How can Intervals Help with Large Numerical Simulations? CoProD'16

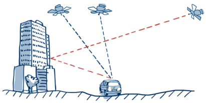
Martine Ceberio

Computer Science Department
The University of Texas at El Paso
mceberio@utep.edu

MOTIVATION

Dynamical systems

model evolving phenomena. Using them allows to model, understand, and control many physical phenomena:



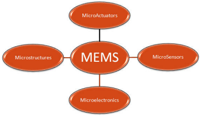


Figure: Heat transfer, signal propagation, wave propagation, and mems systems

Dynamical systems

model evolving phenomena. Using them allows to model, understand, and control many physical phenomena:

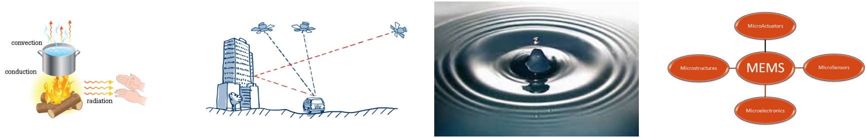


Figure: Heat transfer, signal propagation, wave propagation, and mems systems

Why are Dynamical Systems hard to Handle?

Not all dynamical systems have an analytical solution. So we simulate them...

EXAMPLE

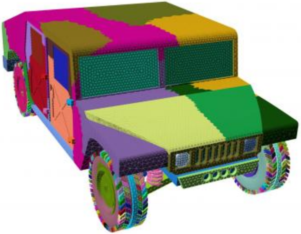
The Lotka-Volterra problem involves a model of a predator-prey system

$$\begin{cases} y_1' = \theta_1 y_1 (1 - y_2), & y_1(0) = 1.2 & \theta_1 = 3, \\ y_2' = \theta_2 y_2 (y_2 - 1), & y_2(0) = 1.1 & \theta_2 = 1 \end{cases}$$

No analytic solution is available.

NUMERICAL SIMULATIONS: OUR ORIGINAL GOAL

ARL problem: Underbody Blasts effet on vehicles



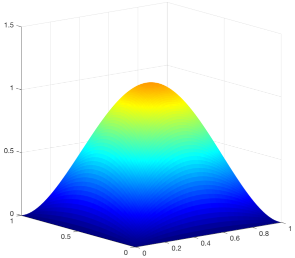
NUMERICAL SIMULATIONS: EXAMPLES

Let us consider the Bratu's equation, which is an example of a PDE in combustion theory. We look at it in the plane (2D):

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + re^u = 0.$$

$(x, y)^T \in \Omega = [0, 1] \times [0, 1]$, $u(x, y) = 0$, if $(x, y) \in \partial\Omega$. The stepsizes $\Delta x = \Delta y = 0.01$, and the parameter $r \in [1, 6.78]$

$r = 6.6176$



NUMERICAL SIMULATIONS: WHAT ARE WE SOLVING?

- They are systems of equations, often nonlinear.

What is challenging about such simulations?

NUMERICAL SIMULATIONS: WHAT ARE WE SOLVING?

- They are systems of equations, often nonlinear.

What is challenging about such simulations?

- They are often **nonlinear** equations.

NUMERICAL SIMULATIONS: WHAT ARE WE SOLVING?

- They are systems of equations, often nonlinear.

What is challenging about such simulations?

- They are often **nonlinear** equations.
- They can be **very large**.

$$\begin{aligned} \{\text{need for accuracy} \rightarrow \text{more details in the modeling}\} &\quad\rightarrow\quad \text{larger dynamical systems} \\ \text{larger dynamical systems} &\quad\rightarrow\quad \text{unmanageably large demands on computational resources} \end{aligned}$$

NUMERICAL SIMULATIONS: WHAT ARE WE SOLVING?

- They are systems of equations, often nonlinear.

What is challenging about such simulations?

- They are often **nonlinear** equations.
- They can be **very large**.

$$\begin{aligned} \{\text{need for accuracy} \rightarrow \text{more details in the modeling}\} &\rightarrow \text{larger dynamical systems} \\ \text{larger dynamical systems} &\rightarrow \text{unmanageably large demands on computational resources} \end{aligned}$$

- **Many simulations** need to be run.

