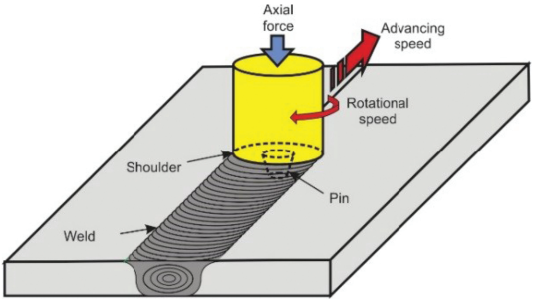
Fast Computation of Friction Stir Welding Process with Model Order Reduction and Machine Learning

Joshua Kay

Research Mentor: Dr. Zilong Song

2023 Student Research Symposium

Friction Stir Welding (FSW)



. **Advantages:**

- Joining method is largely defect free
- No filler materials
- Environmentally friendly (no fumes)
- Automatable and reproducible

Friction Stir Welding

There are many operational parameters in the FSW process.

- Rotational Speed
- Applied Axial Force
- Tool Shoulder Radius
- Rotational and Advancing Speeds
- etc...

Problem:
Finding optimal parameters to avoid product defects in the joints is time consuming. **Fast Simulation is Required!**

Fast Simulation Approach

- FSW process
  - Modeled by coupled system of 2D Navier Stokes and Heat equations [1, 2].
- Numerical modeling
  - Extremely time consuming due to the complexity and non-linearity of the equations.
- Reduced order modeling (ROM)
  - Proper Orthogonal Decomposition (POD) [3].
  - Discrete Empirical Interpolation Method (DEIM) [4, 5].
- Machine learning
  - Predict solutions to FSW process when operating parameters vary, etc...

The ROM techniques will now be used on a simple example.

2D Heat Equation Example

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + u^2 \tag{1}$$

$$u(0, y, t) = u(a, y, t) = 0$$
$$u(x, 0, t) = u(x, b, t) = 0$$
$$u(x, y, 0) = \begin{cases} 50 \text{ if } y < 1 \\ 0 \text{ if } y \geq 1 \end{cases}$$

Where $0 \leq a \leq 2$ and $0 \leq b \leq 2$.

Discretization of equation (1):

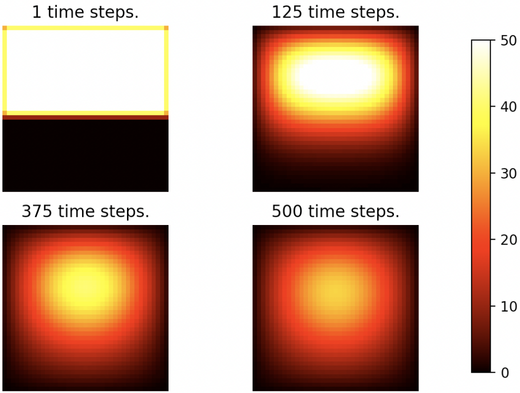
$$\vec{U}^{k+1} = A\vec{U}^{k} + \Delta t(\vec{U}^{k})^2 \tag{2}$$

$$\vec{U}^{k+1}, \vec{U}^{k} \in \mathbb{R}^{N} \text{ and } A \in \mathbb{R}^{NxN}$$

.

Numerical Results:

The dynamics are modeled up to time $t = .25$ on a 39x39 grid with 500 time steps.



Computationally expensive as each time step requires N-dimensional computations.

POD Implementation:

Goal: Construct an optimal, low dimensional, basis matrix Φ_r to approximate $\vec{U}^k$ as:

$$\vec{U}^k \approx \Phi_r \vec{a}^k$$

$$\Phi_r \in \mathbb{R}^{Nxr},\ \vec{a}^k \in \mathbb{R}^r,\ r << N$$

Step 1) Start by constructing snapshot matrix $X \in \mathbb{R}^{NxL}$:

$$X = \begin{bmatrix} | & | & & | \\ \vec{U}^1 & \vec{U}^2 & \dots & \vec{U}^L \\ | & | & & | \end{bmatrix}$$

