Introduction to Simulation - Lecture 15

Methods for Computing Periodic Steady-State

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy

Outline

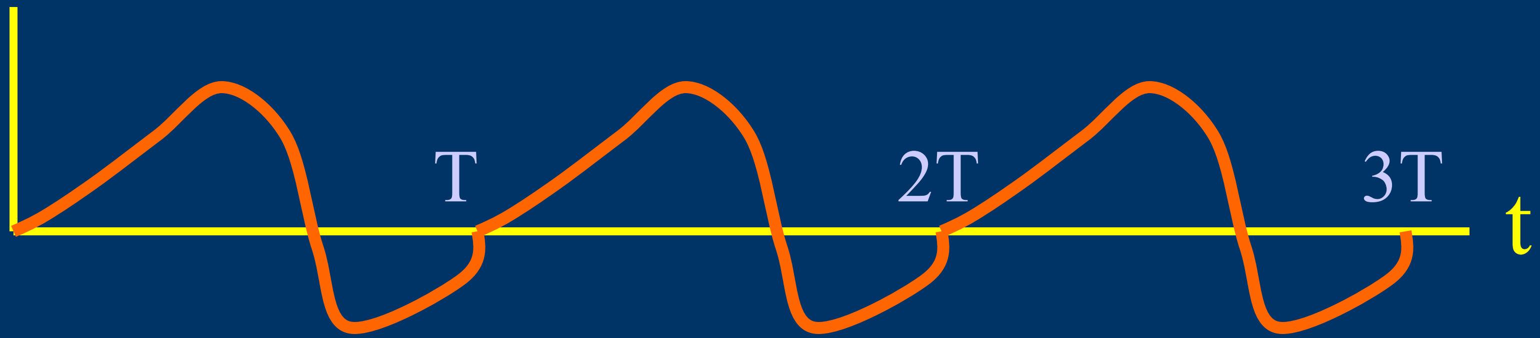
- Periodic Steady-state problems
  - Application examples and simple cases
- Finite-difference methods
  - Formulating large matrices
- Shooting Methods
  - State transition function
  - Sensitivity matrix
- Matrix Free Approach

Periodic Steady-State Basics

Basic Definition

$$\frac{dx(t)}{dt} = F\left(\underbrace{x(t)}_{state}\right) + \underbrace{u(t)}_{input}$$

- Suppose the system has a periodic input



- Many Systems eventually respond periodically

$$x(t+T) = x(t) \quad for \quad t >> 0$$

Periodic Steady-State Basics

Basic Definition

Interesting Property

- If x satisfies a differential equation which has a unique solution for any initial condition

$$\frac{dx(t)}{dt} = F(x(t)) + u(t)$$

- Then if u is periodic with period T and

$$x(t_0 + T) = x(t_0) \quad for \;\; some \; t_0$$

$$\Rightarrow x(t+T) = x(t) \quad for \;\; all \; t > t_0$$

Periodic Steady-State Basics

Application Examples

Swaying Bridge

- Periodic Input
  - Wind
- Response
  - Oscillating Platform
- Desired Info
  - Oscillation Amplitude

Periodic Steady-State Basics

Application Examples

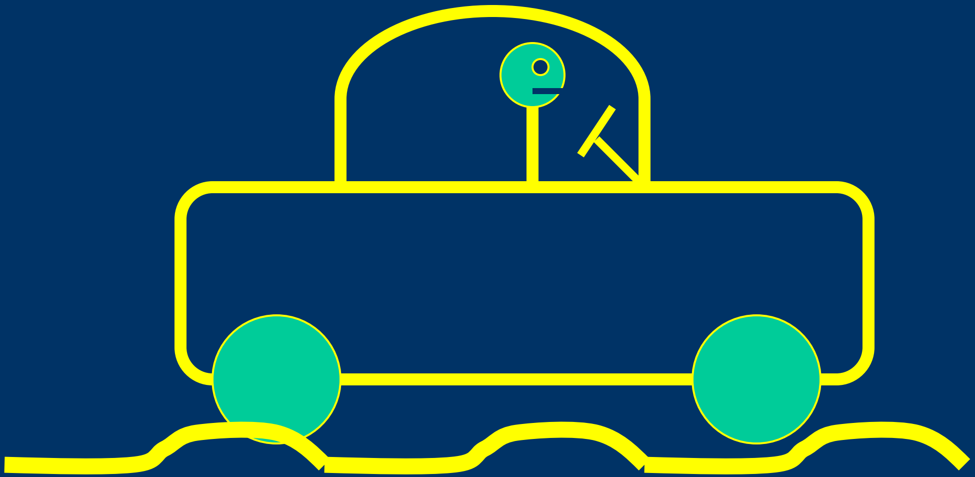
Communication Integrated Circuit

- Periodic Input
  - Received Signal at 900Mhz
- Response
  - filtered demodulated signal
- Desired Info
  - Distortion

