightarrow \begin{cases} (K + i\omega\gamma C - \omega^2 M)x = f \\ y = \ell^* x \end{cases}$$

Theorem The (i, j)-th moments of ∇y and $\nabla \hat{y}$ match if the $(i+1, j)$-th and the $(i, j+1)$-th moments match for y and $\hat{y}$.

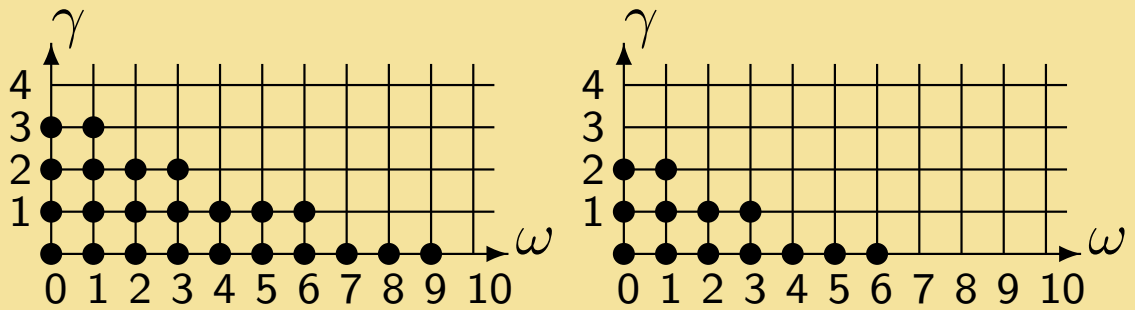


Figure: matched moments of y and ∇y in PIMTAP

Conclusion

Both two-sided SOAR and PIMTAP can compute the gradients with accuracy.

KATHOLIEKE UNIVERSITEIT LEUVEN

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Numerical Results

Table: Example 1

	Direct method	SOAR	PIMTAP
Matrix size	15625	50	61
Optimizer computed	(10.1245, 0.2671)	(10.1245, 0.2671)	(10.1245, 0.2671)
CPU time	3508s	128s	22s

A relatively large example.
The optimizer is a saddle point.

Numerical Results

Table: Example 2

	Direct method	SOAR	PIMTAP
Matrix size	125	16	19
Optimizer computed	(8.8092, 0.9951)	(8.8092, 0.9951)	(8.8092, 0.9951)
CPU time	31s	1s	2s

A relatively small example.

The optimizer is a saddle point.

Numerical Results

Table: Example 3

	Direct method	SOAR	PIMTAP
Matrix size	216	35	44
Optimizer computed	(8.668, 0.2687)	(8.669, 0.2687)	(8.668, 0.2688)
CPU time	162s	27s	18s

A relatively small example.
The optimizer is a kink point.

KATHOLIEKE UNIVERSITEIT LEUVEN

Outline

1. Minimax Optimization
2. Introduction to Krylov based MOR
3. SOAR
4. PIMTAP
5. Using MOR in Minimax Optimization
6. Numerical Results
7. Conclusion

Conclusion

- MOR could drastically reduce the optimization time for large scale systems with high accuracy.
- Even for relatively small scale systems, MOR could also reduce the optimization time.
- In general, PIMTAP is more efficient than SOAR for large scale problems.

Thank you!