



MAX PLANCK INSTITUTE  
FOR DYNAMICS OF COMPLEX  
TECHNICAL SYSTEMS  
MAGDEBURG



COMPUTATIONAL METHODS IN  
SYSTEMS AND CONTROL THEORY

Otto-von-Guericke Universität Magdeburg  
Faculty of Mathematics  
Summer term 2019

# Model Reduction for Dynamical Systems Lecture 7

**Lihong Feng**

Max Planck Institute for Dynamics of Complex Technical Systems  
Computational Methods in Systems and Control Theory  
Magdeburg, Germany

[feng@mpi-magdeburg.mpg.de](mailto:feng@mpi-magdeburg.mpg.de)

[https://www.mpi-magdeburg.mpg.de/3668354/mor\\_ss19](https://www.mpi-magdeburg.mpg.de/3668354/mor_ss19)



# Outline

- Method based on Pade approximation, explicit moment-matching (AWE)
- Method based on Pade, Pade-type approximation, implicit moment-matching
- Method based on rational interpolation



# 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  $\hat{H}(s)$ , such that  $H(s) \approx \hat{H}(s)$ , but  $\hat{H}(s)$  can be computed fastly?***



# Padé approximation

- Padé approximation:

Approximates a (analytic) function  $f(x)$  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:

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

The transfer function can be approximated by Padé approximation!



# Padé approximation

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

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

$$Q_M(x) = 1 + q_1x + q_2x^2 + \cdots + q_Mx^M.$$

- 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: 
$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)} \quad \begin{aligned} P_N(x) &= p_0 + p_1x + p_2x^2 + \cdots + p_Nx^N \\ Q_M(x) &= 1 + q_1x + q_2x^2 + \cdots + q_Mx^M. \end{aligned}$$

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:  $f(x) = f_0 + f_1x + f_2x^2 + \cdots + f_kx^k + \cdots,$

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

This implicates:

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



# How to compute Padé approximation

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

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

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



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



# How to compute Padé approximation

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

M unknowns and M equations in (2),  $q_i$ s can be obtained by solving (2),  
 $p_i$ s can be immediately obtained from (1) without solving equations.





# How to compute Padé approximation

An example:

$$f(x) = \sqrt{x+1}, 0 \leq x \leq 1 \quad f_0 = f(0) = 1; \quad f_1 = f'(0) = 1/2;$$

$$f_2 = f''(0)/2! = -1/16; \quad f_3 = f^{(3)}(0)/3! = 3/48.$$

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

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

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

$$1) \quad q_2 f_0 + q_1 f_1 + f_2 = 0$$

$$q_2 f_1 + q_1 f_2 + f_3 = 0$$

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

$$q_1 f_0 + f_1 - p_1 = 0 \Rightarrow$$

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

$$\begin{pmatrix} f_1 & f_0 \\ f_2 & f_1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} f_2 \\ f_3 \end{pmatrix} \Rightarrow \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0.25 \end{pmatrix}$$

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

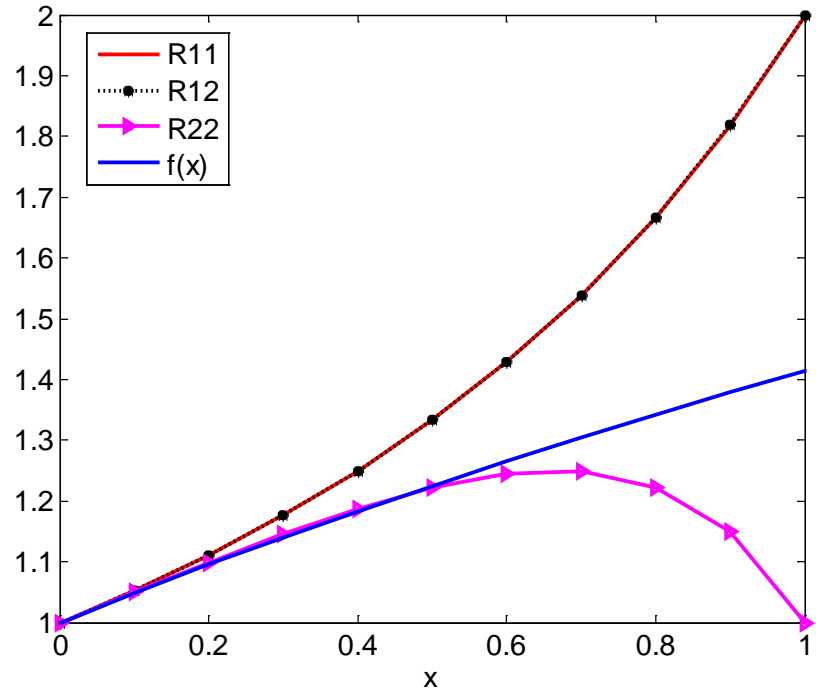
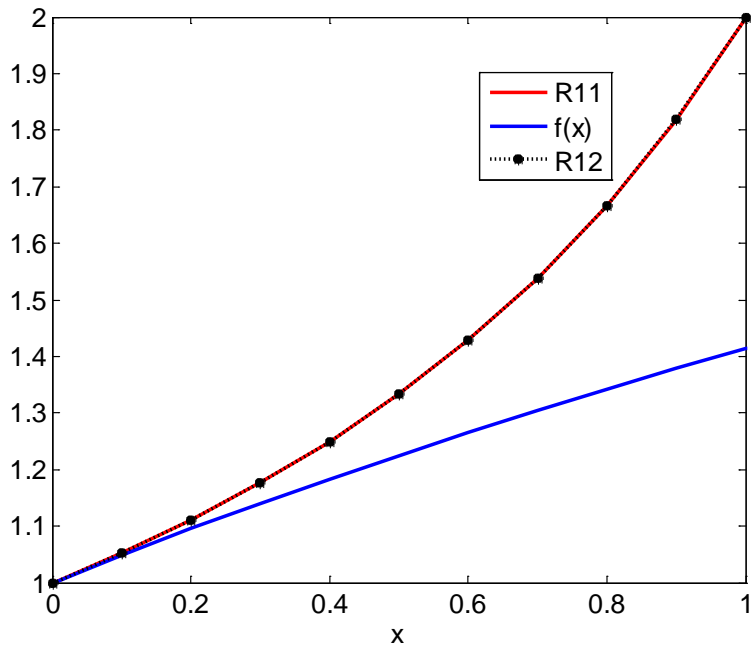
$$2) \quad f_0 - p_0 = 0 \Rightarrow p_0 = 1$$

