Robust Algorithm for Parametric Model Order Reduction

Lihong Feng   Peter Benner

Mathematics in Industry and Technology
Faculty of Mathematics, Chemnitz University of Technology, Germany

{lihong.feng;benner}@mathematik.tu-chemnitz.de

Symposium on Recent advances in MOR
♦ Parametric systems and parametric MOR

♦ Review of Daniel’s method

♦ A robust implementation of Daniel’s method

♦ Simulation results

♦ Conclusions
Parametric systems and Parametric MOR

Model Order Reduction progresses:

non-parametric linear system:

\[
\begin{align*}
C \frac{dx(t)}{dt} &= Gx(t) + Bu(t), \\
y(t) &= L^T x(t), \quad \Rightarrow
\end{align*}
\]

non-parametric nonlinear system:

\[
\begin{align*}
\frac{dg(x(t))}{dt} &= f(x(t)) + Bu(t), \\
y(t) &= L^T x(t), \quad \Rightarrow
\end{align*}
\]

parametric system:

\[
\begin{align*}
\frac{dg(x(t), s_1, s_2, \cdots, s_{\mu-1})}{dt} &= f(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\end{align*}
\]

- \(x\) is the state variable
- \(u(t)\) is the input, \(y(t)\) is the output.

Parametric system can describe the physical system in more detail, but is more complicated than the non-parametric system.
Model Order Reduction progresses:

non-parametric linear system:

\[
\begin{align*}
C \frac{dx(t)}{dt} & = Gx(t) + Bu(t), \\
y(t) & = L^T x(t), \\
\end{align*}
\]

non-parametric nonlinear system:

\[
\begin{align*}
\frac{dg(x(t))}{dt} & = f(x(t)) + Bu(t), \\
y(t) & = L^T x(t), \\
\end{align*}
\]

parametric system:

\[
\begin{align*}
\frac{dg(x(t), s_1, s_2, \cdots, s_{\mu-1})}{dt} & = f(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) & = L^T x(t, s_1, s_2, \cdots, s_{\mu-1}), \\
\end{align*}
\]

- \( x \) is the state variable
- \( u(t) \) is the input, \( y(t) \) is the output.

Parametric system can describe the physical system in more detail, but is more complicated than the non-parametric system.
Model Order Reduction progresses:

non-parametric linear system:

\[ \begin{align*}
C \frac{dx(t)}{dt} &= Gx(t) + Bu(t), \\
y(t) &= L^T x(t), \quad \Rightarrow
\end{align*} \]

non-parametric nonlinear system:

\[ \begin{align*}
\frac{dg(x(t))}{dt} &= f(x(t)) + Bu(t), \\
y(t) &= L^T x(t), \quad \Rightarrow
\end{align*} \]

parametric system:

\[ \begin{align*}
\frac{dg(x(t), s_1, s_2, \cdots, s_{\mu-1})}{dt} &= f(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\end{align*} \]

- \( x \) is the state variable
- \( u(t) \) is the input, \( y(t) \) is the output.

Parametric system can describe the physical system in more detail, but is more complicated than the non-parametric system.
Model Order Reduction progresses:

non-parametric linear system:
\[
\begin{align*}
C \frac{dx(t)}{dt} &= Gx(t) + Bu(t), \\
y(t) &= L^T x(t),
\end{align*}
\]

non-parametric nonlinear system:
\[
\begin{align*}
\frac{dg(x(t))}{dt} &= f(x(t)) + Bu(t), \\
y(t) &= L^T x(t),
\end{align*}
\]

parametric system:
\[
\begin{align*}
\frac{dg(x(t), s_1, s_2, \ldots, s_{\mu-1})}{dt} &= f(x(t), s_1, s_2, \ldots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \ldots, s_{\mu-1}),
\end{align*}
\]

- \(x\) is the state variable
- \(u(t)\) is the input, \(y(t)\) is the output.

Parametric system can describe the physical system in more detail, but is more complicated than the non-parametric system.
Parametric systems and Parametric MOR

Model Order Reduction progresses:

non-parametric linear system:
\[
C \frac{dx(t)}{dt} = Gx(t) + Bu(t), \\
y(t) = L^T x(t), \quad \Rightarrow
\]

non-parametric nonlinear system:
\[
\frac{dg(x(t))}{dt} = f(x(t)) + Bu(t), \\
y(t) = L^T x(t), \quad \Rightarrow
\]

parametric system:
\[
\frac{dg(x(t), s_1, s_2, \cdots, s_{\mu-1})}{dt} = f(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) = L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\]

- \(x\) is the state variable
- \(u(t)\) is the input, \(y(t)\) is the output.

Parametric system can describe the physical system in more detail, but is more complicated than the non-parametric system.
Parametric systems and parametric MOR

Parametric system in time domain:

\[
\begin{align*}
C \frac{dx(t,s_1,s_2,\cdots,s_{\mu-1})}{dt} &= G(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\end{align*}
\]

Parametric system in frequency domain:

\[
\begin{align*}
H(x, s_1, \ldots, s_\mu) &= Bu, \\
y &= L^T x.
\end{align*}
\]

Model reduction by one-sided projection:

\[
\begin{align*}
C \frac{dx(t,s_1,s_2,\cdots,s_{\mu-1})}{dt} &= G(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\end{align*}
\]

\[
\begin{align*}
\Downarrow \quad x &\approx Vz
\end{align*}
\]

\[
\begin{align*}
V^T C V \frac{dz(t,s_1,s_2,\cdots,s_{\mu-1})}{dt} &= V^T G V(z(t), s_1, s_2, \cdots, s_{\mu-1}) + V^T Bu(t), \\
y(t) &= L^T V z(t, s_1, s_2, \cdots, s_{\mu-1}),
\end{align*}
\]
Parametric systems and parametric MOR

Parametric system in time domain:

\[
C \frac{dx(t, s_1, s_2, \cdots, s_{\mu-1})}{dt} = G(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) = L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\]

