Introduction to Simulation - Lecture 16

Methods for Computing Periodic Steady-State - Part II

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy

Outline

- Three Methods so far
  - Time integration until steady-state achieved
  - Finite difference methods
  - Shooting Methods
- Shooting Methods
  - State transition function
  - Sensitivity matrix
  - Matrix-Free Approach
- Spectral Methods
  - Galerkin and Collocation Methods

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

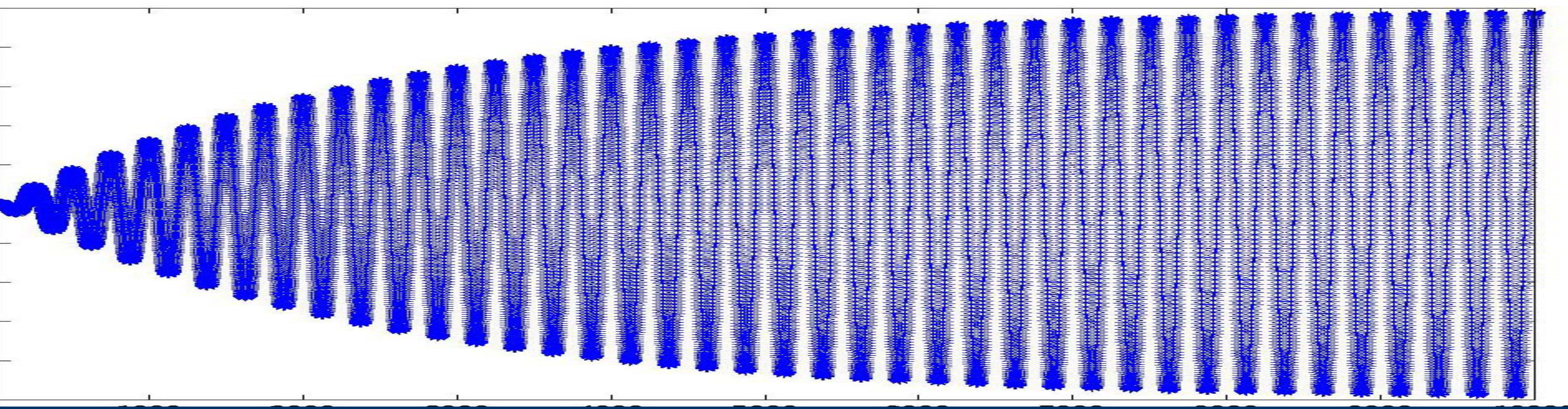
$$x(t+T) = x(t) \quad for \quad t >> 0$$

Periodic Steady-State Basics

- Time-Integrate Until Steady-State Achieved

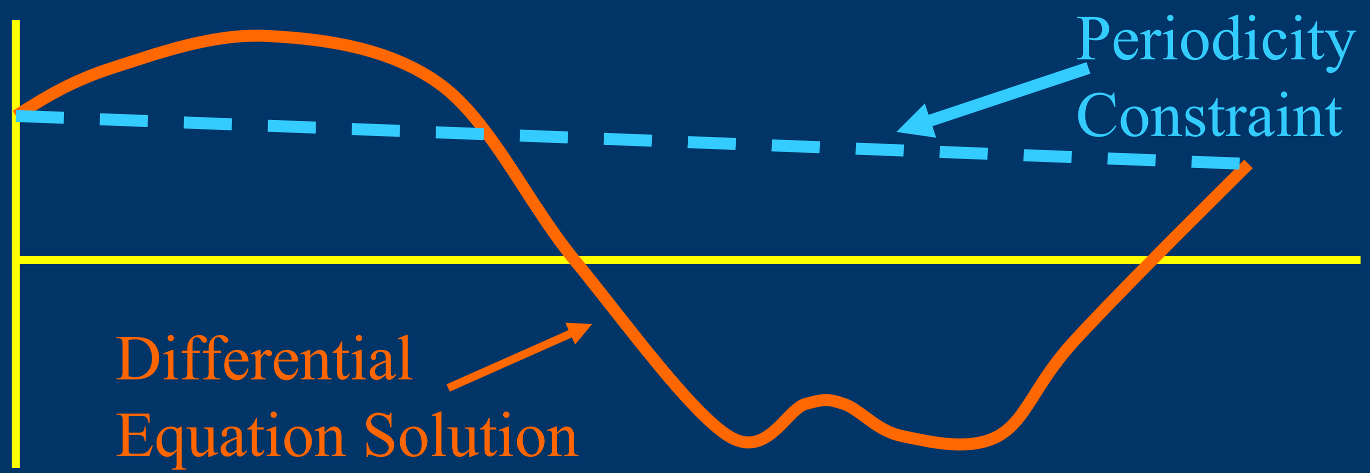
$$\frac{dx(t)}{dt} = F\left(x(t)\right) + u(t) \Rightarrow \hat{x}^{l} = \hat{x}^{l-1} + \Delta t\left(F\left(\hat{x}^{l}\right) + u(l\Delta t)\right)$$