$$q_1 f_0 + f_1 - p_1 = 0 \Rightarrow p_1 = -1 + 1/2 = -0.5$$

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

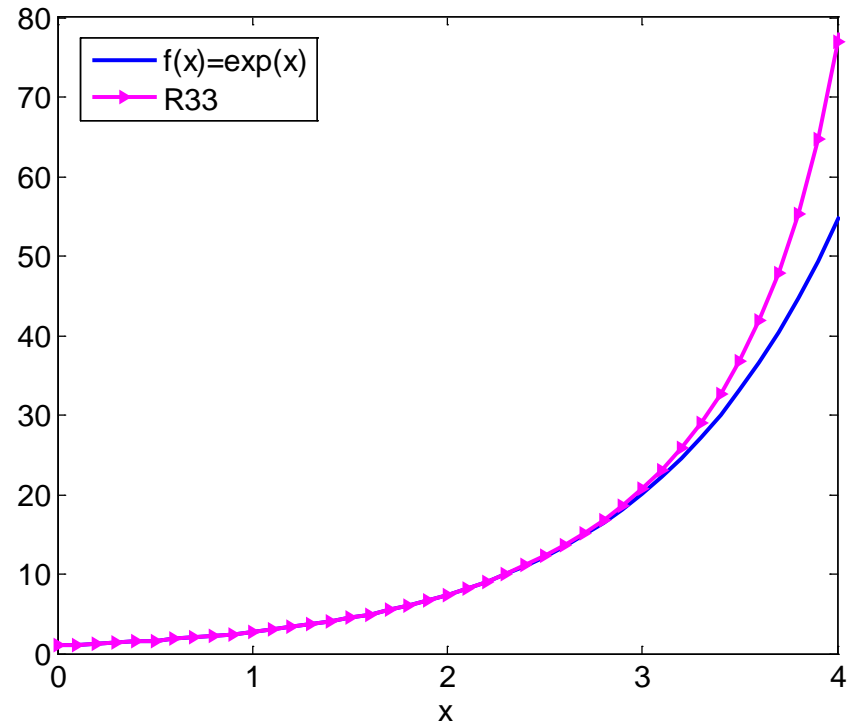
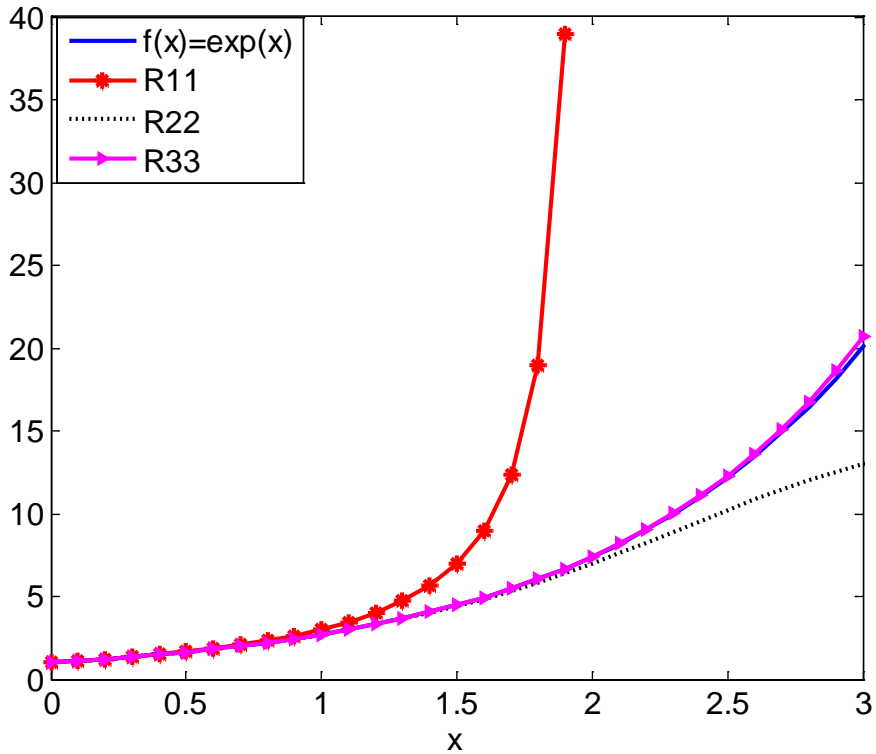


# How to compute Padé approximation





# How to compute Padé approximation



Padé approximation is only accurate around  $x=0$



# MOR: AWE based on Padé approximation

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

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

- 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  $P_N(x)$  is one degree lower than  $Q_M(x)$ , i.e.:  **$N=M-1$** .

\*from the book: John H. Mathews and Kurtis K. Fink, Numerical Methods Using Matlab, 4<sup>th</sup> 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  $H(s) = c(sE - A)^{-1}b = c(-sA^{-1}E + I)^{-1}(-A^{-1}b)$

$$\tilde{A} := A^{-1}E, \quad \tilde{A} = Z\Lambda Z^{-1}, \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$$

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

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



# MOR: AWE based on Padé approximation

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

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

$$\hat{H}(s) = \frac{P_{r-1}(s)}{Q_r(s)} = \frac{p_0 + p_1s + \cdots + p_{r-1}s^{r-1}}{1 + q_1s + \cdots + q_rs^r}$$

Computing the coefficients:

Padé 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  $H(s)$ :

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

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

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

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

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



# MOR: AWE based on Padé approximation

As has been introduced before,  $q_i s$  can be obtained by solving :

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

$p_i s$  can be immediately obtained from:

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

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

Ok ?





# MOR: AWE based on Padé approximation

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

$$\hat{H}(s) = \frac{P_{r-1}(s)}{Q_r(s)} = \frac{p_0 + p_1s + \dots + p_{r-1}s^{r-1}}{1 + q_1s + \dots + q_rs^r}$$

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

$$\hat{H}(s) = \frac{p_0 + p_1s + \dots + p_{r-1}s^{r-1}}{1 + q_1s + \dots + q_rs^r} = \frac{\hat{k}_1}{s - \hat{a}_1} + \frac{\hat{k}_2}{s - \hat{a}_2} + \dots + \frac{\hat{k}_r}{s - \hat{a}_r}$$

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

How to compute the poles and the residues?