Parametric system in frequency domain:

\[
H(x, s_1, \ldots, s_{\mu}) = Bu, \\
y = L^T x.
\]

Model reduction by one-sided projection:

\[
C \frac{dx(t, s_1, s_2, \cdots, s_{\mu-1})}{dt} = G(x(t), s_1, s_2, \cdots, s_{\mu-1}) + Bu(t), \\
y(t) = L^T x(t, s_1, s_2, \cdots, s_{\mu-1}),
\]

\[
\downarrow \quad x \approx Vz
\]

\[
V^T CV \frac{dz(t, s_1, s_2, \cdots, s_{\mu-1})}{dt} = V^T G V(z(t), s_1, s_2, \cdots, s_{\mu-1}) + V^T Bu(t), \\
y(t) = L^T V z(t, s_1, s_2, \cdots, s_{\mu-1}),
\]
Parametric systems and parametric MOR

Parametric system in time domain:

\[
C \frac{dx(t, s_1, s_2, \ldots, s_{\mu-1})}{dt} = G(x(t), s_1, s_2, \ldots, s_{\mu-1}) + Bu(t),
\]
\[
y(t) = L^Tx(t, s_1, s_2, \ldots, s_{\mu-1}),
\]

Parametric system in frequency domain:

\[
H(x, s_1, \ldots, s_{\mu}) = Bu,
\]
\[
y = L^Tx.
\]

Model reduction by one-sided projection:

\[
C \frac{dx(t, s_1, s_2, \ldots, s_{\mu-1})}{dt} = G(x(t), s_1, s_2, \ldots, s_{\mu-1}) + Bu(t),
\]
\[
y(t) = L^Tx(t, s_1, s_2, \ldots, s_{\mu-1}),
\]

\[
\downarrow \quad x \approx Vz
\]

\[
V^TCV \frac{dz(t, s_1, s_2, \ldots, s_{\mu-1})}{dt} = V^TGV(z(t), s_1, s_2, \ldots, s_{\mu-1}) + V^TBu(t),
\]
\[
y(t) = L^TVz(t, s_1, s_2, \ldots, s_{\mu-1}),
\]
Parametric systems and parametric MOR

Parametric system in time domain:

\[
\begin{align*}
C \frac{dx(t,s_1,s_2,\ldots,s_{\mu-1})}{dt} &= G(x(t), s_1, s_2, \ldots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \ldots, s_{\mu-1}),
\end{align*}
\]

Parametric system in frequency domain:

\[
\begin{align*}
H(x, s_1, \ldots, s_{\mu}) &= Bu, \\
y &= L^T x.
\end{align*}
\]

Model reduction by one-sided projection:

\[
\begin{align*}
C \frac{dx(t,s_1,s_2,\ldots,s_{\mu-1})}{dt} &= G(x(t), s_1, s_2, \ldots, s_{\mu-1}) + Bu(t), \\
y(t) &= L^T x(t, s_1, s_2, \ldots, s_{\mu-1}),
\end{align*}
\]

\[
\downarrow \quad x \approx Vz
\]

\[
\begin{align*}
V^T C V \frac{dz(t,s_1,s_2,\ldots,s_{\mu-1})}{dt} &= V^T G V(z(t), s_1, s_2, \ldots, s_{\mu-1}) + V^T Bu(t), \\
y(t) &= L^T V z(t, s_1, s_2, \ldots, s_{\mu-1}),
\end{align*}
\]
Parametric systems and parametric MOR

Review of Daniel’s method

A robust implementation of Daniel’s method

Simulation results

Conclusions
Review of Daniel’s method

Original parametric system:

\[ E(s_1, \ldots, s_\mu)x(s_1, \ldots, s_\mu) = Bu \]
\[ y = L^T x \]

Taylor series expansion of \( E(s_1, \ldots, s_\mu) \) around \( \bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu \),

\[
E(s_1, \ldots, s_\mu) = M_0 + \sum_j (s_j - \bar{s}_j)M_j + \frac{1}{2} \sum_{h,k} (s_h - \bar{s}_h)(s_k - \bar{s}_k)M_{h,k} + \\
(1/3!) \sum_{h,k,l} (s_h - \bar{s}_h)(s_k - \bar{s}_k)(s_l - \bar{s}_l)M_{h,k,l} + \ldots
\]

\[
M_0 = E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu), \\
M_j = \partial E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_j, \\
M_{h,k} = \partial^2 E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_h \partial s_k, \\
\ldots
\]
Review of Daniel’s method

Original parametric system:

\[ E(s_1, \ldots, s_\mu)x(s_1, \ldots, s_\mu) = Bu \]
\[ y = L^T x \]

Taylor series expansion of \( E(s_1, \ldots, s_\mu) \) around \( \bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu \),

\[ E(s_1, \ldots, s_\mu) = M_0 + \sum_j (s_j - \bar{s}_j)M_j + (1/2) \sum_{h,k} (s_h - \bar{s}_h)(s_k - \bar{s}_k)M_{h,k} + \\
(1/3!) \sum_{h,k,l} (s_h - \bar{s}_h)(s_k - \bar{s}_k)(s_l - \bar{s}_l)M_{h,k,l} + \ldots \]

\[ M_0 = E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu), \]
\[ M_j = \partial E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_j, \]
\[ M_{h,k} = \partial^2 E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_h \partial s_k, \]
\[ \ldots \]
Review of Daniel’s method

Original parametric system:

\[
E(s_1, \ldots, s_\mu) x(s_1, \ldots, s_\mu) = Bu \\
y = L^T x
\]

Taylor series expansion of \( E(s_1, \ldots, s_\mu) \) around \( \bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu \),

\[
E(s_1, \ldots, s_\mu) = M_0 + \sum_j (s_j - \bar{s}_j) M_j + (1/2) \sum_{h,k} (s_h - \bar{s}_h)(s_k - \bar{s}_k) M_{h,k} + \\
(1/3!) \sum_{h,k,l} (s_h - \bar{s}_h)(s_k - \bar{s}_k)(s_l - \bar{s}_l) M_{h,k,l} + \ldots
\]

\[
M_0 = E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu), \\
M_j = \partial E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu) / \partial s_j, \\
M_{h,k} = \partial^2 E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu) / \partial s_h \partial s_k, \\
\ldots
\]
Review of Daniel’s method