ADDRESSING THESE CHALLENGES

- **Nonlinear Systems**

ADDRESSING THESE CHALLENGES

- **Nonlinear Systems**
  - Newton-like solving approaches
  - Interval constraint solving techniques

ADDRESSING THESE CHALLENGES

- **Nonlinear Systems**
  - Newton-like solving approaches
  - Interval constraint solving techniques
- **Size of Systems to be solved and Need for many Simulations**

ADDRESSING THESE CHALLENGES

- **Nonlinear Systems**
  - Newton-like solving approaches
  - Interval constraint solving techniques
- **Size of Systems to be solved and Need for many Simulations**
  - Reduction of the Size of the Problem

SOLVING NONLINEAR SYSTEMS WITH NEWTON

For a fixed λ, we solve

$$F(x) = R(x, \lambda) = 0$$

SOLVING NONLINEAR SYSTEMS WITH NEWTON

For a fixed λ, we solve

$$F(x) = R(x, \lambda) = 0$$

Newton's method

- Set $i = 0$
- Guess an approximation of the solution u^0
- **Repeat**
  - Compute $J(u^i)$, Jacobian of F, and $F(u^i)$
  - Solve the linear system $J(u^i)\Delta u = -F(u^i)$
  - Set $u^{i+1} = u^i + \Delta u$
  - Set $i = i + 1$
- **Until convergence**

Branch-and-Bound

It is the underlying principle of search in interval constraint solving techniques and it allows to guarantee completeness of the search.

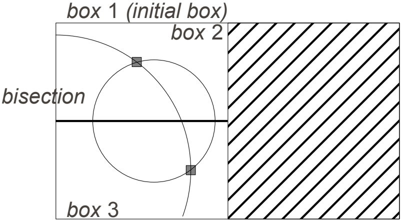


Figure taken from Laurent Granvilliers, RealPaver User's Manual.

Algorithm

Input: System of constraints $C = \{c_1, \ldots, c_k\}$, a search space D_0.
Output: A set Sol of interval solutions

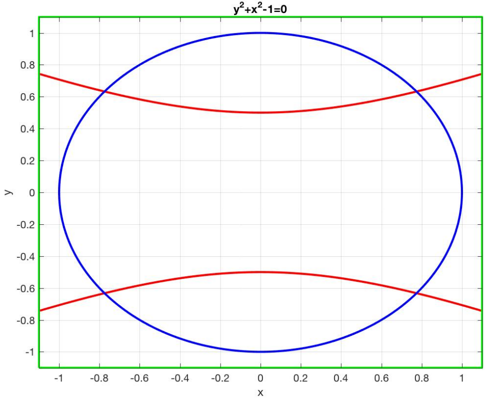
Set Sol to empty
If $\forall i$, $0 \in F_i(D_0)$ then:
    Store D_0 in some storage S
    While (S is not empty) do:
        Take D out of S
        If ($\forall i$, $0 \in F_i(D)$) then:
            If (D is still too large) then:
                Split D in D_1 and D_2
                Store D_1 and D_2 in S
            Else:
                Store D in Sol
Return Sol

EXAMPLE OF A B&B-BASED APPROACH

Let us suppose that we want to solve the following nonlinear system of equations:

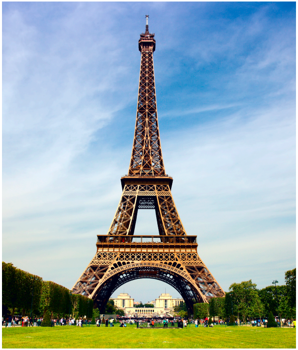
$$\begin{cases} (2y)^2 - x^2 & = & 1 \\ y^2 + x^2 & = & 1 \end{cases}$$

We are in fact going to use B&B but with additional pruning.



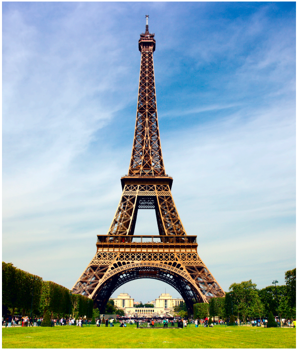
PROBLEMS ARE TOO LARGE?

If we can not take this home:



PROBLEMS ARE TOO LARGE?

If we can not take this home:



At least, we can take this:

HOW WOULD THAT WORK?