$\hat{a}_1, \hat{a}_2, \dots, \hat{a}_r$  are nothing but the roots of  $1 + q_1s + \dots + q_rs^r$

and we have known how to compute  $q_r s^r$  !



# MOR: AWE based on Padé approximation

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

From Pade approximation:

$$\underbrace{\hat{k}_1(s - \hat{a}_1)^{-1} + \hat{k}_2(s - \hat{a}_2)^{-1} + \dots + \hat{k}_r(s - \hat{a}_r)^{-1}}_{\downarrow} - [m_0 + m_1s + \dots + m_{2r-1}s^{2r-1} + m_{2r}s^{2r} + \dots] = e_{2r}s^{2r} + e_{2r+1}s^{2r+1} + \dots$$

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

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



# MOR: AWE based on Padé approximation

$$\begin{aligned} \frac{\hat{k}_1}{\hat{a}_1} + \frac{\hat{k}_2}{\hat{a}_2} + \dots + \frac{\hat{k}_r}{\hat{a}_r} &= -m_0 \\ \frac{\hat{k}_1}{\hat{a}_1^2} + \frac{\hat{k}_2}{\hat{a}_1^2} + \dots + \frac{\hat{k}_r}{\hat{a}_1^2} &= -m_1 \\ &\vdots \\ \frac{\hat{k}_1}{\hat{a}_1^r} + \frac{\hat{k}_2}{\hat{a}_2^r} + \dots + \frac{\hat{k}_r}{\hat{a}_r^r} &= -m_{r-1} \end{aligned} \Rightarrow \begin{pmatrix} \hat{a}_1^{-1} & \hat{a}_2^{-1} & \dots & \hat{a}_r^{-1} \\ \hat{a}_1^{-2} & \hat{a}_2^{-2} & \dots & \hat{a}_r^{-2} \\ & & \vdots & \\ \hat{a}_1^{-r} & \hat{a}_2^{-r} & \dots & \hat{a}_r^{-r} \end{pmatrix} \begin{pmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_r \end{pmatrix} = \begin{pmatrix} -m_0 \\ -m_1 \\ \vdots \\ -m_{r-1} \end{pmatrix} \quad (5)$$

the residues can be obtained by solving the above equations!

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

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



# 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 in truncated Taylor expansion is avoided .
  - Poles and residues of  $H(s)$  can be computed more easily by Padé approximation.
  - $H(s)$  itself is a rational function.



# MOR: AWE based on Padé approximation

Implementation of AWE:

1. Solve (3) to get  $q, s$ .
2. Compute the roots of the polynomial:  $1 + q_1 s + \dots + q_r s^r$

the roots are:  $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_r$  (poles)

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

4. Form the reduced (simpler) transfer function:

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

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  $q,s$ .
2. Get  $p,s$  from (4).
3. Use `*residue(p,q)*` in matlab to get the poles and residues:

$$\hat{a}_1, \hat{a}_2, \dots, \hat{a}_r; \quad \hat{k}_1, \hat{k}_2, \dots, \hat{k}_r$$

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

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



# 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:

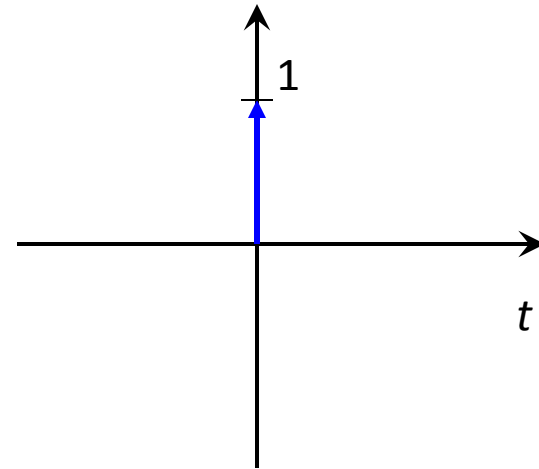
$$H(s) = Y(s)/U(s)$$

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:

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

We obtain:

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

Impulse output  
response  
in time domain

Because:

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





# Numerical instability of AWE

1. Solve (3) to get  $q,s$ .

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



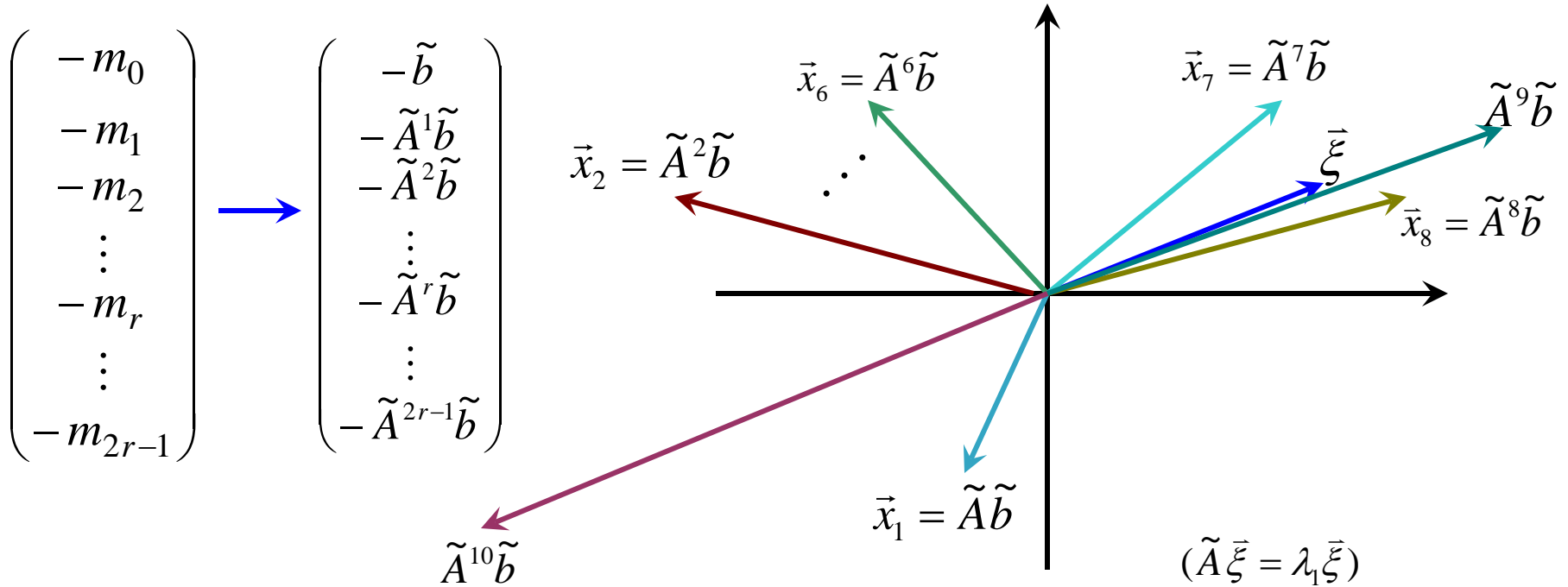
$$\begin{pmatrix} m_0 & m_1 & \cdots & m_{r-1} \\ m_1 & m_2 & \cdots & m_r \\ & & \vdots & \\ m_{r-1} & m_r & \cdots & m_{2r-2} \end{pmatrix} \begin{pmatrix} q_r \\ q_{r-1} \\ \vdots \\ q_1 \end{pmatrix} = \begin{pmatrix} -m_r \\ -m_{r+1} \\ \vdots \\ -m_{2r-1} \end{pmatrix}$$

