# ECEN 667
# Power System Stability

## Lecture 2: Numeric Solution of Differential Equations

Prof. Tom Overbye

Dept. of Electrical and Computer Engineering

Texas A&M University

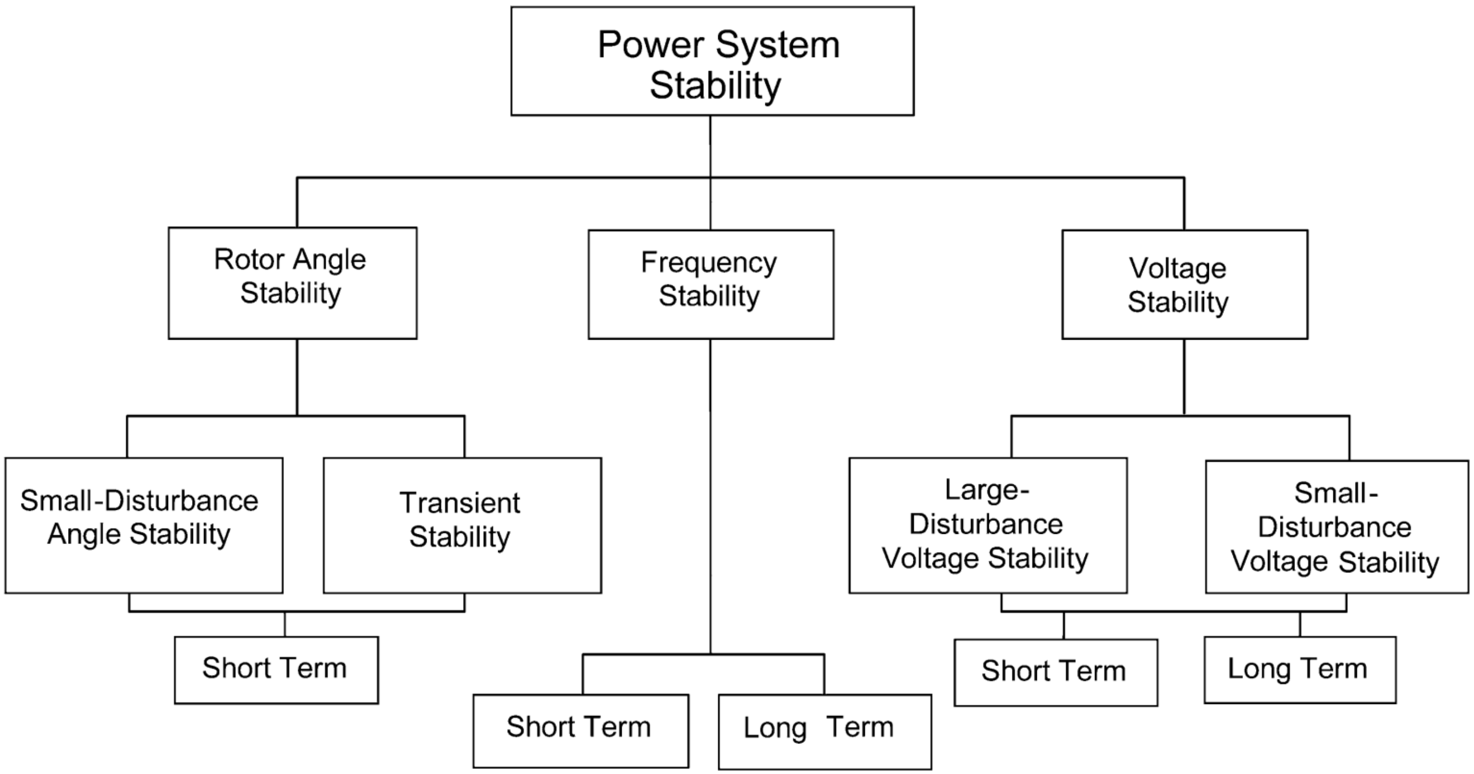
overbye@tamu.edu

# Announcements

- RSVP to Alex at zandra23@ece.tamu.edu for the TAMU ECE Energy and Power Group (EPG) picnic. It starts at 5pm on September 27, 2019
- Be reading Chapters 1 and 2