Given a function

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

Solving a nonlinear system of equations based on F consists in finding $x \in \mathbb{R}^n$ such that:

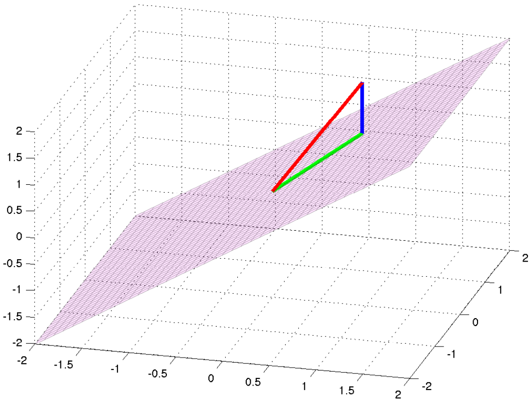
$$F(x) = 0$$

Model-Order Reduction (MOR) is typically performed on the premise that the solution x belongs to an affine subspace, W, of $\mathbb{R}^n$ whose dimension k is orders of magnitude smaller than n, i.e.,

$$x = z + \Phi p$$

where Φ is a basis of a subspace of $\mathbb{R}^n$ associated to W

ILLUSTRATION OF MOR



$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$\begin{aligned} F(X) &= 0 \\ F(\Phi p + z) &= 0 \end{aligned}$$

Assuming

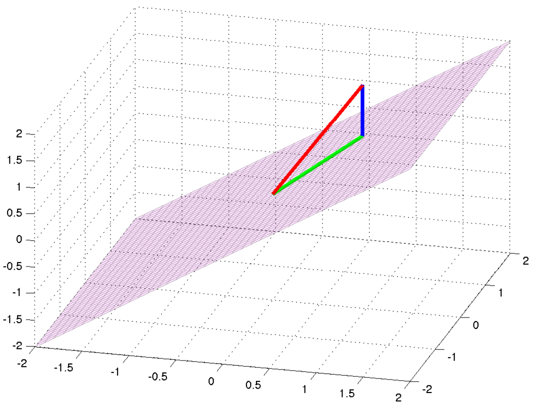
$$z \approx 0$$

we have to solve:

$$F(\Phi p) = 0$$

$$p = \arg\min \frac{1}{2} \|F(\Phi p)\|^2$$

ILLUSTRATION OF MOR



$$F : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$\begin{aligned} F(X) &= 0 \\ F(\Phi p + z) &= 0 \end{aligned}$$

Assuming

$$z \approx 0$$

we have to solve:

$$F(\Phi p) = 0$$

$$p = \arg\min \frac{1}{2} ||F(\Phi p)||^2$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \\ \phi_{31} & \phi_{32} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

USING Φ TO REDUCE THE PROBLEM

It is important to understand that, at this point, it does not matter where Φ is coming from.
For a fixed λ, we solve

$$F(x) = R(x, \lambda) = 0$$

For Full-Order Model, we use:

Newton's method

- Set $i = 0$
- Guess an approximation of the solution u^0
- **Repeat**
  - Compute $J(u^i)$, Jacobian of F, and $F(u^i)$
  - Solve the linear system $J(u^i)\Delta u = -F(u^i)$
  - Set $u^{i+1} = u^i + \Delta u$
  - Set $i = i + 1$
- **Until convergence**

For Reduced-Order Model, we use:

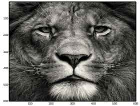
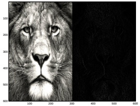
Reduced Newton's method

- Set $i = 0$
- Guess an approximation of the solution p^0
- **Repeat**
  - Compute $J(\Phi p^i)$ and $F(\Phi p^i)$
  - Solve the linear system $J(\Phi p^i)\phi\Delta p = -F(\Phi p^i)$
  - Set $p^{i+1} = p^i + \Delta p$
  - Set $i = i + 1$
- **Until convergence**

HOW TO GO ABOUT MOR?

There are several techniques of MOR, but all of them have the same goal: finding a basis Φ. For example:

1. Krylov method: $\Phi = \mathcal{K}_k(A, b) = \{b,\ Ab\ , A^2 b, \cdots, A^{k-1} b\}$
2. Wavelets method: Let $W = \begin{pmatrix} L \\ H \end{pmatrix}$ a discrete wavelet: $\Phi = L^T$

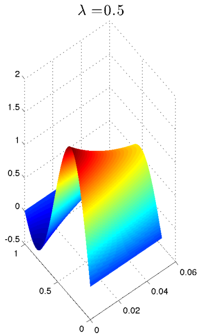



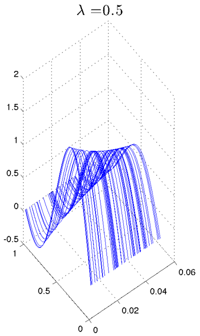
3. Proper Orthogonal Decomposition (POD): Based on Principal Component Analysis (PCA), which is a procedure for identifying a smaller number of uncorrelated variables, called “principal components”, from a large set of data. The goal of PCA is to describe the maximum amount of variance with the fewest number of principal components.

