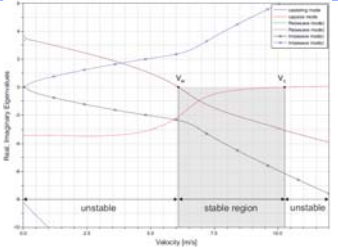
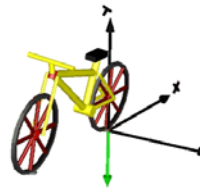


Virtual Physics Equation-Based Modeling

TUM, December 16, 2014

Numerical Stability and Stiffness

Dr. Dirk Zimmer
German Aerospace Center (DLR), Robotics and Mechatronics Centre

Recap: State-Space Form

TUM + DLR
Robotics and Mechatronics Centre

- We have learned that for linear systems, the state-space form can be described by four sub-matrices **A**, **B**, **C**, and **D**.

$$\frac{dx}{dt} = Ax + B(u, t)$$

$$y = Cx + D(u, t)$$

	x_1	x_2	u	t
dx_1/dt	X	X		
dx_2/dt	X	X	X	
y	C	X	D	

© Dirk Zimmer, December 2014, Slide 2

Recap: State-Space Form

TUM + DLR
Robotics and Mechatronics Centre

- For the actual dynamics, the **A** matrix is for primary interest.

$$\frac{dx}{dt} = Ax$$

	x_1	x_2	u	t
dx_1/dt	X	X		
dx_2/dt	X	X	X	
y		X		

© Dirk Zimmer, December 2014, Slide 3

Recap: Stability in 1D Systems

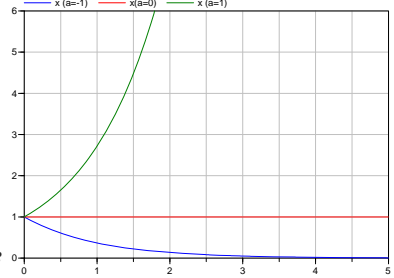
TUM + DLR
Robotics and Mechatronics Centre

Let us look at the general one-dimensional linear system

$$\frac{dx}{dt} = ax \quad (\text{with } x_{\text{start}} = 1)$$

Definition:

- $a > 0$:
The system is **unstable**
- $a < 0$:
The system is **stable**
- $a = 0$:
The system is **marginally stable**



© Dirk Zimmer, December 2014, Slide 4

Recap: Stability in nD Systems

What about the multi-dimensional case?

$$dx/dt = Ax$$

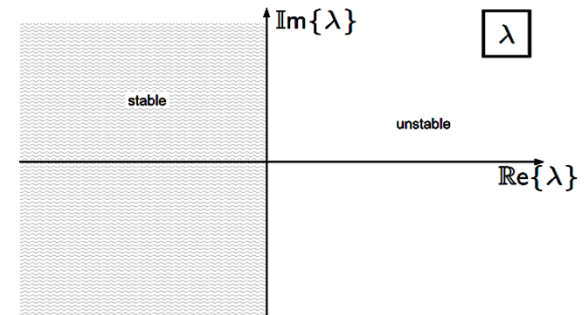
- We can perform an eigenvalue decomposition:

$$A = Q\Lambda Q^{-1}$$

where Q consists in the eigenvectors and Λ is a diagonal matrix containing the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$.

- The system is **asymptotically stable** iff all eigenvalues are smaller than 0.
- The system is **marginally stable** if all eigenvalues are smaller or equal than 0.
- The system is **unstable** otherwise.
- For complex eigenvalues, the real part is decisive. © Dirk Zimmer, December 2014, Slide 5

Domain of Analytical Stability



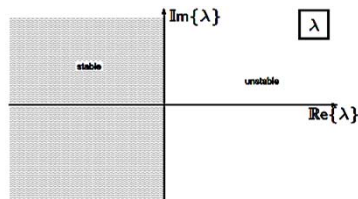
© Dirk Zimmer, December 2014, Slide 6

Analytical vs. Numerical Stability

So far, we have examined the analytical stability of our system. But there is a crucial question for the simulation of a system:

Can the applied scheme for time-integration change the stability of the system?

