Algorithms for eigenvalue problems arising in model reduction

Joost Rommes
Mentor Graphics

MorTrans 2015, Berlin
May 19, 2015
Introduction

Eigenvalue problems

Stability analysis and spurious eigenvalues

Partitioning

Eigenanalysis for model order reduction

Concluding remarks
Acknowledgments

- Jan ter Maten (U Wuppertal)
- Wil Schilders (TU Eindhoven)
- Nelson Martins (CEPEL)
- Francisco Freitas (University of Brasilia)
- Gerard Sleijpen (Utrecht University)
- Pascal Bolcato, Olivier Maury (Mentor Graphics)
Mentor Graphics

- Electronic design automation (EDA) industry pioneer and global innovator of advanced design solutions
- Founded in 1981
- Revenue - ~$1,015B
- Market Share ~24% of worldwide EDA market
- Focused on growth through internal development

Source: EDAC Market Statistics
What is Electronic Design Automation?

- Critical design software used to create the world’s electronic systems
- Comprehensive EDA product portfolios address all levels – from component to systems
From transistor to system
Analog simulation

- Analog simulators are used to design, verify and characterize analog blocks.

- Used by Analog designers to compute the response of a circuit to some given stimuli.
Basic analyses

- **DC**: static operating point

- **AC**: linearized, frequency domain response

- **TRAN**: time domain response
Differential-Algebraic Equations (DAEs)

Modeled by system of differential-algebraic equations:

\[
d\frac{dq(t, x)}{dt} + j(t, x) = bu(t)
\]

- Node voltages and currents \(x \in \mathbb{R}^n\)
- Nonlinear vector valued \(q(t, x), j(t, x) \in \mathbb{R}^n\)
- Input \(bu(t) \in \mathbb{R}^n\) (sources)
- Simulation of schematic (left, \(n\) small): minutes – hours
- Simulation of layout (right, \(n\) large): minutes – \(\infty\)
Stability analysis

- Regulator IC: is the steady-state stable?
- Numerical challenges include
  - Matrices can be large due to parasitic elements
  - Direct methods not applicable
  - Eigenvalues at $\pm \infty$
Behavioral modeling of thermal effects

- Toplevel system simulation should cover all effects
- Computationally often not feasible
- Designers use *handmade* models to replace subsystems
- Automatic construction of behavioral models is open challenge
Oscillator coupling and pulling, phase-noise models

- Perturbation projection vector is eigenvector of large operator
- See e.g. Harutyunyan etal. (IEEE TCAD 2009)
- Also topic in EU project ASIVA14 (TU/e, Mentor)
How to compute the eigenvalues that are most sensitive to parameter changes?
Partitioning

How to partition this graph?
Outline

Eigenvalue problems
Circuit equations

- **Kirchhoff’s Current Law:** \( \sum_k i_k = 0 \)
- **Kirchhoff’s Voltage Law:** \( \sum_{k \in \text{loop}} v_k = 0 \)
- **Branch constitutive equations:**
  - Resistor: \( i = v/R \)
  - Capacitor: \( i = C \frac{dv}{dt} \)
  - Inductor: \( v = L \frac{di}{dt} \)

Leads to system of Differential Algebraic Equations:

\[
\frac{d}{dt} q(t, x) + j(t, x) = b u(t)
\]
Linearization

Let $x_{DC}$ be steady-state solution and

$$E = \left. \frac{\partial q}{\partial x} \right|_{x_{DC}} \quad \text{and} \quad A = - \left. \frac{\partial j}{\partial x} \right|_{x_{DC}}$$

Linearization around steady-state gives dynamical system

$$\begin{cases} 
E \dot{x}(t) &= Ax(t) + bu(t) \\
y(t) &= c^*x(t),
\end{cases}$$

where

$$u(t), y(t) \in \mathbb{R}, \text{ input, output}$$

$$x(t), b, c \in \mathbb{R}^n, \text{ state, input-to-, -to-output}$$

$$E \in \mathbb{R}^{n \times n} \text{ capacitance matrix}$$

$$A \in \mathbb{R}^{n \times n} \text{ conductance matrix}$$
Transfer function

First-order SISO dynamical system:

\[
\begin{align*}
E \dot{x}(t) &= A x(t) + b u(t) \\
y(t) &= c^* x(t)
\end{align*}
\]

with transfer function

\[
H(s) = c^* (sE - A)^{-1} b
\]

Poles are \( \lambda \in \mathbb{C} \) for which

\[
\lim_{s \to \lambda} |H(s)| = \infty,
\]

or, equivalently,

\[
\det(\lambda E - A) = 0,
\]

i.e. the eigenvalues of \((A, E)\)
Eigenvalue problems in practice: Pole-zero analysis

Poles \( \Lambda(A, E) \)

- poles \( \lambda \) with \( \text{real}(\lambda) > 0 \): unstable solution
- dominant poles cause peaks

Bode plot \((\omega, |H(i\omega)|)\)
The generalized eigenvalue problem

Given $A, E \in \mathbb{R}^{n \times n}$, find $(\lambda, x, y)$ that satisfy

$$Ax = \lambda E x, \quad x \neq 0$$
$$y^* A = \lambda y^* E, \quad y \neq 0$$

An eigentriplet $(\lambda, x, y)$ consists of

$\lambda \in \mathbb{C}$ \hspace{1cm} eigenvalue
$x \in \mathbb{C}^n$ \hspace{1cm} right eigenvector
$y \in \mathbb{C}^n$ \hspace{1cm} left eigenvector

- $(A, E)$ has $n$ eigenvalues (real / complex conjugated pairs)
- Corresponding eigenspaces need not be $n$-dimensional
- Bi-orthogonality: $\lambda_i \neq \lambda_j \Rightarrow y_j^* E x_i = 0$
Eigenvalue decompositions