Original parametric system:

\[
E(s_1, \ldots, s_\mu)x(s_1, \ldots, s_\mu) = Bu
\]
\[
y = L^T x
\]

Taylor series expansion of \(E(s_1, \ldots, s_\mu)\) around \(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu\),

\[
E(s_1, \ldots, s_\mu) = M_0 + \sum_j (s_j - \bar{s}_j)M_j + (1/2)\sum_{h,k}(s_h - \bar{s}_h)(s_k - \bar{s}_k)M_{h,k} + (1/3!)\sum_{h,k,l}(s_h - \bar{s}_h)(s_k - \bar{s}_k)(s_l - \bar{s}_l)M_{h,k,l} + \ldots
\]

\[
M_0 = E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu),
\]
\[
M_j = \partial E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_j,
\]
\[
M_{h,k} = \partial^2 E(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_\mu)/\partial s_h \partial s_k,
\]
\[
\ldots
\]
Define a new group of matrices and parameters:

\[ E_i = \begin{cases} 
  M_j \\
  M_{h,k} \\
  M_{h,k,l} \\
  \ldots 
\end{cases} \]

\[ j = 0, \ldots, \mu \]

\[ h = 1, \ldots, \mu; k = 1, \ldots, \mu \]

\[ h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \]

\[ \tilde{s}_i = \begin{cases} 
  (s - \bar{s}_j) \\
  (s - \bar{s}_h)(s - \bar{s}_k), \\
  (s - \bar{s}_h)(s - \bar{s}_k)(s - \bar{s}_l) \\
  \ldots 
\end{cases} \]

\[ j = 0, \ldots, \mu \]

\[ h = 1, \ldots, \mu; k = 1, \ldots, \mu \]

\[ h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \]

Truncate the higher order terms in Taylor series to obtain an approximate linear parametric system:

\[
(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu \\
y = L^T x
\]
Define a new group of matrices and parameters:

$$E_i = \begin{cases} M_j & j = 0, \ldots, \mu \\ M_{h,k} & h = 1, \ldots, \mu; k = 1, \ldots, \mu \\ M_{h,k,l} & h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \\ \ldots \\ \end{cases}$$

$$\tilde{s}_i = \begin{cases} (s - \bar{s}_j) & j = 0, \ldots, \mu \\ (s - \bar{s}_h)(s - \bar{s}_k) & h = 1, \ldots, \mu; k = 1, \ldots, \mu \\ (s - \bar{s}_h)(s - \bar{s}_k)(s - \bar{s}_l) & h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \\ \ldots \\ \end{cases}$$

Truncate the higher order terms in Taylor series to obtain an approximate linear parametric system:

$$\left( E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p \right) x = Bu$$

$$y = L^T x$$
Define a new group of matrices and parameters:

\[ E_i = \begin{cases} 
M_j & j = 0, \ldots, \mu \\
M_{h,k} & h = 1, \ldots, \mu; k = 1, \ldots, \mu \\
M_{h,k,l} & h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \\
\ldots
\end{cases} \]

\[ \tilde{s}_i = \begin{cases} 
(s - \bar{s}_j) & j = 0, \ldots, \mu \\
(s - \bar{s}_h)(s - \bar{s}_k) & h = 1, \ldots, \mu; k = 1, \ldots, \mu \\
(s - \bar{s}_h)(s - \bar{s}_k)(s - \bar{s}_l) & h = 1, \ldots, \mu; k = 1, \ldots, \mu; l = 1, \ldots, \mu \\
\ldots
\end{cases} \]

Truncate the higher order terms in Taylor series to obtain an approximate linear parametric system:

\[
(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu
\]

\[
y = L^T x
\]
Construction of the projection matrix $V$:

$$(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu$$

$$y = L^T x$$

\[\Downarrow\]

$$x = [I - (\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p)]^{-1} E_0^{-1} Bu$$

$$= \sum_{j=0}^{\infty} \left[\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p\right]^j E_0^{-1} Bu$$

$$= \{B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p]$$

$$+ [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M \tilde{s}_1 \tilde{s}_2 +$$

$$+ \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p$$

$$+ (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 +$$

$$+ \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 s_p + (M_p)^2 \tilde{s}_p^2 + \ldots ]$$

$$+ [(M_1)^3 B_M \tilde{s}_1^3 + (M_1^2 M_2 + M_2 M_1^2) B_M \tilde{s}_1^2 \tilde{s}_2 + \ldots + (M_p)^p B_M \tilde{s}_p^3]$$

$$+ \ldots \} u$$
Review of Daniel’s method

Construction of the projection matrix $V$:

$$
(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x =Bu
$$
$$
y = L^T x
$$

\[
\downarrow
\]

$$
x = [I - (\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p)]^{-1} E_0^{-1} Bu
$$

$$
= \sum_{j=0}^{\infty} [\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p]^j E_0^{-1} Bu
$$

$$
= \{B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p]
$$

$$
+ [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M s_1 \tilde{s}_2 +
+ \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p
$$

$$
+ (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 +
+ \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 s_p + (M_p)^2 \tilde{s}_p^2 + \ldots]
$$

$$
+ [(M_1)^3 B_M \tilde{s}_1^3 + (M_1^2 M_2 + M_2 M_1^2) B_M \tilde{s}_1^2 \tilde{s}_2 + \ldots + (M_p)^p B_M \tilde{s}_p^3]
$$

