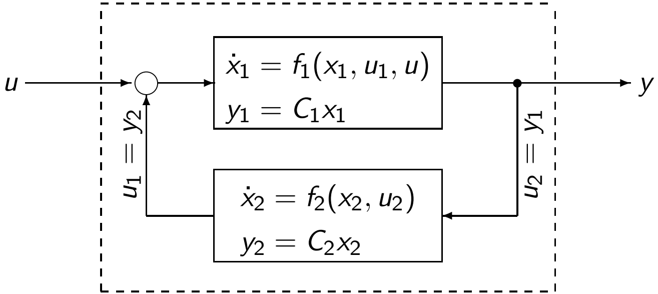
Model order reduction and Dynamic Iteration for coupled systems

Johanna Kerler
joint work with Tatjana Stykel

University of Augsburg

12.12.2013


u
$\dot{x}_1 = f_1(x_1, u_1, u)$
$y_1 = C_1 x_1$
y
$u_1 = y_2$
$u_2 = y_1$
$\dot{x}_2 = f_2(x_2, u_2)$
$y_2 = C_2 x_2$

Problems:

Problems:

- whole system is very large

Problems:

- whole system is very large
- different properties of subsystems

Problems:

- whole system is very large
- different properties of subsystems
- different time scales

Problems:

- whole system is very large
- different properties of subsystems
- different time scales

Approaches:

Problems:

- whole system is very large
- different properties of subsystems
- different time scales

Approaches:

- model order reduction

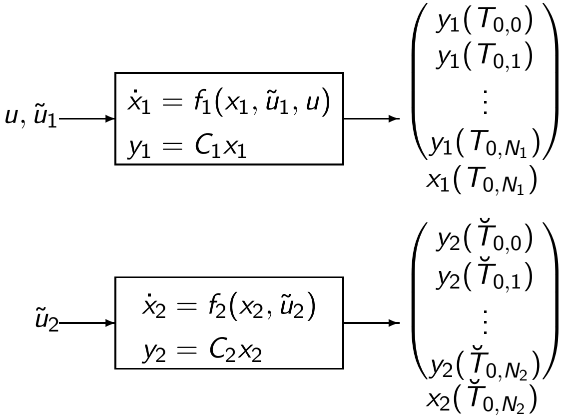
Problems:

- whole system is very large
- different properties of subsystems
- different time scales

Approaches:

- model order reduction
- Dynamic Iteration

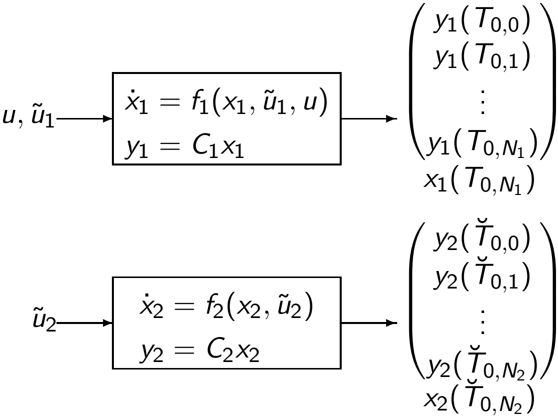
Dynamic Iteration - 1st macro step - initial step



initial step on $[T_0, T_1]$

Dynamic Iteration - 1st macro step - initial step

- approximate $\tilde{u}_1 \equiv y_2(T_0)$ and $\tilde{u}_2 \equiv y_1(T_0)$



initial step on $[T_0, T_1]$

Dynamic Iteration - 1st macro step - initial step

- approximate
  $\tilde{u}_1 \equiv y_2(T_0)$ and
  $\tilde{u}_2 \equiv y_1(T_0)$
- solve the systems separately

