Moment-matching method

Lihong Feng

Outline

- Preliminaries
    - Orthogonality of two vectors
    - Orthogonality of a vector to a group of orthogonal vectors
    - Gram-Schmidt (modified Gram-Schmidt) process
    - Arnoldi algorithm
- Method based on Pade approximation, explicit moment-matching (AWE)
- Method based on Pade, Pade-type approximation, implicit moment-matching
- Method based on rational interpolation

Preliminaries

Angle between two vectors:

Two vectors *u*, *v*: $u \cdot v = \cos(\theta) \, \| u \|_2 \| v \|_2$

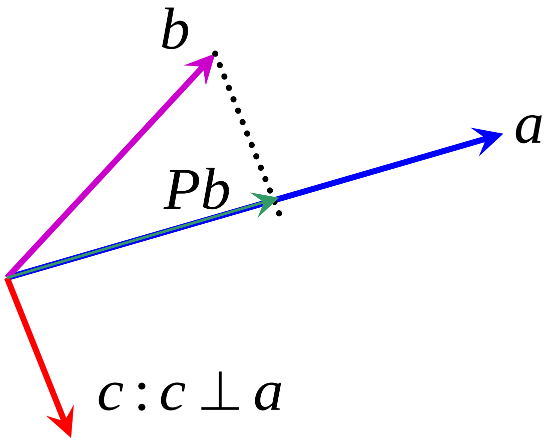
Inner products in R^n: $u \cdot v = v^T u$

Orthogonality of two vectors: $\cos(\theta) = 0 \Leftrightarrow \dfrac{u^T v}{\| u \|_2 \| v \|_2} = 0 \Leftrightarrow u^T v = 0$

Preliminaries

- Orthogonalization of two vectors $a, b \in R^n$

If *Pb* is the projection of *b* onto a, then *c*=*b*-*Pb* is orthogonal to a.



How to compute *c*?

$$Pb = ma$$ (*m* is a scalar)

$$c = b - Pb = b - ma \perp a$$

↓

$$a^T(b - ma) = 0$$

↓

$$m = \frac{a^T b}{a^T a}$$

Finally: $$c = b - \frac{a^T b}{a^T a} a$$

An important information:

$$span\{b, a\} = span\{c, a\}$$

Preliminaries

- Orthogonalization of a vector *b* to a group of orthogonal vectors $a_1, a_2, \cdots, a_l$

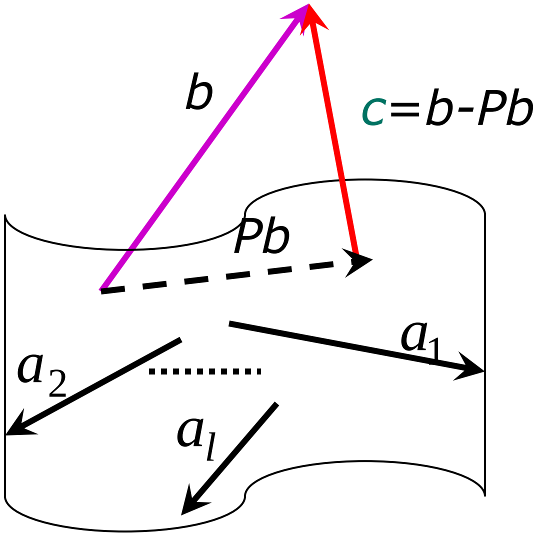
c=b-Pb

$Pb = m_1 a_1 + m_2 a_2 + \cdots + m_l a_l$ and $c \perp a$

$$a_i^T (b - m_1 a_1 - m_2 a_2 - \cdots - m_l a_l) = 0$$

$$a_i^T a_j = 0$$

$$m_i = \frac{a_i^T b}{a_i^T a_i}$$



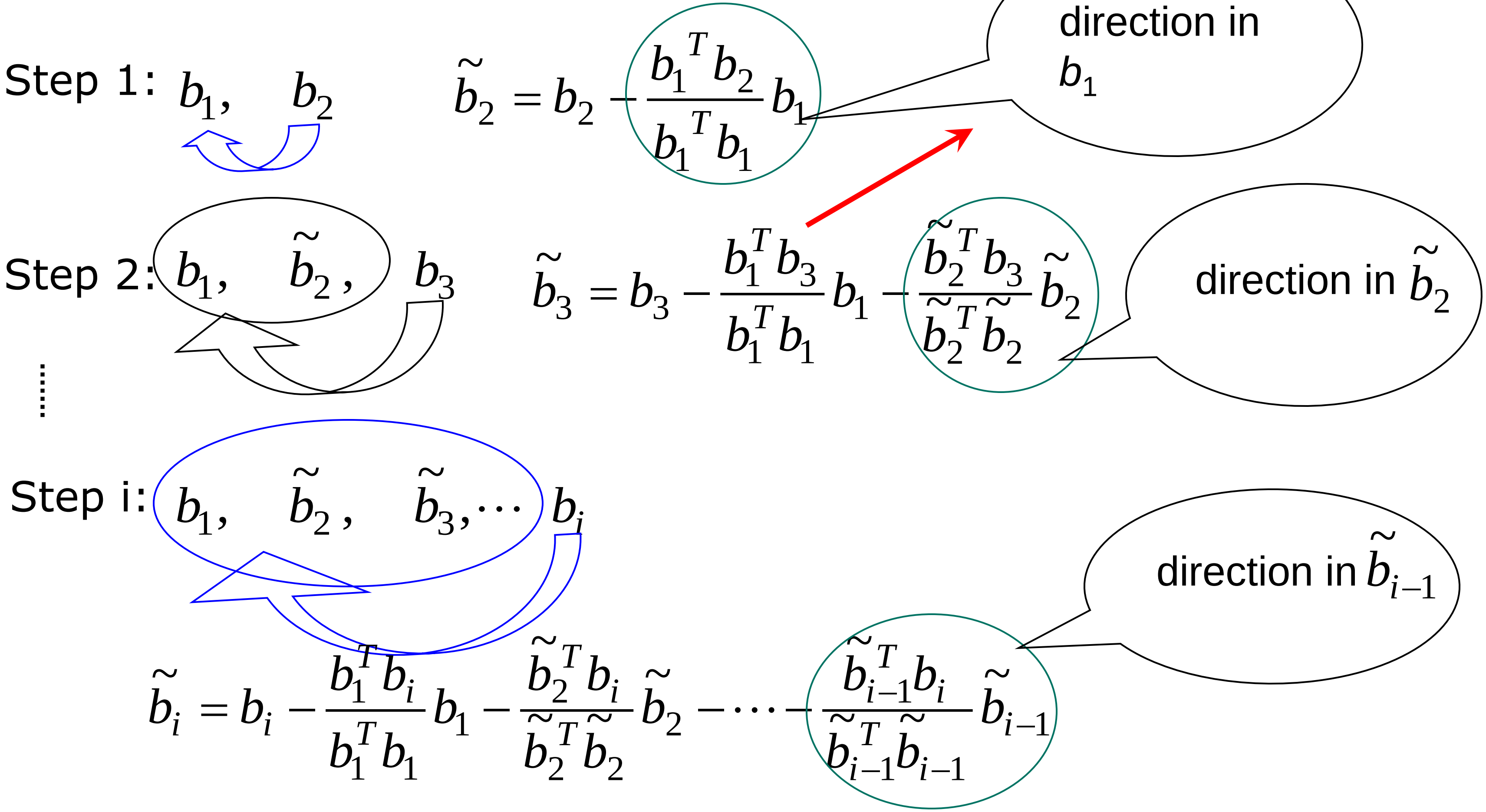
$$c = b - \frac{a_1^T b}{a_1^T a_1} a_1 - \frac{a_2^T b}{a_2^T a_2} a_2 - \cdots - \frac{a_l^T b}{a_l^T a_l} a_l$$

$$span\{b, a_1, a_2, \cdots, a_l\} = span\{c, a_1, a_2, \cdots, a_l\}$$

Preliminaries

- Gram-Schmidt process:

  It is used to orthogonalize a group of vectors $b_1, b_2, \cdots, b_m$



Step 1: $b_1, \quad b_2$

$$\tilde{b}_2 = b_2 - \frac{b_1^T b_2}{b_1^T b_1} b_1$$

direction in b_1

Step 2: $b_1, \quad \tilde{b}_2, \quad b_3$

$$\tilde{b}_3 = b_3 - \frac{b_1^T b_3}{b_1^T b_1} b_1 - \frac{\tilde{b}_2^T b_3}{\tilde{b}_2^T \tilde{b}_2} \tilde{b}_2$$

direction in $\tilde{b}_2$

⋮

Step i: $b_1, \quad \tilde{b}_2, \quad \tilde{b}_3, \cdots \quad b_i$

$$\tilde{b}_i = b_i - \frac{b_1^T b_i}{b_1^T b_1} b_1 - \frac{\tilde{b}_2^T b_i}{\tilde{b}_2^T \tilde{b}_2} \tilde{b}_2 - \cdots - \frac{\tilde{b}_{i-1}^T b_i}{\tilde{b}_{i-1}^T \tilde{b}_{i-1}} \tilde{b}_{i-1}$$

direction in $\tilde{b}_{i-1}$

Preliminaries

Gram-Schmidt process:

```
for i=2,3, ..., m
    $\tilde{b}_i = b_i - \frac{b_1^T b_i}{b_1^T b_1} b_1 - \frac{\tilde{b}_2^T b_i}{\tilde{b}_2^T \tilde{b}_2} \tilde{b}_2 - \cdots - \frac{\tilde{b}_{i-1}^T b_i}{\tilde{b}_{i-1}^T \tilde{b}_{i-1}} \tilde{b}_{i-1}$
end
```

What is the relation between $b_1, b_2, \cdots, b_m$ and $b_1, \tilde{b}_2, \cdots, \tilde{b}_m$?

$$span\{b_1, b_2, \cdots, b_m\} = span\{b_1, \tilde{b}_2, \cdots, \tilde{b}_m\}$$

Preliminaries

Gram-Schmidt process:

```
for $i=2,3, \ldots, m$
    $\tilde{b}_i = b_i - \frac{b_1^T b_i}{b_1^T b_1} b_1 - \frac{\tilde{b}_2^T b_i}{\tilde{b}_2^T \tilde{b}_2} \tilde{b}_2 - \cdots - \frac{\tilde{b}_{i-1}^T b_i}{\tilde{b}_{i-1}^T \tilde{b}_{i-1}} \tilde{b}_{i-1}$
end
```

It is accurate in accurate arithmetic, brings errors in finite arithmetic, not quite orthogonal

Computation with computers is finite arithmetic!