ILLUSTRATION OF THE POD METHOD: SNAPSHOTS

Collect snapshots, i.e.,

1. Solve $R(x, \lambda) = 0$ for different λ values (input)
2. Save the snapshots, i.e., the solution $x(\lambda)$.
3. Organize them in a “snapshots” matrix



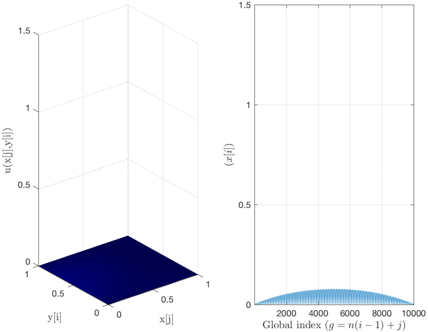


EXAMPLE: BRATU 2D

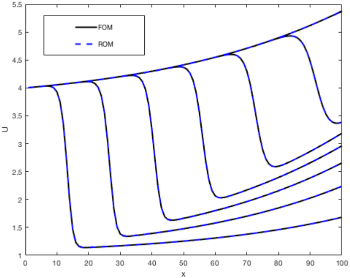
Let us retake the Bratu's equation, which is an example of a PDE in combustion theory, but this time defined in the plane

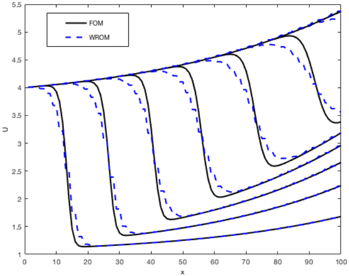
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + re^u = 0$$

$(x, y)^T \in \Omega = [0, 1] \times [0, 1]$, $u(x, y) = 0$, if $(x, y) \in \partial\Omega$. The stepsizes $\Delta x = \Delta y = 0.01$
Collecting Snapshots: The problem is solved 100 times for $r \in [1, 6.78]$



COMPARING POD AND WAVELETS





What is really good about MOR?

- Ability to simulate very large systems with a fraction of the computational power
  - A good candidate to be run in environments with limited resources

MOR IN SHORT

What is really good about MOR?

- Ability to simulate very large systems with a fraction of the computational power
  - A good candidate to be run in environments with limited resources
- Good quality of solutions with (good) POD

MOR IN SHORT

What is really good about MOR?

- Ability to simulate very large systems with a fraction of the computational power
  - A good candidate to be run in environments with limited resources
- Good quality of solutions with (good) POD

What can be improved?

- The need for carefully chosen snapshots in POD can be seen as a limitation

What is really good about MOR?

- Ability to simulate very large systems with a fraction of the computational power
  - A good candidate to be run in environments with limited resources
- Good quality of solutions with (good) POD

What can be improved?

- The need for carefully chosen snapshots in POD can be seen as a limitation
- Uncertainty in the model is not taken into account

MOR IN SHORT

What is really good about MOR?

- Ability to simulate very large systems with a fraction of the computational power
  - A good candidate to be run in environments with limited resources
- Good quality of solutions with (good) POD

What can be improved?

- The need for carefully chosen snapshots in POD can be seen as a limitation
- Uncertainty in the model is not taken into account
- *As they are designed so far, ROM systems can only be used for simulations, not recognition*

ADDED CHALLENGES

Following on our project with the Army, additional problems came up:

- Need to handle uncertainty
- Need to predict behaviors / phenomena
- Need to avoid certain situations
- Need to adjust missions “on the fly”

INTERVAL POD (I-POD)