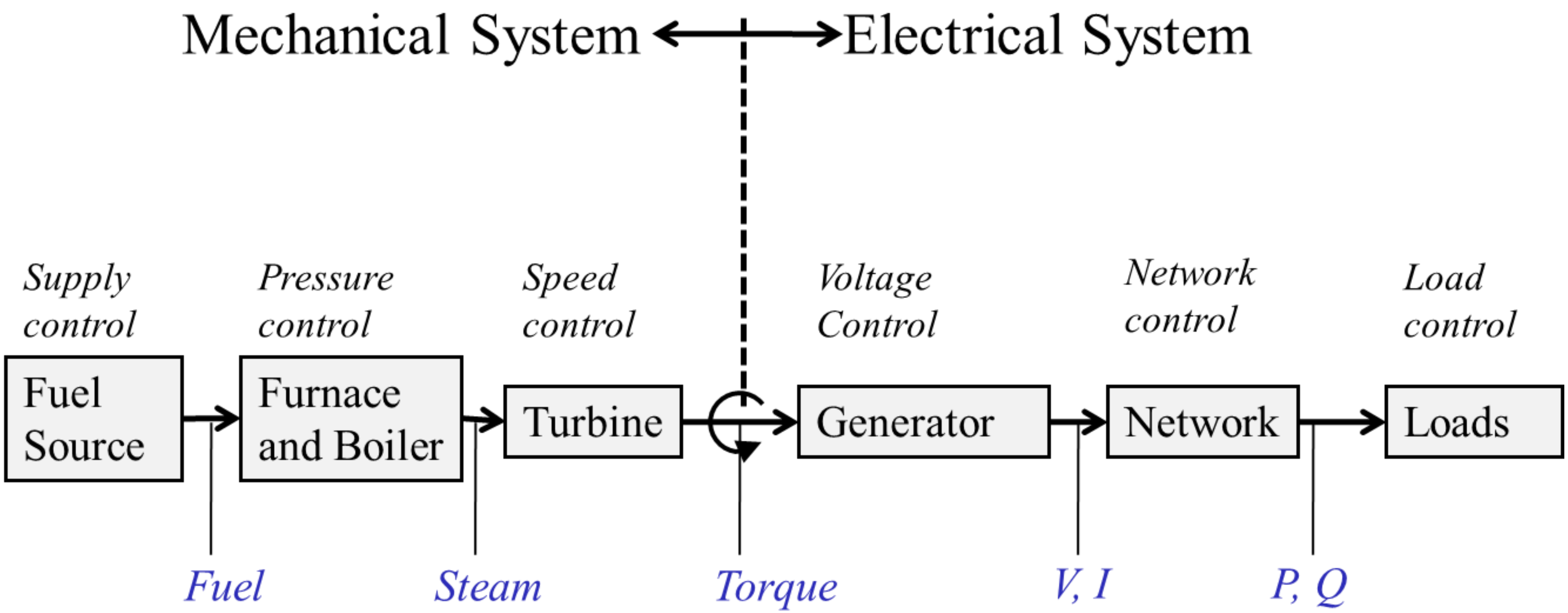
# Power System Stability Terms

- Terms continue to evolve, but a good reference is [1]; image shows Figure 1 from this reference



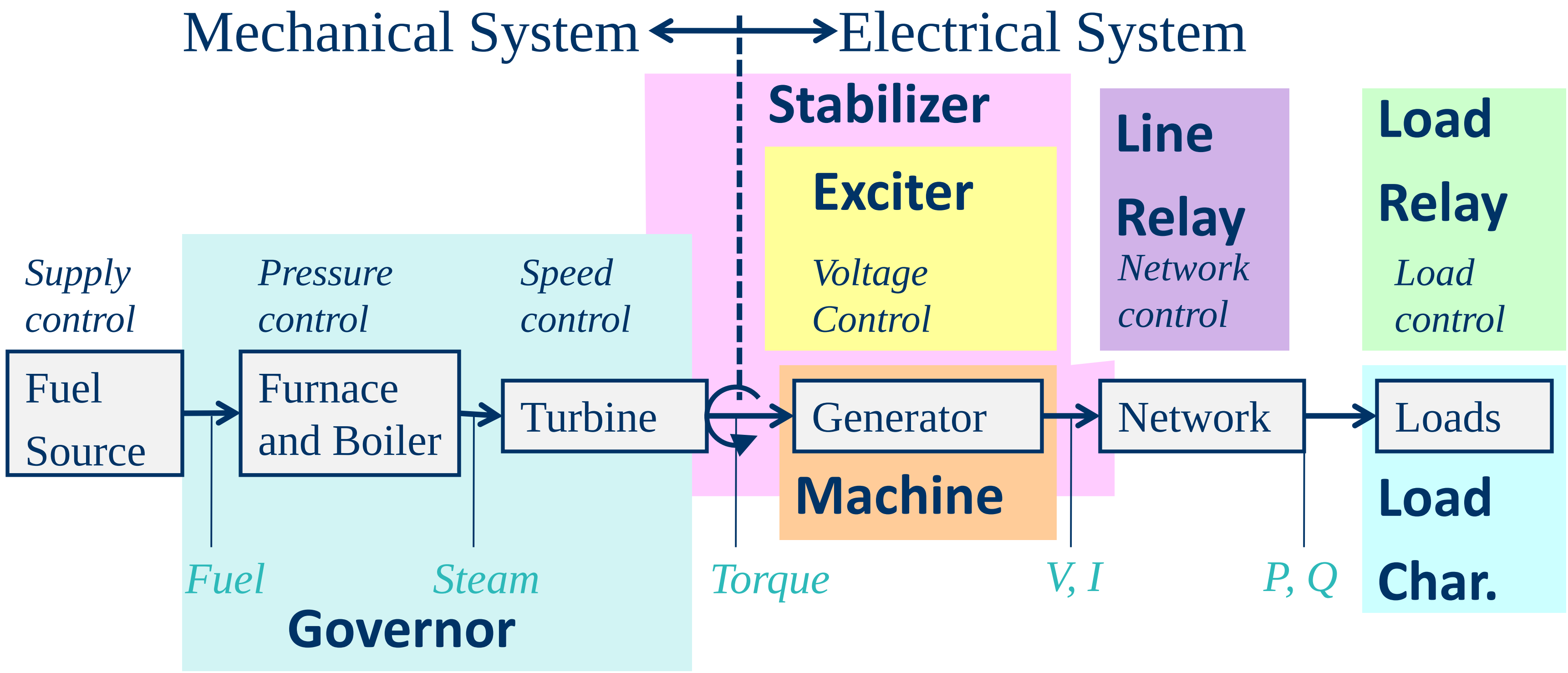

[1] IEEE/CIGRE Joint Task Force on Stability Terms and Definitions, "Definitions and Classification of Power System Stability," *IEEE Transactions Power Systems*, May 2004, pp. 1387-1401