$u, \tilde{u}_1 \longrightarrow$

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \tilde{u}_1, u) \\ y_1 &= C_1 x_1 \end{aligned} \longrightarrow \begin{pmatrix} y_1(T_{0,0}) \\ y_1(T_{0,1}) \\ \vdots \\ y_1(T_{0,N_1}) \end{pmatrix}$$

$$x_1(T_{0,N_1})$$

$\tilde{u}_2 \longrightarrow$

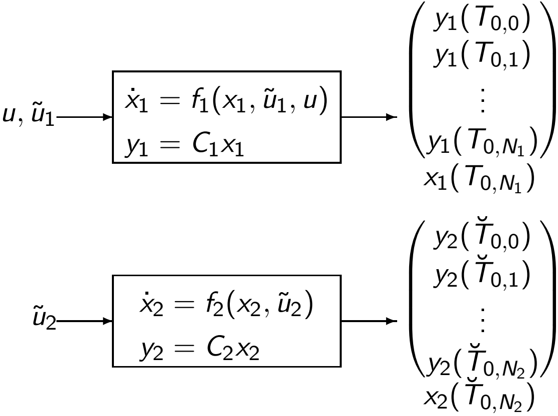
$$\begin{aligned} \dot{x}_2 &= f_2(x_2, \tilde{u}_2) \\ y_2 &= C_2 x_2 \end{aligned} \longrightarrow \begin{pmatrix} y_2(\check{T}_{0,0}) \\ y_2(\check{T}_{0,1}) \\ \vdots \\ y_2(\check{T}_{0,N_2}) \end{pmatrix}$$

$$x_2(\check{T}_{0,N_2})$$

initial step on $[T_0, T_1]$

Dynamic Iteration - 1st macro step - initial step

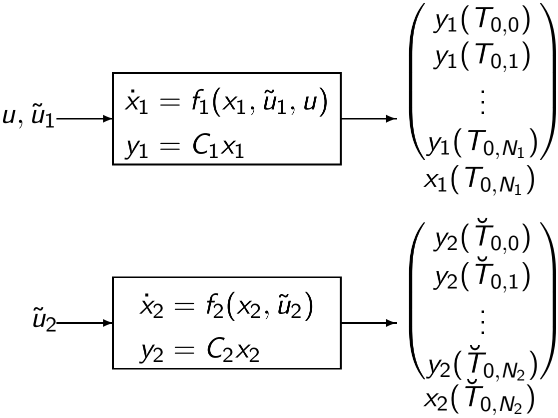
- approximate
  $\tilde{u}_1 \equiv y_2(T_0)$ and
  $\tilde{u}_2 \equiv y_1(T_0)$
- solve the systems separately
- store y_1 and y_2 at all micro steps



initial step on $[T_0, T_1]$

Dynamic Iteration - 1st macro step - iteration step

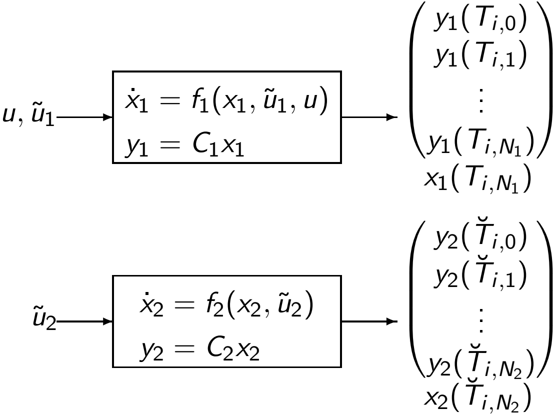
- approximate u_1 and u_2 by interpolation of y_1 and y_2 from the last iteration
- solve the subsystems separately
- store y_1 and y_2 at all micro steps



iteration step on $[T_0, T_1]$

Dynamic Iteration - i-th macro step - initial step

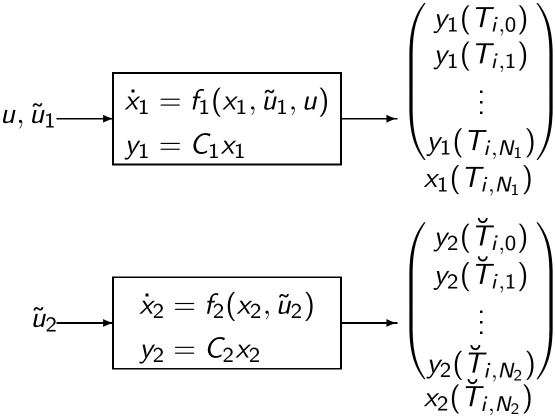
- approximate u_1 and u_2 by extrapolation of y_1 and y_2
- solve the subsystems separately
- store y_1 and y_2 at all micro steps



initial step on $[T_i, T_{i+1}]$

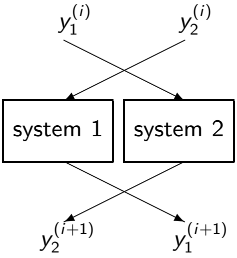
Dynamic Iteration - i-th macro step - iteration step