Complete eigenvalue decomposition \((\Lambda, X, Y)\):

\[
AX = E\Lambda X, \quad Y^* A = \Lambda Y^* E \quad \text{with} \quad Y^* EX = I, \quad Y^* AX = \Lambda
\]

\[
\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{C}^{n\times n}
\]

\[
X = [x_1, x_2, \ldots, x_n] \in \mathbb{C}^{n\times n}
\]

\[
Y = [y_1, y_2, \ldots, y_n] \in \mathbb{C}^{n\times n}
\]

In practice only interest in \(k \ll n\) eigentriplets: partial ED

\[
AX_k = E\Lambda_k X, \quad Y_k^* A = \Lambda_k Y_k^* E \quad \text{with} \quad Y_k^* E X_k = I, \quad Y_k^* A X_k = \Lambda_k
\]

\[
\Lambda_k = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k) \in \mathbb{C}^{k\times k}
\]

\[
X_k = [x_1, x_2, \ldots, x_k] \in \mathbb{C}^{n\times k}
\]

\[
Y_k = [y_1, y_2, \ldots, y_k] \in \mathbb{C}^{n\times k}
\]
Eigenvalue computations

Methods for complete eigendecompositions:

- QR method for $AX = X\Lambda$
- QZ method for $AX = EX\Lambda$
- Complexity $O(n^3)$, practical use limited to small problems

Methods for partial eigendecompositions:

- Krylov methods (Lanczos, Arnoldi)
- Newton based methods ([Ruhe 1973], Jacobi-Davidson [Sleijpen, Van der Vorst (1995)])
- No dense matrix computations needed
- Careful selection strategies needed
Computational problems for large $A\mathbf{x} = \lambda E\mathbf{x}$

Brute force approach:
1. Compute all eigenvalues (and left and right eigenvectors)
2. Select eigenvalues of interest (positive real part, dominant)

Computational complications:
- Matrices can become very large: $n$ of $O(10^3)$ up to $O(10^6)$
- Dense methods QR/QZ too expensive ($O(n^3)$ CPU, memory)
- Spurious eigenvalues

In practice:
- Only few ($k \ll n$) specific eigenvalues of practical interest
- How to compute specifically these eigenvalues?

Similar eigenproblems arise in many other areas:
- Fluid dynamics, structural engineering, power systems
Outline

Stability analysis and spurious eigenvalues
Part II: Eigenvalue problems and purification

- Generalized eigenvalue problems $A\mathbf{x} = \lambda E\mathbf{x}$
- The Arnoldi method
- $E$ can be singular
- Eigenvalues at infinity
- Purification
Pole-zero stability analysis

- Generalized eigenproblem
  \[ Ax = \lambda Ex \]

- Wanted: eigenvalues with largest real part
  \[ \text{Re}(\lambda) > 0 \rightarrow \text{unstable} \]

- \( A \), \( E \) are large, sparse matrices
- \( E \) may be \textit{singular}
- Few \((k \ll n)\) specific eigenvalues are wanted
- Full space methods like QR and QZ too expensive \((O(n^3))\)
Shift-and-Invert

Generalized eigenproblem

\[ Ax = \lambda Ex \]

Choose shift \( \sigma \in \mathbb{C} \):

\[ (A - \sigma E)x = (\lambda - \sigma)Ex \]

and invert:

\[ (A - \sigma E)^{-1}Ex = (\lambda - \sigma)^{-1}x \]

With \( S = (A - \sigma E)^{-1}E \):

\[ Ax = \lambda Ex \iff Sx = \tilde{\lambda}x, \quad \tilde{\lambda} = (\lambda - \sigma)^{-1} \]

\( \lambda(A, E) \) near \( \sigma \) are transformed to outside of spectrum \( \Lambda(S) \)
The Arnoldi method [Arnoldi 1951]

Given $S$, construct orthonormal basis $v_1, \ldots, v_{k+1}$ for

$$Krylov \ space \ \mathcal{K}^{k+1}(S, v_1) = \text{span}(v_1, Sv_1, \ldots, S^k v_1)$$

1. choose $v_1$ with $\|v_1\|_2 = 1$
2. For $i = 1$ to $k$ do
   2.1 compute $w = Sv_i$
   2.2 compute $h_{j,i} = v_j^*w$ for $j = 1, \ldots, i$
   2.3 compute $w = w - Vh$
   2.4 compute $h_{i+1,i} = \|w\|_2$
   2.5 set $v_{i+1} = w/h_{i+1,i}$
The Arnoldi method [Arnoldi 1951]

Orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$ for Krylov space $\mathcal{K}^{k+1}(S, \mathbf{v}_1)$:

$$V_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{C}^{n \times k}$$

$$V_k^* V_k = I,$$

$$S V_k = V_k H_k + h_{k+1,k} \mathbf{v}_{k+1} \mathbf{e}_k^T$$

Require for approximate eigenpair $(\theta, V_k \mathbf{y})$

$$S(V_k \mathbf{y}) - \theta(V_k \mathbf{y}) \perp V_k \quad \text{(Ritz-Galerkin)}$$

1. Compute eigenpairs $(\theta_i, \mathbf{y}_i)$ of $H_k = V_k^* S V_k \in \mathbb{C}^{k \times k}$

$$H_k \mathbf{y}_i = \theta_i \mathbf{y}_i$$

2. Compute Ritz pairs $(\theta_i, V_k \mathbf{y}_i)$ of $S$ and select wanted

3. Check residual norm $\|\mathbf{r}\|_2 = \|S V_k \mathbf{y}_i - \theta_i V_k \mathbf{y}_i\|_2 = |h_{k+1,k} \mathbf{y}_{i(k)}|$
Eigenvalues at infinity

- One finite, one infinite eigenvalue

\[ A = A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \Rightarrow \quad \lambda(A, E) = \{1, \infty\} \]