- Need many timepoints for lightly damped case!



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

Nonlinear Problem

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

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

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

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

<u>D.E. defines a State-Transition Function</u>

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

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

Boundary-Value Problem

Shooting Method

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

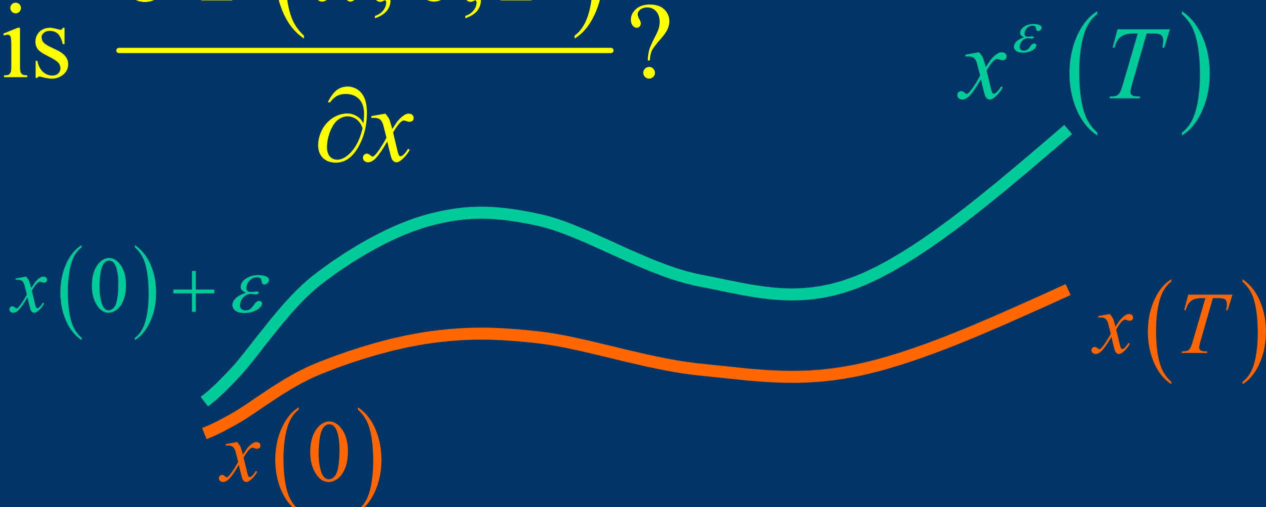
$$J_H(x) = \frac{\partial \Phi\left(x, 0, T\right)}{\partial x} - I$$

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

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

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

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



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

Boundary-Value Problem

Shooting Method

Sensitivity Matrix by Perturbation

$$\frac{\partial \Phi(x, 0, T)}{\partial x} \approx$$

$$\begin{bmatrix} \frac{x_1^{\varepsilon_1}(T) - x_1(T)}{\varepsilon_1} & \cdots & \cdots & \frac{x_1^{\varepsilon_N}(T) - x_1(T)}{\varepsilon_N} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ \frac{x_N^{\varepsilon_1}(T) - x_N(T)}{\varepsilon_1} & \cdots & \cdots & \frac{x_N^{\varepsilon_N}(T) - x_N(T)}{\varepsilon_N} \end{bmatrix}$$