Modified Gram-Schmidt process:

```
for $i=2,3, \ldots, m$
    $\tilde{b}_i = b_i$
    for $j=1,2, \ldots, i-1$
        $\tilde{b}_i = \tilde{b}_i - \frac{\tilde{b}_j^T \tilde{b}_i}{\tilde{b}_j^T \tilde{b}_j} \tilde{b}_j$
    end
end
```

Any difference, and what difference?

Numerically stable.

Preliminaries

Usually the vectors are required to be orthonormalized, so that there will be no overflow in the computation with computers.

Modified Gram-Schmidt process :

```
$b_1 = \frac{b_1}{\| b_1 \|}$
for $i=2,3, \ldots, m$
    for $j=1,2, \ldots, i-1$
        $b_i = b_i - \frac{b_j^T b_i}{b_j^T b_j} b_j$
    end
    $b_i = \frac{b_i}{\| b_i \|}$
end
```

What if $\| b_i \|$ is zero or close to zero? What does it mean?

Preliminaries

Modified Gram-Schmidt process with deflation :

```
for $i=2,3, \ldots, m$
    $b_1 = b_1 / \| b_1 \|$
    for $j=1,2, \ldots, i-1$
        $b_i = b_i - \frac{b_j^T b_i}{b_j^T b_j} b_j$
    end
    $\varepsilon_b = \| b_i \|$
    If $\varepsilon_b \geq tol$
        $b_i = b_i / \varepsilon_b$          deflation
    else
        delete $b_i$
    end
end
```

Preliminaries

- Arnoldi algorithm:

It computes an orthonormal basis $v_1, v_2, \cdots v_q$ for the Krylov subspace:

$$K_p(A, r) = \mathrm{span}\{r, Ar, A^2 r, \cdots, A^{p-1} r\}$$

i.e. $\mathrm{span}\{v_1, v_2, \cdots, v_p\} = K_q(A, r)$

The core in Arnoldi algorithm is Modified Gram-Schmidt process.

Preliminaries

Arnoldi algorithm

```
$v_1 = r / \| r \|$
for $i=2,3, \ldots, p$
    $w = Av_{i-1}$
    for $j=1,2, \ldots, i-1$
        $w = w - \frac{v_j^T w}{v_j^T v_j} v_j$
    end
    $\varepsilon_w = \| w \|$
    If $\varepsilon_w \geq tol$
        $v_i = w / \varepsilon_w$
    else
        stop
    end
end
```

Why?

It is clear:

$$\mathrm{span}\{v_1, v_2, \cdots, v_q\} = K_p(A, r), q \leq p$$

$$K_p(A, r) = \{r, Ar, A^2 r, \cdots, A^{p-1} r\}$$

Motivation of AWE method

AWE method[Pillage,Rohrer ‘90] : Asymptotic waveform evaluation method.

Original large-scale system

$$E\frac{dx(t)}{dt} = Ax(t) + Bu(t),$$
$$y(t) = Cx(t), x(0) = 0.$$

Transfer function $H(s) = C(sE - A)^{-1}B$

The transfer function is a function of *s*.

Does there exist a *easier to compute* $\hat{H}(s)$ ***such that*** $H(s) \approx \hat{H}(s)$ ***?***

Padé approximation

- Padé approximation:

  Approximates a function *f(x)* (analytic) by a rational function, and requires that *f(x)* and its derivatives be continuous at *x=0*.

- Rational function:

  A rational function is the quotient of two polynomials $P_N(x)$ and $Q_M(x)$ of degree N and M respectively:

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)}, \qquad for \quad a \le x \le b$$

The transfer function can be approximated by Padé approximation!

Padé approximation

- $P_N(x)$ and $Q_M(x)$:

$$P_N(x) = p_0 + p_1 x + p_2 x^2 \underline{+\cdots} + p_N x^N$$
$$Q_M(x) = 1 + q_1 x + q_2 x^2 + \cdots q_M x^M \,.$$

- Notice that in $Q_M(x)$, $q_0=1$, which is without loss of generality. Because, $R_{N,M}(x)$ is not changed when both $P_N(x)$ and $Q_M(x)$ are divided by the same constant.

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)}$$

How to compute Padé approximation

Padé approximation: $R_{N,M}(x)=\frac{P_N(x)}{Q_M(x)}$

$$P_N(x)=p_0+p_1x+p_2x^2+\cdots+p_Nx^N$$

$$Q_M(x)=1+q_1x+q_2x+\cdots q_Mx^M\,.$$

The coefficients in $P_N(x)$ and $Q_M(x)$ can be computed by requiring : *f(x)* and $R_{N,M}(x)$ agree at $x=0$ and at their derivatives (at $x=0$) up to N+M degree.

Maclaurin expansion: $f(x) = f_0 + f_1x + f_2x^2 +\cdot\cdot\cdot+f_k\,x^k +\cdot\cdot\cdot\,,$

Maclaurin expansion: $R_{N,M}(x) = r_0 + r_1x + r_2x^2 +\cdot\cdot\cdot+r_k\,x^k +\cdot\cdot\cdot\,,$

This implicates:

$$R_{N,M}(x)-f(x)=:e(x)=\sum_{j=N+M+1}^{\infty}e_jx^j$$

How to compute Padé approximation

$$R_{N,M}(x) - f(x) = P_N(x)/Q_M(x) - f(x)$$

$$R_{N,M}(x) - f(x) = \sum_{j=N+M+1}^{\infty} e_j x^j \Longrightarrow$$

$$\underline{P_N(x) - Q_M(x) f(x)} = Q_M(x) \sum_{j=N+M+1}^{\infty} e_j x^j = \sum_{\boxed{j=N+M+1}}^{\infty} \tilde{e}_j x^j$$

$$\Big\downarrow$$

$$\begin{aligned}
x^0 &: \quad f_0 - p_0 = 0 \\
x^1 &: \quad q_1 f_0 + f_1 - p_1 = 0 \qquad (1) \\
&\quad \vdots \\
x^N &: \quad q_M f_{N-M} + q_{M-1} f_{N-M+1} + \cdots + f_N - p_N = 0
\end{aligned}$$

How to compute Padé approximation

$$
\begin{aligned}
x^{N+1}: &\quad q_M f_{N-M+1} + q_{M-1} f_{N-M+2} + \cdots + q_1 f_N + f_{N+1} = 0 \\
x^{N+2}: &\quad q_M f_{N-M+2} + q_{M-1} f_{N-M+3} + \cdots + q_1 f_{N+1} + f_{N+2} = 0 \qquad (2) \\
&\quad \vdots \\
x^{N+M}: &\quad q_M f_N + q_{M-1} f_{N+1} + \cdots + q_1 f_{N+M-1} + f_{N+M} = 0
\end{aligned}
$$

M unknowns and M equations in (2), q_is can be obtained by solving (2),

p_is can be immediately obtained from (1) without solving equations.

How to compute Padé approximation

An example:

$$f(x)=\sqrt{x+1},0\le x\le 1$$

$$f_0=f(0)=1;\quad f_1=f^{'}(0)=1/2;$$
$$f_2=f^{''}(0)=-1/4;\quad f_3=f^{(3)}(0)=3/8.$$

$$R_{1,1}(x)=\frac{P_1(x)}{Q_1(x)}=\frac{p_0+p_1x}{1+q_1x}$$

1) $q_1f_1+f_2=0\Rightarrow q_1=f_2/f_1=-1/2$

2) $f_0-p_0=0\Rightarrow p_0=1$

$$q_1f_0+f_1-p_1=0\Rightarrow$$
$$p_1=(-1/2)\times 1+1/2=0$$

3) $R_{1,1}(x)=$

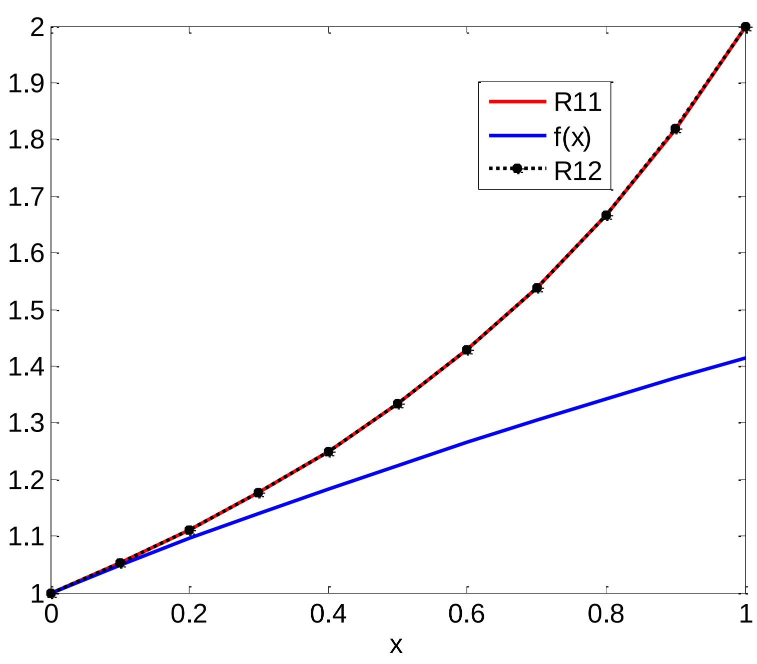
$$R_{1,2}(x)=\frac{P_1(x)}{Q_2(x)}=\frac{p_0+p_1x}{1+q_1x+q_2x^2}$$

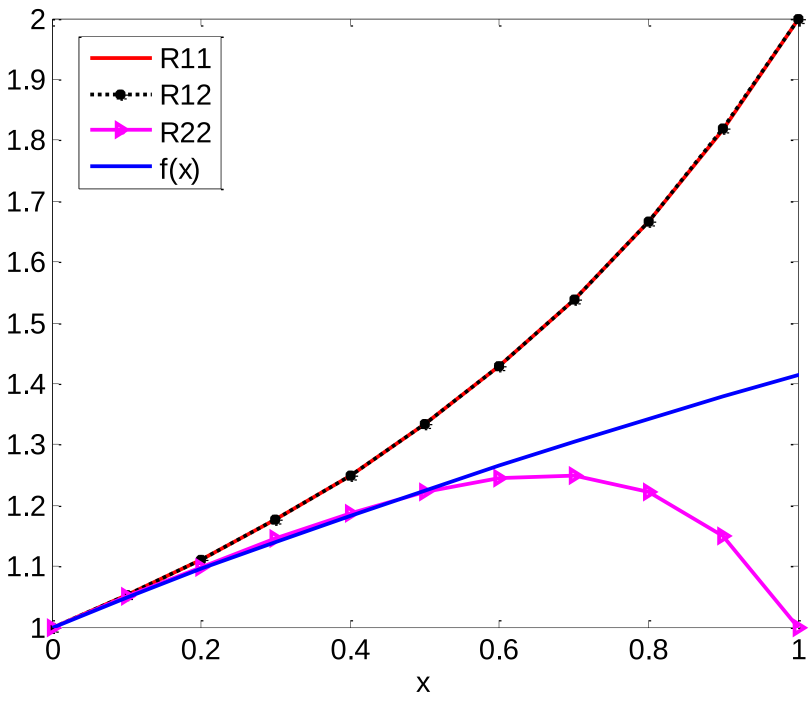
1)
$$q_2f_0+q_1f_1+f_2=0$$
$$q_2f_1+q_1f_2+f_3=0$$
$$\Downarrow$$
$$\begin{pmatrix} f_1 & f_0 \\ f_2 & f_1 \end{pmatrix}\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}=\begin{pmatrix} f_2 \\ f_3 \end{pmatrix}\Rightarrow\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}=\begin{pmatrix} -1 \\ 0.25 \end{pmatrix}$$