- Defective, infinite eigenvalue

\[ A = A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \Rightarrow \quad \lambda(A, E) = \{\infty\} \]

- Note \( \lambda(A, E) = \infty \) becomes \( \tilde{\lambda}(A^{-1}E) = 0 \)

- Eigenvalues at \( \infty \) are not of interest
Numerical problem

- Start Arnoldi with $v_1 = S^2 \mathbf{1} \in \text{range}(S^2)$
- $P_N$: projection on $N = \ker(S)$
- $P_G$: projection on $G = \ker(S^2) \setminus \ker(S)$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$|P_N v_j|_2$</th>
<th>$|P_G v_j|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$3.5 \cdot 10^{-11}$</td>
<td>$7.6 \cdot 10^{-12}$</td>
</tr>
<tr>
<td>2</td>
<td>$7.5 \cdot 10^{-9}$</td>
<td>$1.2 \cdot 10^{-10}$</td>
</tr>
<tr>
<td>3</td>
<td>$2.1 \cdot 10^{-7}$</td>
<td>$2.5 \cdot 10^{-9}$</td>
</tr>
<tr>
<td>4</td>
<td>$5.5 \cdot 10^{-7}$</td>
<td>$5.1 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>5</td>
<td>$1.5 \cdot 10^{-4}$</td>
<td>$1.1 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>15</td>
<td>$3.1 \cdot 10^{+7}$</td>
<td>$3.0 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

One spurious eigenvalue $\theta = 6.4 \cdot 10^{10}$
Numerical problem

- Recall $V_\infty = \mathcal{N}(S) = \mathcal{N}(E) = \{ x \in \mathbb{R}^n \mid Ex = 0 \}$

- In exact arithmetic: $v_1 \in \mathcal{R} \Rightarrow v_j = Sv_{j-1} \in \mathcal{R}$

However, in finite arithmetic

- Rounding errors ($Sv_j$, orth) lead to components in $\mathcal{N} + \mathcal{G}$ in $v_j$

- Arnoldi can find approximations $\theta_i$ to $\tilde{\lambda} = 0$:

$$ (V_k^*SV_k)y_i = \theta_i y_i $$

- Back transformation $\lambda = \theta_i^{-1} + \sigma$ leads to spurious eigenvalues

Purification:

1. Remove/prevent spurious eigenvalue approximations
2. Improve wanted eigenpair approximations by removing components in $\mathcal{N} + \mathcal{G}$ from $v_j$
Consider block structured generalized eigenvalue problem

\[
\begin{bmatrix}
K & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\ p
\end{bmatrix} = \lambda
\begin{bmatrix}
M & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
u \\ p
\end{bmatrix},
\]

with \(C \in \mathbb{R}^{m \times k}\), and \(K, M \in \mathbb{R}^{m \times m}\) (\(n = m + k\))

Corresponding ordinary eigenproblem is

\[
\begin{bmatrix}
S_1 & 0 \\
S_2 & 0
\end{bmatrix}
\begin{bmatrix}
u \\ p
\end{bmatrix} = \tilde{\lambda}
\begin{bmatrix}
u \\ p
\end{bmatrix},
\]

\(S_1 \in \mathbb{R}^{m \times m}\), \(S_2 \in \mathbb{R}^{k \times m}\),

Reduced problem

\[
S_1 u = \tilde{\lambda} u \iff \begin{bmatrix}
S_1 & 0 \\
S_2 & 0
\end{bmatrix}
\begin{bmatrix}
u \\ \tilde{\lambda}^{-1} S_2 u
\end{bmatrix} = \tilde{\lambda}
\begin{bmatrix}
u \\ \tilde{\lambda}^{-1} S_2 u
\end{bmatrix}
\]
Exploiting structure

\[ S_1u = \tilde{\lambda}u \iff \begin{bmatrix} S_1 & 0 \\ S_2 & 0 \end{bmatrix} \begin{bmatrix} u \\ \tilde{\lambda}^{-1}S_2u \end{bmatrix} = \tilde{\lambda} \begin{bmatrix} u \\ \tilde{\lambda}^{-1}S_2u \end{bmatrix} \]

- \( S \), and in particular \( S_1 \), not available explicitly in general
- but MVs \( Sv = (A - \sigma E)^{-1}Ev \) are available:
  - \( LU = A - \sigma E \) (once)
    1. \( w = Ev \)
    2. \( x = L^{-1}w \)
    3. \( y = U^{-1}x \)

Use projectors to compute \( S_1u \)

\[ S_1u = \begin{bmatrix} I_m & 0 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ S_2 & 0 \end{bmatrix} \begin{bmatrix} I_m \\ 0 \end{bmatrix} u \]

- \( \text{dim}(\text{gen ker}(S)) = k \) vs. \( \text{dim}(\text{gen ker}(S_1)) = 0 \)
Figure: The size of $\|\Psi_{k+1}\|_2 = \|V_{k+1}H_k - SV_k\|_2$ for Arnoldi applied to $S = (A - 60E)^{-1}E$, and Arnoldi applied to $S_1$. 
Further improvements

- Implicit restarts [Sorensen 1992]:
  - Additional purification [Meerbergen/Spence 1995]
  - Control convergence [R. 2008/2011]
- Find missed eigenvalues:
  - Cayley transformations [Cliffe/Garratt/Spence 1994, R. 2008]
- Very large problems ($LU = (A - \sigma B)$ not feasible):
  - Jacobi-Davidson methods [Sleijpen/Van der Vorst 1996, R. 2008]
Electro Static Discharge analysis