# Physical Structure
# Power System Components




P. Sauer and M. Pai, *Power System Dynamics and Stability*

# Physical Structure
# Power System Components



# Differential Algebraic Equations

- Many problems, including many in the power area, can be formulated as a set of differential, algebraic equations (DAE) of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y})$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y})$$

- A power example is transient stability, in which **f** represents (primarily) the generator dynamics, and **g** (primarily) the bus power balance equations
- We'll initially consider the simpler problem of just

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

# Ordinary Differential Equations (ODEs)

- Assume we have a problem of the form

  $$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad \text{with } \mathbf{x}(t_0) = \mathbf{x}_0$$

- This is known as an initial value problem, since the initial value of **x** is given at some time $t_0$
  - We need to determine **x**(t) for future time
  - Initial value, $\mathbf{x}_0$, must be either be given or determined by solving for an equilibrium point, $\mathbf{f}(\mathbf{x}) = \mathbf{0}$
  - Higher-order systems can be put into this first order form
- Except for special cases, such as linear systems, an analytic solution is usually not possible – numerical methods must be used

# Equilibrium Points

- An equilibrium point **x*** satisfies

  $$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$$

- An equilibrium point is stable if the response to a small disturbance remains small
  - This is known as Lyapunov stability
  - Formally, if for every $\varepsilon > 0$, there exists a $\delta = \delta(\varepsilon) > 0$ such that if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, then $\|\mathbf{x}(t) - \mathbf{x}^*\| < \varepsilon$ for $t \geq 0$
- An equilibrium point has asymptotic stability if there exists a $\delta > 0$ such that if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, then

  $$\lim_{t \to \infty} \|\mathbf{x}(t) - \mathbf{x}^*\| = \mathbf{0}$$

# Power System Application

- A typical power system application is to assume the power flow solution represents an equilibrium point
- Back solve to determine the initial state variables, $\mathbf{x}(0)$
- At some point a contingency occurs, perturbing the state away from the equilibrium point
- Time domain simulation is used to determine whether the system returns to the equilibrium point

# Initial value Problem Examples

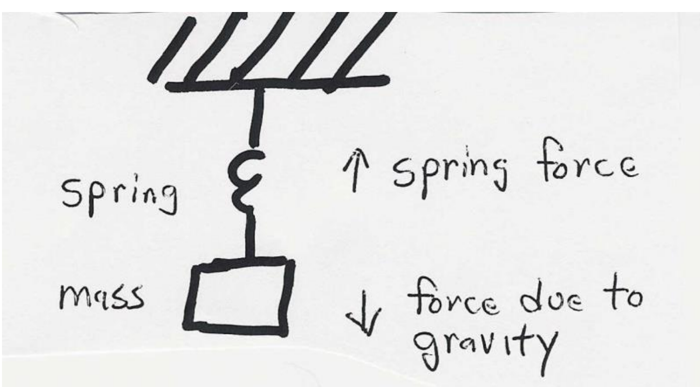
Example 1: Exponential Decay

A simple example with an analytic solution is

$$\dot{x} = -x \qquad \text{with } x(0) = x_0$$

This has a solution $x(t) = x_0 e^{-t}$

Example 2: Mass-Spring System




$$kx - gM = M\ddot{x} + D\dot{x}$$

or

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{M}\left(k\,x_1 - g\,M - D\,x_2\right)$$

# Numerical Solution Methods

- Numerical solution methods do not generate exact solutions; they practically always introduce some error
  - Methods assume time advances in discrete increments, called a stepsize (or time step), $\Delta t$
  - Speed accuracy tradeoff: a smaller $\Delta t$ usually gives a better solution, but it takes longer to compute
  - Numeric roundoff error due to finite computer word size
- Key issue is the derivative of **x, f(x)** depends on **x**, the value we are trying to determine
- A solution exists as long as **f**(**x**) is continuously differentiable

# Numerical Solution Methods

- There are a wide variety of different solution approaches, we will only touch on several
- One-step methods: require information about solution just at one point, **x**(t)
  - Forward Euler
  - Runge-Kutta
- Multi-step methods: make use of information at more than one point, **x**(t), **x**(t-Δt), **x**(t-Δ2t)…
  - Adams-Bashforth
- Predictor-Corrector Methods: implicit
  - Backward Euler

# Error Propagation

- At each time step the total round-off error is the sum of the local round-off at time and the propagated error from steps 1, 2 , … , k − 1
- An algorithm with the desirable property that local round-off error decays with increasing number of steps is said to be numerically stable
- Otherwise, the algorithm is numerically unstable
- Numerically unstable algorithms can nevertheless give quite good performance if appropriate time steps are used
  - This is particularly true when coupled with algebraic equations