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

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



# Numerical instability of AWE



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

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





# Numerical instability of AWE

$\tilde{b} = \alpha_1 \xi_1 + \alpha_2 \xi_2 + \dots + \alpha_n \xi_n$ ,  $\xi_i$  are eigenvectors of  $\tilde{A}$ .

$$\begin{aligned} \tilde{A}^m \tilde{b} &= \alpha_1 \tilde{A}^m \xi_1 + \alpha_2 \tilde{A}^m \xi_2 + \dots + \alpha_n \tilde{A}^m \xi_n \\ &= \alpha_1 \lambda_1^m \xi_1 + \alpha_2 \lambda_2^m \xi_2 + \dots + \alpha_n \lambda_n^m \xi_n \\ &= \alpha_1 \lambda_1^m \left( \xi_1 + \frac{\alpha_2}{\alpha_1} \left( \frac{\lambda_2}{\lambda_1} \right)^m \xi_2 + \dots + \frac{\alpha_n}{\alpha_1} \left( \frac{\lambda_n}{\lambda_1} \right)^m \xi_n \right) \end{aligned}$$

$m \rightarrow \infty$  ( $\lambda_1 > \lambda_2 > \dots > \lambda_n$ )

$$\tilde{A}^m \tilde{b} \rightarrow (\alpha_1 \lambda_1^m) \xi_1 =: \sigma \xi_1$$



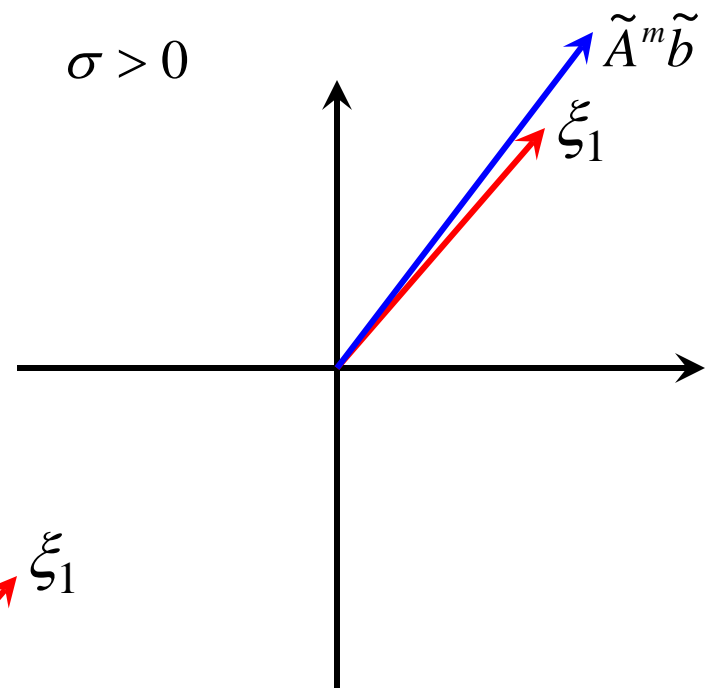
If  $\lambda_1$  and  $\lambda_2$  are not close, round - off error slowly changes  $\tilde{A}^m \tilde{b}$  to  $\sigma \xi_1$



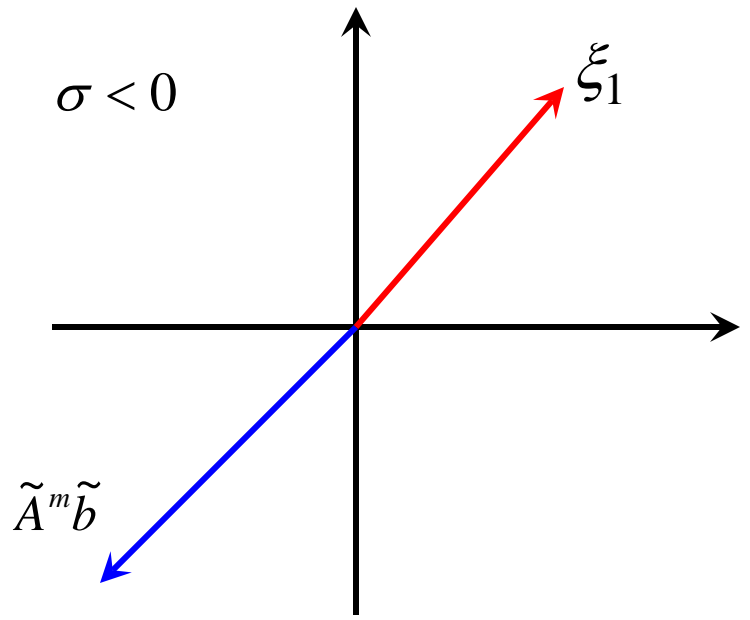
# Numerical instability of AWE

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

$$\sigma > 0$$



$$\sigma < 0$$





# Numerical instability of AWE

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

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

Therefore  $m_i, i > 8$  cannot be computed accurately due to round - off errors, such that the computed poles and residues are inaccurate.