+ \ldots \} u
Review of Daniel’s method

Construction of the projection matrix $V$:

$$(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu$$

$$y = L^T x$$

$$\Downarrow$$

$$x = [I - (\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p)]^{-1} E_0^{-1} Bu$$

$$= \sum_{j=0}^{\infty} [\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p]^j E_0^{-1} Bu$$

$$= \{B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p]$$

$$+ [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M \tilde{s}_1 \tilde{s}_2 +$$

$$+ \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p$$

$$+ (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 +$$

$$+ \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 \tilde{s}_p + (M_p)^2 \tilde{s}_p^2 + \ldots]$$

$$+ [(M_1)^3 B_M \tilde{s}_1^3 + (M_1^2 M_2 + M_2 M_1^2) B_M \tilde{s}_1^2 \tilde{s}_2 + \ldots + (M_p)^p B_M \tilde{s}_p^3]$$

$$+ \ldots \} u$$
Construction of the projection matrix $V$:

$$(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu$$

$$y = L^T x$$

\[\downarrow\]

$$x = [I - (\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p)]^{-1} E_0^{-1} Bu$$

$$= \sum_{j=0}^{\infty} [\tilde{s}_1 E_0^{-1} E_1 + \ldots + \tilde{s}_p E_0^{-1} E_p]^j E_0^{-1} Bu$$

$$= \{B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p] + [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M s_1 \tilde{s}_2 + \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p + (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 + \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 s_p + (M_p)^2 \tilde{s}_p^2 + \ldots] + [(M_1)^3 B_M \tilde{s}_1^3 + (M_1^2 M_2 + M_2 M_1^2) B_M \tilde{s}_1^2 \tilde{s}_2 + \ldots + (M_p)^p B_M \tilde{s}_p^3] + \ldots \} u$$
Review of Daniel’s method

Construction of the projection matrix $V$:

$$x = \{ B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p]$$

$$+ [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M \tilde{s}_1 \tilde{s}_2 + \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p]$$

$$+ (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 +$$

$$+ \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 \tilde{s}_p + (M_p)^2 \tilde{s}_p^2 + \ldots \} u$$

$$\downarrow$$

$$\text{range}\{V\}$$

$$= \text{span}\{ B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots,$$

$$\ldots \{ M_1 M_p + M_p M_1 \} B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, \ldots \}$$
Review of Daniel’s method

Construction of the projection matrix $V$:

$$x = \{B_M + [M_1 B_M \tilde{s}_1 + M_2 B_M \tilde{s}_2 + \ldots + M_p B_M \tilde{s}_p]$$

$$+ [(M_1)^2 B_M \tilde{s}_1^2 + (M_1 M_2 + M_2 M_1) B_M \tilde{s}_1 \tilde{s}_2 + \ldots + (M_1 M_p + M_p M_1) B_M \tilde{s}_1 \tilde{s}_p$$

$$+ (M_2)^2 B_M \tilde{s}_2^2 + (M_2 M_3 + M_3 M_2) B_M \tilde{s}_2 \tilde{s}_3 +$$

$$+ \ldots + (M_2 M_p + M_p M_2) B_M \tilde{s}_2 \tilde{s}_p + (M_p)^2 \tilde{s}_p^2 + \ldots]$$

$$+ [(M_1)^3 B_M \tilde{s}_1^3 + (M_1^2 M_2 + M_2 M_1^2) B_M \tilde{s}_1^2 \tilde{s}_2 + \ldots + (M_p)^p B_M \tilde{s}_p^3]$$

$$+ \ldots \} u$$

$$\downarrow$$

$\text{range}\{V\}$

$= \text{span}\{B_M, \underbrace{M_1 B_M, M_2 B_M, \ldots, M_p B_M}_{(M_1^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots),}$

$\underbrace{(M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, \ldots}_{}\}$
Review of Daniel’s method

Derivation of the reduced model:

\[(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu\]
\[y = L^T x\]

\[\Downarrow x \approx Vz\]

\[(\hat{E}_0 + \tilde{s}_1 \hat{E}_1 + \tilde{s}_2 \hat{E}_2 + \ldots + \tilde{s}_p \hat{E}_p)z = \hat{B}u\]
\[\hat{y} = \hat{L}^T z\]

\[\hat{E}_i = V^T E_i V, \quad i = 0, 1, 2, \ldots, p, \quad \hat{B} = V^T B, \quad \hat{L} = V^T L.\]
Derivation of the reduced model:

\[
(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu
\]
\[
y = L^T x
\]

\[
\Downarrow \quad x \approx Vz
\]

\[
(\hat{E}_0 + \tilde{s}_1 \hat{E}_1 + \tilde{s}_2 \hat{E}_2 + \ldots + \tilde{s}_p \hat{E}_p)z = \hat{B}u
\]
\[
\hat{y} = \hat{L}^T z
\]

\[
\hat{E}_i = V^T E_i V, \quad i = 0, 1, 2, \ldots, p, \quad \hat{B} = V^T B, \quad \hat{L} = V^T L.
\]
Derivation of the reduced model:

\[
(E_0 + \tilde{s}_1 E_1 + \tilde{s}_2 E_2 + \ldots + \tilde{s}_p E_p)x = Bu
\]
\[
y = L^T x
\]

\[
\Downarrow x \approx Vz
\]

\[
(\hat{E}_0 + \hat{s}_1 \hat{E}_1 + \hat{s}_2 \hat{E}_2 + \ldots + \hat{s}_p \hat{E}_p)z = \hat{B}u
\]
\[
\hat{y} = \hat{L}^T z
\]

\[
\hat{E}_i = V^T E_i V, \quad i = 0, 1, 2, \ldots, p, \quad \hat{B} = V^T B, \quad \hat{L} = V^T L.
\]
Review of Daniel’s method

How to compute $V$?

Direct computation of $V$

$V := [B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots, (M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, (M_p)^q B_M].$

Theoretically,

$$V = [v_1, v_2, \ldots, v_{100}]$$

Numerically, higher order moments become linear dependent very quickly,

$$V = [v_1, v_2, \ldots, v_{10}]$$
Review of Daniel’s method

How to compute $V$?

Direct computation of $V$

$$V := [B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots, (M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, (M_p)^q B_M].$$

Theoretically,

$$V = [v_1, v_2, \ldots, v_{100}]$$

Numerically, higher order moments become linear dependent very quickly,

$$V = [v_1, v_2, \ldots, v_{10}]$$
Review of Daniel’s method

How to compute $V$?