Boundary-Value Problem

Shooting Method

Efficient Sensitivity Evaluation

Differentiate the first step of Backward-Euler

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

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

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

Applying the same trick on the l-th step

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

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

Observations on Sensitivity Matrix

Newton at each timestep uses same matrices

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

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

Example

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

Shooting-Newton Jacobian

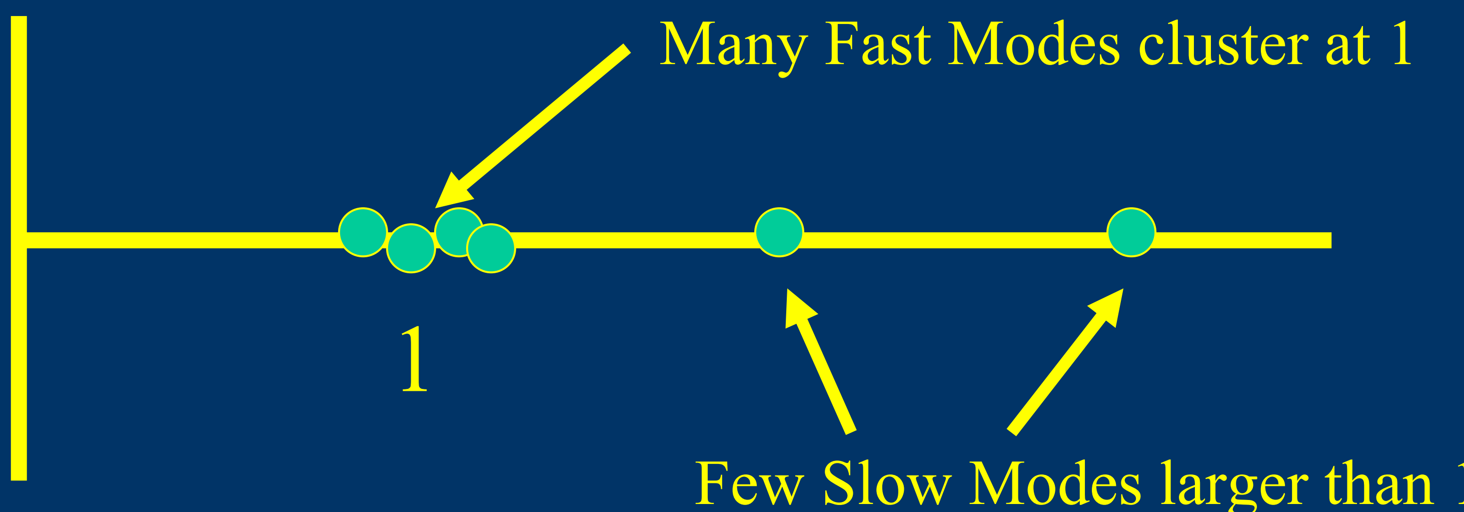
$$\frac{\partial \Phi(x, 0, T)}{\partial x} - I = e^{AT} - I$$