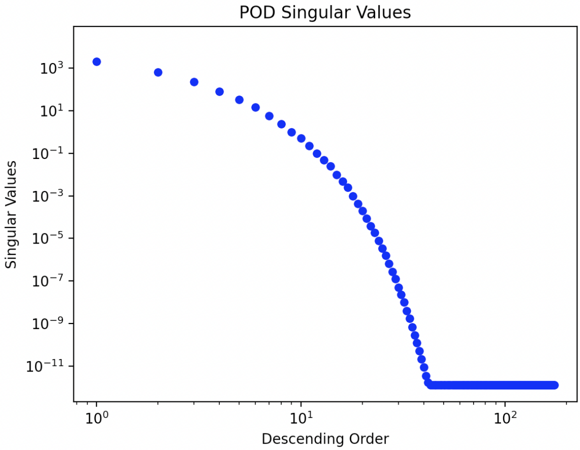
With $L < m$. m is the number of time-steps.

Step 2) Compute the SVD of X:

$$X = \Phi \Sigma V^T$$

. $\Phi \in \mathbb{R}^{NxN}$ is an orthogonal matrix that holds the spatial dynamics of the problem.

Singular Values:



. **Step 3)** Truncate Φ to its first r columns. This gives $\Phi_r \in \mathbb{R}^{Nxr}$.

POD Implementation:

Applying the POD method to equation (1). This gives us:

$$
\begin{aligned}
\vec{U}^{k+1} &= A\vec{U}^{k} + \Delta t(\vec{U}^{k})^2 \\
\Phi_r \vec{a}^{k+1} &= A\Phi_r \vec{a}^{k} + \Delta t(\Phi_r \vec{a}^{k})^2 \\
\vec{a}^{k+1} &= \underbrace{\Phi_r^T A \Phi_r}_{(rxr)} \vec{a}^{k} + \Delta t \Phi_r^T (\Phi_r \vec{a}^{k})^2
\end{aligned}
$$

.

Each $\vec{a}^{k} \in \mathbb{R}^r$ can be converted back to *n*-dimensional space by computing:

$$\Phi_r \vec{a}^{k} \in \mathbb{R}^n$$

$\Phi_r^T A \Phi_r$ can be computed once and re-used. Non-linear term still high dimensional. This problem motivates DEIM.

DEIM and POD Implementation:

Approximate the non-linearity with the DEIM approximation:

$$\begin{aligned} N(\vec{U}^k) &\approx \Xi_p (P^T \Xi_p)^{-1} P^T N(\vec{U}^k) \\ &= \Xi_p (P^T \Xi_p)^{-1} N(P^T \vec{U}^k) \end{aligned}$$

$N(\vec{U}^k)$ non-linear function.

$\Xi_p \in \mathbb{R}^{Nxp}$ is the optimal low rank basis for the non-linearity.

$P \in \mathbb{R}^{Nxp}$ is an interpolation matrix.

$p << N$, $p \sim r$

Non-linear approximation applied to the POD method:

$$\Phi_r^T N(\Phi_r \vec{a}^k) \approx \underbrace{\Phi_r^T \Xi_p (P^T \Xi_p)^{-1}}_{(rxr)} N(\underbrace{(P^T \Phi_r)}_{(rxr)} \vec{a}^k)$$

DEIM and POD Implementation:

To construct Ξ_p and P:

Step 1) Compute snapshot $\mathbf{N} \in \mathbb{R}^{Nxl}$ of non-linearity:

$$\mathbf{N} = \begin{bmatrix} | & | & & | \\ N(\vec{U}^1) & N(\vec{U}^2) & \dots & N(\vec{U}^l) \\ | & | & & | \end{bmatrix}$$