**Direct computation of $V$**

$$V := [B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots, (M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, (M_p)^q B_M].$$

Theoretically,

$$V = [v_1, v_2, \ldots, v_{100}]$$

Numerically, higher order moments become linear dependent very quickly,

$$V = [v_1, v_2, \ldots, v_{10}]$$
How to compute $V$?

**Direct computation of $V$**

$$V := [B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots, (M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, (M_p)^q B_M].$$

Theoretically,

$$V = [v_1, v_2, \ldots, v_{100}]$$

Numerically, higher order moments become linear dependent very quickly,

$$V = [v_1, v_2, \ldots, v_{10}]$$
Review of Daniel’s method

How to compute $V$?

Direct computation of $V$

$$V := [B_M, M_1 B_M, M_2 B_M, \ldots, M_p B_M, (M_1)^2 B_M, (M_1 M_2 + M_2 M_1) B_M, \ldots, (M_1 M_p + M_p M_1) B_M, (M_2)^2 B_M, (M_2 M_3 + M_3 M_2) B_M, \ldots, (M_p)^q B_M].$$

Theoretically,

$$V = [v_1, v_2, \ldots, v_{100}]$$

Numerically, higher order moments become linear dependent very quickly,

$$V = [v_1, v_2, \ldots, v_{10}]$$
Review of Daniel’s method

Analog to the analysis in PVL [Feldmann and Freund 95] (v.s. AWE [Pillage and Rohrer 90]), direct computation of $V$ includes computation of the following terms:

$$M_1 B_M, M_1^2 B_M, M_1^3 B_M, \ldots, M_1^j B_M \rightarrow \xi^{(1)}, \quad M_1 \xi^{(1)} = \lambda_{max}^{(1)} \xi^{(1)}$$

$$M_2 B_M, M_2^2 B_M, M_2^3 B_M, \ldots, M_2^j B_M \rightarrow \xi^{(2)}, \quad M_2 \xi^{(2)} = \lambda_{max}^{(2)} \xi^{(2)}$$

$$\ldots$$

$$M_p B_M, M_p^2 B_M, M_p^3 B_M, \ldots, M_p^j B_M \rightarrow \xi^{(p)}, \quad M_p \xi^{(p)} = \lambda_{max}^{(p)} \xi^{(p)}$$

$\implies$ We actually run $p$ power iterations for $M_1, M_2, \ldots, M_p$; most of the vectors above will become parallel with increasing $j$

Goal: Derive a numerical stable algorithm for computation of $V$

i.e., orthomormal basis of the subspace spanned by the coefficients in the series expansion of $x$
Review of Daniel’s method

Analog to the analysis in PVL [Feldmann and Freund 95] (v.s. AWE [Pillage and Rohrer 90]), **direct computation of** $V$ **includes computation of the following terms:**

\[
M_1 B_M, M_1^2 B_M, M_1^3 B_M, \ldots, M_1^j B_M \rightarrow \xi^{(1)}, \quad M_1 \xi^{(1)} = \lambda^{(1)} \xi^{(1)}
\]

\[
M_2 B_M, M_2^2 B_M, M_2^3 B_M, \ldots, M_2^j B_M, \rightarrow \xi^{(2)}, \quad M_2 \xi^{(2)} = \lambda^{(2)} \xi^{(2)}
\]

\[
\vdots
\]

\[
M_p B_M, M_p^2 B_M, M_p^3 B_M, \ldots, M_p^j B_M, \rightarrow \xi^{(p)}, \quad M_p \xi^{(p)} = \lambda^{(p)} \xi^{(p)}
\]

\[\implies\] **We actually run** $p$ **power iterations for** $M_1, M_2, \ldots, M_p$; **most of the vectors above will become parallel with increasing** $j$

**Goal:** Derive a numerical stable algorithm for computation of $V$ i.e., orthomormal basis of the subspace spanned by the coefficients in the series expansion of $x$
Review of Daniel’s method

Analog to the analysis in PVL [Feldmann and Freund 95] (v.s. AWE [Pillage and Rohrer 90]), direct computation of $V$ includes computation of the following terms:

$$M_1 B_M, M_1^2 B_M, M_1^3 B_M, \ldots, M_1^j B_M \longrightarrow \xi^{(1)}, \quad M_1 \xi^{(1)} = \lambda^{(1)} \max \xi^{(1)}$$

$$M_2 B_M, M_2^2 B_M, M_2^3 B_M, \ldots, M_2^j B_M, \longrightarrow \xi^{(2)}, \quad M_2 \xi^{(2)} = \lambda^{(2)} \max \xi^{(2)}$$

$$\vdots$$

$$M_p B_M, M_p^2 B_M, M_p^3 B_M, \ldots, M_p^j B_M, \longrightarrow \xi^{(p)}, \quad M_p \xi^{(p)} = \lambda^{(p)} \max \xi^{(p)}$$

$\implies$ We actually run $p$ power iterations for $M_1, M_2, \ldots, M_p$; most of the vectors above will become parallel with increasing $j$.

Goal: Derive a numerical stable algorithm for computation of $V$ i.e., orthonormal basis of the subspace spanned by the coefficients in the series expansion of $x$. 

12/29

L. Feng, P. Benner

Robust Algorithm for Parametric Model Order Reduction
Analog to the analysis in PVL [Feldmann and Freund 95] (v.s. AWE [Pillage and Rohrer 90]), direct computation of $V$ includes computation of the following terms:

$$
M_1 B_M, M_1^2 B_M, M_1^3 B_M, \ldots, M_1^j B_M \rightarrow \xi^{(1)}, \quad M_1 \xi^{(1)} = \lambda_{\text{max}}^{(1)} \xi^{(1)}
$$

$$
M_2 B_M, M_2^2 B_M, M_2^3 B_M, \ldots, M_2^j B_M, \rightarrow \xi^{(2)}, \quad M_2 \xi^{(2)} = \lambda_{\text{max}}^{(2)} \xi^{(2)}
$$

\[\vdots\]

$$
M_p B_M, M_p^2 B_M, M_p^3 B_M, \ldots, M_p^j B_M, \rightarrow \xi^{(p)}, \quad M_p \xi^{(p)} = \lambda_{\text{max}}^{(p)} \xi^{(p)}
$$

$\Rightarrow$ We actually run $p$ power iterations for $M_1, M_2, \ldots, M_p$; most of the vectors above will become parallel with increasing $j$.