Damaged interconnect that was too small to conduct current

<table>
<thead>
<tr>
<th>W/E</th>
<th>Resistor</th>
<th>Index</th>
<th>Layer</th>
<th>X:Y</th>
<th>Width</th>
<th>Current</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>PRm1957</td>
<td>1023</td>
<td>&amp;r#1</td>
<td>58 458:157:32</td>
<td>1.106</td>
<td>1</td>
<td>3.62</td>
</tr>
<tr>
<td>E</td>
<td>PRm1956</td>
<td>1022</td>
<td>&amp;r#1</td>
<td>59 142:157:086</td>
<td>1.79</td>
<td>1</td>
<td>2.23</td>
</tr>
<tr>
<td>E</td>
<td>PRm1959</td>
<td>1025</td>
<td>&amp;r#1</td>
<td>56 452:157:36</td>
<td>2.26</td>
<td>1</td>
<td>1.75</td>
</tr>
<tr>
<td>E</td>
<td>PRm1968</td>
<td>1024</td>
<td>&amp;r#1</td>
<td>57 262:157:36</td>
<td>2.26</td>
<td>1</td>
<td>1.75</td>
</tr>
<tr>
<td>E</td>
<td>PRm1980</td>
<td>1747</td>
<td>&amp;r#1</td>
<td>58 051:159:1</td>
<td>0.6</td>
<td>-0.243099</td>
<td>1.66</td>
</tr>
<tr>
<td>E</td>
<td>PRm1965</td>
<td>1021</td>
<td>&amp;r#1</td>
<td>59 982:158:636</td>
<td>2.63</td>
<td>1</td>
<td>1.52</td>
</tr>
<tr>
<td>E</td>
<td>PRm1971</td>
<td>1739</td>
<td>&amp;r#1</td>
<td>58 052:159:82</td>
<td>0.6</td>
<td>-0.213122023</td>
<td>1.42</td>
</tr>
<tr>
<td>E</td>
<td>PRm1964</td>
<td>1020</td>
<td>&amp;r#1</td>
<td>60 352:156:81</td>
<td>3.08</td>
<td>1</td>
<td>1.3</td>
</tr>
<tr>
<td>E</td>
<td>PRm1762</td>
<td>896</td>
<td>&amp;r#1</td>
<td>66 492:135:16</td>
<td>1.66</td>
<td>-0.493862312</td>
<td>1.27</td>
</tr>
<tr>
<td>E</td>
<td>PRm1960</td>
<td>1026</td>
<td>&amp;r#1</td>
<td>56 002:153:04</td>
<td>2.28</td>
<td>0.710632606</td>
<td>1.25</td>
</tr>
<tr>
<td>E</td>
<td>PRm1970</td>
<td>1736</td>
<td>&amp;r#1</td>
<td>58 052:160:82</td>
<td>0.6</td>
<td>-0.182341944</td>
<td>1.22</td>
</tr>
<tr>
<td>E</td>
<td>PRm1953</td>
<td>1019</td>
<td>&amp;r#1</td>
<td>60 882:156:166</td>
<td>3.53</td>
<td>1</td>
<td>1.75</td>
</tr>
</tbody>
</table>

• Worst r=3.62 in M1
Partitioning of electrical circuits

Elementary graph algorithms like biconnected components suffice
Example: difficult network for reduceR

How to partition this network?
Spectral partitioning

- Given undirected graph $G$ with equally weighted edges $g_{ij}$
- Note $\text{diag}(G) = 0$
- Define $D = \text{diag}(\text{degree}_i)$
- Laplacian of $G$ is defined as $L = D - G$

Partitioning $G$ with fewest cut edges:

$$\min_{y_i \in \{-\frac{1}{2}, \frac{1}{2}\}} \sum_{i,j} (y_i - y_j)^2 g_{ij}$$

Difficult, so relax

$$\min_{y_i \in \mathbb{R}^n} \sum_{i,j} (y_i - y_j)^2 g_{ij}$$
Spectral partitioning

- Given undirected graph $G$ with equally weighted edges $g_{ij}$
- Note $\text{diag}(G) = 0$
- Define diagonal $D$ with $d_{ii} = \text{degree}(\text{node } i)$
- Laplacian of $G$ is defined as $L = D - G$

Partitioning $G$ with fewest cut edges:

$$\min_{y_i \in \mathbb{R}^n} \sum_{i,j} (y_i - y_j)^2 g_{ij}$$  \hspace{1cm} (1)

- Note $0 = \lambda_1(L) < \lambda_2(L) < \ldots < \lambda_n(L)$
- Eigenvector $v_2$ corresponding to $\lambda_2$ is called Fiedler vector
- Fiedler vector solves (1): partitioning reduces to eigenproblem!
- See [Fiedler, Pothen, D. Higham]
How to use this in electrical networks?

- Construct Laplacian $L$ (cheap)
- Deflate $\lambda_1 = 0$ (we know the eigenvector)
- Compute eigenpair $(\lambda_2, v_2)$ of $L$
  - If small network, use QR
  - If large, use Arnoldi or Subspace Accelerated RQI [R. 2010]
- Partition by inspecting $v_2$
  - If $v_2(i) > 0$, put node $i$ in cluster 1
  - Else if $v_2(i) \leq 0$, put node $i$ in cluster 2
  - Find cut nodes by inspecting subgraphs (cheap)
- This can be applied recursively...
Example: biconnected component

Fiedler vector (2nd eigenvector)
Example: biconnected component
Example: difficult network for reduceR

76 terminals vs. 43 and 33 terminals, (3 and 2) cutnodes
Outline

Eigenanalysis for model order reduction
Transfer function \( H(s) = c^*(sE - A)^{-1}b \)