# Forward Euler's Method

- The simplest technique for numerically integrating such equations is known as the Euler's Method (sometimes the Forward Euler's Method)
- Key idea is to approximate

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \frac{\mathrm{d}\mathbf{x}}{\mathrm{dt}} \text{ as } \frac{\Delta\mathbf{x}}{\Delta \mathrm{t}}$$

Then

$$\mathbf{x}(t+\Delta t) \approx \mathbf{x}(t) + \Delta t\, \mathbf{f}(\mathbf{x}(t))$$

- In general, the smaller the $\Delta$t, the more accurate the solution, but it also takes more time steps

# Euler's Method Algorithm

Set $t = t_0$ (usually 0)

$\mathbf{x}(t_0) = \mathbf{x}_0$

Pick the time step $\Delta t$, which is problem specific

While $t \leq t^{end}$ Do

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\mathbf{f}(\mathbf{x}(t))$$

$$t = t + \Delta t$$

End While

# Euler's Method Example 1

Consider the Exponential Decay Example

$$\dot{x} = -x \qquad \text{with } x(0) = x_0$$

This has a solution $x(t) = x_0 e^{-t}$

Since we know the solution we can compare the accuracy of Euler's method for different time steps

# Euler's Method Example 1, cont'd

| t | $x^{actual}(t)$ | $x(t)$  $\Delta t=0.1$ | $x(t)$  $\Delta t=0.05$ |
|---|---|---|---|
| 0 | 10 | 10 | 10 |
| 0.1 | 9.048 | 9 | 9.02 |
| 0.2 | 8.187 | 8.10 | 8.15 |
| 0.3 | 7.408 | 7.29 | 7.35 |
| … | … | … | … |
| 1.0 | 3.678 | 3.49 | 3.58 |
| … | … | … | … |
| 2.0 | 1.353 | 1.22 | 1.29 |

# Euler's Method Example 2

Consider the equations describing the horizontal position of a cart attached to a lossless spring:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1$$

Assuming initial conditions of $x_1(0) = 1$ and $x_2(0) = 0$, the analytic solution is $x_1(t) = \cos t$.

We can again compare the results of the analytic and numerical solutions

# Euler's Method Example 2, cont'd

Starting from the initial conditions at t =0 we next calculate the value of x(t) at time t = 0.25.

$$
\begin{aligned}
x_1(0.25) &= x_1(0) + 0.25\,x_2(0) &&= 1.0 \\
x_2(0.25) &= x_2(0) - 0.25\,x_1(0) &&= -0.25
\end{aligned}
$$

Then we continue on to the next time step, t = 0.50

$$
\begin{aligned}
x_1(0.50) &= x_1(0.25) + 0.25\,x_2(0.25) &&= \\
&= 1.0 + 0.25 \times (-0.25) &&= 0.9375 \\
x_2(0.50) &= x_2(0.25) - 0.25\,x_1(0.25) &&= \\
&= -0.25 - 0.25 \times (1.0) &&= -0.50
\end{aligned}
$$

# Euler's Method Example 2, cont'd

| t | $x_1^{actual}(t)$ | $x_1(t)$ $\Delta t=0.25$ |
|---|---|---|
| 0 | 1 | 1 |
| 0.25 | 0.9689 | 1 |
| 0.50 | 0.8776 | 0.9375 |
| 0.75 | 0.7317 | 0.8125 |
| 1.00 | 0.5403 | 0.6289 |
| … | … | … |
| 10.0 | -0.8391 | -3.129 |
| 100.0 | 0.8623 | -151,983 |

Since we know from the exact solution that $x_1$ is bounded between -1 and 1, clearly the method is numerically unstable

# Euler's Method Example 2, cont'd

Below is a comparison of the solution values for $x_1(t)$ at time $t = 10$ seconds

| $\Delta t$ | $x_1(10)$ |
|---|---|
| **actual** | **-0.8391** |
| **0.25** | **-3.129** |
| **0.10** | **-1.4088** |
| **0.01** | **-0.8823** |
| **0.001** | **-0.8423** |

# Second Order Runge-Kutta Method

- Runge-Kutta methods improve on Euler's method by evaluating **f**(**x**) at selected points over the time step
- Simplest method is the second order method in which

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \frac{1}{2}\left(\mathbf{k}_1 + \mathbf{k}_2\right)$$

where

$$\mathbf{k}_1 = \Delta t\ \mathbf{f}\left(\mathbf{x}(t)\right)$$

$$\mathbf{k}_2 = \Delta t\ \mathbf{f}\left(\mathbf{x}(t) + \mathbf{k}_1\right)$$

- That is, $\mathbf{k}_1$ is what we get from Euler's; $\mathbf{k}_2$ improves on this by reevaluating at the estimated end of the time step