**Goal:** Derive a numerical stable algorithm for computation of $V$ i.e., orthonormal basis of the subspace spanned by the coefficients in the series expansion of $x$.
♦ Parametric systems and Parametric MOR

♦ Review of Daniel’s method

♦ A robust implementation of Daniel’s method

♦ Simulation results

♦ Conclusions
A Robust implementation of Daniel’s method

**Observation:** power series expansion of $x$ can be re-ordered as,

$$
x = \sum_{m=0}^{\infty} [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^m B_M
$$

$$
= B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M + \ldots
+ [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M + \ldots
$$

Define:

$$
x_0 = B_M
$$

$$
x_1 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M,
$$

$$
x_2 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M,
$$

$$
\vdots
$$

$$
x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M
$$

$$
\vdots
$$

We get:

$$
x = x_0 + x_1 + x_2 + \ldots + x_j + \ldots
$$
A Robust implementation of Daniel’s method

Observation: power series expansion of $x$ can be re-ordered as,

$$
x = \sum_{m=0}^{\infty} [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^m B_M
= B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M + \ldots + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M + \ldots
$$

Define:

$$
x_0 = B_M
x_1 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M,
x_2 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M,
\vdots
x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M
\vdots
$$

We get:

$$
x = x_0 + x_1 + x_2 + \ldots + x_j + \ldots
$$
A Robust implementation of Daniel's method

Observation: power series expansion of \( x \) can be re-ordered as,

\[
x = \sum_{m=0}^{\infty} \left[ \tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p \right]^m B_M
\]

\[
= B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M + \ldots
\]

\[
+ [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M + \ldots
\]

Define:

\[
\begin{align*}
  x_0 &= B_M \\
  x_1 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M, \\
  x_2 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M, \\
  &\vdots \\
  x_j &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M \\
  &\vdots
\end{align*}
\]

\[
\Rightarrow \quad x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_{j-1}
\]

We get:

\[
x = x_0 + x_1 + x_2 + \ldots + x_j + \ldots
\]
A Robust implementation of Daniel’s method

Observation: power series expansion of $x$ can be re-ordered as,

$$ x = \sum_{m=0}^{\infty} [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^m B_M $$

$$ = B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M + \ldots + [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M + \ldots $$

Define:

$$ x_0 = B_M $$
$$ x_1 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] B_M, $$
$$ x_2 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^2 B_M, $$
$$ \vdots $$
$$ x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]^j B_M $$

$$ \Rightarrow $$

$$ x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_{j-1} $$
$$ \vdots $$

We get:

$$ x = x_0 + x_1 + x_2 + \ldots + x_j + \ldots $$
A Robust implementation of Daniel’s method

\[
\begin{align*}
    x_0 &= B_M, \\
    x_1 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_0, \\
    x_2 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_1, \\
    \vdots \\
    x_j &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_{j-1}, \\
    \vdots \\
    x &\approx x_0 + x_1 + x_2 + \ldots + x_q \iff \text{range}(V) = \text{span}\{R_0, R_1, \ldots, R_q\}
\end{align*}
\]

\[R_0 = B_M, \quad R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \quad R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \quad R_j = [M_1 R_{j-1}, M_2 R_{j-1}, \ldots, M_p R_{j-1}]\]
A Robust implementation of Daniel’s method

\[
x_0 = B_M, \quad R_0 = B_M,
\]

\[
x_1 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]x_0, \quad R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0],
\]

\[
x_2 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]x_1, \quad R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1],
\]

\[
\vdots
\]

\[
x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p]x_{j-1} \quad \iff \quad R_j = [M_1 R_{j-1}, M_2 R_{j-1}, \ldots, M_p R_{j-1}]
\]

\[
\vdots
\]

\[
x \approx x_0 + x_1 + x_2 + \ldots + x_q \iff \text{range}(V) = \text{span}\{R_0, R_1, \ldots, R_q\}
\]
A Robust implementation of Daniel’s method

\[ x_0 = B_M, \]
\[ x_1 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_0, \]
\[ x_2 = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_1, \]
\[ \vdots \]
\[ x_j = [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_{j-1} \]
\[ \vdots \]

\[ x \approx x_0 + x_1 + x_2 + \ldots + x_q \iff \text{range}(V) = \text{span}\{R_0, R_1, \ldots, R_q\} \]
A Robust implementation of Daniel’s method

\[
\begin{align*}
    x_0 &= B_M, \\
    x_1 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_0, \\
    x_2 &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_1, \\
    \vdots \\
    x_j &= [\tilde{s}_1 M_1 + \ldots + \tilde{s}_p M_p] x_{j-1} \\
    \vdots \\

    x &\approx x_0 + x_1 + x_2 + \ldots + x_q \iff \text{range}(V) = \text{span}\{R_0, R_1, \ldots, R_q\}
\end{align*}
\]
Algorithm 1: Compute an orthonormal projection matrix $V = [v_1, v_2, \ldots, v_{nr}]$ for linear parametric system.

1. Compute the first columns in $V$
   
   $V_0 = \text{orth}\{R_0\}$,

2. Compute the orthonormal columns in $R_1, R_2, \ldots, R_q$ iteratively based on the Modified Gram-Schmidt orthogonalization process with deflation to get $\tilde{V}$,

3. Orthogonalize the columns in $\tilde{V}$ again to delete possible dependent columns to get the final $V$. 
Algorithm 1: Compute an orthonormal projection matrix $V = [v_1, v_2, \ldots, v_{nr}]$ for linear parametric system.