## 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

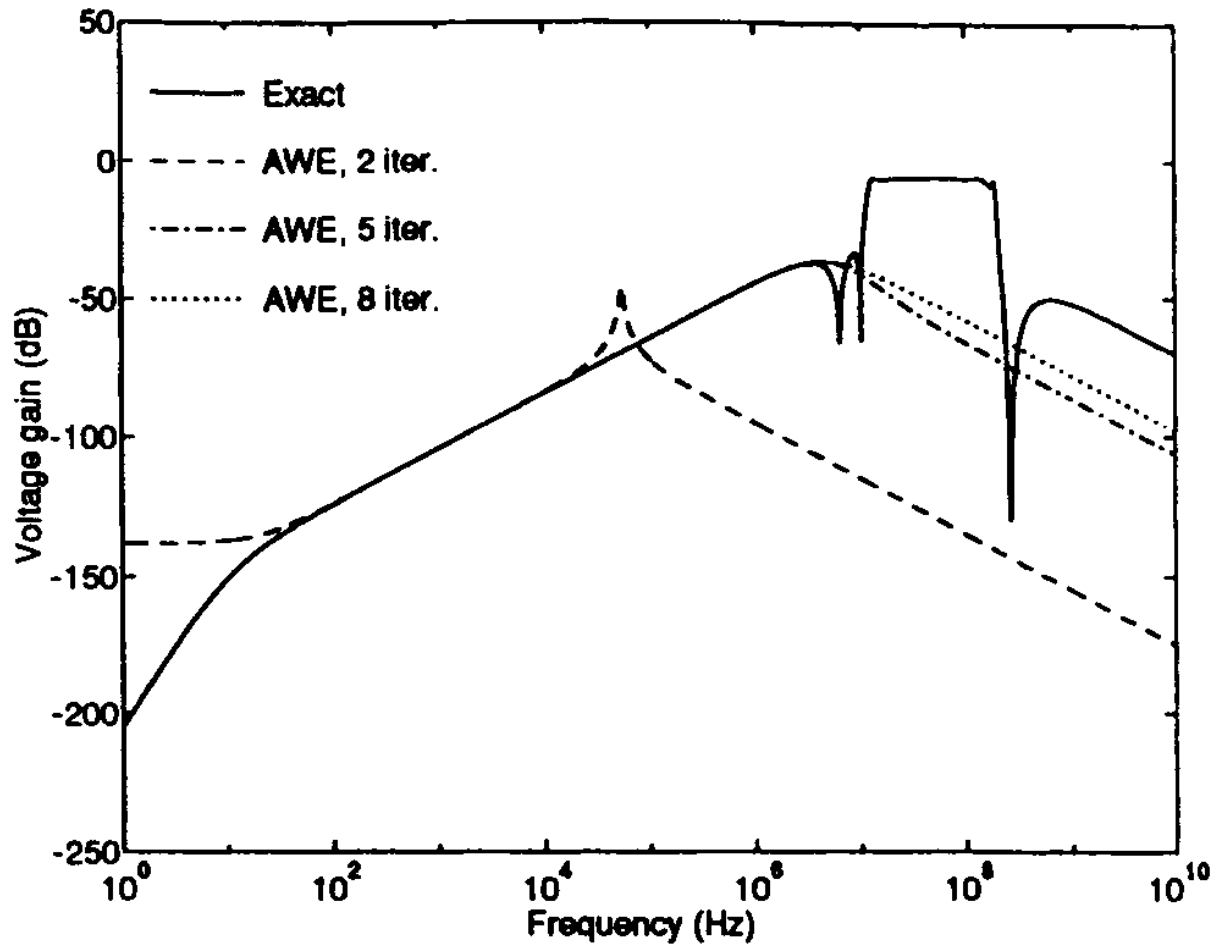


Fig. 1. Results for simulation of voltage gain with AWE.



# Outline

- Method based on Pade, Pade-type approximation, implicit moment-matching
- Method based on rational interpolation



## Angle between two vectors:

Dot product of two vectors  $u$ ,  $v$  is defined as:  $u \cdot v = \cos(\theta) \|u\|_2 \|v\|_2$

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

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





# Preliminaries

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

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

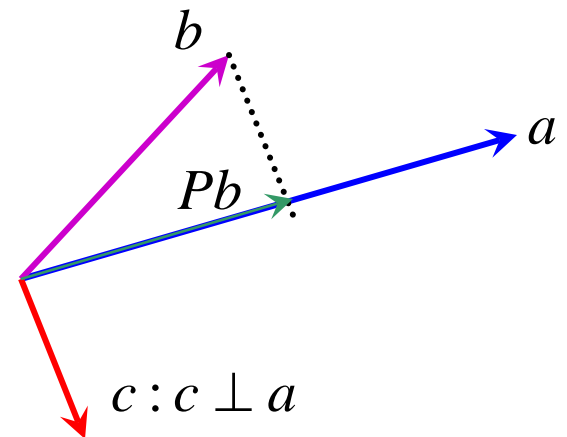
How to compute  $c$ ?

$$Pb = ma \quad (m \text{ is a scalar})$$

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

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

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



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

An important information:

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



# Preliminaries

- Orthogonalization of a vector  $b$  to the subspace spanned by a group of orthogonal vectors  $a_1, a_2, \dots, a_l$

$$c = b - Pb$$

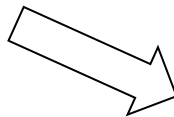
$$Pb = m_1 a_1 + m_2 a_2 + \dots + m_l a_l \quad \text{and} \quad c \perp a_i$$