2)
$$\begin{aligned} f_0-p_0&=0 \\ q_1f_0+f_1-p_1&=0 \end{aligned}\Rightarrow\begin{aligned} p_0&=1 \\ p_1&=-1+1/2 \\ &=-0.5 \end{aligned}$$

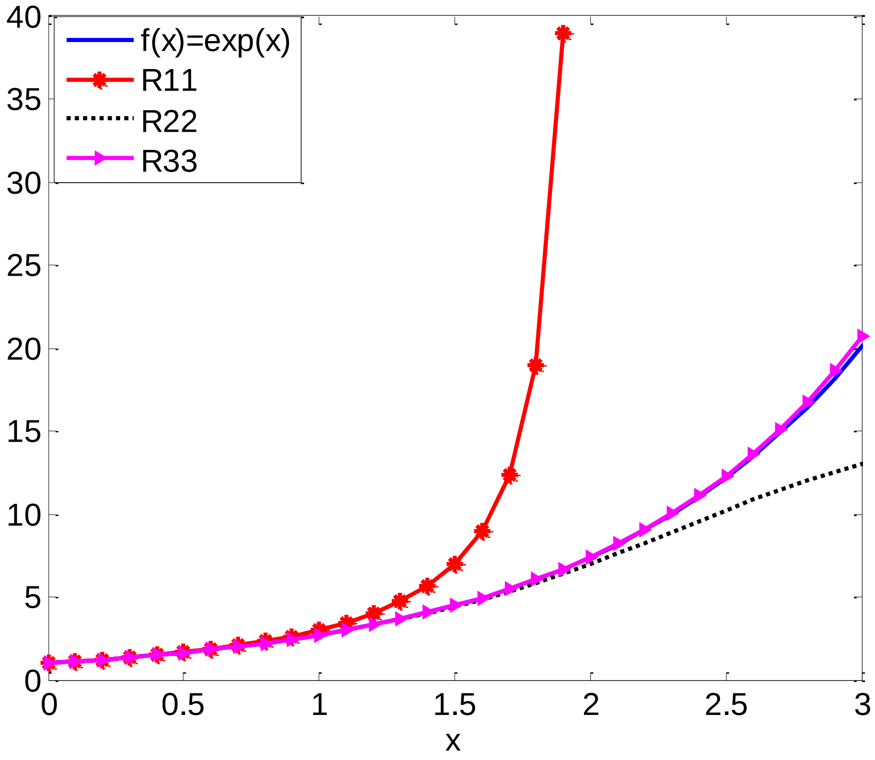
3) $R_{1,2}(x)=$

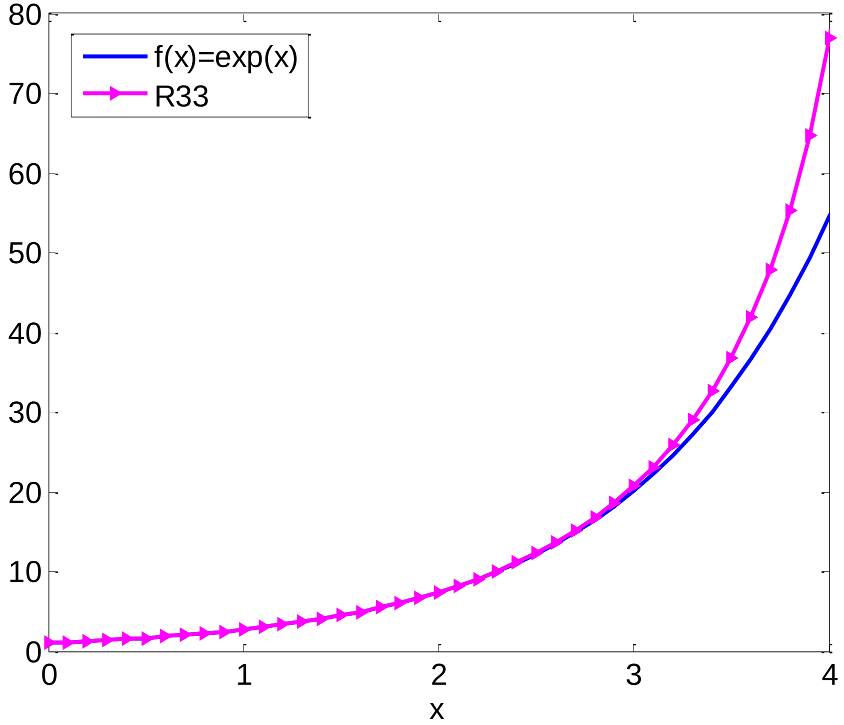
How to compute Padé approximation





How to compute Padé approximation





Padé approximation is only accurate around 0

MOR: AWE based on Padé approximation

MOR: AWE tries to find a Padé approximation $\hat{H}(s)$ of $H(s)$.

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)} \qquad \text{N, M ?}$$

- How to choose N and M in $P_N(x)$ and $Q_M(x)$?:

Proposition*:
For a fixed value of N+M, the error is smallest when $P_N(x)$ and $Q_M(x)$ have the same degree: N=M or when $P_N(x)$ has degree one higher than $Q_M(x)$: N=M+1.

*from the book: John H. Mathews and Kurtis K. Fink, Numerical Methods Using Matlab, 4th Edition, Prentice-Hall Inc. Upper Saddle River, New Jersey, USA 2004.

MOR: AWE based on Padé approximation

Given a system

$$E\frac{dx}{dt} = Ax(t) + Bu(t)$$

$$y = cx(t)$$

The transfer function is $H(s) = c^T(sE - A)^{-1}b = c^T(-sA^{-1}E + I)^{-1}(-A^{-1}b)$

Assuming $\tilde{A} = A^{-1}E$ is diagonalizable: $\tilde{A} = Z\Lambda Z^{-1}, \Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$

$$\begin{aligned} H(s) &= c^T(I - s\tilde{A})^{-1}(-A^{-1}b) \\ &= c^T Z(I - s\Lambda)^{-1} Z^{-1}\tilde{b} \\ &= \tilde{c}^T(I - s\Lambda)^{-1}\tilde{b} \end{aligned} \quad \Longrightarrow \quad H(s) = \sum_{j=1}^{n} \frac{\tilde{l}_j \tilde{b}_j}{1 - s\lambda_j}$$

- *H*(*s*) is a rational function.
- Numerator polynomial is of degree at most n-1, denominator polynomial is of degree at most n.

MOR: AWE based on Padé approximation

Therefore, it is natural to take *M*=*r*, and *N*=*r*-1

$$R_{r-1,r}(x) = \frac{P_N(x)}{Q_M(x)} = \frac{P_{r-1}(x)}{Q_r(x)}$$

$$\hat{H}(s) = \frac{P_{r-1}(s)}{Q_r(s)} = \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^r}$$

Computing the coefficients:

Pade approximation requires:

The values of H(s) at s=0, and the derivatives of H(s) at s=0 till r-1+r degree should be the same as those of $\hat{H}(s)$.

MOR: AWE based on Padé approximation

Derivatives of H(s) at s=0 are the coefficients of Maclaurin series of *H*(*s*):

$$H(s) = c^T (sE - A)^{-1} b = c^T (I - \underbrace{sA^{-1}E}_{\tilde{A}})^{-1} \underbrace{(-A^{-1})b}_{\tilde{b}} = \sum_{i=0}^{\infty} c^T \tilde{A}^i \tilde{b} s^i$$

$\boxed{m_i = c^T \tilde{A}^i \tilde{b}}, i = 0,1,\cdots$ are the moments of the transfer function.

$$H(s) = \sum_{i=0}^{\infty} l^T \tilde{A}^i \tilde{b} s^i = \sum_{i=0}^{\infty} m_i s^i = m_0 + m_1 s + m_2 s^2 + \cdots$$

$$\hat{H}(s) = \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^r} = \hat{m}_0 + \hat{m}_1 s + \hat{m}_2 s^2 + \cdots$$

$$(m_0 + m_1 s + m_2 s^2 + \cdots) - \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^r} = e_{2r} s^{2r} + e_{2r+1} s^{2r+1} + \cdots$$

MOR: AWE based on Padé approximation

As has been introduced above, q_is can be obtained by solving :

$$\begin{aligned} q_r m_0 + q_{r-1} m_1 + \cdots + q_1 m_{r-1} + m_r &= 0 \\ q_r m_1 + q_{r-1} m_2 + \cdots + q_1 m_r + m_{r+1} &= 0 \\ &\vdots \\ q_r m_{r-1} + q_{r-1} m_r + \cdots + q_1 m_{2r-2} + m_{2r-1} &= 0 \end{aligned} \qquad (3)$$

p_is can be immediately obtained from:

$$\begin{aligned} m_0 - p_0 &= 0 \\ q_1 m_0 + m_1 - p_1 &= 0 \\ &\vdots \\ q_{r-1} m_0 + q_{r-2} m_1 + \cdots + m_{r-1} - p_{r-1} &= 0 \end{aligned} \qquad (4)$$

$$\boxed{\hat{H}(s) = \frac{P_{r-1}(s)}{Q_r(s)} = \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^r}}$$

Ok ?