Shooting Method

Matrix-Free Approach

Convergence for GCR-evals

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



- Periodic function → fourier series

$$x(t) = \sum_{l=-\infty}^{\infty} X_l e^{-i2\pi l \frac{t}{T}}$$

- Approximate a function with truncated series

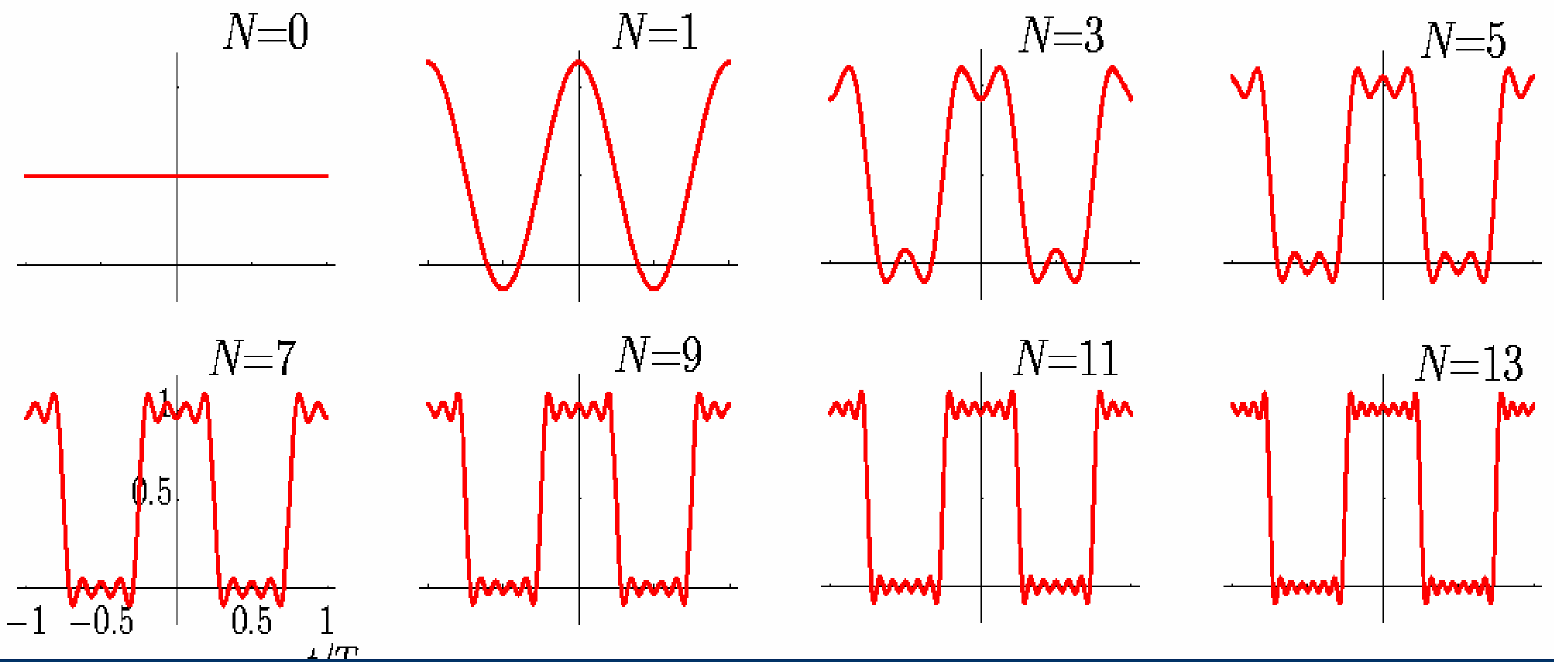
$$x(t) \approx \sum_{l=-L}^{L} X_l e^{-i2\pi l \frac{t}{T}}$$

Spectral Methods

Fourier Representation

Square Wave Example

$$x_N(t) = \frac{1}{2} + \sum_{n=1}^{N} \left(\frac{\sin(n\pi/2)}{n\pi/2} \right) \cos(2\pi n t/T).$$



Annoyance for Real Functions

- Real x →Fourier Coeffs complex conjugate

$$X_{-l} = X_l^*$$

- Can rewrite series with fewer unknowns

$$x(t) = \sum_{l=1}^{\infty} \underbrace{X_l e^{-i2\pi l \frac{t}{T}} + X_l^* e^{+i2\pi l \frac{t}{T}}}_{\text{Real}} + \underbrace{X_0}_{l=0}$$

Spectral Methods

Fourier Representation

Orthogonality

- Terms in Fourier Series are orthogonal

$$\int_0^T e^{-i2\pi l\frac{t}{T}} e^{-i2\pi m\frac{t}{T}} dt = 0 \quad l \neq m$$

- Simple formula for computing coefficients

$$\int_0^T e^{-i2\pi m\frac{t}{T}} x(t) dt = \int_0^T e^{-i2\pi m\frac{t}{T}} \sum_{l=-\infty}^{\infty} X_l e^{-i2\pi l\frac{t}{T}} dt = TX_m$$

Advantages

- For smooth functions (infinitely cont. diff)
  - Fourier Coefficients decay exponentially fast

$$\lim_{m->\infty} \frac{1}{T}\int_0^T e^{-i2\pi m\frac{t}{T}} x(t)dt = \lim_{m->\infty} X_m = O\left(c^m\right)$$

- Automatically satisfies periodicity

$$x(t+T) = \sum_{l=-L}^{L} X_l e^{-i2\pi l\left(\frac{t+T}{T}\right)} = \sum_{l=-L}^{L} X_l e^{-i2\pi l\frac{t}{T}} = x\left(t\right)$$

Residual

- Plug representation into differential equation

$$\underbrace{R\left(\vec{X},t\right)}_{\text{Residual}} = \frac{d}{dt}\left(\sum_{l=-L}^{L} X_l e^{-i2\pi l\frac{t}{T}}\right) - F\left(\sum_{l=-L}^{L} X_l e^{-i2\pi l\frac{t}{T}}\right) - u\left(t\right)$$

- Simplify by differentiating representation

$$\underbrace{R\left(\vec{X},t\right)}_{\text{Residual}} = \sum_{l=-L}^{L} \frac{-i2\pi l}{T} X_l e^{-i2\pi l\frac{t}{T}} - F\left(\sum_{l=-L}^{L} X_l e^{-i2\pi l\frac{t}{T}}\right) - u\left(t\right)$$