Periodic Steady-State Basics

Application Examples

Automobile Vibration

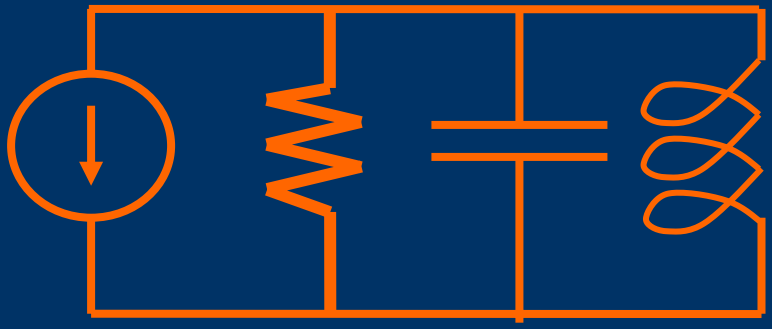


- Periodic Input
  - Regularly Spaced Road Bumps
- Response
  - Car Shakes
- Desired Info
  - Shake amplitude

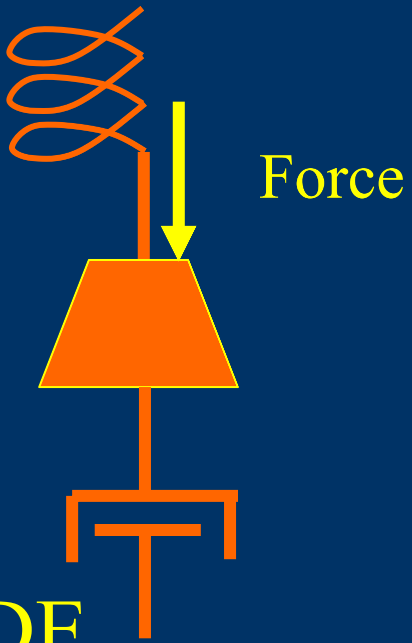
Simple Example

RLC Filter, Spring+Mass+Dashpot

RLC Circuit



Spring-Mass-Dashpot



- Both Described by Second-Order ODE

$$M\frac{d^2x}{dt^2}+D\frac{dx}{dt}+x=\underbrace{u(t)}_{input}$$

Simple Example

RLC Filter, Spring+Mass+Dashpot Cont.