MOR: AWE based on Padé approximation

$\hat{H}(s)$ is inaccurate at high frequency due to **floating point overflow.**

$$\hat{H}(s) = \frac{P_{r-1}(s)}{Q_r(s)} = \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^r}$$

Any other possible way? Yes!: Parial Fractions Decomposition:

$$\hat{H}(s) = \frac{p_0 + p_1 s + \cdots + p_{r-1} s^{r-1}}{1 + q_1 s + \cdots q_r s^{r-1}} = \frac{\hat{k}_1}{s - \hat{a}_1} + \frac{\hat{k}_2}{s - \hat{a}_2} + \cdots + \frac{\hat{k}_r}{s - \hat{a}_r}$$

If we know $\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_r$ (approximate poles) and (approximate residues) $\hat{k}_1, \hat{k}_2, \cdots, \hat{k}_r$ then $\hat{H}(s)$ is known and is easily computed.

How to compute the poles and the residues?

$\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_r$ are nothing but the roots of $1 + q_1 s + \cdots + q_r s^r$

and we have known how to compute $q_i s$!

MOR: AWE based on Padé approximation

What is left? computation of the residues: $\hat{k}_1, \hat{k}_2, \cdots, \hat{k}_r$

From Pade approximation:

$$\underbrace{\hat{k}_1(s-\hat{a}_1)^{-1} + \hat{k}_2(s-\hat{a}_2)^{-1} + \cdots + \hat{k}_r(s-\hat{a}_r)^{-1}} - [m_0 + m_1 s + \cdots + m_{2r-1}s^{2r-1} + m_{2r}s^{2r} + \cdots]$$

$$= e_{2r}s^{2r} + e_{2r+1}s^{2r+1} + \cdots$$

$$\downarrow$$

$$-\frac{\hat{k}_1}{\hat{a}_1}(1-\frac{s}{\hat{a}_1})^{-1} - \frac{\hat{k}_2}{\hat{a}_2}(1-\frac{s}{\hat{a}_2})^{-1} - \cdots - \frac{\hat{k}_r}{\hat{a}_r}(1-\frac{s}{\hat{a}_r})^{-1}$$

$$\downarrow$$

$$-\frac{\hat{k}_1}{\hat{a}_1}(1+\frac{s}{\hat{a}_1}+\frac{s^2}{\hat{a}_1^2}+\cdots) - \cdots - \frac{\hat{k}_r}{\hat{a}_r}(1+\frac{s}{\hat{a}_r}+\frac{s^2}{\hat{a}_r^2}+\cdots)$$

MOR: AWE based on Padé approximation

$$\begin{gathered}
\frac{\hat{k}_1}{\hat{a}_1}+\frac{\hat{k}_2}{\hat{a}_2}+\cdots+\frac{\hat{k}_r}{\hat{a}_r}=-m_0 \\
\frac{\hat{k}_1}{\hat{a}_1^2}+\frac{\hat{k}_2}{\hat{a}_1^2}+\cdots+\frac{\hat{k}_r}{\hat{a}_r^2}=-m_1 \\
\vdots \\
\frac{\hat{k}_1}{\hat{a}_1^r}+\frac{\hat{k}_2}{\hat{a}_2^r}+\cdots+\frac{\hat{k}_r}{\hat{a}_r^r}=-m_{r-1}
\end{gathered}$$

$$\Longrightarrow \begin{pmatrix} \hat{a}_1^{-1} & \hat{a}_2^{-1} & \cdots & \hat{a}_r^{-1} \\ \hat{a}_1^{-2} & \hat{a}_2^{-2} & \cdots & \hat{a}_r^{-2} \\ & & \vdots & \\ \hat{a}_1^{-r} & \hat{a}_2^{-r} & \cdots & \hat{a}_r^{-r} \end{pmatrix} \begin{pmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_r \end{pmatrix} = \begin{pmatrix} -m_0 \\ -m_1 \\ \vdots \\ -m_{r-1} \end{pmatrix} \text{(5)}$$

the residues can be obtained by solving the above euqations!

Once $\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_r$ and $\hat{k}_1, \hat{k}_2, \cdots, \hat{k}_r$ are computed, we have:

$$\hat{H}(s)=\frac{\hat{k}_1}{s-\hat{a}_1}+\frac{\hat{k}_2}{s-\hat{a}_2}+\cdots+\frac{\hat{k}_r}{s-\hat{a}_r}$$

MOR: AWE based on Padé approximation

- Why not approximation by *truncated* Taylor expansion?
  - The Padé approximant often gives better approximation of the function than truncating its Taylor series, and it may still work where the Taylor series does not converge.
  - Round off error or overflow of in truncated Taylor expansion is avoided .
  - Poles and residues of $H(s)$ can be computed more easily by Pade approximation.
  - $H(s)$ itself is a rational function.

MOR: AWE based on Padé approximation

Implementation of AWE:

1. Solve (3) to get q_is.

2. Compute the roots of the polynomial: $1+q_1 s+\cdots+q_r s^r$

   the roots: $\hat{a}_1,\hat{a}_2,\cdots,\hat{a}_r$

3. Solve (5) to get $\hat{k}_1,\hat{k}_2,\cdots,\hat{k}_r$

4. Form the reduced (simpler) transfer function:

$$\hat{H}(s)=\frac{\hat{k}_1}{s-\hat{a}_1}+\frac{\hat{k}_2}{s-\hat{a}_2}+\cdots+\frac{\hat{k}_r}{s-\hat{a}_r}$$

Much Easier to be computed than H(s)!

In MATLAB step 2. and 3. are implemented in the function: residue.m

MOR: AWE based on Padé approximation

In MATLAB:

1. Solve (3) to get q_i*s.*

2. Get p_i*s* from (4).

3. Use *residue(p,q)* in matlab to get the poles and residues:

$$\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_r; \qquad \hat{k}_1, \hat{k}_2, \cdots, \hat{k}_r$$

4. Form the Pade approximation (approximate transfer function):

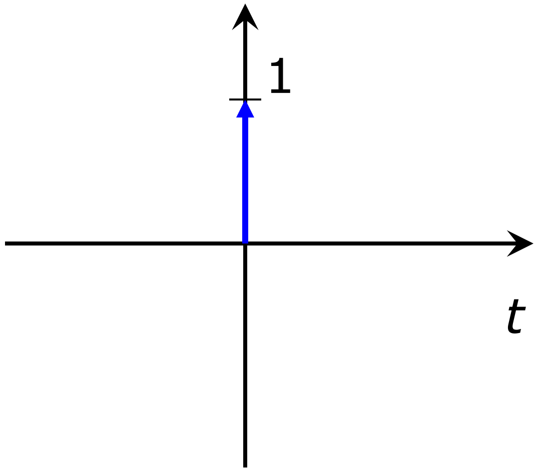
$$\hat{H}(s) = \frac{\hat{k}_1}{s-\hat{a}_1} + \frac{\hat{k}_2}{s-\hat{a}_2} + \cdots + \frac{\hat{k}_r}{s-\hat{a}_r}$$

MOR: AWE based on Padé approximation

How to compute the output response $y(t)$ in time domain from $\hat{H}(s)$?

Definition of transfer function:

$$H(s) = Y(s)/U(s)$$



If the input is the unit impulse function:

$$u(t) = \delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

then $$U(s) = \int_0^{\infty} \delta(t) e^{-st} dt = 1$$

Therefore with impulse input: $H(s) = Y(s)/U(s) = Y(s)$

$$y(t) = \frac{1}{2\pi j} \int_{\gamma - j\infty}^{\gamma + j\infty} Y(s) e^{st} ds = \frac{1}{2\pi j} \int_{\gamma - j\infty}^{\gamma + j\infty} H(s) e^{st} ds$$

MOR: AWE based on Padé approximation

We get:

$$y(t) = \frac{1}{2\pi j}\int_{\gamma - j\infty}^{\gamma + j\infty} H(s)e^{st}ds \approx \frac{1}{2\pi j}\int_{\gamma - j\infty}^{\gamma + j\infty} \hat{H}(s)e^{st}ds$$

Replace $\hat{H}(s)$ with:

$$\hat{H}(s) = \frac{\hat{k}_1}{s-\hat{a}_1} + \frac{\hat{k}_2}{s-\hat{a}_2} + \cdots + \frac{\hat{k}_r}{s-\hat{a}_r}$$

We obtain:

$$y(t) \approx \sum_{i=1}^{r} \hat{k}_i e^{\hat{a}_i t}$$

Impulse output response
in time domain

Because:

$$\frac{1}{2\pi j}\int_{\gamma - j\infty}^{\gamma + j\infty} \frac{\hat{k}_i}{s-\hat{a}_i} e^{st}ds = L^{-1}(\frac{\hat{k}_i}{s-\hat{a}_i}) = \hat{k}_i e^{\hat{a}_i t}$$

Numerical instability of AWE

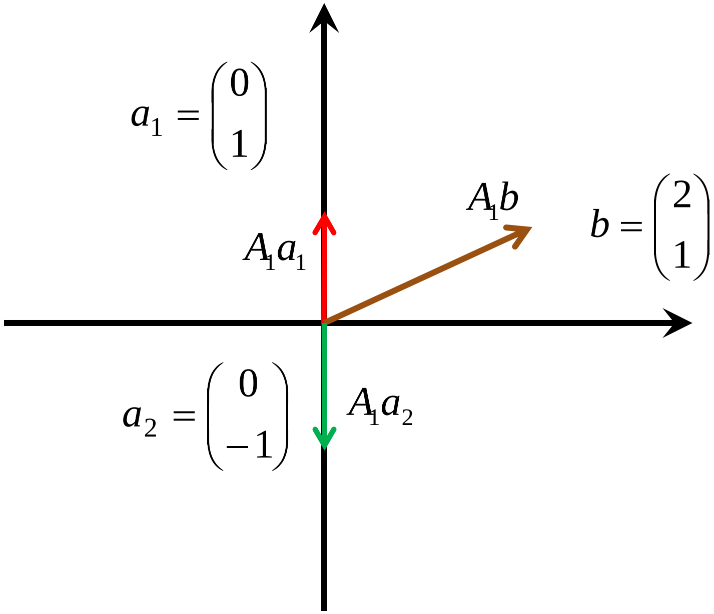
Eigenvalue and eigenvectors of a matrix A

$$A_1 = \begin{pmatrix} 1 & 0 \\ -0.5 & 0.5 \end{pmatrix}$$

$$A_1 a_1 =$$

$$A_1 a_2 =$$

$$A_1 b =$$



$a_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$A_1 a_1$
$A_1 b$
$b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$
$a_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$
$A_1 a_2$

An eigenvector either does not change direction by *A* or is reversed by *A*.

Numerical instability of AWE

Eigenvalue and eigenvectors of a matrix $A \in R^{n \times n}$:

$$A\xi_i = \lambda_i \xi_i, i = 1,2,\cdots n$$

$\xi_i, i = 1,2,\cdots n$ are eigenvectors of A.

$\lambda_i, i = 1,2,\cdots n$ are eigenvalues of A.