Spectral Methods

- Collocation – Residual = 0 at test points

$$\underbrace{R\left(\vec{X}, t_l\right)}_{\text{Residual}} = 0 \quad l = \{1, \ldots, 2L+1\}$$

- Galerkin – Residual orthog to Fourier Terms

$$\int_0^T e^{-i2\pi m \frac{t}{T}} \underbrace{R\left(\vec{X}, t\right)}_{\text{Residual}} dt = 0 \quad m \in \{-L, \ldots, 0, \ldots L\}$$

- Galerkin – Residual orthog to Fourier Terms

$$-\left(\int_0^T e^{-i2\pi m\frac{t}{T}}\left(\sum_{l=-L}^{L}\frac{-i2\pi l}{T}X_l e^{-i2\pi l\frac{t}{T}} - F\left(\sum_{l=-L}^{L}X_l e^{-i2\pi l\frac{t}{T}}\right)-u(t)\right)dt\right)=$$

$$i2\pi m X_l + \int_0^T e^{-i2\pi m\frac{t}{T}} F\left(\sum_{l=-L}^{L}X_l e^{-i2\pi l\frac{t}{T}}\right)dt + \int_0^T e^{-i2\pi m\frac{t}{T}} u(t)dt = 0$$

$$m \in \{-L,\ldots,0,\ldots L\}$$

Spectral Methods

Computing Coefficients

Linear Galerkin *F(x)=Ax*

$$i2\pi m X_l + \int_0^T e^{-i2\pi m\frac{t}{T}} A\left(\sum_{l=-L}^{L} X_l e^{-i2\pi l\frac{t}{T}}\right)dt + \underbrace{\int_0^T e^{-i2\pi m\frac{t}{T}} u(t)dt}_{U_m} = 0$$

$$\underbrace{\begin{bmatrix} \frac{i2\pi L}{T}+A & 0 & 0 & 0 \\ 0 & \frac{i2\pi(L-1)}{T}+A & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & -\frac{i2\pi L}{T}+A \end{bmatrix}}_{Diagonal} \begin{bmatrix} X_{-L} \\ X_{-(L-1)} \\ \vdots \\ \vdots \\ X_L \end{bmatrix} = - \begin{bmatrix} U_{-L} \\ U_{-(L-1)} \\ \vdots \\ \vdots \\ U_L \end{bmatrix}$$

Spectral Methods

Computing Coefficients

Collocation Equations

- Collocation – Residual zero at test times

$$\underbrace{R\left(\vec{X}, t_l\right)}_{\text{Residual}} = 0 = \sum_{l=-L}^{L} \frac{-i2\pi l}{T} X_l e^{-i2\pi l \frac{t_l}{T}} - F\left(\sum_{l=-L}^{L} X_l e^{-i2\pi l \frac{t_l}{T}}\right) - u\left(t_l\right)$$

$$l = \{1, \ldots, 2L+1\}$$

Computing Coefficients

Discrete Fourier Transform

$$\underbrace{\begin{bmatrix} e^{i2\pi\frac{L}{T}t_1} & \cdots & \cdots & e^{-i2\pi\frac{L}{T}t_1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ e^{i2\pi\frac{L}{T}t_{(2L+1)}} & \cdots & \cdots & e^{-i2\pi\frac{L}{T}t_{(2L+1)}} \end{bmatrix}}_{\text{Discrete Fourier Transform(DFT)}} \begin{bmatrix} X_{-L} \\ X_{-(L-1)} \\ \vdots \\ \vdots \\ X_L \end{bmatrix} = \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ \vdots \\ x(t_{(2L+1)}) \end{bmatrix}$$