- approximate u_1 and u_2 by interpolation of y_1 and y_2 from the last iteration
- solve the subsystems separately
- store y_1 and y_2 at all micro steps



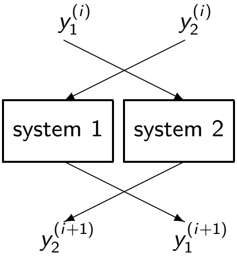
iteration step on $[T_i, T_{i+1}]$

Communication between the subsystems

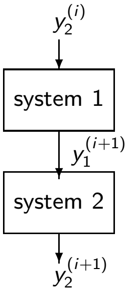


(a) Jacobi

Communication between the subsystems



(a) Jacobi



(b) Gaus-Seidel

Known theory:

- for ODEs:

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller
  - preconditioning to speed up the convergence

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller
  - preconditioning to speed up the convergence
- for DAEs:

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller
  - preconditioning to speed up the convergence
- for DAEs:
  - converges not for all systems. Convergence depends on the system and on the data flow

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller
  - preconditioning to speed up the convergence
- for DAEs:
  - converges not for all systems. Convergence depends on the system and on the data flow
  - step-size controller

Known theory:

- for ODEs:
  - converges for all systems if macro step size is small enough and mild assumptions for the data flow
  - error bounds are known
  - step-size controller
  - preconditioning to speed up the convergence
- for DAEs:
  - converges not for all systems. Convergence depends on the system and on the data flow
  - step-size controller
  - regularization to enforce the convergence

DIRM

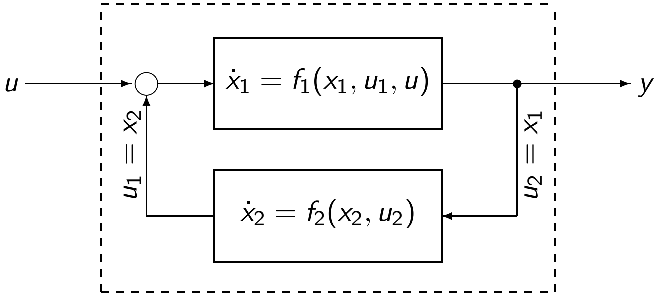
[RATHINAM/PETZOLD '2002]

Idea: combine model reduction with Dynamic Iteration

DIRM

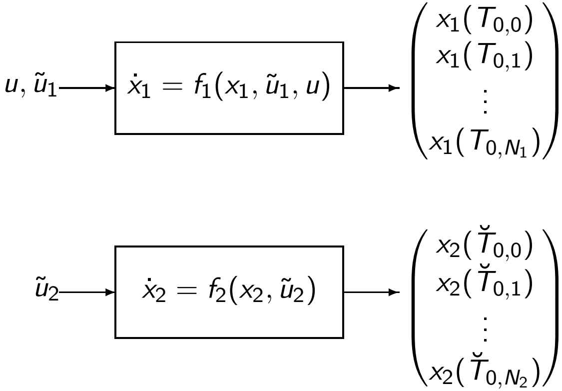
[RATHINAM/PETZOLD '2002]