- Both Described by Second-Order ODE

$$M\frac{d^2x}{dt^2} + D\frac{dx}{dt} + x = u(t)$$

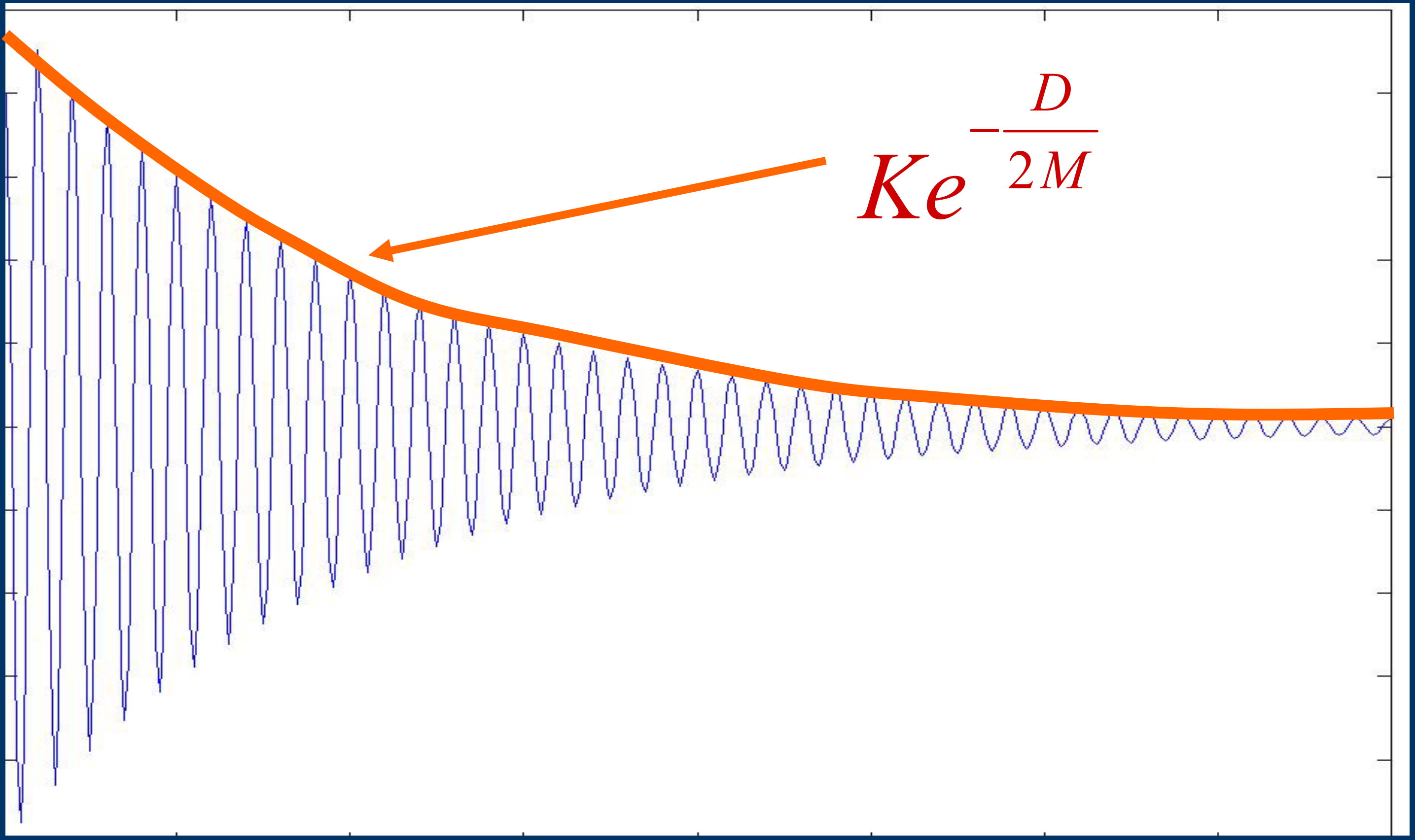
- u(t) = 0 lightly damped (D<<M) Response

$$x(t) \approx Ke^{-\frac{D}{2M}} \cos\left(\frac{t}{\sqrt{M}} + \phi\right)$$

Periodic Steady-State Basics

Simple Example

RLC Filter, Spring+Mass+Dashpot Cont.



$$Ke^{-\frac{D}{2M}}$$

- A lightly damped system oscillates many times before settling to a steady-state