Let us recall the problem we are solving

$$R(x,\lambda) = 0, \qquad \lambda \in \mathbf{I}$$

POD method

$$\begin{aligned} R(x,\lambda_1) &= 0, \\ R(x,\lambda_2) &= 0, \\ \vdots \quad & \quad \vdots \\ R(x,\lambda_n) &= 0, \end{aligned}$$

where $\lambda_i \in \mathbf{I}$, for $i = 1, 2, \ldots, n$

I-POD method

we propose an **interval version of POD**

$$R(x,\mathbf{I}) = 0$$

NUMERICAL RESULTS

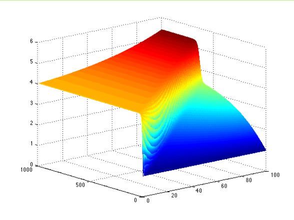
Consider the Burgers' equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{1}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\,\mathtt{exp}(0.02x)$. The initial and boundary conditions used with the above PDE are: $U(x;0) \equiv 1$; $U(0;t) = \lambda$, for all $x \in [0;100]$, and $t > 0$.

POD method

We solve the Burgers' equation for $\lambda_i \in [3.5, 4.5]$



NUMERICAL RESULTS

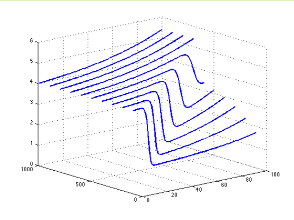
Consider the Burgers' equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{2}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\,\mathtt{exp}(0.02x)$. The initial and boundary conditions used with the above PDE are: $U(x;0) \equiv 1$; $U(0;t) = \lambda$, for all $x \in [0;100]$, and $t > 0$.

POD method

We solve the Burgers' equation for $\lambda_i \in [3.5, 4.5]$



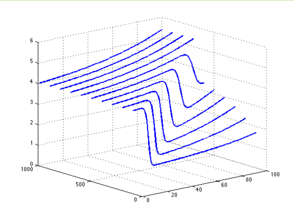
Consider the Burgers' equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \qquad (2)$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\,\mathtt{exp}(0.02x)$. The initial and boundary conditions used with the above PDE are: $U(x;0) \equiv 1$; $U(0;t) = \lambda$, for all $x \in [0;100]$, and $t > 0$.

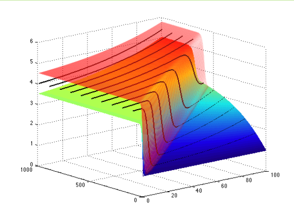
POD method

We solve the Burgers' equation for $\lambda_i \in [3.5, 4.5]$



I-POD method

We solve the Burgers' equation for $\lambda_i = [3.5, 4.5]$



NUMERICAL RESULTS

Method	Tag 1	Tag 2	Tag 3	Tag 4
POD	300 secs	37	0.75 secs	$4.85\mathrm{E}-4$
I-POD	94.45	36	0.75 secs	$5.76\mathrm{E}-4$

Tag 1: Time computing the Reduced basis

Tag 2: Dimension of the Subspace

Tag 3: Time solving Burgers' equation using the obtained basis

Tag 4: $||u_{\mathrm{fom}} - u_{\mathrm{rom}}||/||u_{\mathrm{fom}}||$

EXAMPLE OF I-POD: BRATU 2D

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + re^u = 0$$

$(x, y)^T \in \Omega = [0, 1] \times [0, 1]$, $u(x, y) = 0$, if $(x, y) \in \partial\Omega$. The stepsizes $\Delta x = \Delta y = 0.01$

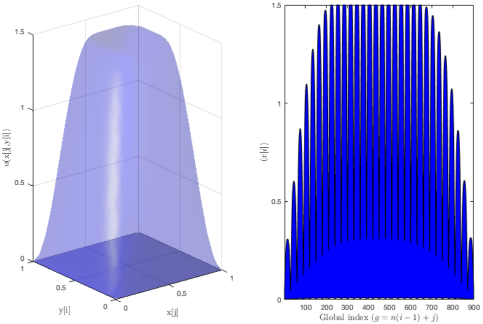
Collecting Snapshots: The problem is solved once using ICST for $r = [1, 6.78]$

```
x2 - 4*x1 + x31 + (r*exp(x1))/961=0,
x1 - 4*x2 + x3 + x32 + (r*exp(x2))/961=0,
x2 - 4*x3 + x4 + x33 + (r*exp(x3))/961=0,
x3 - 4*x4 + x5 + x34 + (r*exp(x4))/961=0,
x4 - 4*x5 + x6 + x35 + (r*exp(x5))/961=0,
x5 - 4*x6 + x7 + x36 + (r*exp(x6))/961=0,
x6 - 4*x7 + x8 + x37 + (r*exp(x7))/961=0,
x7 - 4*x8 + x9 + x38 + (r*exp(x8))/961=0,
x8 - 4*x9 + x10 + x39 + (r*exp(x9))/961=0,
x9 - 4*x10 + x11 + x40 + (r*exp(x10))/961=0,
x10 - 4*x11 + x12 + x41 + (r*exp(x11))/961=0,
x11 - 4*x12 + x13 + x42 + (r*exp(x12))/961=0,
x12 - 4*x13 + x14 + x43 + (r*exp(x13))/961=0,
x13 - 4*x14 + x15 + x44 + (r*exp(x14))/961=0,
x14 - 4*x15 + x16 + x45 + (r*exp(x15))/961=0,
x15 - 4*x16 + x17 + x46 + (r*exp(x16))/961=0,
x16 - 4*x17 + x18 + x47 + (r*exp(x17))/961=0
```