Idea: combine model reduction with Dynamic Iteration



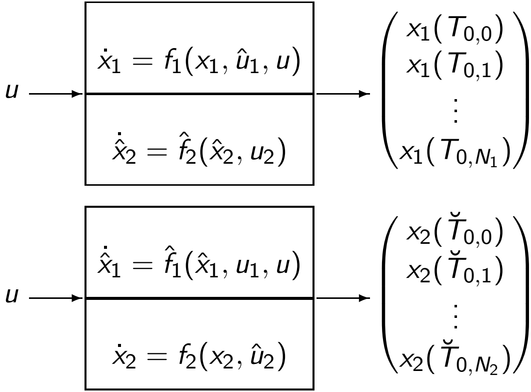
DIRM - 1st macro step - initial step

- approximate $\tilde{u}_1 \equiv x_2(T_0)$ and $\tilde{u}_2 \equiv x_1(T_0)$
- solve the systems separately
- store x_1 and x_2 at all micro steps



Initial step on $[T_0, T_1]$

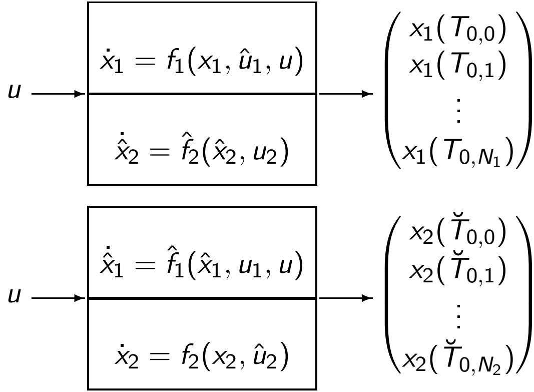
DIRM - 1st macro step - iteration step



Iteration step on $[T_0, T_1]$

DIRM - 1st macro step - iteration step

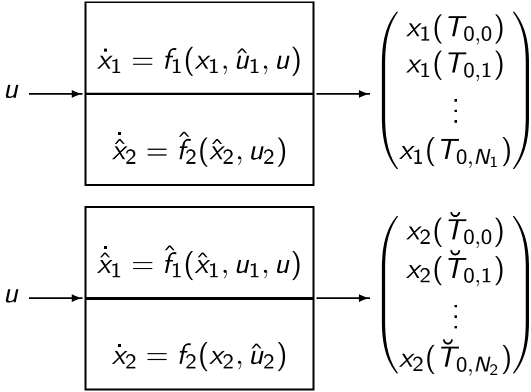
- calculate the reduced-order subsystems from snapshots x_1 and x_2 from the last iteration



Iteration step on $[T_0, T_1]$

DIRM - 1st macro step - iteration step

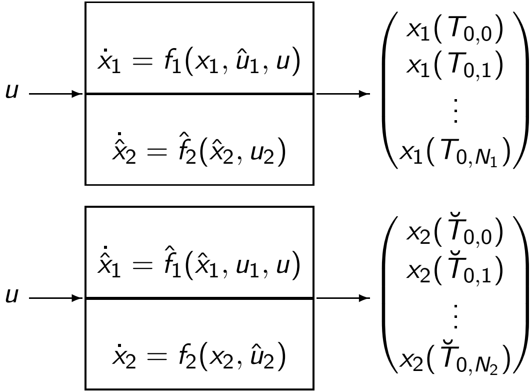
- calculate the reduced-order subsystems from snapshots x_1 and x_2 from the last iteration
- solve every subsystem coupled with other reduced subsystems



Iteration step on $[T_0, T_1]$

DIRM - 1st macro step - iteration step

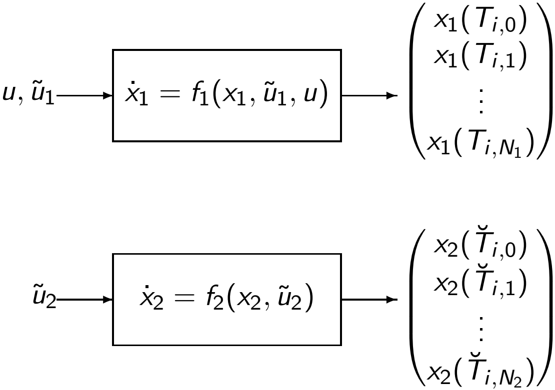
- calculate the reduced-order subsystems from snapshots x_1 and x_2 from the last iteration
- solve every subsystem coupled with other reduced subsystems
- store x_1 and x_2