Periodic Steady-State Basics

Computing Steady State

Frequency Domain Approach

- Sinusoidally excited linear time-invariant system

$$\frac{dx(t)}{dt} = Ax(t) + \underbrace{e^{i\omega t}}_{input}$$

- Steady-State Solution simple to determine

$$x(t) = (i\omega - A)^{-1} e^{i\omega t}$$

Not useful for nonlinear or time-varying systems

Periodic Steady-State Basics

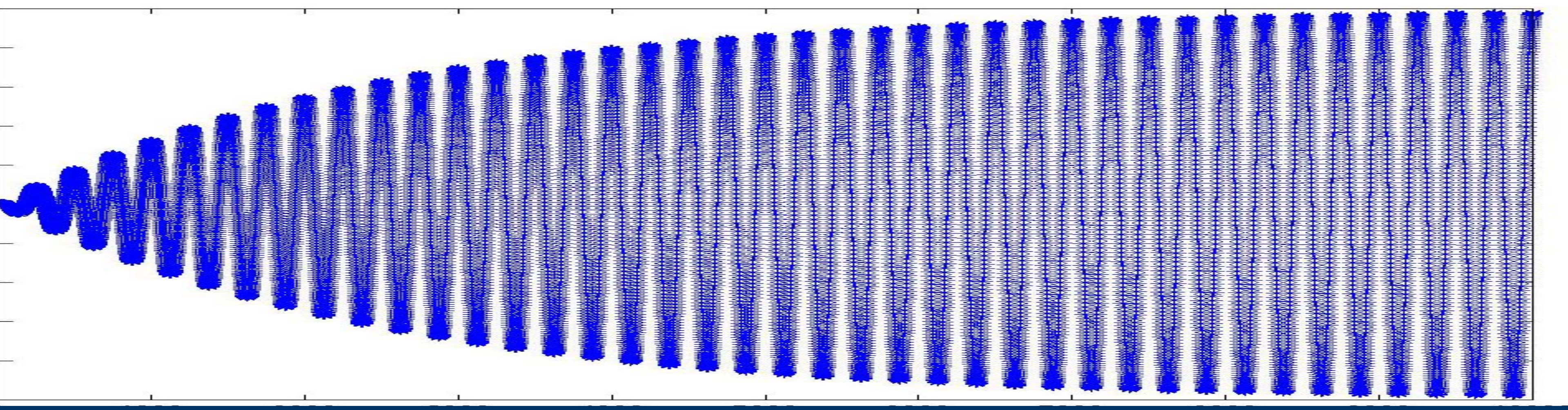
Computing Steady State

Time Integration Method

- Time-Integrate Until Steady-State Achieved

$$\frac{dx(t)}{dt} = F\left(x(t)\right) + u(t) \Rightarrow \hat{x}^{l} = \hat{x}^{l-1} + \Delta t\left(F\left(\hat{x}^{l}\right) + u(l\Delta t)\right)$$

- Need many timepoints for lightly damped case!



Aside Reviewing Integration Methods

Solve with Backward-Euler

- Nonlinear System

$$\frac{dx(t)}{dt} = F\left(\underbrace{x(t)}_{state}\right) + \underbrace{u(t)}_{input} \qquad \underbrace{x(0) = x_0}_{Initial\,Condition}$$

- Backward Euler Equation for timestep l

$$\hat{x}^{l} - \hat{x}^{l-1} = \Delta t\left(F\left(\hat{x}^{l}\right) + u(l\Delta t)\right)$$

How do we solve the backward-Euler Equation?

Aside Reviewing Integration Methods

Implicit Methods

Backward-Euler Example

Forward-Euler

$$x(t_1) \approx \hat{x}^1 = x(0) + \Delta t \ f\left(x\left(0\right), u\left(0\right)\right)$$

$$x(t_2) \approx \hat{x}^2 = \hat{x}^1 + \Delta t \ f\left(\hat{x}^1, u\left(t_1\right)\right)$$

$$\vdots$$

$$x(t_L) \approx \hat{x}^L = \hat{x}^{L-1} + \Delta t \ f\left(\hat{x}^{L-1}, u\left(t_{L-1}\right)\right)$$

Requires just function Evaluations

Backward-Euler

$$x(t_1) \approx \hat{x}^1 = x(0) + \Delta t \ f\left(\hat{x}^1, u\left(t_1\right)\right)$$

$$x(t_2) \approx \hat{x}^2 = \hat{x}^1 + \Delta t \ f\left(\hat{x}^2, u\left(t_2\right)\right)$$

$$\vdots$$

$$x(t_L) \approx \hat{x}^L = \hat{x}^{L-1} + \Delta t \ f\left(\hat{x}^L, u\left(t_L\right)\right)$$

Nonlinear equation solution at each step

Stepwise Nonlinear equation solution needed whenever $\beta_0 \neq 0$

Aside Reviewing Integration Methods

Implicit Methods

Solution with Newton

Rewrite the multistep Equation

$$\alpha_0 \hat{x}^l - \Delta t \beta_0 f\left(\hat{x}^l, u\left(t_l\right)\right) + \underbrace{\sum_{j=1}^{k} \alpha_j \hat{x}^{l-j} - \Delta t \sum_{j=1}^{k} \beta_j f\left(\hat{x}^{l-j}, u\left(t_{l-j}\right)\right)}_{b} = 0$$

Independent of $\hat{x}^l$

Solve with Newton

$$\underbrace{\left(\alpha_0 I - \Delta t \beta_0 \frac{\partial f\left(\hat{x}^{l,j}, u\left(t_l\right)\right)}{\partial x}\right)}_{\text{Jacobian}} \left(\hat{x}^{l,j+1} - \hat{x}^{l,j}\right) = -\underbrace{\left(\alpha_0 \hat{x}^{l,j} - \Delta t \beta_0 f\left(\hat{x}^{l,j}, u\left(t_l\right)\right) + b\right)}_{F\left(x^{l,j}\right)}$$