Can be expressed as

\[
H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i},
\]

where residues \( R_i \) are

\[
R_i = (c^*x_i)(y_i^*b),
\]

and \((\lambda_i, x_i, y_i)\) are eigentriples \((i = 1, \ldots, n)\)

\[
\begin{align*}
Ax_i &= \lambda_i Ex_i, & \text{right eigenpairs} \\
y_i^*A &= \lambda_i y_i^*E, & \text{left eigenpairs} \\
y_i^*Ex_i &= 1, & \text{normalization} \\
y_j^*Ex_i &= 0 \ (i \neq j), & \text{E-orthogonality}
\end{align*}
\]
Dominant poles cause peaks in Bode-plot

\[ H(s) = c^*(sE - A)^{-1}b = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i} \quad \text{with} \quad R_i = (c^*x_i)(y_i^*b) \]

Bode-plot is graph of \((\omega, |H(i\omega)|)\)

- frequency \(\omega \in \mathbb{R}\)
- magnitude \(|H(i\omega)|\) usually in dB (note dB(\(x\)) = 20 \cdot 10 \log(\(x\)))

Consider pole \(\lambda = \alpha + \beta i\) with residue \(R\), then

\[
\lim_{\omega \to \beta} H(i\omega) = \lim_{\omega \to \beta} \frac{R}{i\omega - (\alpha + \beta i)} + \sum_{j=1}^{n-1} \frac{R_j}{i\omega - \lambda_j}
\]

\[
= -\frac{R}{\alpha} + H_{n-1}(i\beta)
\]

Pole \(\lambda\) with large \(\left|\frac{R}{\text{Re}(\lambda)}\right|\) is dominant and causes peak
Dominant poles cause peaks in Bode-plot
Figure: Bode plot $(\omega, |H(i\omega)|)$. Pole $\lambda_j$ dominant if $\frac{|R_j|}{|\text{Re}(\lambda_j)|}$ large.
Dominant poles of transfer functions

\[ H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i} \quad \text{with} \quad R_i = (c^* x_i)(y_i^* b) \]

- Pole \( \lambda_i \) dominant if \( \frac{|R_i|}{|\text{Re}(\lambda_i)|} \) large
- Dominant poles cause peaks in Bode-plot \((\omega, |H(i\omega)|)\)
- Effective transfer function behavior:

\[ H_k(s) = \sum_{i=1}^{k} \frac{R_i}{s - \lambda_i}, \]

where \( k \ll n \) and \((\lambda_i, R_i)\) ordered by decreasing dominance

- Early work modal approximation [Davison, Marschall (1966)]
Dominant Pole Algorithm [Martins (1996)]

\[ H(s) = c^*(sE - A)^{-1}b \]

- Pole \( \lambda \): 
  \[ \lim_{s \to \lambda} |H(s)| = \infty, \text{ or } \lim_{s \to \lambda} \frac{1}{H(s)} = 0 \]

Apply Newton’s Method to \( 1/H(s) \):

\[
\begin{align*}
    s_{k+1} &= s_k + \frac{1}{H(s_k)} \frac{H^2(s_k)}{H'(s_k)} \\
    &= s_k - \frac{c^*(s_k E - A)^{-1}b}{c^*(s_k E - A)^{-1}E(s_k E - A)^{-1}b}
\end{align*}
\]

Note \( \frac{dH}{ds} = -c^*(s_k E - A)^{-1}E(s_k E - A)^{-1}b \)
Dominant Pole Algorithm

1: Initial pole estimate $s_1$, tolerance $\epsilon \ll 1$
2: for $k = 1, 2, \ldots$ do
3: Solve $v_k \in \mathbb{C}^n$ from $(s_k E - A)v_k = b$
4: Solve $w_k \in \mathbb{C}^n$ from $(s_k E - A)^*w_k = c$
5: Compute the new pole estimate

$$s_{k+1} = s_k - \frac{c^*v_k}{w_k^*E v_k}$$

6: The pole $\lambda = s_{k+1}$ with $x = v_k/\|v_k\|_2$ and $y = w_k/\|w_k\|$ has converged if

$$\|(s_{k+1} E - A)x\|_2 < \epsilon$$

7: end for
Twosided Rayleigh quotient iteration

Note that with $v \equiv v_k$ and $w \equiv w_k$

\[
s_{k+1} = s_k - \frac{c^*(s_k E - A)^{-1}b}{w^*Ev}
= s_k \frac{w^*Ev}{w^*Ev} - \frac{c^*(s_k E - A)^{-1}(s_k E - A)(s_k E - A)^{-1}b}{w^*Ev}
= \frac{w^*Av}{w^*Ev}
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>DPA</th>
<th>Twosided RQI</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>solve $(s_k E - A)v_k = b$</td>
<td>solve $(s_k E - A)v_k = E v_{k-1}$</td>
</tr>
<tr>
<td>4</td>
<td>solve $(s_k E - A)^*w_k = c$</td>
<td>solve $(s_k E - A)^*w_k = E^*w_{k-1}$</td>
</tr>
</tbody>
</table>

Original work on twosided RQI [Ostrowski (1958), Parlett (1974)]
Convergence behavior: DPA vs. RQI
Convergence behavior: DPA vs. RQI

Typically, with initial pole guess $s_0$,

- DPA converges to *dominant* pole closest to $s_0$
  - with $\angle(c, x)$ and $\angle(b, y)$ small
  - i.e., large $|R|$ with $R = (c^*x)(y^*b)$
- Quadratic rate of convergence
- See also [R., Sleijpen (2006)]

while

- RQI converges to pole *closest* to $s_0$
- Originally intended for refinement of eigenpairs
- Cubic rate of convergence
- See also [Ostrowski (1958), Parlett (1974)]
Subspace acceleration and selection

- Keep approximations \( \mathbf{v}_k \) and \( \mathbf{w}_k \) in search spaces \( V \) and \( W \)
- Petrov-Galerkin leads to projected eigenproblem

\[
\tilde{A}\tilde{x} = \theta\tilde{E}\tilde{x},
\]
\[
\tilde{y}^*\tilde{E} = \theta\tilde{y}^*\tilde{A}
\]

where \( \tilde{E} = W^*EV \in \mathbb{C}^{k \times k} \) and \( \tilde{A} = W^*AV \in \mathbb{C}^{k \times k} \)