Iteration step on $[T_0, T_1]$

DIRM - i-th macro step - initial step

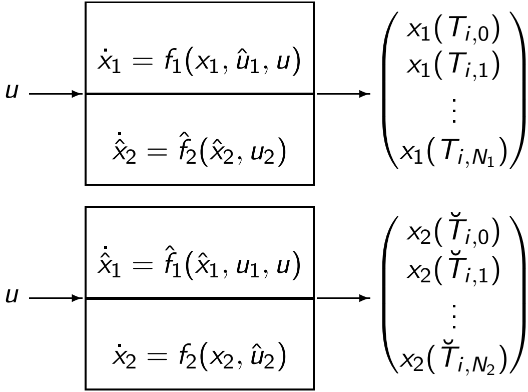
- approximate u_1 and u_2 by extrapolation of x_2 and x_1
- solve the systems separately
- store x_1 and x_2 at all micro steps



Initial step on $[T_i, T_{i+1}]$

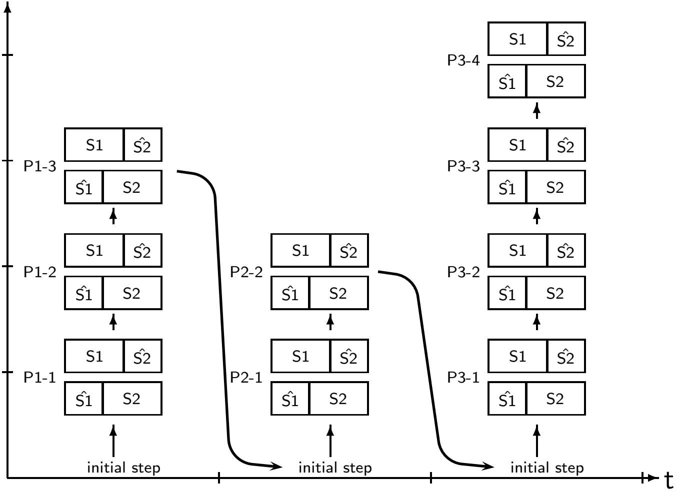
DIRM - i-th macro step - iteration step

- calculate the reduced-order subsystems from snapshots x_1 and x_2 from last iteration
- solve every subsystem coupled with other reduced subsystems
- store x_1 and x_2



Iteration step on $[T_i, T_{i+1}]$

iterations


P1-1
P1-2
P1-3
P2-1
P2-2
P3-1
P3-2
P3-3
P3-4
S1
Ŝ2
Ŝ1
S2
initial step
initial step
initial step
t

Problems:

- convergence proof only for linear systems with “weak coupling”

Problems:

- convergence proof only for linear systems with "weak coupling"
- no error estimates

Problems:

- convergence proof only for linear systems with “weak coupling”
- no error estimates
- no strategy to choose macro step size, number of iterations or reduced dimension

$$\begin{aligned} \dot{x} &= f(x, u) \\ x &\in \mathbb{R}^n,\ u \in \mathbb{R}^m. \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x}, u) \\ \hat{x} &\in \mathbb{R}^k,\ u \in \mathbb{R}^m. \end{aligned}$$

$$\begin{aligned} \dot{x} &= f(x,u) \\ x &\in \mathbb{R}^n,\ u \in \mathbb{R}^m. \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x},u) \\ \hat{x} &\in \mathbb{R}^k,\ u \in \mathbb{R}^m. \end{aligned}$$

Modelreduction by projection.
Let $V \in \mathbb{R}^{n,k}$. Then $x \approx V\hat{x}$ where $\hat{x}$ solves

$$\dot{\hat{x}} = V^T f(V\hat{x}, u)$$

Error in one macro step at one arbitrary iteration

an coupled system with 2 subsystems:

$$\dot{x}_1 = f_1(x_1, x_2), \qquad \dot{x}_2 = f_2(x_1, x_2).$$

Error in one macro step at one arbitrary iteration

an coupled system with 2 subsystems:

$$\dot{x}_1 = f_1(x_1, x_2), \qquad \dot{x}_2 = f_2(x_1, x_2).$$

Start with one iteration step at one macro step i:

$$\begin{aligned} \dot{x}_1^{D1} &= f_1(x_1^{D1}, V_2\hat{x}_2^{D1}), & \dot{\hat{x}}_1^{D2} &= \hat{f}_1(\quad \hat{x}_1^{D2}, x_2^{D2}), \\ \dot{\hat{x}}_2^{D1} &= \hat{f}_2(x_1^{D1}, \quad \hat{x}_2^{D1}), & \dot{x}_2^{D2} &= f_2(V_1\hat{x}_1^{D2}, x_2^{D2}). \end{aligned}$$