Here j is the Newton iteration index

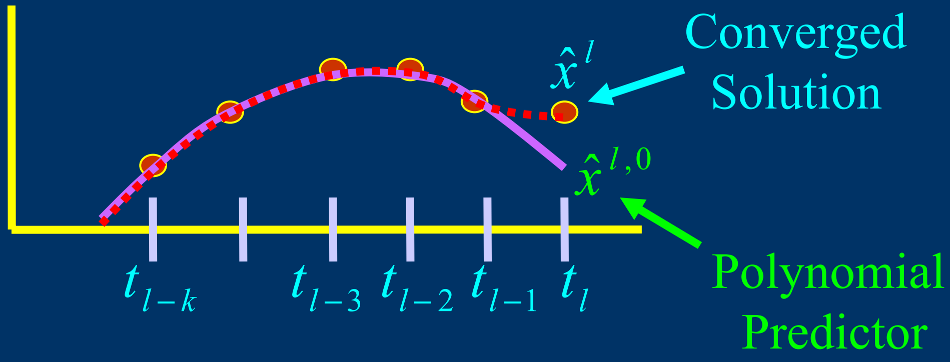
Aside Reviewing Integration Methods

Implicit Methods

Solution with Newton Cont.

Newton Iteration: $$\left(\alpha_0 I - \Delta t \beta_0 \frac{\partial f\left(\hat{x}^{l,j}, u(t_l)\right)}{\partial x}\right)\left(\hat{x}^{l,j+1} - \hat{x}^{l,j}\right) = -F(x^{l,j})$$

Solution with Newton is very efficient



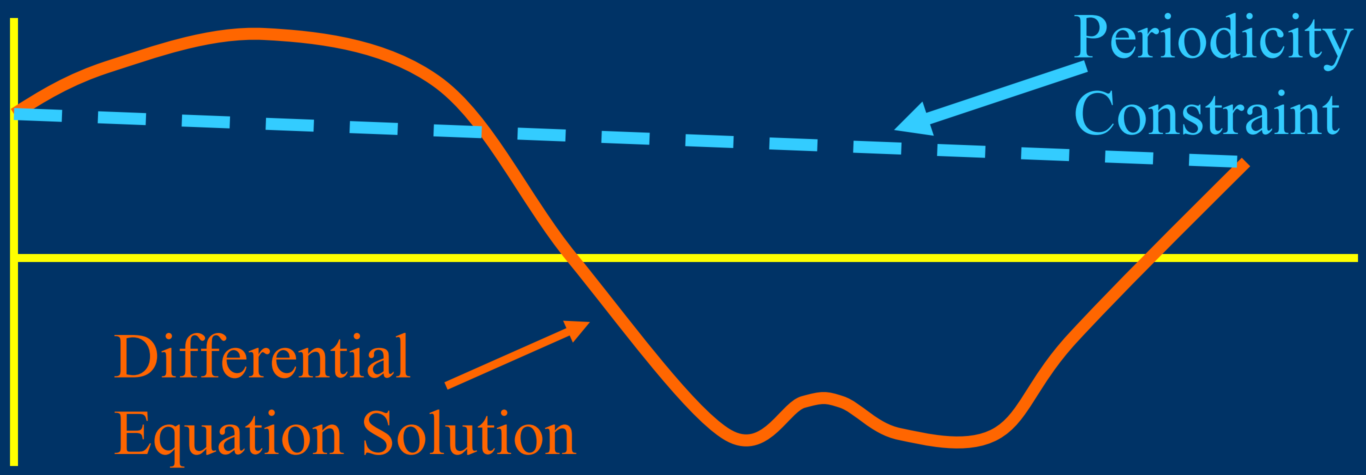
Easy to generate a good initial guess using polynomial fitting

$$\alpha_0 I - \Delta t \beta_0 \frac{\partial f\left(\hat{x}^{l,j}, u(t_l)\right)}{\partial x} \Rightarrow \alpha_0 I \text{ as } \Delta t \rightarrow 0$$

Jacobian become easy to factor for small timesteps

Boundary-Value Problem

Basic Formulation



N Differential Equations: $\frac{d}{dt}x_i(t) = F_i(x(t))$

N Periodicity Constraints: $x_i(T) = x_i(0)$

Boundary-Value Problem

Finite Difference Methods

Linear Example Problem

$$\frac{dx(t)}{dt} = Ax(t) + \underbrace{u(t)}_{input} \qquad t \in [0, T] \qquad \underbrace{x(T) = x(t)}_{\text{periodicity constraint}}$$