- Gives \( k \) approximations \( (\theta_i, \hat{x}_i = V\tilde{x}_i, \hat{y}_i = W\tilde{y}_i) \) in iter \( k \)
- Select approximation with largest residue as next shift:

\[
s_{k+1} = \arg\max_i \left| \frac{(c^*\hat{x}_i)(\hat{y}_i^*b)}{\text{Re}(\theta_i)} \right|
\]

- Similarities with twosided Jacobi-Davidson ([Hochstenbach (2003), Stathopoulos (2002)])
Deflation for $H(s) = c^*(sE - A)^{-1}b$

- Triplet $(\lambda, x, y)$: $Ax = \lambda Ex$ and $y^*A = \lambda y^*E$
- New search spaces: $V \perp E^*y$ and $W \perp Ex$
- Usual deflation (every iteration):
  \[
  v_k \leftarrow (I - xy^*E)v_k \\
  w_k \leftarrow (I - yx^*E^*)w_k
  \]
- More efficient: deflate only once
  \[
  b_d \leftarrow (I - Exy^*)b \quad \Rightarrow \quad v_k = (s_kE - A)^{-1}b_d \perp E^*y \\
  c_d \leftarrow (I - E^*yx^*)c \quad \Rightarrow \quad w_k = (s_kE - A)^{-*}c_d \perp Ex
  \]
- Note that $y^*b_d = c_d^*x = 0$
Computation of dominant zeros

Figure: Dominant zeros of $H(s)$ are dominant poles of $H^{-1}(s)$. 
Computation of dominant zeros

Computation of dominant zeros of $H(s) = c^*(sE - A)^{-1}b + d$:

1. Realize inverse transfer function $H_z(s) = c_z^*(sE_z - A_z)^{-1}b_z$:

$$A_z = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix}, \quad E_z = \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix},$$

$$b_z = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad c_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad d_z = 0,$$

2. Apply DPA to $H_z(s)$

See [Martins, Pellanda, R., IEEE TPWRS 2007] for more details
Computation of most sensitive eigenvalues

- Suppose system matrix $A$ depends on parameter $p$
- Sensitivity of eigenvalue is given by
  \begin{equation}
  \frac{\partial \lambda}{\partial p} = y^* \frac{\partial A}{\partial p} x
  \end{equation}
  where $y$ and $x$ are left and right eigenvectors
- If $\frac{\partial A}{\partial p}$ has rank 1:
  \begin{equation}
  \frac{\partial \lambda}{\partial p} = y^* \frac{\partial A}{\partial p} x = (y^* b)(c^* x) = (c^* x)(y^* b)
  \end{equation}
- Apply DPA to $(A, b, c)$ to compute sensitive eigenvalues!
- See [R., Martins, IEEE TPWRS 2008] for more details
Computation of most sensitive eigenvalues

Figure: Root locus of most sensitive eigenvalues for 13k state system.
DPA: Complexity

Every iteration solves of

$$(s_k E - A)v_k = b \text{ and } (s_k E - A)^*w_k = c$$

Can be done efficiently:

- Fill-in minimizing reordering: AMD [Davis, Duff]
- Reuse $LU = s_k E - A$ as $U^*L^* = (s_k E - A)^*$
- Scalable up to millions of unknowns for sparse systems

If solves cannot be done exactly (CPU/MEM):

- Jacobi-Davidson style methods [Sleijpen/Van der Vorst, Hochstenbach, R.]
- Recycling Krylov spaces [De Sturler]
- Inexact variants [Kürschner et al]
Dominant Pole Algorithm (DPA) and extensions

DPA computes dominant poles of $H(s) = c^*(sE - A)^{-1}b$

1. Newton scheme [M., Lima, Pinto (IEEE TPWRS 11(1) 1996)]
2. Convergence analysis [R., Sleijpen (SIMAX 30(1) 2008)]
3. Subspace acceleration, selection, deflation: SADPA [R., Martins (IEEE TPWRS 21(3) 2006)]
5. Dominant zeros [R., Martins, Pellanda (IEEE TPWRS 22(4) 2007)]
6. Poles of second-order systems: QDPA [R., Martins (SISC 30(4) 2008)]
7. Spectral zeros [Ionutiu, R., Antoulas (IEEE TCAD 27(12) 2008)]
8. Sensitive poles: SPA [R., Martins (IEEE TPWRS 23(2) 2008)]
9. Time-delay systems [Meerbergen etal 2012]
10. Parameterized systems [Saadvandi etal 2014]
Computation of rightmost eigenvalues

Typical problem in stability analysis $Ax = \lambda Ex$:

Are there any eigenvalues in the right half-plane?

or

What is the rightmost eigenvalue?

or

Compute eigenvalues with specified damping ratio

Damping ratio of $\lambda = \alpha + \beta i$:

$$\zeta = -\frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}$$
Numerical difficulties

\[ Ax = \lambda E x \]

1. The problems are large: \( n \gg 1000 \)
   - Hence full space (QR, QZ) do not apply
2. \( E \) is singular: there are eigenvalues at infinity
3. How do we know for sure we have the rightmost eigenvalue?
4. We cannot (?) solve this problem in single-vector-iterations like RQI and DPA
   - Unlike dominance index \( r = (c^T x)(y^T b) \ldots \)
   - \ldots info on being rightmost is not in eigenvector!
   - Info is in the eigenvalues only! \( (\text{Re}(\lambda) > 0) \)

See also [Meerbergen, Spence, Sleijpen/Van der Vorst]
Solution methods

Previous work on $Ax = \lambda E x$ with singular $E$

- Shift-and-Invert Arnoldi with purification
  - Compute eigenvalues $\mu = \frac{1}{\lambda - \sigma}$ of $(A - \sigma E)^{-1}E$
  - Can be done cheaply due to sparsity
  - $\lambda = \infty$ transformed to $\mu = 0$
  - However, also affects order of eigenvalues
  - Still problems with infinity eigenvalues (use purification)
  - See [Spence, Meerbergen, R.,...]