1. Compute the first columns in $V$
   \[ V_0 = \text{orth}\{R_0\}, \]

2. Compute the orthonormal columns in $R_1, R_2, \ldots, R_q$ iteratively based on the Modified Gram-Schmidt orthogonalization process with deflation to get $\tilde{V}$,

3. Orthogonalize the columns in $\tilde{V}$ again to delete possible dependent columns to get the final $V$. 
Algorithm 1: Compute an orthonormal projection matrix $V = [v_1, v_2, ..., v_{nr}]$ for linear parametric system.

1. Compute the first columns in $V$
   $V_0 = \text{orth}\{R_0\}$,

2. Compute the orthonormal columns in $R_1, R_2, ..., R_q$ iteratively based on the Modified Gram-Schmidt orthogonalization process with deflation to get $\tilde{V}$,

3. Orthogonalize the columns in $\tilde{V}$ again to delete possible dependent columns to get the final $V$. 
**Algorithm 1**: Compute an orthonormal projection matrix \( V = [v_1, v_2, ..., v_{n_r}] \) for linear parametric system.

1. Compute the first columns in \( V \)
   \[ V_0 = \text{orth}\{R_0\}, \]

2. Compute the orthonormal columns in \( R_1, R_2, ..., R_q \) iteratively based on the Modified Gram-Schmidt orthogonalization process with deflation to get \( \tilde{V} \),

3. Orthogonalize the columns in \( \tilde{V} \) again to delete possible dependent columns to get the final \( V \).
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
R_0 = B_M, \\
R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots \\
R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \\
\]

\[\downarrow\]

\[
R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \\
R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \\
R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \\
\vdots \\
R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \\
\vdots \\
R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \Downarrow \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \downarrow \]

\[ R_0 \implies V_0 \implies \text{the 1st iteration step,} \]
\[ R_1 \implies M_1 V_0, M_2 V_0, \ldots, M_p V_0 \implies V_1 \implies \text{the 2nd iteration step,} \]
\[ R_2 \implies M_1 V_1, M_2 V_1, \ldots, M_p V_1 \implies V_2 \implies \text{the 3rd iteration step,} \]
\[ \vdots \]
\[ R_j \implies M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \implies V_j \implies \text{the } j\text{th iteration step,} \]
\[ \vdots \]
\[ R_q \implies M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \implies V_q \implies \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
\begin{align*}
R_0 &= B_M, \\
R_1 &= [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 &= [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots
R_q &= [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\end{align*}
\]

\[\downarrow\]

\[
\begin{align*}
R_0 &\Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \\
R_1 &\Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \\
R_2 &\Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \\
\vdots
R_j &\Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \\
\vdots
R_q &\Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\end{align*}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Step 2:

\[
R_0 = B_M, \\
R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots \\
R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\]

\[
\downarrow
\]

\[
R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step,} \\
R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step,} \\
R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step,} \\
\vdots \\
R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step,} \\
\vdots \\
R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \Rightarrow \]
\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{\mathcal{V}} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
\begin{align*}
R_0 & = B_M, \\
R_1 & = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 & = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
& \vdots \\
R_q & = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \\
\end{align*}
\]

\[\Rightarrow \]

\[
\begin{align*}
R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step,} \\
R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step,} \\
R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step,} \\
& \vdots \\
R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step,} \\
& \vdots \\
R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\end{align*}
\]

\[\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].\]
Step 2:

\[
R_0 = B_M,
\]
\[
R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0],
\]
\[
R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1],
\]
\[
\vdots
\]
\[
R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\]

\[
\Rightarrow \quad R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step},
\]
\[
R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step},
\]
\[
R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step},
\]
\[
\vdots
\]
\[
R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step},
\]
\[
\vdots
\]
\[
R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
R_0 = B_M, \\
R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots \\
R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\]

\[\downarrow\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \Downarrow \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \therefore \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \Downarrow \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j \text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Step 2:

\[
\begin{align*}
R_0 &= B_M, \\
R_1 &= [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 &= [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
&\vdots \\
R_q &= [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \\
\downarrow \\
R_0 &\implies V_0 \implies \text{the 1st iteration step,} \\
R_1 &\implies M_1 V_0, M_2 V_0, \ldots, M_p V_0 \implies V_1 \implies \text{the 2nd iteration step,} \\
R_2 &\implies M_1 V_1, M_2 V_1, \ldots, M_p V_1 \implies V_2 \implies \text{the 3rd iteration step,} \\
&\vdots \\
R_j &\implies M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \implies V_j \implies \text{the } j\text{th iteration step,} \\
&\vdots \\
R_q &\implies M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \implies V_q \implies \text{the last iteration step}
\end{align*}
\]

\[\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].\]
Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \downarrow \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
R_0 = B_M, \\
R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots \\
R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\]

\[
R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step,} \\
R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step,} \\
R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step,} \\
\vdots \\
R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step,} \\
\vdots \\
R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[ R_0 = B_M, \]
\[ R_1 = [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \]
\[ R_2 = [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \]
\[ \vdots \]
\[ R_q = [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}] \]

\[ \downarrow \]

\[ R_0 \Rightarrow V_0 \Rightarrow \text{the 1st iteration step}, \]
\[ R_1 \Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step}, \]
\[ R_2 \Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step}, \]
\[ \vdots \]
\[ R_j \Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j \text{th iteration step}, \]
\[ \vdots \]
\[ R_q \Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step} \]

\[ \tilde{V} = [V_0, V_1, V_2, \ldots, V_q]. \]
Robust algorithm based on modified Gram-Schmidt