Applications to Engineering:

- Vibration of a beam.
- Stability of a system.
-

$$\vdots$$

Numerical instability of AWE

1. Solve (3) to get *q*

$$\begin{aligned} q_r m_0 + q_{r-1} m_1 + \cdots + q_1 m_{r-1} + m_r &= 0 \\ q_r m_1 + q_{r-1} m_2 + \cdots + q_1 m_r + m_{r+1} &= 0 \\ &\vdots \\ q_r m_{r-1} + q_{r-1} m_r + \cdots + q_1 m_{2r-2} + m_{2r-1} &= 0 \end{aligned} \tag{3}$$

$$\Downarrow$$

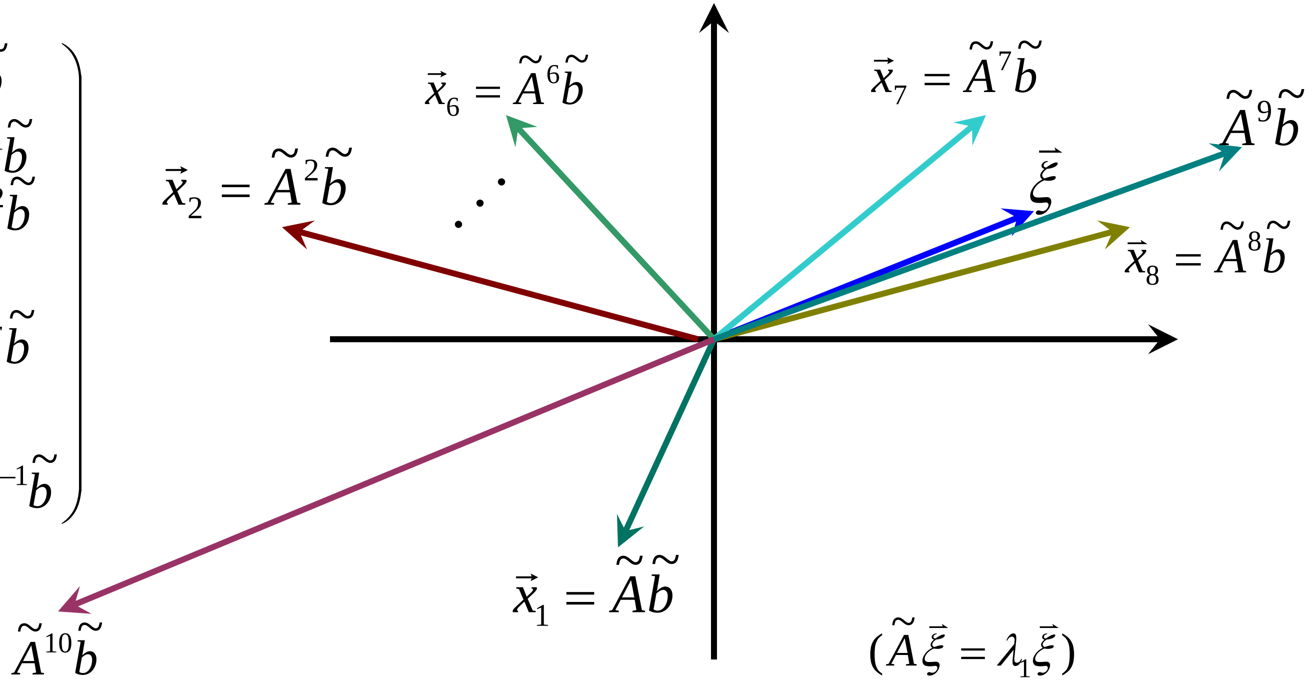
$$\begin{pmatrix} m_0 & m_1 & \cdots & m_{r-1} \\ m_1 & m_2 & \cdots & m_r \\ & & \vdots & \\ m_{r-1} & m_r & \cdots & m_{2r-2} \end{pmatrix} \begin{pmatrix} q_r \\ q_{r-1} \\ \vdots \\ q_1 \end{pmatrix} = \begin{pmatrix} -m_r \\ -m_{r+1} \\ \vdots \\ -m_{2r-1} \end{pmatrix}$$

$$\begin{pmatrix} -m_0 \\ -m_1 \\ -m_2 \\ \vdots \\ -m_r \\ \vdots \\ -m_{2r-1} \end{pmatrix} = c^T \begin{pmatrix} -\tilde{b} \\ -\tilde{A}^1 \tilde{b} \\ -\tilde{A}^2 \tilde{b} \\ \vdots \\ -\tilde{A}^{r-1} \tilde{b} \\ \vdots \\ -A^{2r-1} \tilde{b} \end{pmatrix}$$

$$\tilde{A} = A^{-1} E, \tilde{b} = -A^{-1} b$$

Numerical instability of AWE

$$\begin{pmatrix} -m_0 \\ -m_1 \\ -m_2 \\ \vdots \\ -m_r \\ \vdots \\ -m_{2r-1} \end{pmatrix} \rightarrow \begin{pmatrix} -\tilde{b} \\ -\tilde{A}^1\tilde{b} \\ -\tilde{A}^2\tilde{b} \\ \vdots \\ -\tilde{A}^r\tilde{b} \\ \vdots \\ -\tilde{A}^{2r-1}\tilde{b} \end{pmatrix}$$


$\vec{x}_6 = \tilde{A}^6\tilde{b}$
$\vec{x}_7 = \tilde{A}^7\tilde{b}$
$\tilde{A}^9\tilde{b}$
$\vec{x}_2 = \tilde{A}^2\tilde{b}$
$\vec{\xi}$
$\vec{x}_8 = \tilde{A}^8\tilde{b}$
$\vec{x}_1 = \tilde{A}\tilde{b}$
$\tilde{A}^{10}\tilde{b}$
$(\tilde{A}\vec{\xi} = \lambda_1\vec{\xi})$

$\vec{x}_i = \tilde{A}^i\tilde{b}$, *i=1,2,...* run parallel to $\vec{\xi}$ soon !

Usually, after *i*=8, all $\vec{x}_i = A^i\tilde{b}$ will in the same direction with $\vec{\xi}$.

$\Downarrow$

Numerical instability of AWE

$$\tilde{b} = c_1\xi_1 + c_2\xi_2 + \cdots + c_n\xi_n$$

$$\begin{aligned}\tilde{A}^m\tilde{b} &= c_1\tilde{A}^m\xi_1 + c_2\tilde{A}^m\xi_2 + \cdots + c_n\tilde{A}^m\xi_n \\ &= c_1\lambda_1^m\xi_1 + c_2\lambda_2^m\xi_2 + \cdots + c_n\lambda_n^m\xi_n \\ &= c_1\lambda_1^m\left(\xi_1 + \frac{c_2}{c_1}\left(\frac{\lambda_2}{\lambda_1}\right)^m\xi_2 + \cdots + \frac{c_n}{c_1}\left(\frac{\lambda_n}{\lambda_1}\right)^m\xi_n\right)\end{aligned}$$

$m \to \infty \ (\lambda_1 > \lambda_2 > \cdots > \lambda_n)$

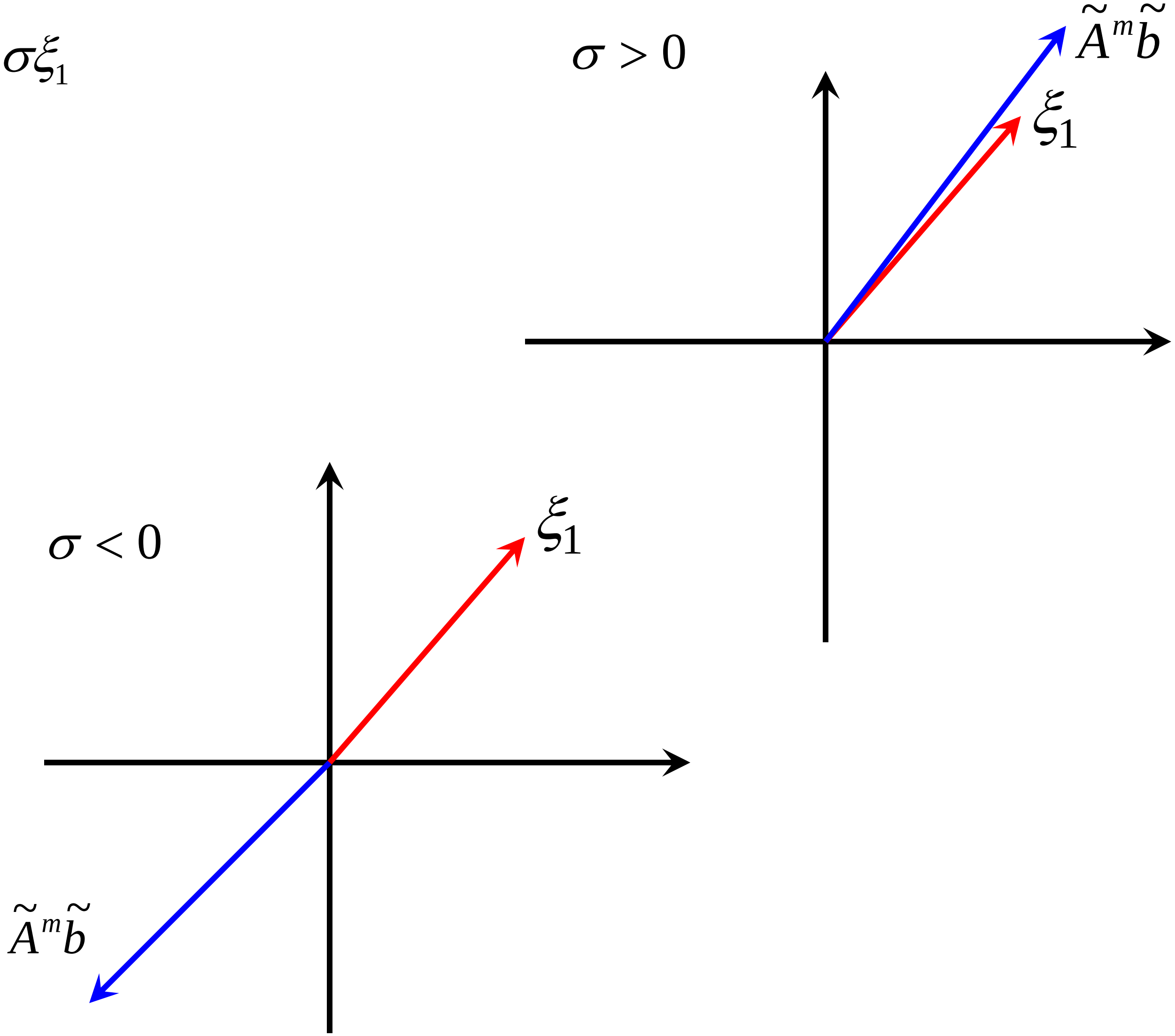
$$\tilde{A}^m\tilde{b} \to (c_1\lambda_1^m)\xi_1 = \sigma\xi_1$$

↓

If λ_1 and λ_2 are not close, round-off error changes $\tilde{A}^m\tilde{b}$ to $\sigma\zeta_1$

Numerical instability of AWE

$\tilde{A}^m \tilde{b} \rightarrow \sigma \xi_1$



Numerical instability of AWE

For many examples, when $i > 8$, all $\tilde{A}^i\tilde{b}$ will be almost on the same line with ξ_1, this means $\tilde{A}^i\tilde{b}$ only contain the information of λ_1, $\tilde{A}\xi_1 = \lambda_1\xi_1$.