# Second Order Runge-Kutta Algorithm

**$t = 0$, $x(0) = x_0$, $\Delta t$ = step size**

**While $t \leq t^{final}$ Do**

$$k1 = \Delta t \, f(x(t))$$

$$k2 = \Delta t \, f(x(t) + k1)$$

$$x(t+\Delta t) = x(t) + (k1 + k2)/2$$

$$t = t + \Delta t$$

**End While**

# RK2 Oscillating Cart

- Consider the same example from before the position of a cart attached to a lossless spring. Again, with initial conditions of $x_1(0) = 1$ and $x_2(0) = 0$, the analytic solution is $x_1(t) = \cos(t)$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 \end{aligned}$$

- With $\Delta t = 0.25$ at $t = 0$

$$\mathbf{k}_1 = (0.25) \times \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.25 \end{bmatrix}$$

$$\mathbf{x}(0) + \mathbf{k}1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.25 \end{bmatrix}$$

# RK2 Oscillating Cart

$$\mathbf{k}_2 = (0.25) \times \mathbf{f}\left(\mathbf{x}(0) + \mathbf{k}_1\right) = \begin{bmatrix} -0.0625 \\ -0.25 \end{bmatrix}$$

$$\mathbf{x}(0.25) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2}\left(\mathbf{k}_1 + \mathbf{k}_2\right) = \begin{bmatrix} 0.96875 \\ -0.25 \end{bmatrix}$$

# Comparison

- The below table compares the numeric and exact solutions for $x_1(t)$ using the RK2 algorithm

| time | actual $x_1(t)$ | $x_1(t)$ with RK2 $\Delta t=0.25$ |
|---|---|---|
| 0 | 1 | 1 |
| 0.25 | 0.9689 | 0.969 |
| 0.50 | 0.8776 | 0.876 |
| 0.75 | 0.7317 | 0.728 |
| 1.00 | 0.5403 | 0.533 |
| 10.0 | -0.8391 | -0.795 |
| 100.0 | 0.8623 | 1.072 |

# Comparison of $x_1(10)$ for varying $\Delta t$

- The below table compares the $x_1(10)$ values for different values of $\Delta t$; recall with Euler's with $\Delta t$=0.1 was -1.41 and with 0.01 was -0.8823

| $\Delta t$ | $x_1(10)$ |
|---|---|
| actual | -0.8391 |
| 0.25 | -0.7946 |
| 0.10 | -0.8310 |
| 0.01 | -0.8390 |
| 0.001 | -0.8391 |

# RK2 Versus Euler's

- RK2 requires twice the function evaluations per iteration, but gives much better results
- With RK2 the error tends to vary with the cube of the step size, compared with the square of the step size for Euler's
- The smaller error allows for larger step sizes compared to Eulers

# Fourth Order Runge-Kutta

- Other Runge-Kutta algorithms are possible, including the fourth order

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \frac{1}{6}\left( \mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4 \right)$$

where

$$\mathbf{k}_1 = \Delta t \ \mathbf{f}\left(\mathbf{x}(t)\right)$$

$$\mathbf{k}_2 = \Delta t \ \mathbf{f}\left( \mathbf{x}(t) + \frac{1}{2}\mathbf{k}_1 \right)$$

$$\mathbf{k}_3 = \Delta t \ \mathbf{f}\left( \mathbf{x}(t) + \frac{1}{2}\mathbf{k}_2 \right)$$

$$\mathbf{k}_4 = \Delta t \ \mathbf{f}\left(\mathbf{x}(t) + \mathbf{k}_2 \right)$$

# RK4 Oscillating Cart Example

- RK4 gives much better results, with error varying with the time step to the fifth power

| time | actual $x_1(t)$ | $x_1(t)$ with RK4 $\Delta t=0.25$ |
|---|---|---|
| 0 | 1 | 1 |
| 0.25 | 0.9689 | 0.9689 |
| 0.50 | 0.8776 | 0.8776 |
| 0.75 | 0.7317 | 0.7317 |
| 1.00 | 0.5403 | 0.5403 |
| 10.0 | -0.8391 | -0.8392 |
| 100.0 | 0.8623 | 0.8601 |

# Multistep Methods

- Euler's and Runge-Kutta methods are single step approaches, in that they only use information at **x**(t) to determine its value at the next time step
- Multistep methods take advantage of the fact that using we have information about previous time steps **x**(t-$\Delta$t), **x**(t-2$\Delta$t), etc
- These methods can be explicit or implicit (dependent on **x**(t+$\Delta$t) values; we'll just consider the explicit Adams-Bashforth approach

# Multistep Motivation

- In determining $\mathbf{x}$(t+Δt) we could use a Taylor series expansion about $\mathbf{x}$(t)

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\dot{\mathbf{x}}(t) + \frac{\Delta t^2}{2}\ddot{\mathbf{x}}(t) + O(\Delta t^3)$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\mathbf{f}(t) + \frac{\Delta t^2}{2}\left(\frac{\mathbf{f}(\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(t-\Delta t))}{\Delta t} + O(\Delta t)\right)$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\left(\frac{3}{2}\mathbf{f}\left(\mathbf{x}(t)\right) - \frac{1}{2}\mathbf{f}\left(\mathbf{x}(t-\Delta t)\right)\right) + O(\Delta t^3)$$

(note Euler's is just the first two terms on the right-hand side)

# Adams-Bashforth

- What we derived is the second order Adams-Bashforth approach.  Higher order methods are also possible, by approximating subsequent derivatives.  Here we also present the third order Adams-Bashforth