Discretize with Backward-Euler

$$\hat{x}^1 = \hat{x}^0 + \Delta t\left(A\hat{x}^1 + u(\Delta t)\right)$$

$$\hat{x}^2 = \hat{x}^1 + \Delta t\left(A\hat{x}^2 + u(2\Delta t)\right)$$

$$\vdots$$

$$\hat{x}^L = \hat{x}^{L-1} + \Delta t\left(A\hat{x}^L + u(L\Delta t)\right)$$

$$\Delta t = \frac{T}{L}$$

Periodicity implies $\hat{x}^L = \hat{x}^0$

Boundary-Value Problem

Finite Difference Methods

Linear Example Matrix Form

$$\begin{array}{cc} & \xleftrightarrow{\qquad \text{NxL} \qquad} \\ \text{NxL} \updownarrow & \begin{bmatrix} \frac{1}{\Delta t}I - A & 0 & 0 & -\frac{1}{\Delta t}I \\ -\frac{1}{\Delta t}I & \frac{1}{\Delta t}I - A & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -\frac{1}{\Delta t}I & \frac{1}{\Delta t}I - A \end{bmatrix} \end{array} \begin{bmatrix} \hat{x}^1 \\ \hat{x}^2 \\ \vdots \\ \vdots \\ \hat{x}^L \end{bmatrix} = \begin{bmatrix} u(0) \\ u(\Delta t) \\ \vdots \\ \vdots \\ u(L\Delta t) \end{bmatrix}$$

Matrix is almost lower triangular

Boundary-Value Problem

Finite Difference Methods

Nonlinear Problem

$$\frac{dx(t)}{dt} = F(x(t)) + \underbrace{u(t)}_{input} \qquad t \in [0,T] \qquad \underbrace{x(T) = x(t)}_{\text{periodicity constraint}}$$

Discretize with Backward-Euler

$$H_{FD}\left(\begin{bmatrix} \hat{x}^1 \\ \hat{x}^2 \\ \vdots \\ \hat{x}^L \end{bmatrix}\right) = \begin{matrix} \hat{x}^1 - \hat{x}^L - \Delta t\left(F\left(\hat{x}^1\right) + u(\Delta t)\right) \\ \hat{x}^2 - \hat{x}^1 - \Delta t\left(F\left(\hat{x}^2\right) + u(2\Delta t)\right) \\ \vdots \\ \hat{x}^L - \hat{x}^{L-1} - \Delta t\left(F\left(\hat{x}^L\right) + u(L\Delta t)\right) \end{matrix} = 0$$

Solve Using Newton's Method

Boundary-Value Problem

Shooting Method

Basic Definitions

Start with $\frac{dx(t)}{dt} = F(x(t)) + u(t)$

And assume x(t) is unique given x(0).

D.E. defines a State-Transition Function

$$\Phi(y, t_0, t_1) \equiv x(t_1)$$

where $x(t)$ is the D.E. solution given $x(t_0) = y$

Boundary-Value Problem

Shooting Method

State Transition function Example

$$\frac{dx(t)}{dt} = \lambda x(t)$$

$$\Phi(y, t_0, t_1) \equiv e^{\lambda(t_1 - t_0)} y$$

Boundary-Value Problem

Shooting Method

Abstract Formulation

Solve

$$H\left(x(0)\right) = \underbrace{\Phi\left(x(0), 0, T\right)}_{x(T)} - x(0) = 0$$