Notice: $m_i = c^T\tilde{A}^i\tilde{b}$

Therefore $m_i, i > 8$ also only contain the information of λ_1.

However, the original transfer function contains the information of all the eigenvalues:

$$H(s) = \sum_{j=1}^{n} \frac{\tilde{l}_j\tilde{b}_j}{s-\lambda_j} = \sum_{j=1}^{n} \frac{\tilde{l}_j\tilde{b}_j}{s-\lambda_j} = \sum_{j=1}^{n} \frac{k_j}{s-a_j}$$

Conclusion:

Although theoretically, employing more moments to compute more q_i will match more moments, and will lead to more accurate $\hat{H}(s)$, numerically, the accuracy of $\hat{H}(s)$ cannot be improved by using more moments!

Numerical instability of AWE

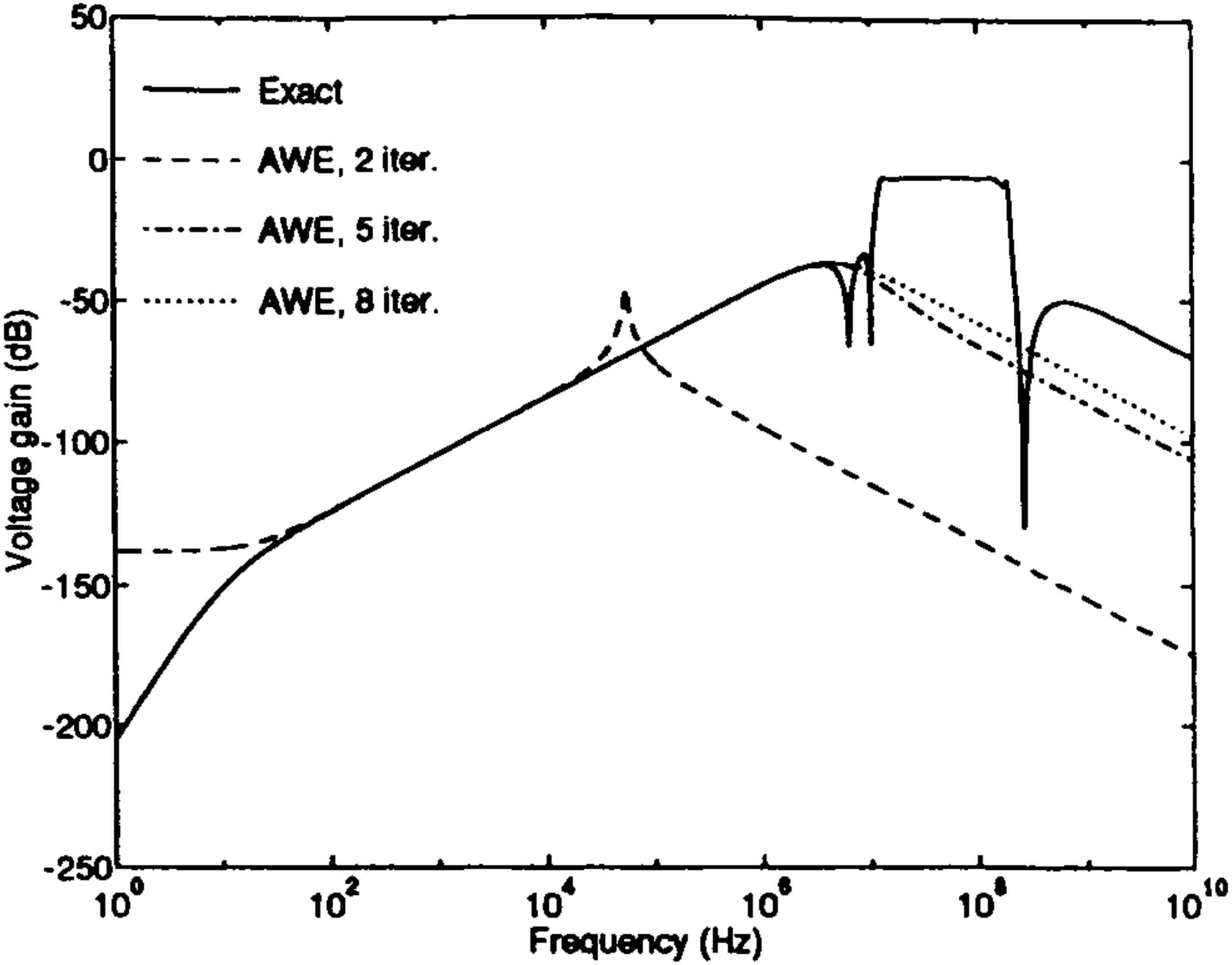


Fig. 1. Results for simulation of voltage gain with AWE.

L. T. Pillage, R. A. Rohrer, Asymptotic Waveform Evaluation for Timing Analysis, IEEE Transactions on computer-aided design, Vol. 9, No. 4, 1990.

Implicit moment-matching(Pade, Pade-type approximation)

Recall that for projection based MOR:

$$\begin{aligned} E\frac{dx}{dt} &= Ax(t)+Bu(t) \\ y &= Cx(t) \end{aligned} \xrightarrow{x \approx Vz} \begin{aligned} W^T EVdz/dt &= W^T AVz(t)+W^T Bu(t) \\ \hat{y} &= CVz(t) \end{aligned}$$

By definition: $H(s) = y(s)/u(s) = Cx(s)/u(s)$

Taylor expansion at s$_0$:

$$\begin{aligned} H(s) &= c(sE-A)^{-1}B \\ &= c(sE - s_0E + s_0E - A)^{-1}B \\ &= c((s-s_0)(s_0E-A)^{-1}E + I)^{-1}(s_0E-A)^{-1}B \\ &= c\sum_{i=0}^{\infty}\tilde{A}(s_0)\tilde{B}(s_0)(s-s_0)^i \end{aligned}$$

where $\tilde{A}(s_0) = -(s_0E-A)^{-1}E$, $\tilde{B}(s_0) = (s_0E-A)^{-1}B$

Definition (moments are defined for any expansion point $s_0 < \infty$)

$M_i(s_0) = C\tilde{A}^i(s_0)\tilde{B}(s_0)$ ($c\tilde{A}^i(s_0)\tilde{b}(s_0)$ for SISO system), $i = 0,\ldots,$ are called the moments of the transfer function.

Implicit moment-matching(Pade, Pade-type approximation)

Recall that in AWE method, the moments

$$M_i = c\tilde{A}^i(0)\tilde{b}(0), i = 0,\ldots, 2r-1$$

are computed explicitly. It is numerically unstable.

Observe that $M_t = c\tilde{A}^t(s_0)\tilde{b}(s_0), t = 0,\ldots,$ can be seen as being composed either of :

$$\tilde{A}^i(s_0)\tilde{b}(s_0), i = 0,1\ldots \qquad (4)$$

$$\text{and } c\tilde{A}^j(s_0),\ j = 0,1,\ldots \qquad (5)$$

or of

$$\tilde{A}^t(s_0)\tilde{b}(s_0), t = 0,1\ldots \qquad (6)$$

$$\text{and } c \qquad (7)$$

Instead of explicitly computing the terms in (4)(5) or in (6)(7) as is done by AWE method, one can compute the basis of

$$\text{range(V)} = \text{span}\{\tilde{b}(s_0), \tilde{A}(s_0)\tilde{b}(s_0), \ldots, \tilde{A}^{p-1}(s_0)\tilde{b}(s_0)\} \qquad (8)$$

$$\text{range }(\tilde{\text{W}}) = \text{span}\{c^T, \tilde{A}^T(s_0)c, \ldots, (\tilde{A}^T(s_0))^{p-1}c^T\} \qquad (9)$$

Implicit moment-matching(Pade approximation)

How to compute W, V? [Feldmann, Freund ’95]

Recall that

$$\mathrm{range}(V) = \mathrm{span}\{\tilde{b}(s_0), \tilde{A}(s_0)\tilde{b}(s_0), \ldots, \tilde{A}^{p-1}(s_0)\tilde{b}(s_0)\} = K_p(\tilde{A}(s_0), \tilde{b}(s_0)) \tag{8}$$

$$\mathrm{range}\,(\tilde{W}) = \mathrm{span}\{c^T, \tilde{A}^T(s_0)c, \ldots, (\tilde{A}^T(s_0))^{p-1}c^T\} = K_p(\tilde{A}^T(s_0), c^T) \tag{9}$$

W, V span two Krylov subspaces, so that they can be simultaneously computed by (band)Lanczos algorithm, such that $\tilde{W}^T V = I$.

The outputs of Lanczos algorithm are

$$T = \tilde{W}^T (A - s_0 E)^{-1} E V$$

$$\rho = \tilde{W}^T (A - s_0 E)^{-1} b$$

$$\eta = V^T c^T$$

Implicit moment-matching (Pade approximation)

Applying Petrov-Galerkin projection (using $\tilde{W}, V$) to the transformed system

$$
\begin{aligned}
&(A-s_0E)^{-1}E\frac{dx}{dt} = (A-s_0E)^{-1}Ax(t) + (A-s_0E)^{-1}bu(t),\\
&y = cx(t),
\end{aligned}
\tag{10}
$$

One gets the reduced order model (ROM)

$$
\begin{aligned}
&\tilde{W}^T(A-s_0E)^{-1}EV\frac{dz}{dt} = \tilde{W}^T(A-s_0E)^{-1}AVz(t) + \tilde{W}^T(A-s_0E)^{-1}bu(t),\\
&y = cVz(t),
\end{aligned}
\tag{11}
$$

By studying the ROM in (11) [Freund '03]

one can see that (10) - (11) is equivalent to applying Petrov - Galerkin projection to the original system with $W = (A-s_0E)^{-T}\tilde{W}$, and V.