Second Order

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \frac{\Delta t}{2}\left(3\mathbf{f}(\mathrm{x}(t)) - \mathbf{f}(x(t-\Delta t))\right) + O(\Delta t^3)$$

Third Order

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \frac{\Delta t}{12}\left(23\mathbf{f}(\mathrm{x}(t)) - 16\mathbf{f}(x(t-\Delta t)) + 5\mathbf{f}(x(t-2\Delta t))\right) + O(\Delta t^4)$$

# Adams-Bashforth Versus Runge-Kutta

- The key Adams-Bashforth advantage is the approach only requires one function evaluation per time step while the RK methods require multiple evaluations
- A key disadvantage is when discontinuities are encountered, such as with limit violations;
- Another method needs to be used until there are sufficient past solutions
- They also have difficulties if variable time steps are used

# Numerical Instability

- All explicit methods can suffer from numerical instability if the time step is not correctly chosen for the problem eigenvalues

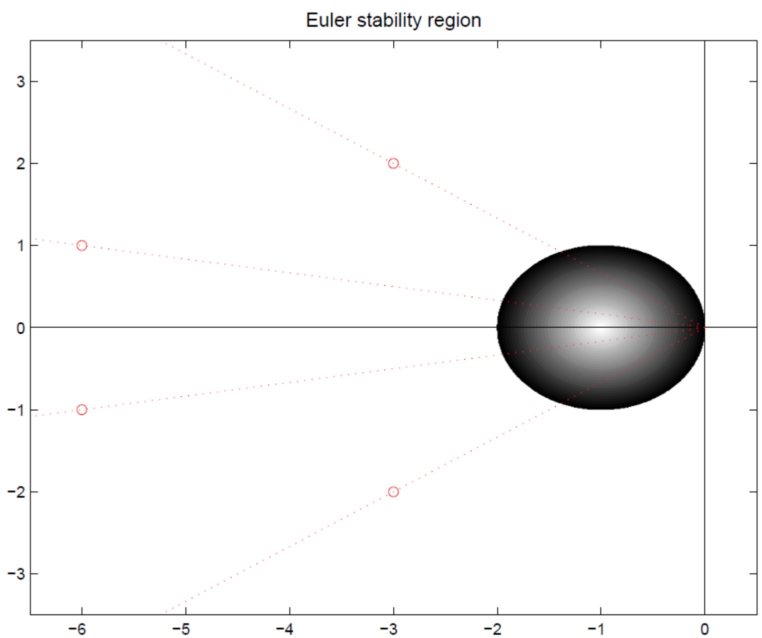



Figure 10.2: The spectrum of $A$ is scaled by $h$. Stability of the origin is recovered if $h\lambda$ is in t[he] region of absolute stability $|1+z|<1$ in the complex plane.

Values are scaled by the time step; the shape for RK2 has similar dimensions but is closer to a square. Key point is to make sure the time step is small enough relative to the eigenvalues

Image source: http://www.staff.science.uu.nl/~frank011/Classes/numwisk/ch10.pdf

# Stiff Differential Equations

- Stiff differential equations are ones in which the desired solution has components the vary quite rapidly relative to the solution
- Stiffness is associated with solution efficiency: in order to account for these fast dynamics we need to take quite small time steps

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -1000x_1 - 1001x_2\end{aligned}$$

$$\dot{\mathbf{x}} \rightarrow = \begin{bmatrix} 0 & 1 \\ -1000 & -1000 \end{bmatrix}\mathbf{x}$$

$$x_1(t) = Ae^{-t} + Be^{-1000t}$$

# Implicit Methods

- Implicit solution methods have the advantage of being numerically stable over the entire left half plane
- Only methods considered here are the is the Backward Euler and Trapezoidal

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t))$$

Then using backward Euler

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\mathbf{A}(\mathbf{x}(t+\Delta t))$$

$$\left[I - \Delta t\,\mathbf{A}\right]\mathbf{x}(t+\Delta t) = \mathbf{x}(t)$$

$$\mathbf{x}(t+\Delta t) = \left[I - \Delta t\,\mathbf{A}\right]^{-1}\mathbf{x}(t)$$

# Implicit Methods

- The obvious difficulty associated with these methods is **x**(t) appears on both sides of the equation
- Easiest to show the solution for the linear case:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t))$$

Then using backward Euler

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\mathbf{A}(\mathbf{x}(t+\Delta t))$$

$$\left[I - \Delta t\,\mathbf{A}\right]\mathbf{x}(t+\Delta t) = \mathbf{x}(t)$$

$$\mathbf{x}(t+\Delta t) = \left[I - \Delta t\,\mathbf{A}\right]^{-1}\mathbf{x}(t)$$

# Backward Euler Cart Example

- Returning to the cart example

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{x}(t))$$

Then using backward Euler with $\Delta t = 0.25$

$$\mathbf{x}(t+\Delta t) = \left[ I - \Delta t\, \mathbf{A} \right]^{-1} \mathbf{x}(t) = \begin{bmatrix} 1 & -0.25 \\ 0.25 & 1 \end{bmatrix}^{-1} \mathbf{x}(t)$$

# Backward Euler Cart Example

- Results with $\Delta t = 0.25$ and 0.05

| time | actual $x_1(t)$ | $x_1(t)$ with $\Delta t=0.25$ | $x_1(t)$ with $\Delta t=0.05$ |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 0.25 | 0.9689 | 0.9411 | 0.9629 |
| 0.50 | 0.8776 | 0.8304 | 0.8700 |
| 0.75 | 0.7317 | 0.6774 | 0.7185 |
| 1.00 | 0.5403 | 0.4935 | 0.5277 |
| 2.00 | -0.416 | -0.298 | -0.3944 |

Note: Just because the method is numerically stable doesn't mean it is error free! RK2 is more accurate than backward Euler.