If $t_l = \frac{l}{2L+1}T$ then DFT Matrix has orthog columns

Spectral Methods

Computing Coefficients

Collocation using timepoints

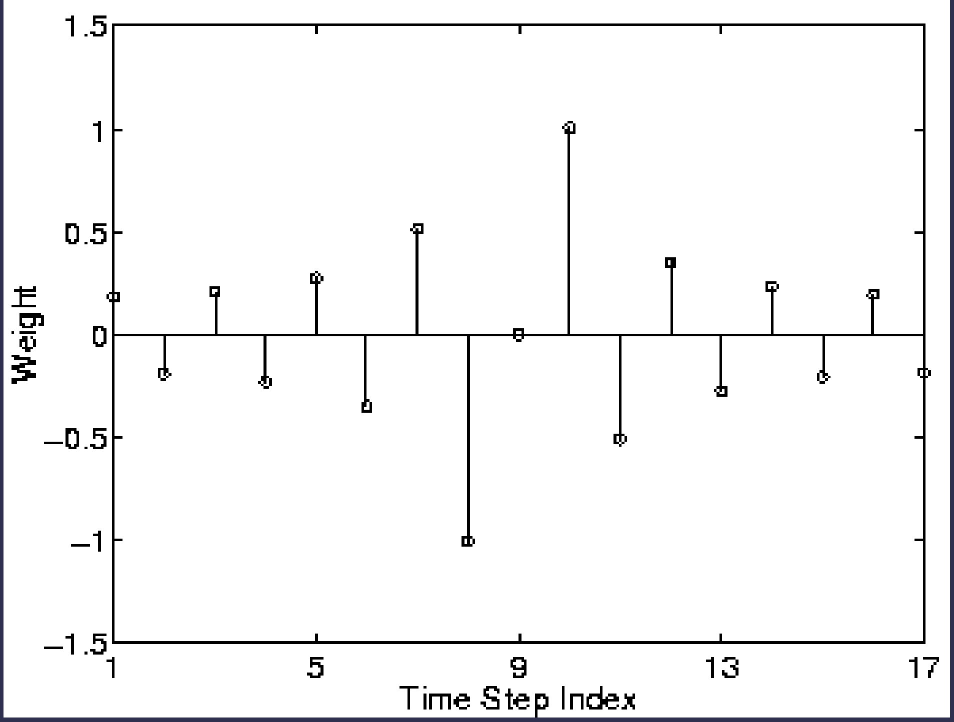
$$\underbrace{DFT\begin{bmatrix} \frac{i2\pi L}{T} & 0 & 0 & 0 \\ 0 & \frac{i2\pi(L-1)}{T} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & -\frac{i2\pi L}{T} \end{bmatrix}(DFT)^{-1}}_{\text{Spectral Differentiation}} \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ \vdots \\ x(t_{(2L+1)}) \end{bmatrix} - \begin{bmatrix} F(x(t_1)) \\ F(x(t_2)) \\ \vdots \\ \vdots \\ F(x(t_{(2L+1)})) \end{bmatrix} = \begin{bmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ \vdots \\ u(t_{(2L+1)}) \end{bmatrix}$$

Converting timepoint into Fourier Coeffs,
Differentiating, and then returning to time

Spectral Methods

Computing Coefficients

Spectral Differentiation Example



Middle row, T = 17 and 2L+1 = 17

Spectral Methods

Computing Coefficients

Spectral Colloc vs. F-D

$$\begin{bmatrix} \frac{1}{\Delta t} & 0 & 0 & -\frac{1}{\Delta t} \\ -\frac{1}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -\frac{1}{\Delta t} & \frac{1}{\Delta t} \end{bmatrix} \begin{bmatrix} \hat{x}^1 \\ \hat{x}^2 \\ \vdots \\ \vdots \\ \hat{x}^{2L+1} \end{bmatrix} + \begin{bmatrix} F(x(t_1)) \\ F(x(t_2)) \\ \vdots \\ \vdots \\ F(x(t_{(2L+1)})) \end{bmatrix} = \begin{bmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ \vdots \\ u(t_{(2L+1)}) \end{bmatrix}$$

$$DFT \begin{bmatrix} \frac{i2\pi L}{T} & 0 & 0 & 0 \\ 0 & \frac{i2\pi(L-1)}{T} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & -\frac{i2\pi L}{T} \end{bmatrix} (DFT)^{-1} \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ \vdots \\ x(t_{(2L+1)}) \end{bmatrix} - \begin{bmatrix} F(x(t_1)) \\ F(x(t_2)) \\ \vdots \\ \vdots \\ F(x(t_{(2L+1)})) \end{bmatrix} = \begin{bmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ \vdots \\ u(t_{(2L+1)}) \end{bmatrix}$$

Summary

- Four Methods
  - Time integration until steady-state achieved
  - Finite difference methods
  - Shooting Methods
  - Spectral Methods
- Shooting Methods
  - State transition function
  - Sensitivity matrix
  - Matrix-Free Approach
- Spectral Methods
  - Galerkin and Collocation Methods