Implicit moment-matching (Pade approximation)

Actually the ROM in (11) can be implicitly derived using the outputs of the Lanczos algorithm[Freund '03] :

$$T\frac{dz}{dt} = (I + s_0 T)z(t) + \rho u(t),$$
$$y = \eta^T z(t), \qquad (12)$$

Theorem 1 [Feldmann, Freund '95]

If $\tilde{\mathrm{W}}$, V are the basis of the subspace in (8)(9), and satisfy $\tilde{W}^T V = I$,
then for a SISO system in (10), the first $2p$ moments of the transfer function
of the ROM in (11) match the first $2p$ moments of $H(s)$, i.e.

$$M_i(s_0) = \hat{M}_i(s_0), i = 0,1,\ldots,2p-1.$$

Therefore, $\hat{H}(s)$ is a Pade approximation of $H(s)$.

Drawbacks of Lanczos method of computing the projection matrices:

- The ROM computed by W, V may be unstable, there are eigenvalues with positive real parts.
- The Lanczos method does not maintain precise biorthogonality given limited numerical precision.

Implicit moment-matching(Pade approximation)

Number of moments matched for MIMO systems

In [Freund '00], it is shown that if $\tilde{\mathrm{W}}$, V are any basis of the subspace in (8)(9) (they are not necessarily generated by Lanczos algorithm), and satisfy $\tilde{W}^T V = I$, then $\hat{H}(s)$ matches at least the first $\lfloor \mathrm{r/n_O} \rfloor + \lfloor \mathrm{r/n_I} \rfloor$ moments of $H(s)$, and it is a matrix Pade approximant of $H(s)$. Here r is the order of the reduced model, or equivalently, the number of the columns in V or W. There are n_I inputs and n_O outputs.

It is immediately seen from the above statement that for SISO systems, there are at least *2r* moments matched. Moreover, for SISO systems, *r* = *p*. **Therefore, there is no contradiction with Theorem 1.**

Implicit moment-matching(Pade-type approximation)

Pade-type approximation [Odabasioglu,et.al ’97, Freund ‘03]

A passive (therefore stable) ROM can be obtained by using W=V.

$$\mathrm{range}(V) = \mathrm{span}\{\tilde{b}(s_0), \tilde{A}(s_0)\tilde{b}(s_0), \ldots, \tilde{A}^{p-1}(s_0)\tilde{b}(s_0)\} = K_p(\tilde{A}(s_0), \tilde{b}(s_0))$$

V satisfies $V^T V = I$. Therefore it can be computed by Arnoldi algorithm.

Theorem 2 [Odabasioglu, et.al ’97, Freund ‘03]

If the columns in V constitute an orthogonal basis of the Krylov subspace in (8), then for a SISO system in (10), the first p moments of $H(s)$ are matched by $\hat{H}(s)$,

$$M_i(s_0) = \hat{M}_i(s_0), i = 0,1,\ldots, p-1.$$

$\hat{H}(\mathrm{s})$ is a Pade - type approximation of $H(s)$.

Implicit moment-matching (Pade, Pade-type approximation)

The ROM is obtained by Galerkin projection onto the original system.

$$\begin{aligned} &V^T E V dz/dt = V^T A V z(t) + V^T b u(t) \\ &\hat{y} = cVz(t) \end{aligned} \qquad (11)$$

Why using $x \approx Vz$, rather than $x \approx Wz$?

Recall: $$H(s) = c(sE - A)^{-1} b = c\sum_{i=0}^{\infty} \tilde{A}(s_0)\tilde{b}(s_0)(s - s_0)^i$$

$$cx(s) = H(s)u(s) \Rightarrow x(s) = \sum_{i=0}^{\infty} \tilde{A}^i(s_0)\tilde{b}(s_0)(s - s_0)^i u(s) = \sum_{i=0}^{\infty} g_i(s)\tilde{A}^i(s_0)\tilde{b}(s_0)$$

$$\Longrightarrow x(s) \approx \sum_{i=0}^{p-1} g_i(s)\tilde{A}^i(s_0)\tilde{b}(s_0)$$

$$\text{range}(V) = \text{span}\{\tilde{b}(s_0), \tilde{A}(s_0)\tilde{b}(s_0), \ldots, \tilde{A}^p(s_0)\tilde{b}(s_0)\} = K_{p-1}(\tilde{A}(s_0), \tilde{b}(s_0))$$

$$x \approx Vz \Leftrightarrow x(s) \approx \sum_{i=0}^{p-1} g_i(s)\tilde{A}^i(s_0)\tilde{b}(s_0)$$

Implicit moment-matching(Pade, Pade-type approximation)

Expansion point

The expansion point $s_0 : s = s_0 + \sigma$ can be chosen as zero or nonzero.

If the interesting frequency range is far away from zero, then a nonzero expansion point s_0 is preferred.

Implicit moment-matching(rational interpolation)

- A **single expansion point** is used in the method based on Pade approximation. **Multiple expansion points** are used in rational interpolation method.
- Rational interpolation views computing the transfer function from the viewpoint of **solving linear systems**.

$$H(s) = c(sE - A)^{-1}b = c(sE - A)^{-1}(sE - A)(sE - A)^{-1}b = x_c^T (sE - A)x_b$$

where

$$(sE - A)^T x_c = c^T, \qquad (sE - A)x_b = b$$

Applying a preconditioner to each of the linear systems,

$$(A - s_0E)^{-T}(sE - A)^T x_c = (A - s_0E)^{-T} c^T, \qquad (A - s_0E)^{-1}(sE - A)x_b = (A - s_0E)^{-1}b$$

Implicit moment-matching(rational interpolation)

If using Krylov-subspace iterative methods to solve the preconditioned linear systems, we have (this is the property of Krylov-subpace iterative methods, e.g. CG, GMERS, MINRES. etc..)

$$x_b \approx \hat{x}_b \in K_q((A-s_0E)^{-1}(sE-A),(A-s_0E)^{-1}b)$$
$$x_c \approx \hat{x}_c \in K_q((A-s_0E)^{-T}(sE-A)^T,(A-s_0E)^{-T}c^T)$$

Lemma 2.2 [Grimme '97] Krylov subspace shift-invariance

For any matrix G, vector g and nonzero η,

$$K_q((\eta G+I,g))=K_q(G,g).$$

Since $(A-s_0E)^{-1}(A-sE)=I+(s_0-s)(A-s_0E)^{-1}E$

$$K_q((A-s_0E)^{-1}(sE-A),(A-s_0E)^{-1}b)=K_q((A-s_0E)^{-1}E,(A-s_0E)^{-1}b)$$

$$K_q((A-s_0E)^{-T}(sE-A)^T,(A-s_0E)^{-T}c)=K_q((A-s_0E)^{-T}E^T,(A-s_0E)^{-T}c^T)$$

Implicit moment-matching(rational interpolation)

Compute V, such that

$$\text{range}(V) = K_p((A - s_0E)^{-1}E, (A - s_0E)^{-1}b) \qquad (12)$$

Compute W, such that

$$\text{range}(W) = K_p((A - s_0E)^{-T}E^T, (A - s_0E)^{-T}c^T) \qquad (13)$$

Then $x_b \approx \hat{x}_b = Vz_b,\ x_c \approx \hat{x}_c = Wz_c$

$$H(s) \approx \hat{H}(s) = \hat{x}_c^T(sE - A)\hat{x}_b = z_c^TW^T(sE - A)Vz_b = z_c^T(sW^TEV - W^TAV)z_b$$

Therefore the reduced matrices are: $\hat{E} = W^TEV,\ \hat{A} = W^TAV,$

Theorem ([Grimme '97])

For both SISO and MIMO systems, if the projection matrices V, W satisfy (12)(13), then $2p$ moments of $H(s)$ are matched by $\hat{H}(s)$, i.e. $M_i(s_0) = \hat{M}_i(s_0),\ i = 0,1,\ldots,2p-1.$

Implicit moment-matching(rational interpolation)

Instead of using a single expansion point, **multiple expansion points** are used in rational interpolation method, such that

$$\mathrm{range}(V) = \bigcup_{i=0}^{k} K_{p_i}((A - s_i E)^{-1} E, (A - s_i E)^{-1} b) \qquad (14)$$

$$\mathrm{range}(W) = \bigcup_{i=0}^{k} K_{p_i}((A - s_i E)^{-T} E^T, (A - s_i E)^{-T} c^T) \qquad (15)$$

Theorem ([Grimme '97])

For both SISO and MIMO systems, if the projection matrices V, W satisfy (14)(15), then $2p_j$ moments of $H(s)$ are matched by $\hat{H}(s)$ at each expansion point s_j, i.e. $M_i(s_j) = \hat{M}_i(s_j), i = 0,1,\ldots,2p_j - 1; j = 0,\ldots,k.$

Remark

For the moment-matching property of the rational interpolation method, it is not required that $W^T V = I$.

Implicit moment-matching(rational interpolation)

Computation of V, W in (14)(15)

- Rational Arnoldi method or rational Lanczos method in [Grimme '97]
- Repeated modified Gram-Schmidt algorithm (Repeated Arnoldi algorithm).

How to decide the expansion points?

- Some heuristic methods
- Using error estimation and a greedy algorithm.
- Locally optimal algorithm: IRKA.

Implicit moment-matching(rational interpolation)

How to decide the expansion points?

Using error estimation and a greedy algorithm

Error estimation, e.g. $\Delta(s)$:

$$\text{Residual} \| r \|_2 = \| B - (sE - A)\hat{x}(s) \|_2$$

$$\text{Error between } x \text{ and } \hat{x} : \| x(s) - \hat{x}(s) \|_2 \le \| r \|_2 / \sigma_{\min}(sE - A)$$

$$\text{Error between } y \text{ and } \hat{y} : \| y - \hat{y} \|_2 = \| r^{pr} \|_2 \| r^{du} \|_2 / \sigma_{\min}(sE - A)$$

$$\begin{aligned} \| x - \hat{x} \|_2 &= \| (sE - A)^{-1} B - \hat{x} \|_2 = \| (sE - A)^{-1} B - (sE - A)^{-1}(sE - A)\hat{x} \|_2 \\ &= \| (sE - A)^{-1}(B - (sE - A)\hat{x} \|_2 \\ &\le \| (sE - A)^{-1} \|_2 \| r \|_2 = \| r \| / \sigma_{\min}(sE - A)_2 \end{aligned}$$

$r^{pr} = r, r^{du} = -C^T - (sE^T - A^T)$ (Proof in [Feng, Benner, Antoulas ’14])

Implicit moment-matching(rational interpolation)

How to decide the expansion points?