Step 2:

\[
\begin{align*}
R_0 &= B_M, \\
R_1 &= [M_1 R_0, M_2 R_0, \ldots, M_p R_0], \\
R_2 &= [M_1 R_1, M_2 R_1, \ldots, M_p R_1], \\
\vdots \\
R_q &= [M_1 R_{q-1}, M_2 R_{q-1}, \ldots, M_p R_{q-1}]
\end{align*}
\]

\[\downarrow\]

\[
\begin{align*}
R_0 &\Rightarrow V_0 \Rightarrow \text{the 1st iteration step,} \\
R_1 &\Rightarrow M_1 V_0, M_2 V_0, \ldots, M_p V_0 \Rightarrow V_1 \Rightarrow \text{the 2nd iteration step,} \\
R_2 &\Rightarrow M_1 V_1, M_2 V_1, \ldots, M_p V_1 \Rightarrow V_2 \Rightarrow \text{the 3rd iteration step,} \\
\vdots \\
R_j &\Rightarrow M_1 V_{j-1}, M_2 V_{j-1}, \ldots, M_p V_{j-1} \Rightarrow V_j \Rightarrow \text{the } j\text{th iteration step,} \\
\vdots \\
R_q &\Rightarrow M_1 V_{q-1}, M_2 V_{q-1}, \ldots, M_p V_{q-1} \Rightarrow V_q \Rightarrow \text{the last iteration step}
\end{align*}
\]

\[
\tilde{V} = [V_0, V_1, V_2, \ldots, V_q].
\]
◊ Parametric system and Parametric MOR

◊ Review of Daniel’s method

◊ A robust implementation of Daniel’s method

◊ Simulation results

◊ Conclusions
Example 1: Electro-Chemical Reaction

- Model description:

\[ E \frac{d}{dt} c + Gc + (s_1 D_1 + s_2 D_2) c = bu(t), \quad y = l^T c, \]

where \( c \in \mathbb{R}^n \) \((n = 16912)\) is the unknown concentrations.

- The output \( y \) is the current changing with the input voltage \( u \).

- \( s_1, s_2 \) are functions of the parameter \( \alpha \), the interesting values of \( \alpha \): \([0.0005, 0.5]\).

- The Chemical reaction takes place on the electrode:

\[ Ox + e^- \iff Red \]
Example 1, Cyclic Voltammogramm for $\alpha = 0.5$:

Multi-Moments matched up to 4th order ($s^4, s_1^4, s_2^4$), $\hat{n} = 26$.

Multi-Moments matched up to 6th order ($s^6, s_1^6, s_2^6$), $\hat{n} = 50$. 
Example 1, Cyclic Voltammogramm for $\alpha = 0.5$:

Multi-Moments matched up to 8th order ($s^8, s_1^8, s_2^8$), $\hat{n} = 84$. 
Example 1, Cyclic Voltammogramm for different values of $\alpha$:

Multi-Moments matched up to 8th order ($s^8, s^8_1, s^8_2$), $\alpha = 0.05$, $\hat{n} = 84$.

Multi-Moments matched up to 8th order ($s^8, s^8_1, s^8_2$), $\alpha = 0.005$, $\hat{n} = 84$. 

---

L. Feng, P. Benner
Robust Algorithm for Parametric Model Order Reduction
Example 2: Thermodynamical Model, a microthruster unit

**Model Description:**

\[ C \frac{d}{dt} x(t) + (G + kD)x(t) = bu(t), \quad y = l^T x(t), \]

where \( x \in \mathbb{R}^n \) \((n = 4725)\) is the unknown temperature distributed on the unit.

- The output \( y \) is the temperature in the middle of the heater.
- The parameter \( k \) is the so called film coefficient. Interesting values of \( k \): \( 1 \leq k \leq 10^9 \).

\[ \text{PolySi} \quad \text{SOG} \]

\[ \text{SiNx} \]

\[ \text{SiO2} \]

\[ \text{Fuel} \quad \text{Si-substrate} \]

---

Example 2, Simulation results of the thermal model:

Output response of the original model corresponding to different values of $k$.

Relative errors of $y_r$ by three different reduced models.
Conclusions

- Advantages and drawbacks of the parametric MOR in [Daniel04] is analyzed.

- A numerical stable algorithm is proposed based on [Daniel04], guaranteeing the accuracy of the projection matrix.

- The algorithm applies to system with any number of distinct parameters.

- Simulation results have shown the robustness of the algorithm.

- Two sided projection in the style of dual Arnoldi is also possible, yielding twice the number of matched moments. ([Weile et al. 99]).
Advantages and drawbacks of the parametric MOR in [Daniel04] is analyzed.

A numerical stable algorithm is proposed based on [Daniel04], guaranteeing the accuracy of the projection matrix.

The algorithm applies to systems with any number of distinct parameters.

Simulation results have shown the robustness of the algorithm.

Two-sided projection in the style of dual Arnoldi is also possible, yielding twice the number of matched moments. ([Weile et al. 99]).
Advantages and drawbacks of the parametric MOR in [Daniel04] is analyzed.

A numerical stable algorithm is proposed based on [Daniel04], guaranteeing the accuracy of the projection matrix.

The algorithm applies to systems with any number of distinct parameters.

Simulation results have shown the robustness of the algorithm.

Two-sided projection in the style of dual Arnoldi is also possible, yielding twice the number of matched moments. ([Weile et al. 99]).
Advantages and drawbacks of the parametric MOR in [Daniel04] is analyzed.

A numerical stable algorithm is proposed based on [Daniel04], guaranteeing the accuracy of the projection matrix.

The algorithm applies to system with any number of distinct parameters.

Simulation results have shown the robustness of the algorithm.

Two sided projection in the style of dual Arnoldi is also possible, yielding twice the number of matched moments. ([Weile et al. 99]).
Advantages and drawbacks of the parametric MOR in [Daniel04] is analyzed.

A numerical stable algorithm is proposed based on [Daniel04], guaranteeing the accuracy of the projection matrix.

The algorithm applies to system with any number of distinct parameters.

Simulation results have shown the robustness of the algorithm.

Two sided projection in the style of dual Arnoldi is also possible, yielding twice the number of matched moments. ([Weile et al. 99]).
Two different parametric MOR algorithms treating numerical instability of direct computation of moments are proposed in:


The advantages and weak points compared with our method will be studied in the future.


The speaker’s attendance at this conference was sponsored by the Alexander von Humboldt Foundation.

http://www.humboldt-foundation.de

Thank you!