Use Newton's method

$$J_H(x) = \frac{\partial \Phi\left(x, 0, T\right)}{\partial x} - I$$

$$J_H\left(x^k\right)\left(x^{k+1} - x^k\right) = -H\left(x^k\right)$$

Boundary-Value Problem

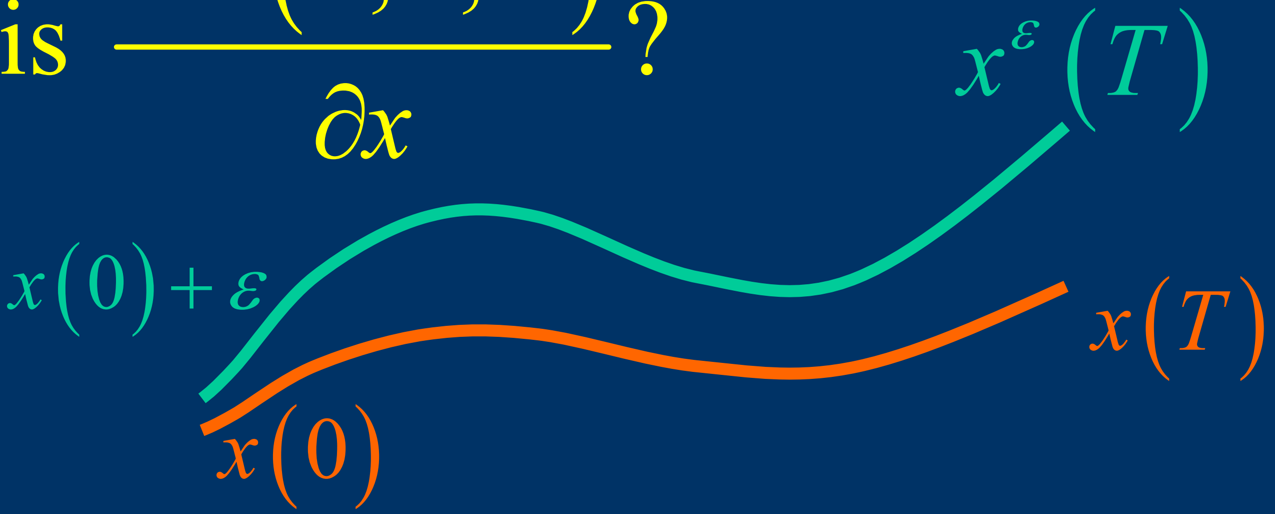
Shooting Method

Computing Newton

To Compute $\Phi\left(x(0),0,T\right)$

Integrate $\frac{dx(t)}{dt} = F\left(x(t)\right) + u(t)$ on [0,T]

What is $\frac{\partial\Phi\left(x,0,T\right)}{\partial x}$?



Indicates the sensitivity of x(T) to changes in x(0)

Boundary-Value Problem

Shooting Method

Sensitivity Matrix by Perturbation

$$\frac{\partial \Phi\left(x,0,T\right)}{\partial x} \approx \begin{bmatrix} \frac{x_1^{\varepsilon_1}\left(T\right)-x_1\left(T\right)}{\varepsilon_1} & \cdots & \cdots & \frac{x_1^{\varepsilon_N}\left(T\right)-x_1\left(T\right)}{\varepsilon_N} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ \frac{x_N^{\varepsilon_1}\left(T\right)-x_N\left(T\right)}{\varepsilon_1} & \cdots & \cdots & \frac{x_N^{\varepsilon_N}\left(T\right)-x_N\left(T\right)}{\varepsilon_N} \end{bmatrix}$$

Boundary-Value Problem

Shooting Method

Efficient Sensitivity Evaluation

Differentiate the first step of Backward-Euler

$$\frac{\partial}{\partial x(0)}\left(\hat{x}^1 - x(0) - \Delta t\left(F\left(\hat{x}^1\right) + u\left(\Delta t\right)\right) = 0\right)$$

$$\Rightarrow \frac{\partial \hat{x}^1}{\partial x(0)} - \frac{\partial x(0)}{\partial x(0)} - \Delta t \frac{\partial F\left(\hat{x}^1\right)}{\partial x} \frac{\partial \hat{x}^1}{\partial x(0)} = 0$$

$$\Rightarrow \left(I - \Delta t \frac{\partial F\left(\hat{x}^1\right)}{\partial x}\right) \frac{\partial \hat{x}^1}{\partial x(0)} = \cancelto{I}{\frac{\partial x(0)}{\partial x(0)}}$$

Efficient Sensitivity Matrix Cont

Applying the same trick on the l-th step

$$\Rightarrow \left(I - \Delta t \frac{\partial F\left(\hat{x}^{l}\right)}{\partial x} \right) \frac{\partial \hat{x}^{l}}{\partial x(0)} = \frac{\partial \hat{x}^{l-1}}{\partial x(0)}$$

$$\frac{\partial \Phi\left(x, 0, T\right)}{\partial x} \approx \prod_{l=1}^{L} \left(I - \Delta t \frac{\partial F\left(\hat{x}^{l}\right)}{\partial x} \right)^{-1}$$

Boundary-Value Problem

Shooting Method

Observations on Sensitivity Matrix

Newton at each timestep uses same matrices

$$\frac{\partial \Phi\left(x,0,T\right)}{\partial x} \approx \prod_{l=1}^{L} \underbrace{\left(I - \Delta t \frac{\partial F\left(\hat{x}^{l}\right)}{\partial x} \right)^{-1}}_{\textit{Timestep Newton Jacobian}}$$