A greedy algorithm: Selection of expansion points

```
Initial expansion point : $s_0 = \hat{s}; i = -1;$
$\Xi_{train}$ : a large set of samples of $s$
WHILE $\varepsilon > \varepsilon_{tol}$
$i = i + 1;$
$s_i = \hat{s};$
$\mathrm{range}(V_i) = K_p((s_iE - A)^{-1}E, (s_iE - A)^{-1}B); \mathrm{range}(W_i) = K_p((s_iE - A)^{-T}E^T, (s_iE - A)^{-T}C^T)$
$V = [V, V_i]; W = [W, W_i];$
$\hat{s} = \arg\max_{s \in \Xi_{train}} \Delta(s);$
$\varepsilon = \Delta(\hat{s});$
ENDWHILE
```

Implicit moment-matching(rational interpolation)

How to decide the expansion points?

- Locally optimal algorithm: IRKA for both SISO and MIMO system:

An Iterative Rational Krylov Algorithm (IRKA)

1. Make an initial selection of σ_i, for $i = 1, \ldots, r$, closed under conjugation, fix a tol. Choose initial directions $\tilde{B}_1, \ldots \tilde{B}_r, \tilde{C}_1, \ldots \tilde{C}_r$.

2. Choose V_r and W_r so that $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 E - A)^{-1} B\tilde{B}_1, \ldots, (\sigma_{\mathrm{r}} E - A)^{-1} B\tilde{B}_r\}$,

   $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 E - A^T)^{-1} C^T \tilde{C}_1, \ldots, (\sigma_r E - A)^{-1} C^T \tilde{C}_r\}$. and $W_r = (W_r^T V_r)^{-1} V_r$

3. WHILE ($\max\limits_{\mathrm{j}=1,\ldots,\mathrm{r}} \left\{ \dfrac{\sigma_j - \sigma_j^{old}}{\sigma_j} \right\} > \mathrm{tol}$)

   (a) $\hat{E} = W_r^T E V_r, \hat{A} = W_r^T A V_r$

   (b) Solve $(\lambda_i \hat{E} - \hat{A}) y_i = \lambda_i y_i, i = 1, \cdots, r$

   (c) Assign $\sigma_i \leftarrow -\lambda_i$ for $i = 1, \ldots, r; Y = (y_1, \ldots, y_r)$

   (d) $\tilde{B} = \hat{B}^T Y^{-T}, \tilde{C} = \hat{C} Y, \tilde{B} = (\tilde{B}_1, \ldots \tilde{B}_r), \tilde{C} = (\tilde{C}_1, \ldots, \tilde{C}_r)$

   (e) Update V_r and W_r so $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 E - A)^{-1} B\tilde{B}_1, \ldots, (\sigma_{\mathrm{r}} I - A)^{-1} B\tilde{B}_r\}$,

   $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 E - A^T)^{-1} C^T \tilde{C}_1, \ldots, (\sigma_r E - A)^{-1} C^T \tilde{C}_r\}$.

   (f) $W_r = (W_r^T V_r)^{-1} W_r$

4. $\hat{E} = W_r^T E V_r \hat{A} = W_r^T A V_r, \hat{B} = W_r^T B, \hat{C} = C V_r$

Implicit moment-matching(rational interpolation)

How to decide the expansion points?

- Locally optimal algorithm: IRKA for both SISO and MIMO system:

```
An Iterative Rational Krylov Algorithm (IRKA)
1. Make an initial selection of $\sigma_i$, for $i = 1,\ldots,r$, closed under conjugation, fix a
   tol. Choose initial directions $\tilde{B}_1,\ldots\tilde{B}_r, \tilde{C}_1,\ldots\tilde{C}_r$.
2. Choose $V_r$ and $W_r$ so that $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 E - A)^{-1} B\tilde{B}_1,\ldots,(\sigma_{\mathrm{r}} E - A)^{-1} B\tilde{B}_r\}$,
   $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 E - A^T)^{-1} C^T\tilde{C}_1,\ldots,(\sigma_r E - A)^{-1} C^T\tilde{C}_r\}$. and $W_r = (W_r^T V_r)^{-1} V_r$
3. WHILE $(\max_{\mathrm{j}=1,\ldots,\mathrm{r}}\left\{\frac{\sigma_j - \sigma_j^{old}}{\sigma_j}\right\} > \mathrm{tol})$
    (a) $\hat{E} = W_r^T E V_r, \hat{A} = W_r^T A V_r$
    (b) Solve $(\lambda_i \hat{E} - \hat{A}) y_i = \lambda_i y_i, (\lambda_i \hat{E}^T - \hat{A}^T)\tilde{y}_i = \lambda_i \tilde{y}_i i = 1,\cdots,r$
    (c) Assign $\sigma_i \leftarrow -\lambda_i$ for $i = 1,\ldots,r; Y = (y_1,\ldots,y_r)$
    (d) $\tilde{B} = \hat{B}^T\tilde{Y}, \tilde{C} = \hat{C}Y, \tilde{B} = (\tilde{B}_1,\ldots\tilde{B}_r), \tilde{C} = (\tilde{C}_1,\ldots,\tilde{C}_r)$
    (e) Update $V_r$ and $W_r$ so $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 E - A)^{-1} B\tilde{B}_1,\ldots,(\sigma_{\mathrm{r}} I - A)^{-1} B\tilde{B}_r\}$,
                    $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 E - A^T)^{-1} C^T\tilde{C}_1,\ldots,(\sigma_r E - A)^{-1} C^T\tilde{C}_r\}$.
    (f) $W_r = (W_r^T V_r)^{-1} W_r$
4. $\hat{E} = W_r^T E V_r \hat{A} = W_r^T A V_r, \hat{B} = W_r^T B, \hat{C} = C V_r$
```

Implicit moment-matching(rational interpolation)

In (d) $\tilde{B} = \hat{B}^T \tilde{Y}, \tilde{C} = \hat{C}Y, \tilde{B} = (\tilde{B}_1, \dots \tilde{B}_r), \tilde{C} = (\tilde{C}_1, \dots, \tilde{C}_r)$
if σ_i is complex, y_i in Y is the corresponding eigenvector, then $\bar{y}_i$ is in Y
and corresponds to $\overline{\sigma}_i$.

Residual σ_i are closed under conjugation, then W, V can be taken as *real* matrices.*why?*
For any complex variable σ_i, we have

$$(\sigma_i E - A)^{-1}\tilde{b} = \sum_{k=0}^{\infty} (A^{-1}E)^k (-A^{-1}\tilde{b})\sigma_i^k$$

$$(\sigma_i^* E - A)^{-1}\tilde{b}^* = \sum_{k=0}^{\infty} (A^{-1}E)^k (-A^{-1}\tilde{b}^*)(\sigma_i^*)^k$$

Since $(\sigma_i^*)^k = (\sigma_i^k)^*$, $(\sigma_i E - A)^{-1}\tilde{b}$ and $(\sigma_i^* E - A)^{-1}\tilde{b}^*$ have the same real and imaginary parts.
Therefore

$$\text{span}\{(\sigma_i E - A)^{-1}\tilde{b}, (\sigma_i^* E - A)^{-1}\tilde{b}^*\} = \text{span}\{\text{Re}[(\sigma_i E - A)^{-1}\tilde{b}], \text{Im}[(\sigma_i E - A)^{-1}\tilde{b}]\}.$$

So that

$$\text{span}\{(\sigma_1 E - A)^{-1}\tilde{b}, \dots, (\sigma_i E - A)^{-1}\tilde{b}, (\sigma_i^* E - A)^{-1}\tilde{b}^*, \dots, (\sigma_r E - A)^{-1}\tilde{b}\} =$$
$$\text{span}\{(\sigma_1 E - A)^{-1}\tilde{b}, \dots, \text{Re}[(\sigma_i E - A)^{-1}\tilde{b}], \text{Im}[(\sigma_i E - A)^{-1}\tilde{b}], \dots, (\sigma_r E - A)^{-1}\tilde{b}\}$$

Implicit moment-matching(rational interpolation)

- Locally optimal algorithm: IRKA for SISO system

Upon convergence, Algorithm IRKA leads to:

$$\hat{H}(-\hat{\lambda}_i) = H(-\hat{\lambda}_i) \text{ and } \hat{H}'(-\hat{\lambda}_i) = H'(-\hat{\lambda}_i) \text{ for } i = 1, \ldots, r.$$

Theorem 3.4. [Gugercin et al '08] Given a stable SISO system $H(s) = c(sI - A)^{-1}b$, let $\hat{H}(s) = \hat{c}(sI - \hat{A})^{-1}\hat{b}$ be a local minimizer of dimension r for the optimal H_2 model reduction problem

$$\| H - \hat{H} \|_{H_2} = \min_{\substack{\dim(\tilde{H})=r \\ \tilde{H}:\text{stable}}} \| H - \tilde{H} \|_{H_2}$$

and suppose that $\hat{H}(s)$ has simple poles at $\hat{\lambda}_i, i = 1, \ldots, r$. Then $\hat{H}(s)$ interpolates both $H(s)$ and its first derivative at $\hat{\lambda}_i, i = 1, \ldots, r$:

$$\hat{H}(\hat{\lambda}_i) = H(\hat{\lambda}_i) \text{ and } \hat{H}'(\hat{\lambda}_i) = H'(\hat{\lambda}_i) \text{ for } i = 1, \ldots, r.$$

Therefore IRKA obtains a reduced model satisfies the local optimal necessary conditions in Theorem 3.4 in [Gugercin et al. '08].

Implicit moment-matching(rational interpolation)

References

Pade- and Pade-type approximation;

[**Freund ‘00**] R. W. Freund, Krylov-subspace methods for reduced-order modeling in circuit simulation, *Journal of Computational and Applied Mathematics*, 123: 395-421, 2000.

[**Freund ‘03**] R. W. Freund, Model reduction methods based on Krylov subspaces, *Acta Numerica*, 12: 267-319, 2003.

[**Feldmann, Freund ‘95**] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Pade approximation via the Lanczos process. *IEEE Transactions on Computer-aided design of Integrated Circuits and Systems*. 14(5), 1995.

[Odabasioglu et al ‘97] Odabasioglu, M. Celik, L. T. Pileggi, PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE/ACM ICCAD*, 58-65, 1997.

Rational interpolation:

[**Grimme ‘97**] E. J. Grimme, Krylov projection methods for model reduction, *PhD thesis*, University of Illinois at Urbana-Champaign, 1994.

IRKA:

[**Gugercin et al ‘08]** S. Gugercin, A. C. Antoulas, C. Beattie, H_2 Model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Analysis,* 30(2): 609-638, 2008.