# Trapezoidal Linear Case

- For the trapezoidal with a linear system we have

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t))$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \frac{\Delta t}{2}\left[\mathbf{A}(\mathbf{x}(t)) + \mathbf{A}(\mathbf{x}(t+\Delta t))\right]$$

$$\left[I - \frac{\Delta t}{2}\mathbf{A}\right]\mathbf{x}(t+\Delta t) = \left[I + \frac{\Delta t}{2}\mathbf{A}\right]\mathbf{x}(t)$$

$$\mathbf{x}(t+\Delta t) = \left[I - \Delta t\,\mathbf{A}\right]^{-1}\left[I + \frac{\Delta t}{2}\mathbf{A}\right]\mathbf{x}(t)$$

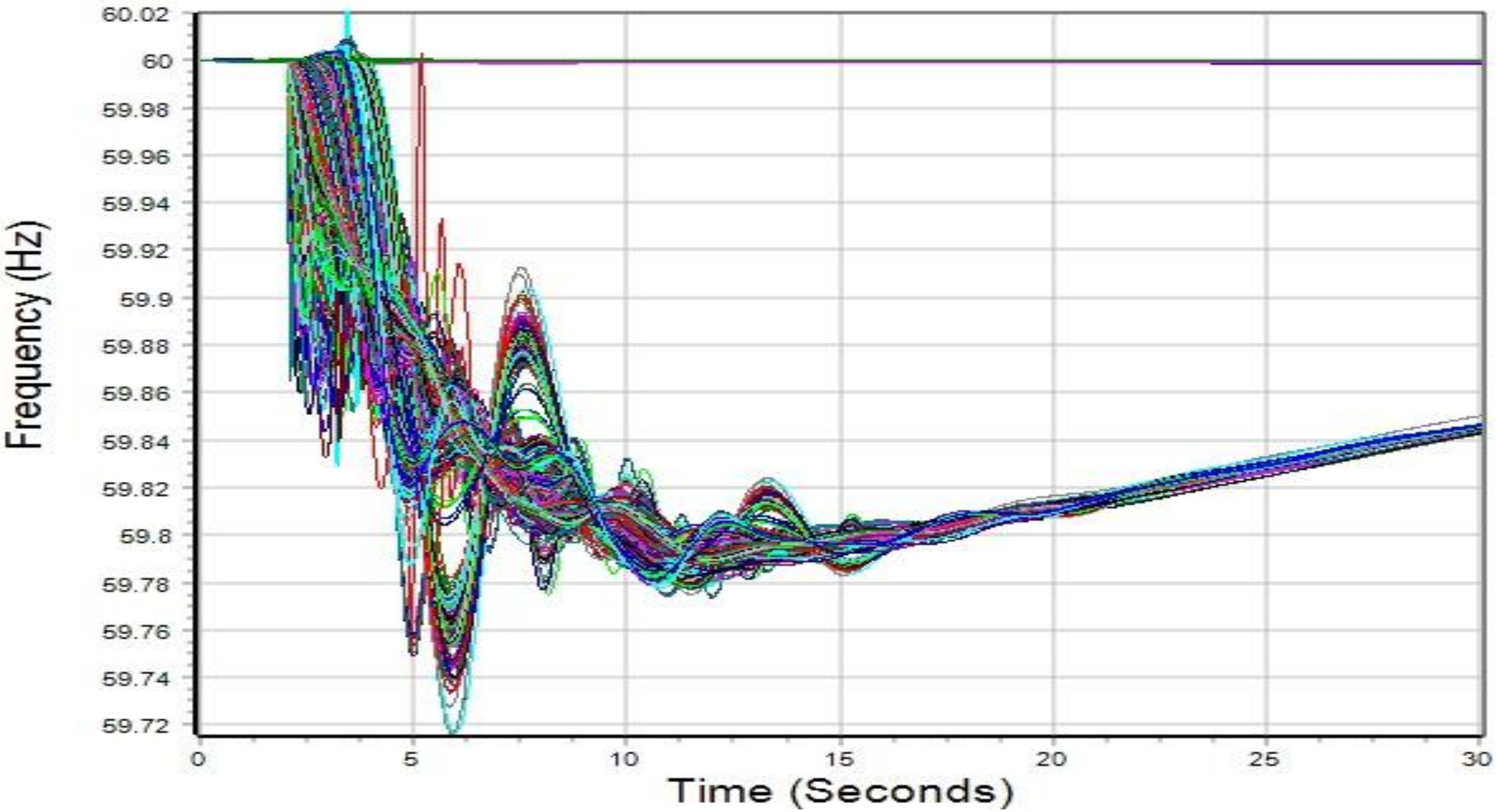
# Trapezoidal Cart Example

- Results with $\Delta t = 0.25$, comparing between backward Euler and trapezoidal