- Let us start with an analysis of Forward Euler (FE):
- Remember:



© Dirk Zimmer, December 2014, Slide 7

Numerical Stability: FE

Let us apply the Forward Euler scheme on the system:

$$dx/dt = f(x(t),t) = Ax(t)$$

- By discretizing time with step-width h we get:

$$\begin{aligned} \rightarrow x(t+h) &= x(t) + h \cdot f(x(t),t) \\ \rightarrow x(t+h) &= x(t) + h \cdot Ax(t) \\ \rightarrow x(t+h) &= (I + hA) x(t) \end{aligned}$$

© Dirk Zimmer, December 2014, Slide 8

Numerical Stability: FE

Let us apply the Forward Euler scheme on the system:

$$dx/dt = f(x(t),t) = Ax(t)$$

- By discretizing time with step-width h we get:

$$\begin{aligned} \rightarrow x(t+h) &= x(t) + h \cdot f(x(t),t) \\ \rightarrow x(t+h) &= x(t) + h \cdot Ax(t) \\ \rightarrow x(t+h) &= (I + hA) x(t) \end{aligned}$$

- Applying FE transforms the continuous function $x(t)$ into a series x_k .

$$\begin{aligned} \rightarrow x_{k+1} &= (I + hA) x_k \\ \rightarrow x_{k+1} &= Fx_k \quad \text{with} \quad F = I + hA \end{aligned}$$

© Dirk Zimmer, December 2014, Slide 9

Numerical Stability: FE

When is this series numerically stable?

$$x_{k+1} = Fx_k$$

- This is the case when all eigenvalues of F are located inside the unit circle of complex numbers.
- Each integration scheme has its own formula for F . In case of FE, it is:

$$F = I + hA$$

© Dirk Zimmer, December 2014, Slide 10

Numerical Stability: FE

When is this series numerically stable?

$$x_{k+1} = Fx_k$$

- This is the case when all eigenvalues of F are located inside the unit circle of complex numbers.
- Each integration scheme has its own formula for F . In case of FE, it is:

$$F = I + hA$$

- We would like to express the eigenvalues of F in terms of the eigenvalues of A

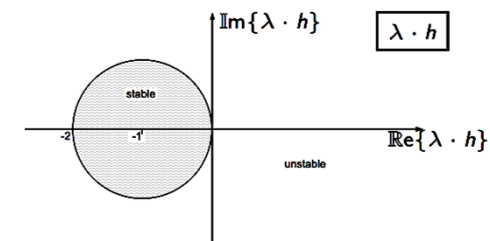
$$\begin{aligned} \rightarrow F &= I + hA \\ \rightarrow F &= QIQ^{-1} + hQAQ^{-1} \\ \rightarrow F &= Q(I + hA)Q^{-1} \end{aligned}$$

- By assuming $h=1$, the eigenvalues of F are the eigenvalues of A plus $1.0 + 0i$.

© Dirk Zimmer, December 2014, Slide 11

Numerical Stability: FE

By applying FE, the stability region of the system A has been reduced from the half-plane to a circle of radius $1/h$ (in the plot, the axes are scaled by h).

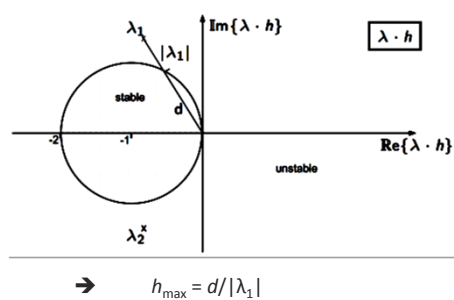


- Forward Euler is potentially destabilizing.

© Dirk Zimmer, December 2014, Slide 12

Numerical Stability: FE

To get a qualitative correct result for a stable system, we have to choose h small enough, so that all eigenvalues are located inside the stable region.