Error in one macro step at one arbitrary iteration

an coupled system with 2 subsystems:

$$\dot{x}_1 = f_1(x_1, x_2), \qquad \dot{x}_2 = f_2(x_1, x_2).$$

Start with one iteration step at one macro step i:

$$\begin{aligned} \dot{x}_1^{D1} &= f_1(x_1^{D1}, V_2\hat{x}_2^{D1}), & \dot{\hat{x}}_1^{D2} &= \hat{f}_1(\quad \hat{x}_1^{D2}, x_2^{D2}), \\ \dot{\hat{x}}_2^{D1} &= \hat{f}_2(x_1^{D1}, \quad \hat{x}_2^{D1}), & \dot{x}_2^{D2} &= f_2(V_1\hat{x}_1^{D2}, x_2^{D2}). \end{aligned}$$

Define $e := \begin{pmatrix} x_1 - x_1^{D1} \\ x_2 - x_2^{D2} \end{pmatrix}$.

Error in one macro step at one arbitrary iteration

an coupled system with 2 subsystems:

$$\dot{x}_1 = f_1(x_1, x_2), \qquad \dot{x}_2 = f_2(x_1, x_2).$$

Start with one iteration step at one macro step i:

$$\begin{aligned} \dot{x}_1^{D1} &= f_1(x_1^{D1}, V_2\hat{x}_2^{D1}), & \dot{\hat{x}}_1^{D2} &= \hat{f}_1(\quad \hat{x}_1^{D2}, x_2^{D2}), \\ \dot{\hat{x}}_2^{D1} &= \hat{f}_2(x_1^{D1}, \quad \hat{x}_2^{D1}), & \dot{x}_2^{D2} &= f_2(V_1\hat{x}_1^{D2}, x_2^{D2}). \end{aligned}$$

Define $e := \begin{pmatrix} x_1 - x_1^{D1} \\ x_2 - x_2^{D2} \end{pmatrix}$.

Idea: solve $\frac{d\|e\|}{dt} = \frac{\langle \dot{e}, e \rangle}{\|e\|}$ for $\|e\|$

$$\langle \dot{e}, e \rangle = \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$\langle \dot{e}, e \rangle = \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$= \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) \\ f_2(x_1^{D1}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$+ \left\langle \begin{pmatrix} f_1(x_1^{D1}, x_2^{D1}) \\ f_2(x_1^{D1}, x_2^{D2}) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$\langle \dot{e}, e \rangle = \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$= \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) \\ f_2(x_1^{D1}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$
$$+ \left\langle \begin{pmatrix} f_1(x_1^{D1}, x_2^{D1}) \\ f_2(x_1^{D1}, x_2^{D2}) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$

$$\leqslant \left\langle \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} - \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) \\ f_2(x_1^{D1}, x_2^{D2}) \end{pmatrix}, e \right\rangle$$
$$+ \left\| \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) - f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(x_1^{D1}, x_2^{D2}) - f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix} \right\| \|e\|$$

Logarithmic constant:

$$L_G[f] := \sup_{x \neq y \in \mathbb{R}^d} \frac{\langle x - y, f(x) - f(y) \rangle}{\|x - y\|^2}$$

Logarithmic constant:

$$L_G[f] := \sup_{x \neq y \in \mathbb{R}^d} \frac{\langle x - y, f(x) - f(y) \rangle}{\|x - y\|^2}$$