EXAMPLE OF I-POD: BRATU 2D

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + re^u = 0$$

$(x, y)^T \in \Omega = [0, 1] \times [0, 1]$, $u(x, y) = 0$, if $(x, y) \in \partial\Omega$. The stepsizes $\Delta x = \Delta y = 0.01$
Collecting Snapshots: The problem is solved once using ICST for $r = [1, 6.78]$



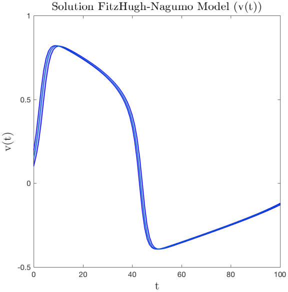
EXAMPLE OF I-POD: FITZ-HUGH NAGUMO

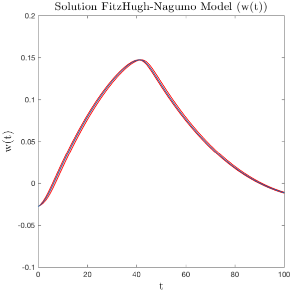
The following nonlinear model is based on the classical FitzHugh-Nagumo oscillator.

$$f(v) = v(v-\alpha)(1-v)$$

$$\begin{cases} \dfrac{dv}{dt} = f(v+v_{eq}) - f(v_{eq}) - w \\[2ex] \dfrac{dw}{dt} = \varepsilon(v-\gamma w) \end{cases}$$

- $\alpha = 0.139$, $\varepsilon = 0.008$, $\gamma = 2.54$
- $v_0 = v_{eq} = 0.15$, $w_0 = w_{eq} = -0.028$
- $t = [0, 10]$ and $\Delta_t = 0.1$
- Uncertainty in $v_0 = [0.1, 0.2]$





FHN. POD VS I-POD

Pr.	Dimension			Precision. Relative Error	
	FOM	RomPOD	RomI-POD	RomPOD	RomI-POD
FHN	200	3	3	0.0208	0.0118

I-POD IN SHORT

- The ability to handle uncertainty: Whether it is the interval that contains λ or another source of uncertainty, the solution is obtained the same way.
- The main limitation of I-POD (or POD with uncertainty) is the time it takes to solve it...
  - On the up side, I-POD is run offline, so the time to run I-POD is not so relevant as the time to run the MORs is

NUMERICAL RESULTS. YAMAMURA

Let us consider the nonlinear system of equations

$$F : \mathbb{R}^{16} \to \mathbb{R}^{16},\ X \to (F_i(X))_{1 \leq i \leq 16}$$

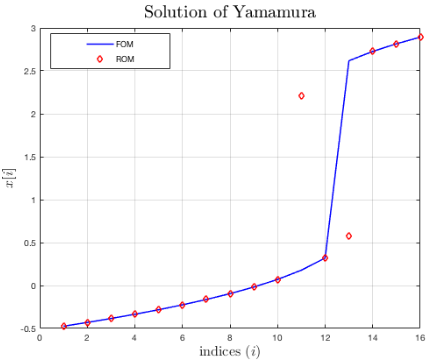
where $F : \mathbb{R}^{16} \to \mathbb{R}^{16}$ is a function defined:

$$F_i = \mu x_i^3 - 10.5 x_i^2 + 11.8 x_i - i + \sum_{i=1}^{i=n} x_i = 0; \qquad 0 \leq i \leq 16$$