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

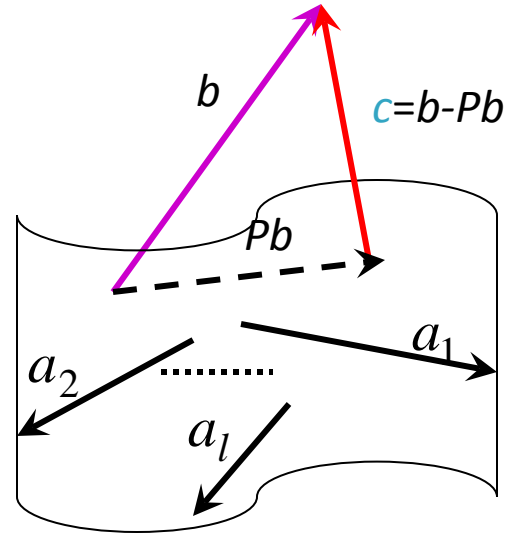
$$a_i^T a_j = 0$$

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

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



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

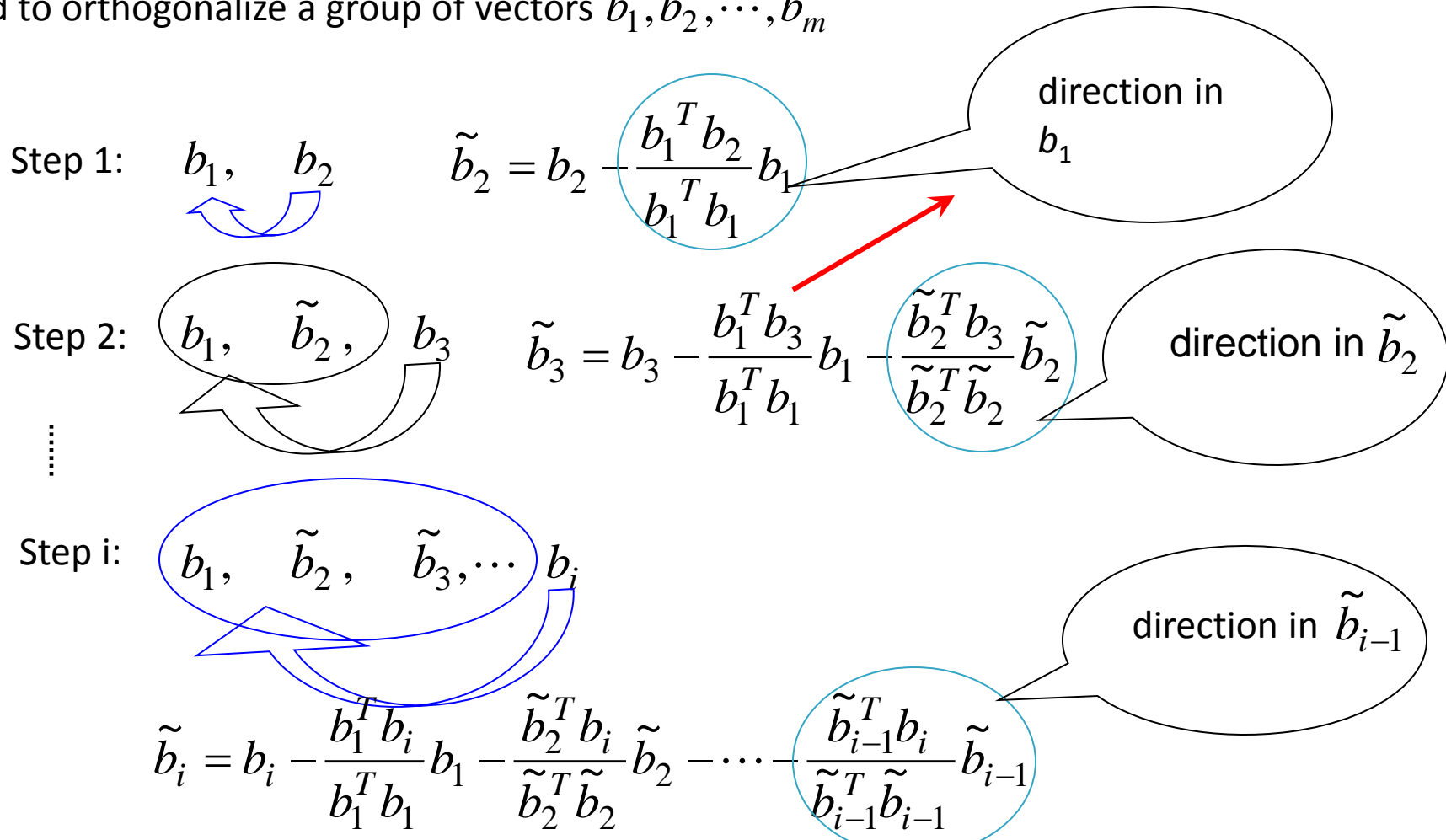




# Preliminaries

- Gram-Schmidt process:

used to orthogonalize a group of vectors  $b_1, b_2, \dots, b_m$





# Preliminaries

Gram-Schmidt process:

*for*  $i=2,3, \dots, m$

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

*end*

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

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



# Preliminaries

Gram-Schmidt process:

for  $i=2,3, \dots, m$

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

end

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

Computation with computers is finite arithmetic!

Modified Gram-Schmidt process:

for  $i=2,3, \dots, m$

$$\tilde{b}_i = b_i$$

for  $j=1,2, \dots, i-1$

$$\tilde{b}_i = \tilde{b}_i - \frac{\tilde{b}_j^T \tilde{b}_i}{\tilde{b}_j^T \tilde{b}_j} \tilde{b}_j$$

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  $i=2,3, \dots, m$

for  $j=1,2, \dots, i-1$

$$b_i = b_i - \frac{b_j^T b_i}{b_j^T b_j} b_j$$

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  $i=2,3, \dots, m$

$$b_1 = b_1 / \| b_1 \|$$

for  $j=1,2, \dots, i-1$

$$b_i = b_i - \frac{b_j^T b_i}{b_j^T b_j} b_j$$

end

$$\varepsilon_b = \| b_i \|$$

if  $\varepsilon_b \geq tol$   
 $b_i = b_i / \varepsilon_b$   
else  
delete  $b_i$

} deflation

end

end



- Arnoldi algorithm:

It computes an orthonormal basis  $v_1, v_2, \dots, v_q$  of the Krylov subspace:

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

i.e.  $\text{span}\{v_1, v_2, \dots, v_q\} = K_p(A, r)$

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





# Preliminaries

Arnoldi algorithm

$$v_1 = r / \| r \|$$

for  $i=2,3, \dots, p$

$$w = Av_{i-1}$$

for  $j=1,2, \dots, i-1$

$$w = w - \frac{v_j^T w}{v_j^T v_j} v_j$$

end

$$\varepsilon_w = \| w \|$$

If  $\varepsilon_w \geq tol$

$$v_i = w / \varepsilon_w$$

else

stop

end

end

It is clear:

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

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

Why?