Formula simplifies in the linear case

$$\frac{\partial \Phi\left(x,0,T\right)}{\partial x} \approx \left(I - \Delta t A \right)^{-L}$$

Shooting Method

Matrix-Free Approach

Basic Setup

Start with $\frac{dx(t)}{dt} = F(x(t)) + u(t)$

$$H(x(0)) = \Phi(x(0), 0, T) - x(0) = 0$$

Use Newton's method

$$J_H(x) = \frac{\partial \Phi(x, 0, T)}{\partial x} - I$$

$$J_H(x^k)(x^{k+1} - x^k) = -H(x^k)$$

Shooting Method

Matrix-Free Approach

Matrix-Vector Product

Solve Newton equation with Krylov-subspace method

$$\underbrace{\left(\frac{\partial\Phi\left(x^{k},0,T\right)}{\partial x}-I\right)}_{A}\underbrace{\left(x^{k+1}-x^{k}\right)}_{x}=\underbrace{x^{k}-\Phi\left(x^{k},0,T\right)}_{b}$$

Matrix-Vector Product Computation

$$\left(\frac{\partial\Phi\left(x^{k},0,T\right)}{\partial x}-I\right)p^{j}\approx\frac{\Phi\left(x^{k}+\varepsilon p^{j},0,T\right)-\Phi\left(x^{k},0,T\right)}{\varepsilon}-p^{j}$$

Krylov method search direction

Convergence for GCR

Example

$$\frac{dx}{dt} - Ax = 0 \quad eig\left(A\right) \text{ real and negative}$$

Shooting-Newton Jacobian

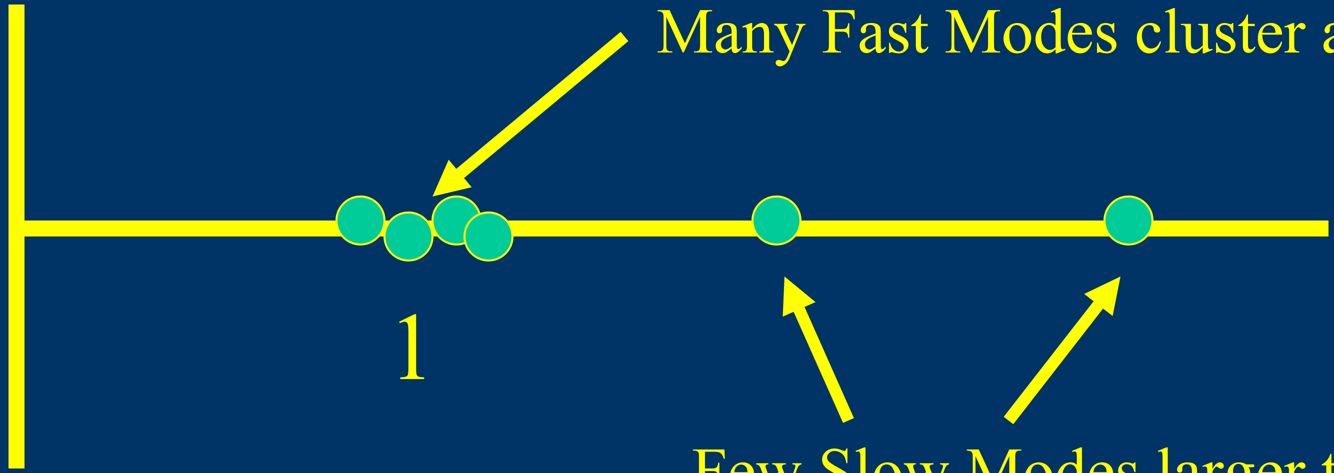
$$\frac{\partial\Phi\left(x,0,T\right)}{\partial x} - I = e^{AT} - I$$

Shooting Method

Matrix-Free Approach

Convergence for GCR-evals

$$e^{AT} - I = S \begin{bmatrix} e^{\lambda_1 T} - 1 & & \\ & \ddots & \\ & & e^{\lambda_N T} - 1 \end{bmatrix} S^{-1}$$



Many Fast Modes cluster at 1

Few Slow Modes larger than 1

Summary

- Periodic Steady-state problems
  - Application examples and simple cases
- Finite-difference methods
  - Formulating large matrices
- Shooting Methods
  - State transition function
  - Sensitivity matrix