Approximation the log-constant by the log-constant of the Jacobian. [WIRTZ/SORENSEN/HAASDONK '2012]

$$L_G[J(f)] = \sup_{\mathbb{R}^d \setminus \{0\}} \frac{\langle x, J(f)x \rangle}{\langle x, x \rangle} = \lambda_{\max}\left(\frac{1}{2}(J(f) + J(f)^T)\right)$$

Logarithmic constant:

$$L_G[f] := \sup_{x \neq y \in \mathbb{R}^d} \frac{\langle x - y, f(x) - f(y) \rangle}{\|x - y\|^2}$$

Approximation the log-constant by the log-constant of the Jacobian. [WIRTZ/SORENSEN/HAASDONK '2012]

$$L_G[J(f)] = \sup_{\mathbb{R}^d \setminus \{0\}} \frac{\langle x, J(f)x \rangle}{\langle x, x \rangle} = \lambda_{\max} \left(\frac{1}{2}(J(f) + J(f)^T) \right)$$

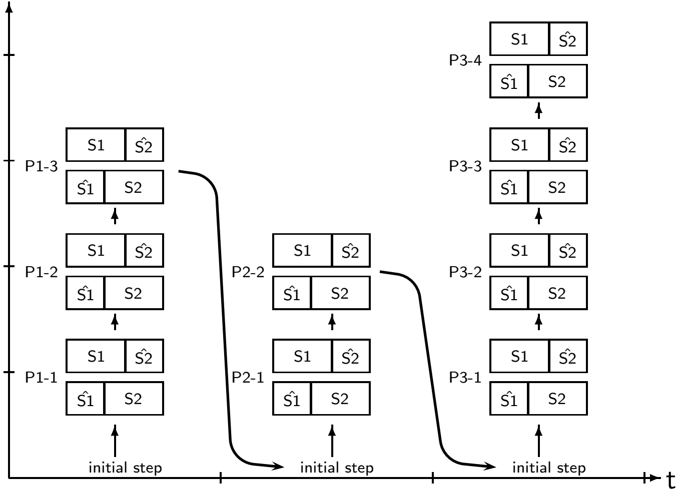
We can sum this up with

$$\frac{d\|e\|}{dt} \leqslant \alpha \|e\| + \beta$$

with

$$\alpha = L_G[J(f)](x_1^{D1}, x_2^{D2}),$$
$$\beta = \left\| \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) - f_1(x_1^{D1}, V_2 \hat{x}_2^{D1}) \\ f_2(x_1^{D1}, x_2^{D2}) - f_2(V_1 \hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix} \right\|.$$

iterations


S1
Ŝ2
Ŝ1
S2
P1-1
P1-2
P1-3
P2-1
P2-2
P3-1
P3-2
P3-3
P3-4
initial step
initial step
initial step
t

Error for DIRM

$\|e(T_0)\| = 0$;

for *every macro step* **do**

in the last iteration solve

$$\|\dot{e}\| = \alpha\|e\| + \beta,$$

with

$$\alpha = L_G[J(f)](x_1^{D1}, x_2^{D2}),$$

$$\beta = \left\| \begin{pmatrix} f_1(x_1^{D1}, x_2^{D2}) - f_1(x_1^{D1}, V_2\hat{x}_2^{D1}) \\ f_2(x_1^{D1}, x_2^{D2}) - f_2(V_1\hat{x}_1^{D2}, x_2^{D2}) \end{pmatrix} \right\|.$$

end

example:

$$\begin{aligned} \dot{x} &= \nu \Delta x + a x \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\ x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\ x(0, s) &= x_0 \end{aligned}$$

example:

$$\begin{aligned} \dot{x} &= \nu \Delta x + ax \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\ x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\ x(0, s) &= x_0 \end{aligned}$$

discretize and split in two systems:

$$\dot{x}_1 = Ax_1 + f(x_1) + b_1(x_1, x_2) \quad \dot{x}_2 = Ax_2 + f(x_2) + b_2(x_1, x_2)$$

example:

$$
\begin{aligned}
\dot{x} &= \nu \Delta x + a x \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\
x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\
x(0, s) &= x_0
\end{aligned}
$$

discretize and split in two systems:

$$
\dot{x}_1 = A x_1 + f(x_1) + b_1(x_1, x_2) \quad \dot{x}_2 = A x_2 + f(x_2) + b_2(x_1, x_2)
$$