with input parameter $\mu = 2.5 \in [1.5, 3.5]$

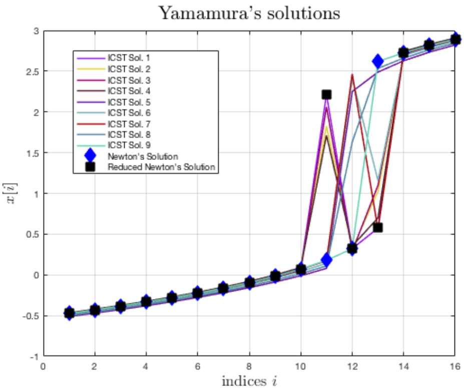
NUMERICAL RESULTS. YAMAMURA

Newton's Method and Reduced Newton's Method:



Method	# Iterations	Time	$\|\|F(x)\|\|$
Newton's	1	21.9 ms	8.9367e-4
Reduced Newton's	1	5.7 ms	5.5775e-6

ICST:



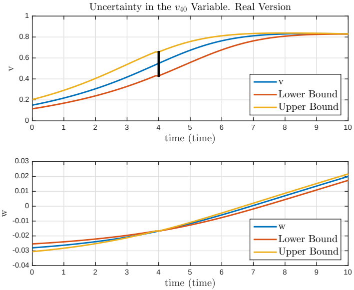
Method	# Solutions	Time	$\|\|F(x)\|\|$
Newton's	1	21.9 ms	8.9367e-4
Reduced Newton's	1	5.7 ms	5.5775e-6
ICST	9	–	$< 1.9418e-11,$

HOW TO UNDERSTAND AND AVOID?

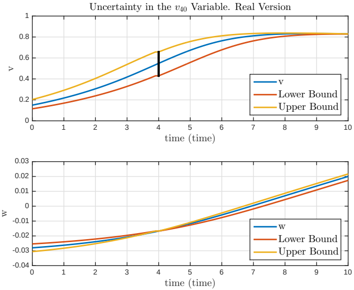
Problems that are of interest are the following:

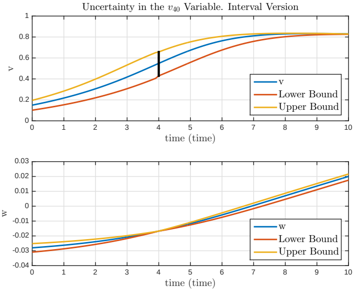
- Can we understand initial (or other) conditions that are consistent with observed data? And use this new data to predict what is going to happen?
- Can we identify initial (or other) conditions or values of parameters that are undesirable?

PREDICTIONS ON FITZ-HUGH NAGUMO



PREDICTIONS ON FITZ-HUGH NAGUMO

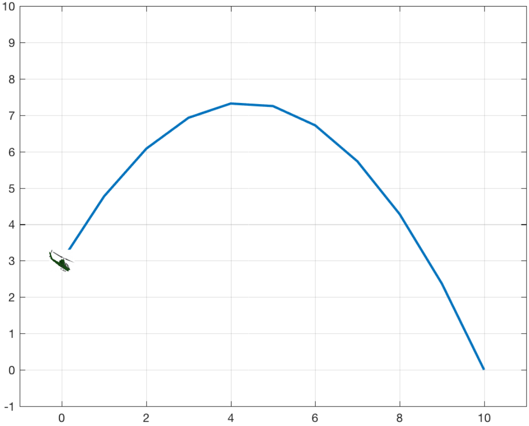




HOW TO ADAPT / ADJUST?

- Goal: adjust parameters of a dynamical system ”in flight” to ensure a new mission is completed
- Example of an Army problem: a helicopter is hit during mission.
  - How to adjust parameters to make sure it lands in a safe place?
  - What are the possible landing zones?

APPLICATIONS: PREDICTION



CONCLUSIONS

1. We are able to handle Interval POD: this means that we can handle uncertainty in models
    - The time it takes to run I-POD is not very relevant because it would normally be run “offline”
2. We are able to run Newton on the reduced problem
    - However, we get only one solution...
    - In general that’s all we would need (in an “online” situation)
3. In the context of prediction or adjustment “on the fly”, Reduced Newton is not going to be sufficient
    - We need to figure out how to improve: the focus has to be on an efficient ROM