- Variants with Jacobi-Davidson (see [R. 2008])
Subspace accelerated Rayleigh Quotient Iteration

Compute rightmost \((\lambda, x)\) of \(Ax = \lambda Ex\)

- We cannot (?) solve this problem in single-vector-iterations like DPA
  - Unlike dominance index \(r = (c^T x)(y^T b)\ldots\)
  - \(\ldots\) info on being rightmost is not in eigenvector!
  - Info is in the eigenvalues only! \((\text{Re}(\lambda) > 0)\)
- Hence we need subspace acceleration and selection strategy
Subspace accelerated Rayleigh Quotient Iteration

Sketch of SARQI

1: for \( k = 1, 2, \ldots \) until convergence do
2: \( \text{Given approximate eigenvalue } \sigma_k, \text{ solve } v_k \text{ and } w_k \)
   \[
   v_k = (\sigma_k E - A)^{-1}(E\tilde{x}) \quad \text{and} \quad w_k = (\sigma_k E - A)^{-*}(E^*\tilde{y})
   \]
3: \( \text{Expand the search spaces } V \text{ and } W \text{ with } v_k \text{ and } w_k \)
4: \( \text{Project: } E_k = W^*EV \text{ and } A_k = W^*AV \)
5: \( \text{Select approximate eigentriplet } (\tilde{\lambda}, \tilde{x}, \tilde{y}) \text{ from } \text{QZ}(A_k, E_k) \)
   according to selection criterion
6: \( \text{Check convergence and deflate if converged} \)
7: end for
Rightmost: How to select the best approximant?

- Keep approximations \( \mathbf{v}_k \) and \( \mathbf{w}_k \) in search spaces \( V \) and \( W \)
- Petrov-Galerkin leads to projected eigenproblem

\[
\begin{align*}
\tilde{A}\tilde{x} &= \theta \tilde{E}\tilde{x}, \\
\tilde{y}^*\tilde{E} &= \theta \tilde{y}^*\tilde{A}
\end{align*}
\]

where \( \tilde{E} = W^*EV \in \mathbb{C}^{k \times k} \) and \( \tilde{A} = W^*AV \in \mathbb{C}^{k \times k} \)

- Gives \( k \ll n \) approximations \((\theta_i, \hat{x}_i = V\tilde{x}_i, \hat{y}_i = W\tilde{y}_i)\)
- Select rightmost approximation as next shift:

\[
\sigma_{k+1} = \arg\max_i \left| \frac{(c^*\tilde{x})(\tilde{y}^*b)}{\Re(\tilde{\lambda})} \right|
\]

- Here \( b = c = [1, 1, \ldots, 1]^T \)
Damping ratio target: How to select the best approximant?

- Keep approximations $v_k$ and $w_k$ in search spaces $V$ and $W$
- Petrov-Galerkin leads to projected eigenproblem

$$\tilde{A}\tilde{x} = \theta\tilde{E}\tilde{x},$$
$$\tilde{y}^*\tilde{E} = \theta\tilde{y}^*\tilde{A}$$

where $\tilde{E} = W^*EV \in \mathbb{C}^{k \times k}$ and $\tilde{A} = W^*AV \in \mathbb{C}^{k \times k}$
- Gives $k \ll n$ approximations ($\theta_j, \hat{x}_j = V\tilde{x}_j, \hat{y}_j = W\tilde{y}_j$)
- Select approximation closest to damping ratio as next shift:

$$\sigma_{k+1} = \arg\min_j |\zeta_0 - \frac{-\alpha_j}{\sqrt{\alpha_j^2 + \beta_j^2}}|$$

- Here $\zeta_0$ is target damping ratio and $\theta_j = \alpha_j + \beta_j i$
Properties of SARQI

- Easy to implement (compared to e.g., Jacobi-Davidson)
- Purification is done automatically by selection strategy
- Shift selection is automatic (SI-Arnoldi needs predefined shift)
- Structure can be exploited
- Fast convergence due to RQI
- Requires fast solutions of large sparse linear systems

See [R., Freitas, Martins, IEEE TPWRS 2010] for more details
Example \((n = 40366)\)

**Figure:** Relevant part of spectrum with damping ratio lines
Example run of SARQI ($n = 40366$)
Model order reduction

Given large-scale dynamical system

\[
\begin{align*}
E\dot{x}(t) &= Ax(t) + bu(t) \\
y(t) &= c^*x(t) + du(t)
\end{align*}
\]

where \( x(t), b, c \in \mathbb{R}^n \) and \( E, A \in \mathbb{R}^{n \times n} \), find

\[
\begin{align*}
E_k\dot{x}_k(t) &= A_kx_k(t) + b_ku(t) \\
y_k(t) &= c_k^*x_k(t) + du(t)
\end{align*}
\]

where \( x_k(t), b_k, c_k \in \mathbb{R}^k \), \( E_k, A_k \in \mathbb{R}^{k \times k} \) and

- \( k \ll n \)
- approximation error \( \|y - y_k\| \) small

Antoulas (2005) and Schilders, Van der Vorst, R. (2008)
Additional constraints on reduced order model

\[
\begin{align*}
E\dot{x}(t) &= Ax(t) + bu(t) \\
y(t) &= c^*x(t) + du(t)
\end{align*}
\Rightarrow
\begin{align*}
E_k\dot{x}_k(t) &= A_kx_k(t) + b_ku(t) \\
y_k(t) &= c_k^*x_k(t) + du(t)
\end{align*}
\]

Size may be reduced, but what about complexity?