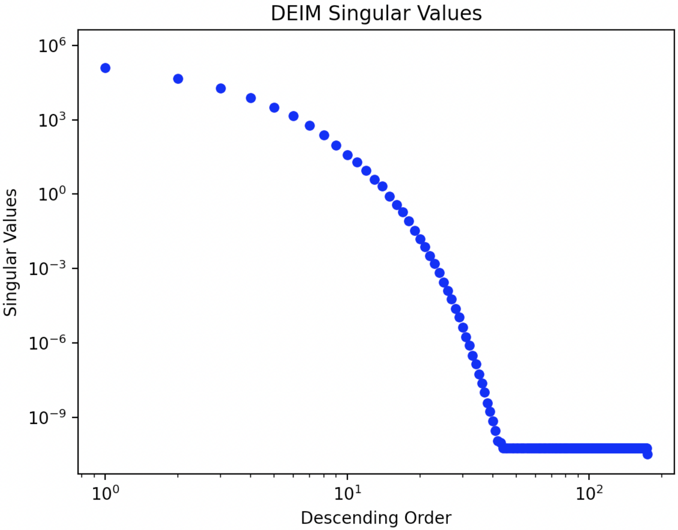
with $l < m$.

Step 2) Compute the SVD of $\mathbf{N}$:

$$\mathbf{N} = \Xi \hat{\Sigma} \hat{V}^T$$

. where $\Xi \in \mathbb{R}^{NxN}$.

DEIM and POD Implementation:



DEIM and POD Implementation:

Let p denote the number of dominant singular values and their corresponding column vectors in Ξ.

Step 3) Truncate Ξ to its first p columns which provides $\Xi_p \in \mathbb{R}^{Nxp}$.

Step 4) The interpolation matrix P is constructed through the DEIM algorithm by using as input Ξ_p.

DEIM and POD Implementation:

The DEIM/POD method applied to equation (2) is:

$$\bar{a}^{k+1} = \underbrace{\Phi_r^T A \Phi_r}_{(rxr)} \bar{a}^k + \underbrace{\Phi_r^T \Xi_p (P^T \Xi_p)^{-1}}_{(rxr)} ((\underbrace{P^T \Phi_r}_{(rxr)}) \bar{a}^k)^2$$

- Every computation is low-dimensional resulting in massive time savings.
- POD and DEIM can be applied to a variety of problems to reduce computational time while preserving accuracy.

Errors:

$$||U - E|| = max_k \sqrt{\Delta x \Delta y \sum_{i=1}^{N} |\vec{U}_i^k - \vec{E}_i^k|^2}, \text{ with } 0 \leq k \leq m$$

.

ROM Dimensions	Maximum Error	
	||U - U_POD||	||U_POD - U_DEIM||
r = 10, p = 10	0.00199	4.07157E-06
r = 20, p = 20	0.00051	5.94124E-07
r = 30, p = 30	0.00021	2.61611E-07
r = 40, p = 40	5.54206E-05	1.90929E-07

.

Timing Study of Similar Problem:

	Offline (s)	Online (s)
Full Model	—	85.060
POD Model	4.233	39.266
DEIM Model	11.506	0.306

- Offline time is the time required to construct the basis and interpolation matrices.
- Online time is the time required to complete numerical scheme.

Future Work:

- Apply ROM techniques to heat transfer model for FSW.
- Conduct literature review of additional machine learning approaches suitable for the FSW process.
- Incorporate ROM techniques with machine learning approaches to predict solutions in FSW process when operating parameters vary.

References

[1] Xiulei Cao, Kirk Fraser, Zilong Song, Chris Drummond, and Huaxiong Huang. Ma- chine learning and reduced order computation of a friction stir welding model. Jour- nal of Computational Physics, 454:110863, 2022.

[2] R Nandan, GG Roy, and T Debroy. Numerical simulation of three-dimensional heat transfer and plastic flow during friction stir welding. Metallurgical and materials transactions A, 37(4):1247–1259, 2006.

[3] Steven L Brunton and J Nathan Kutz. Data-driven science and engineering: Ma- chine learning, dynamical systems, and control. Cambridge University Press, 2022.

[4] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An ˚Æempirical interpolation˚Ømethod: application to efficient reduced-basis discretiza- tion of partial differential equations. Comptes Rendus Mathematique, 339(9):667– 672, 2004.

[5] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. SIAM Journal on Scientific Computing, 32(5):2737–2764, 2010.