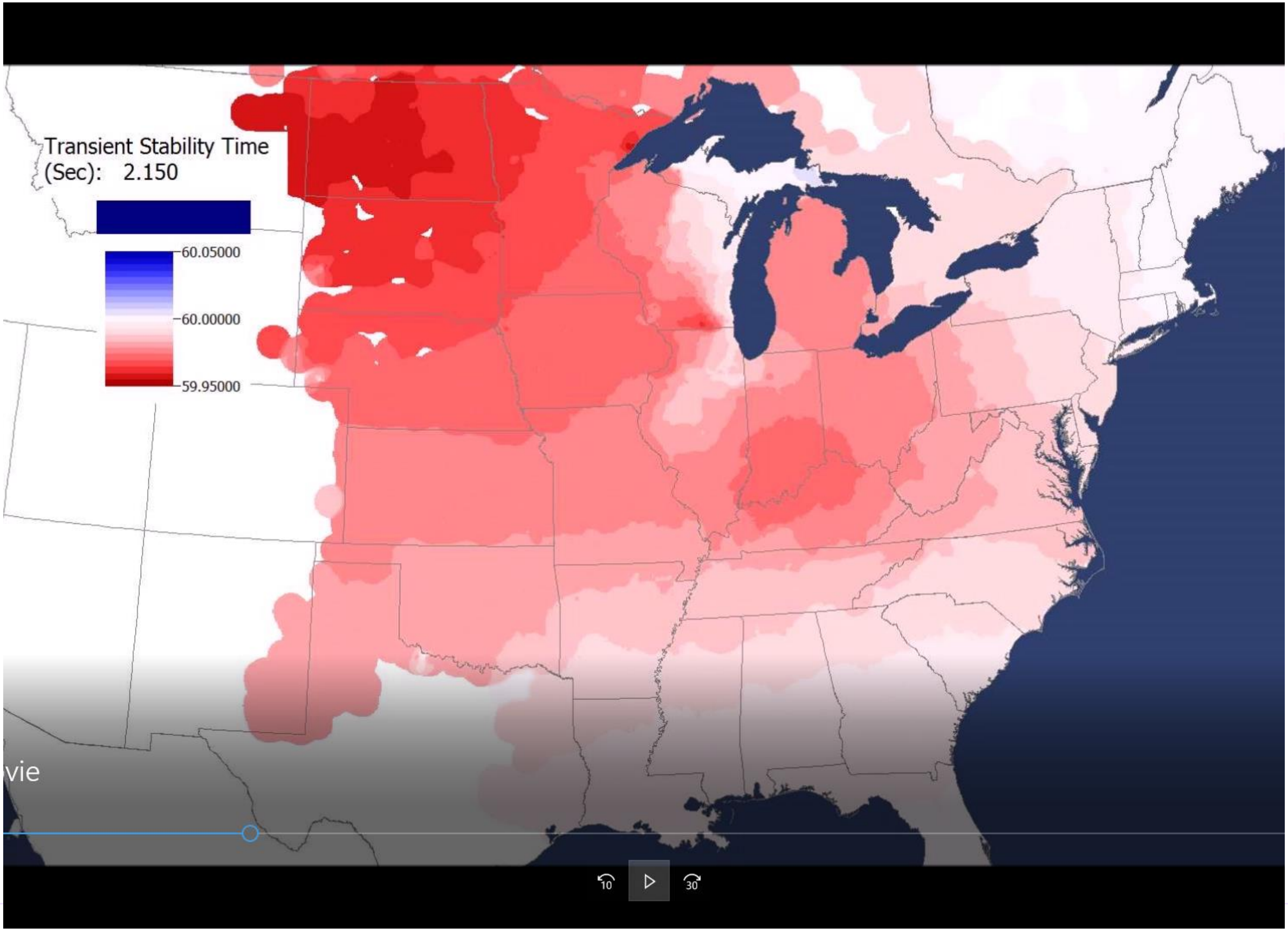
| time | actual $x_1(t)$ | Backward Euler | Trapezoidal |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 0.25 | 0.9689 | 0.9411 | 0.9692 |
| 0.50 | 0.8776 | 0.8304 | 0.8788 |
| 0.75 | 0.7317 | 0.6774 | 0.7343 |
| 1.00 | 0.5403 | 0.4935 | 0.5446 |
| 2.00 | -0.416 | -0.298 | -0.4067 |

# Example Transient Stability Results

- Figure shows simulated generator frequencies after a large generator outage contingency



# Spatial Variation of Frequency Response: EI Model



# Electromagnetic Transients

- The modeling of very fast power system dynamics (much less than one cycle) is known as electromagnetics transients program (EMTP) analysis
  - Covers issues such as lightning propagation and switching surges
- Concept originally developed by Prof. Hermann Dommel for his PhD in the 1960's (now emeritus at Univ. British Columbia)
  - After his PhD work Dr. Dommel worked at BPA where he was joined by Scott Meyer in the early 1970's
  - Alternative Transients Program (ATP) developed in response to commercialization of the BPA code