- Original model may have sparse system matrices, while reduced order model has dense system matrices
- Time domain simulation may become more expensive
- Reuse: ROM must be available as, e.g., netlist
- Simulators and software may introduce additional constraints
Model order reduction

Model order reduction via projection:

1. Construct matrices $V, W \in \mathbb{R}^{n \times k}$ whose columns form a basis for the dominant dynamics
2. Project using $V$ and $W$:

$$E_k = W^* EV, \quad A_k = W^* AV, \quad b_k = W^* b, \quad c_k = V^* c$$

Various projection based methods:

- **Modal truncation**: columns $V, W$ are eigenvectors of $(A, E)$
- **Moment matching**: columns $V, W$ are bases for Krylov spaces
- **Balanced truncation**: $V, W$ part of balancing transformation
Modal approximation

General framework for modal approximation of

\[ H(s) = c^*(sE - A)^{-1}b = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i} = \sum_{i=1}^{n} \frac{(c^*x_i)(y_i^*b)}{s - \lambda_i} \]

where \( y_i \) and \( x_i \) are left and right eigenvectors of \((A, E)\):

1. Sort \((\lambda_i, R_i)\) in decreasing \(|R_i|/\text{Re}(\lambda_i)\) order
2. Truncate at \(|R_i|/\text{Re}(\lambda_i) < R_{\text{min}}\)
3. Project with \( Y_k = [y_1, \ldots, y_k] \) and \( X_k = [x_1, \ldots, x_k] \)

\[
\begin{align*}
\dot{\tilde{x}} &= \Lambda_k \tilde{x}(t) + \tilde{b}u(t) \\
y(t) &= \tilde{c}^* \tilde{x}(t)
\end{align*}
\]

\[ H_k(s) = \sum_{i=1}^{k} \frac{R_i}{s - \lambda_i} \]

Use SADPA [R., Martins (2006)] to compute dominant poles
Moment matching

Series expansion of \( H(s) = c^*(sE - A)^{-1}b \) around \( s_0 \) is

\[
H(s) = \sum_{i=0}^{\infty} m_i(s - s_0)^i
\]

with moments \( m_i = c^*G^i(s_0E - A)^{-1}b \) and \( G = (s_0E - A)^{-1}E \)

Model order reduction: Match only \( 2k \ll n \) moments:

1. Compute bases \( V \in \mathbb{R}^{n \times k} \) and \( W \in \mathbb{R}^{n \times k} \) for (Arnoldi)

\[
\mathcal{K}^k((s_0E - A)^{-1}E, b) \text{ and } \mathcal{K}^k((s_0E - A)^{-1}E^*, c)
\]

2. Petrov-Galerkin projection gives \( k \)-th order system:

\[
\begin{align*}
E\dot{x} &= Ax(t) \quad \Rightarrow \quad (W^*EV)\dot{x} = (W^*AV)\ddot{x}(t) \\
+ bu(t) \quad &+ (W^*b)u(t) \\
y(t) &= c^*x(t) \quad \Rightarrow \quad \ddot{y}(t) = (c^*V)\ddot{x}(t)
\end{align*}
\]
Advantages and disadvantages

- Different applications require different methods
- Best method not always known in advance

- **Modal approximation** [Davison, Varga, Martins, R.]
  - easy realization of ROM, natural interpretation
  - may be expensive, requires decay in residues

- **Moment matching** [Freund, Van Dooren, Grimme]
  - cheap and robust implementations
  - dense ROMs, requires decay in moments

- **Balanced truncation** [Moore, Glover, Penzl, Stykel, Benner]
  - a priori error estimate
  - complications for descriptor systems
  - requires decay in Hankel singular values
Modal approximation and moment matching

Figure: Frequency response of complete system \((n = 66)\), modal approximation \((k = 12)\), and dual Arnoldi model \((k = 30)\).
Dominant poles: location in complex plane

Figure: Pole spectrum of complete system ($n = 66$), modal approximation ($k = 12$), and dual Arnoldi model ($k = 30$).
Dominant poles: location in complex plane (zoom)

Dominant poles not necessarily at outside of spectrum

Figure: Pole spectrum (zoom) of complete system \((n = 66)\), modal approximation \((k = 12)\), and dual Arnoldi model \((k = 30)\).
Rational Krylov methods [Ruhe (1998)]

General approach:

1. Choose $m$ interpolation points $s_i$
2. Construct $V_i, W_i \in \mathbb{C}^{n \times k_i}$ such that
   
   $\text{colspan}(V_i) = \mathcal{K}^{k_i}((s_i E - A)^{-1} E, (s_i E - A)^{-1} E b)$
   $\text{colspan}(W_i) = \mathcal{K}^{k_i}((s_i E - A)^{-*} E^*, (s_i E - A)^{-*} E^* c)$
3. Project with $V = [V_1, \ldots, V_m]$ and $W = [W_1, \ldots, W_m]$

Open question:

- How to choose interpolation points $s_i$?
- See also PhD thesis Grimme (1997)
Figure: Breathing sphere ($n = 17611$). Exact transfer function (solid), 70th order SOAR [Bai/Su 2005] RKA model (dash) using interpolation points based on dominant poles, and relative error (dash-dot).
Outline

Concluding remarks
Concluding remarks

- Eigenproblems arise in many application domains
- Nature and difficulties vary
  - Stability analysis (rightmost eigenvalues)
  - MOR (dominant modes)
  - Phase noise analysis (left eigenvector for $\lambda = 1$)
  - Partitioning (Fiedler vector)
- Open challenges include
  - How to know we did not miss any eigenvalues?
  - Avoiding piling up of rounding errors (deflation)
  - Robustness and performance for inexact solvers
Thank you!

joost_rommes@mentor.com
www.mentor.com
sites.google.com/site/rommes