# Preliminaries

Arnoldi algorithm

$$v_1 = r / \| r \|_2$$

for  $i=2,3, \dots, p$

$$v_i = Av_{i-1}$$

for  $j=1,2, \dots, i-1$

$$h_{j,i-1} = v_j^T v_i$$

$$v_i = v_i - (h_{j,i-1})v_j \quad (v_j^T v_j = 1)$$

end

$$h_{i,i-1} = \| v_i \|_2$$

If  $h_{i,i-1} \geq tol$

$$v_i = v_i / h_{i,i-1}$$

else

stop

end

end

output : Hessenberg matrix  $\tilde{H}_i \in R^{i+1,i}$ , and  $v_1, \dots, v_i, i < p$

$$AV_i = V_{i+1} \tilde{H}_i$$



## Generalized minimal residual method (GMRES)

GMRES is an iterative method of solving a nonsymmetric system of linear equations. The method approximates the solution by the vector in a Krylov subspace with minimal residual. The **Arnoldi algorithm** is used to find this vector.

$$Ax = b$$

initial guess  $x_0$

initial residual  $r_0 = b - Ax_0$

$$K_p(A, r_0) = \{r_0, Ar_0, A^2r_0, \dots, A^{p-1}r_0\}, r_0 = b - Ax_0 \text{ (} r_0 = b, \text{ when } x_0 = 0)$$

Find  $x_i = x_0 + \tilde{x}_i$ ,  $\tilde{x}_i \in K_i$  such that  $x_i$  minimizes the 2-norm of the residual:

$$r = b - Ay, x_i = \arg \min_y \|r\|_2$$

The first vector to be orthonormalized is  $r_0 = b - Ax_0$ , i.e.  $v_1 = r_0 / \|r_0\|_2$



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

Recall that for Petrov-Galerkin projection based MOR:

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

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

Taylor expansion at  $s_0$ :

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

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 ),  $i = 0, \dots$ , are called the moments of the transfer function.



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

Recall that in AWE method, the moments

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

are computed explicitly. It is numerically unstable.

Observe that  $M_t = c\tilde{A}^t(s_0)\tilde{b}(s_0), t = 0, \dots$ , can be divided either into :

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

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

or into

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

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

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

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

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



# Implicit moment-matching (Pade approximation)

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

Recall that

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

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

$W, V$  span two Krylov subspaces, so that they can be simultaneously computed by (Band) Lanczos algorithm, such that  $\tilde{W}^T V = \text{diag}(d_1, \dots, d_r) =: \Delta$ .

The outputs of Band Lanczos algorithm are [Freund'03]

$\tilde{W}, V$

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

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

$$\eta = \Delta^{-T} V^T c^T$$



# Implicit moment-matching (Pade approximation)

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

$$\begin{aligned} (A - s_0 E)^{-1} E \frac{dx}{dt} &= (A - s_0 E)^{-1} Ax(t) + (A - s_0 E)^{-1} bu(t), \\ y &= cx(t), \end{aligned} \tag{10}$$

One gets the reduced order model (ROM)

$$\begin{aligned} \tilde{W}^T (A - s_0 E)^{-1} EV \frac{dz}{dt} &= \tilde{W}^T (A - s_0 E)^{-1} AVz(t) + \tilde{W}^T (A - s_0 E)^{-1} bu(t), \\ y &= cVz(t), \end{aligned} \tag{11}$$

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



# Implicit moment-matching (Pade approximation)

Actually the ROM in (11) can be **implicitly** derived using the outputs of the Band Lanczos algorithm<sub>[Freund '03]</sub>: (It can be easily proved that (12) is exactly (11))

$$\begin{aligned} T \frac{dz}{dt} &= (I + s_0 T)z(t) + \rho u(t), \\ y &= \eta^T \Delta z(t), \end{aligned} \tag{12}$$

**Theorem 1** <sub>[Feldmann, Freund '95]</sub>

If the columns of  $\tilde{W}$ ,  $V \in \mathbb{R}^r$  are the basis of the subspace in (8)(9), and satisfy  $\tilde{W}^T V = \text{diag}(d_1, \dots, d_r)$ , 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.

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

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.





## Number of moments matched for MIMO systems

In [Freund '00], it is shown that if  $\tilde{W}$ ,  $V$  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  $\lfloor r/n_o \rfloor + \lfloor r/n_i \rfloor$  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. It is in agreement with **Theorem 1**.



# Implicit moment-matching (Pade-type approximation)

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

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

$V$  can be computed by Arnoldi algorithm.

The resulting reduced model is passive, if  $E^T + E$  is positive semi-definite, and  $A^T + A$  is negative semi-definite [Odabasioglu, et.al '97].

Many models from circuit simulation meet this condition.

**Theorem 2** [Odabasioglu, et.al '97, Freund '03]

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

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

$\hat{H}(s)$  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**.

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

For moment - matching with Petrov - Galerkin projection, why using  $x \approx Vz$ , rather than  $x \approx Wz$ ?

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

$$x(s) = (sE - A)^{-1} b u(s) \Rightarrow x(s) = \sum_{i=0}^{\infty} \tilde{A}^i(s_0) \tilde{b}(s_0) (s - s_0)^i u(s) = \sum_{i=0}^{\infty} g_i(s) \tilde{A}^i(s_0) \tilde{b}(s_0)$$

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

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



$$x \approx Vz$$



# 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 wide, then a nonzero expansion point  $s_0$  is preferred.



# Implicit moment-matching<sub>(rational interpolation)</sub>

- 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 as **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

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

Applying a preconditioner to each of the linear systems,

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



# Implicit moment-matching<sub>(rational interpolation)</sub>

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

CG, GMRES, MINRES. etc..) [Grimme '97]:

$$x_b \approx x_b^q \in x_b^0 + K_q((s_0 E - A)^{-1}(sE - A), (s_0 E - A)^{-1}b)$$

$$x_c \approx x_c^q \in x_c^0 + K_q((s_0 E - A)^{-T}(sE - A)^T, (s_0 E - A)^{-T}c^T)$$