- dimension of subsystems: 50

example:

$$\begin{aligned} \dot{x} &= \nu \Delta x + a x \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\ x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\ x(0, s) &= x_0 \end{aligned}$$

discretize and split in two systems:

$$\dot{x}_1 = A x_1 + f(x_1) + b_1(x_1, x_2) \quad \dot{x}_2 = A x_2 + f(x_2) + b_2(x_1, x_2)$$

- dimension of subsystems: 50
- reduced dimensions: 2

example:

$$\begin{aligned} \dot{x} &= \nu \Delta x + ax \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\ x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\ x(0, s) &= x_0 \end{aligned}$$

discretize and split in two systems:

$$\dot{x}_1 = Ax_1 + f(x_1) + b_1(x_1, x_2) \quad \dot{x}_2 = Ax_2 + f(x_2) + b_2(x_1, x_2)$$

- dimension of subsystems: 50
- reduced dimensions: 2
- number of iterations: 3

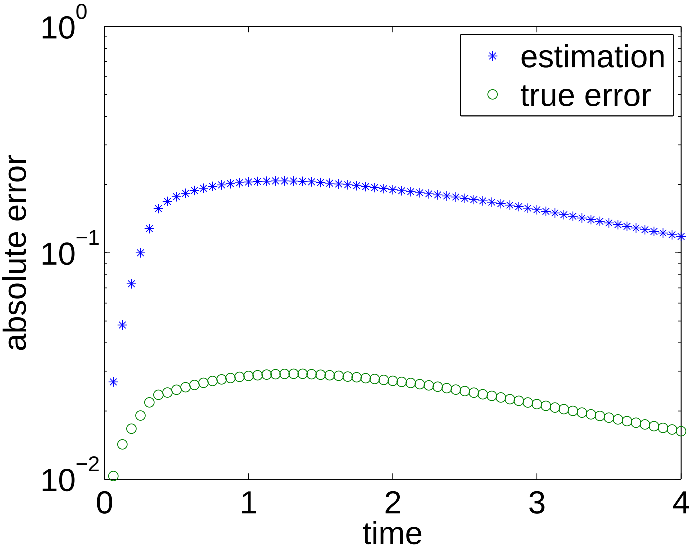
example:

$$\begin{aligned} \dot{x} &= \nu \Delta x + a x \cdot x_s && \text{in } (\Omega_1 \cup \Omega_2) \times [0, T] \\ x &= 0 && \text{on } \partial(\Omega_1 \cup \Omega_2) \times [0, T] \\ x(0, s) &= x_0 \end{aligned}$$

discretize and split in two systems:

$$\dot{x}_1 = Ax_1 + f(x_1) + b_1(x_1, x_2) \quad \dot{x}_2 = Ax_2 + f(x_2) + b_2(x_1, x_2)$$

- dimension of subsystems: 50
- reduced dimensions: 2
- number of iterations: 3
- $-0.3 \leqslant L_G(J(f)) \leqslant 0.01$


estimation
true error
absolute error
time
10^0
10^{-1}
10^{-2}
0
1
2
3
4

Conclusion

Presented:

- Dynamic Iteration

Conclusion

Presented:

- Dynamic Iteration
- DIRM

Conclusion

Presented:

- Dynamic Iteration
- DIRM
- a posteori error estimate

Conclusion

Presented:

- Dynamic Iteration
- DIRM
- a posteori error estimate

Future work:

- cheap computation of the error estimator (employing the structure of f? DEIM?)

Conclusion

Presented:

- Dynamic Iteration
- DIRM
- a posteori error estimate

Future work:

- cheap computation of the error estimator (employing the structure of f? DEIM?)
- conditions for convergence of DIRM

Conclusion

Presented:

- Dynamic Iteration
- DIRM
- a posteori error estimate

Future work:

- cheap computation of the error estimator (employing the structure of f? DEIM?)
- conditions for convergence of DIRM
- strategy to choose macro step size, reduced dimensions and number of DIRM iterations