© Dirk Zimmer, December 2014, Slide 13

Backward Euler

There is an alternative Euler scheme: Backward Euler.

- In contrast to Forward Euler, we take the time-derivative of $t+h$:

$$\dot{\mathbf{x}}(t+h) = \mathbf{x}(t) + h \cdot \mathbf{f}(\mathbf{x}(t+h), t+h)$$

- If we use Backward Euler to integrate forward in time, we have to solve this equation system for $\mathbf{x}(t+h)$ by an iterative solver (Newton's method or similar) with the residual equation:

$$0 = \mathbf{x}(t) + h \cdot \mathbf{f}(\mathbf{x}(t+h), t+h) - \mathbf{x}(t+h)$$

- In contrast to Forward Euler, there is no **explicit** computational form available. Hence Backward Euler is an **implicit** method. (FE is an explicit method)

© Dirk Zimmer, December 2014, Slide 14

Numerical Stability: BE

What is the numerical stability domain of Backward Euler?

- The answer can be derived without needing any formulas.
- If we compute backwards in time, stable systems become unstable and vice versa.
- If we compute backwards in time, BE becomes computational identical to FE (computed forward in time)
- Hence the stability domain is simply flipped to the other side.

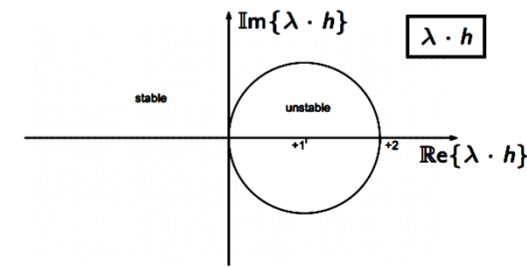
$$\begin{aligned} \rightarrow & \quad \dot{\mathbf{x}}(t+h) = \mathbf{x}(t) + h \cdot \mathbf{f}(\mathbf{x}(t+h), t+h) \\ \rightarrow & \quad \mathbf{x}(t+h) = \mathbf{x}(t) + h \cdot \mathbf{A}\mathbf{x}(t+h) \\ \rightarrow & \quad (\mathbf{I} - h\mathbf{A}) \mathbf{x}(t+h) = \mathbf{x}(t) \\ \rightarrow & \quad \mathbf{x}(t+h) = (\mathbf{I} - h\mathbf{A})^{-1} \mathbf{x}(t) \end{aligned}$$

$$\begin{aligned} \rightarrow & \quad \mathbf{x}_{k+1} = (\mathbf{I} - h\mathbf{A})^{-1} \mathbf{x}_k \\ \rightarrow & \quad \mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k \quad \text{with} \quad \mathbf{F} = (\mathbf{I} - h\mathbf{A})^{-1} \end{aligned}$$

© Dirk Zimmer, December 2014, Slide 15

Numerical Stability: BE

By applying BE, the unstable region in the right-half plane collapses to a circle of radius $1/h$.

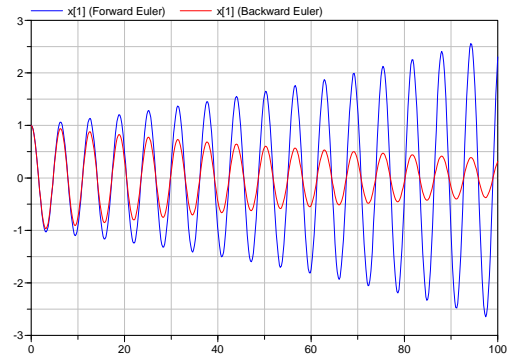


- Backward Euler causes a numerical stabilization.

© Dirk Zimmer, December 2014, Slide 16

FE vs. BE

FE and BE ($h=0.02s$) are applied on a marginally stable system.



© Dirk Zimmer, December 2014, Slide 17

Fight the Drift

For Real-Time Applications that use FE, it is sometimes possible to fight the destabilizing drift and get better solutions.