**Lemma 2.2** [Grimme '97] Krylov subspace is shift-invariant

For any matrix  $G$ , vector  $g$  and nonzero  $\eta$ ,

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

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

$$K_q((s_0 E - A)^{-1}(sE - A), (s_0 E - A)^{-1}b) = K_q((s_0 E - A)^{-1}E, (s_0 E - A)^{-1}b)$$

$$K_q((s_0 E - A)^{-T}(sE - A)^T, (s_0 E - A)^{-T}c^T) = K_q((s_0 E - A)^{-T}E^T, (s_0 E - A)^{-T}c^T)$$



# Implicit moment-matching<sub>(rational interpolation)</sub>

Compute  $V$ , such that

$$\text{range}(V) = K_p((A - s_0 E)^{-1} E, (A - s_0 E)^{-1} b) \quad (12)$$

Compute  $W$ , such that

$$\text{range}(W) = K_p((A - s_0 E)^{-T} E^T, (A - s_0 E)^{-T} c^T) \quad (13)$$

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

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

Therefore the reduced matrices are:  $\hat{E} = W^T EV$ ,  $\hat{A} = W^T AV$ ,

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



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

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

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

## 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.

$$M_i(s_j) = \hat{M}_i(s_j), i = 0, 1, \dots, 2p_j - 1; j = 0, \dots, k.$$

## Remark

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





# Implicit moment-matching<sub>(rational interpolation)</sub>

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

- Rational Arnoldi method or rational Lanczos method. [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.



## How to decide the expansion points?

Using error estimation and a greedy algorithm

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

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

$$\text{State vector error : error between } x \text{ and } \hat{x} : \| x(s) - \hat{x}(s) \|_2 \leq \| r \|_2 / \sigma_{\min}(sE - A)$$

$$\text{Output error : error between } y \text{ and } \hat{y} : \| y - \hat{y} \|_2 \leq \| r^{pr} \|_2 \| r^{du} \|_2 / \sigma_{\min}(sE - A) + | \hat{x}_{du} r_{pr} |$$

[Feng, Antoulas, Benner '17]

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



## How to decide the expansion points?

A greedy algorithm: Selection of expansion points

Initial expansion point :  $s_0 = \hat{s}; i = -1; V = []; W = [];$

$\Xi_{train}$  : a set of samples of  $s$  covering the range of interest

WHILE  $\varepsilon > \varepsilon_{tol}$

$i = i + 1;$

$s_i = \hat{s};$

$\text{range}(V_i) = K_p((s_i E - A)^{-1} E, (s_i E - A)^{-1} B); \text{range}(W_i) = K_p((s_i E - A)^{-T} E^T, (s_i E - A)^{-T} C^T)$

$V = \text{orth}[V, V_i]; W = \text{orth}[W, W_i];$

$\hat{s} = \arg \max_{s \in \Xi_{train}} \Delta(s);$

$\varepsilon = \Delta(\hat{s});$

ENDWHILE

Arnoldi algorithm  
is used.

orth: M-G-S



# Implicit moment-matching<sub>(rational interpolation)</sub>

## How to decide the expansion points?

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  $\hat{\lambda}_i, i = 1, \dots, r$ . Then  $\hat{H}(s)$  interpolates both  $H(s)$  and its first derivative at  $\hat{\lambda}_i, i = 1, \dots, r$ :

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



## 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  $i = 1, \dots, r$ , closed underconjugation for a fixed *tol*.

Choose initial directions  $\tilde{b}_1, \dots, \tilde{b}_r, \tilde{c}_1, \dots, \tilde{c}_r$ .

2. Choose  $V_r$  and  $W_r$  so that  $\text{Ran}(V_r) = \text{span}\{(\sigma_1 E - A)^{-1} B \tilde{b}_1, \dots, (\sigma_r E - A)^{-1} B \tilde{b}_r\}$ ,

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

3. WHILE  $\left( \max_{j=1, \dots, r} \left\{ \frac{\sigma_j - \sigma_j^{old}}{\sigma_j} \right\} > \text{tol} \right)$

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

(b) Solve  $(\hat{\lambda}_i \hat{E} - \hat{A}) y_i = 0, i = 1, \dots, r$

(c) Let  $\sigma_i = -\hat{\lambda}_i$  for  $i = 1, \dots, r$ ;  $Y = (y_1, \dots, y_r)$

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

(e) Update  $V_r$  and  $W_r$  so  $\text{Ran}(V_r) = \text{span}\{(\sigma_1 E - A)^{-1} B \tilde{b}_1, \dots, (\sigma_r E - A)^{-1} B \tilde{b}_r\}$ ,

$\text{Ran}(W_r) = \text{span}\{(\sigma_1 E - A)^{-T} C^T \tilde{c}_1, \dots, (\sigma_r E - A)^{-T} C^T \tilde{c}_r\}$ .

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

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<sub>(rational interpolation)</sub>

If  $\sigma_i$  are closed under conjugation, then  $W, V$  can be computed as *real* matrices. *why?*

For any complex variable  $\sigma_i$ , we have

$$(\overline{\sigma_i} E - A)^{-1} \tilde{b} = \overline{(\sigma_i E - A)^{-1} \tilde{b}} = \overline{(\sigma_i E - A)^{-1}} \tilde{b}$$

since for any complex matrix  $M : I = \overline{M^{-1} M} = \overline{M^{-1}} \cdot \overline{M} \Rightarrow (\overline{M})^{-1} = \overline{M^{-1}}$ .

Therefore

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

So that

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



# Implicit moment-matching<sub>(rational interpolation)</sub>

- Locally optimal algorithm: IRKA for SISO system

Upon convergence, Algorithm IRKA leads to:

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

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



# Implicit moment-matching<sub>(rational interpolation)</sub>

## References

Pade- and Pade-type approximation;

[**Pillage, Rohrer' 90**] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.

[**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<sub>2</sub> Model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Analysis*, 30(2): 609-638, 2008.

Error estimation:

[**Feng, Benner, Antoulas '14**] L. Feng, A. C. Antoulas, and P. Benner, *Some a Posteriori Error Bounds for Reduced Order Modelling of (Non-)Parametrized Linear Systems*, Preprint MPIMD/15-17, Max Planck Institute Magdeburg, Germany, 2015.