- The following system represents a marginally stable spring-damper system:

$$\begin{aligned} dv/dt &= -s \cdot C/M \\ ds/dt &= v \end{aligned}$$

- The total energy in this system is the potential energy in the spring plus the kinetic energy:

$$E = \frac{1}{2} \cdot C \cdot s^2 + \frac{1}{2} \cdot M \cdot v^2$$

© Dirk Zimmer, December 2014, Slide 18

Fight the Drift

- We know that energy is conserved hence:

$$E = \frac{1}{2} \cdot C \cdot s^2 + \frac{1}{2} \cdot M \cdot v^2 = \text{const}$$

- This means that s and v are always placed on an ellipse. To fight the drift, we can project on this ellipse after each integration step. For this example, it is particularly easy.

- We compute a scaling factor out the current total energy E and the initial total energy E_0 ...

$$p = (E_0/E)^{1/2}$$

- ...and apply this factor to s and v

$$s' = s \cdot p$$

$$v' = v \cdot p$$

© Dirk Zimmer, December 2014, Slide 19

Fight the Drift

This is the corresponding Python Code

```
open file for ouput
fh = open("out.dat","w")

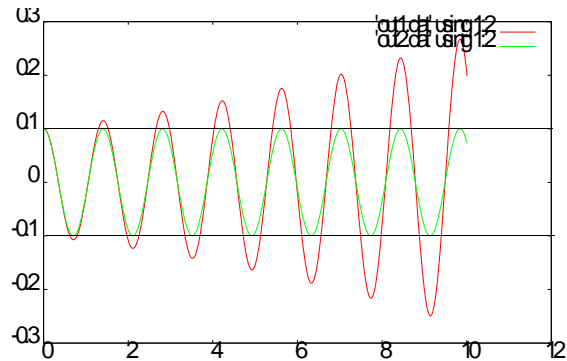
#perform time-integration
while time < tStop:
    dv_dt = -s*C/M
    ds_dt = v
    v += h*dv_dt
    s += h*ds_dt
    E = 0.5*C*s*s + 0.5*M*v*v
    p = sqrt(E0/E)
    s *= p
    v *= p
    time += h
    print(time,"t",s,"t",v,file=fh)

print("See out.dat for simulation result")
fh.close()
```

© Dirk Zimmer, December 2014, Slide 20

Fight the Drift

And here the simulation results for FE and FE with projection.



© Dirk Zimmer, December 2014, Slide 21

FE vs. BE

Forward Euler and Backward Euler are the origin of all higher-order methods

- All explicit higher-order methods degenerate to FE for $h \rightarrow 0$ (supposing a finite numerical precision).
- All implicit higher-order methods degenerate to BE for $h \rightarrow 0$ (supposing a finite numerical precision).
- Most explicit and implicit methods share somehow their behavior w.r.t stability with their first-order counterparts.
- Both FE and BE are totally unsuited for marginally stable systems, since the stability domain shares only a single point with the imaginary axis. Fortunately, there are higher-order methods that share larger fractions of their stability domain with the imaginary axis.

© Dirk Zimmer, December 2014, Slide 22

Stiff Systems

What are stiff systems?

- We call a linear system stiff, if it is stable and its eigenvalues vary a lot in terms of their real parts.
- Non-linear systems are called stiff, if they are stable and exhibit both fast and slow modes in their behavior. The linearization of such systems leads to stiff linear systems.
- These systems cannot be simulated efficiently by means of any explicit method (FE), because we would need very small time steps to move those eigenvalues (of either the system itself or of its linearization) into the numerical stability domain.
- **For the efficient simulation of stiff systems, implicit integration algorithms are required!**

© Dirk Zimmer, December 2014, Slide 23

Stiff Systems: Example

Example:

$$A = \begin{pmatrix} -5.05 & -4.95 \\ -4.95 & -5.05 \end{pmatrix}$$

- We can perform an eigenvalue decomposition:

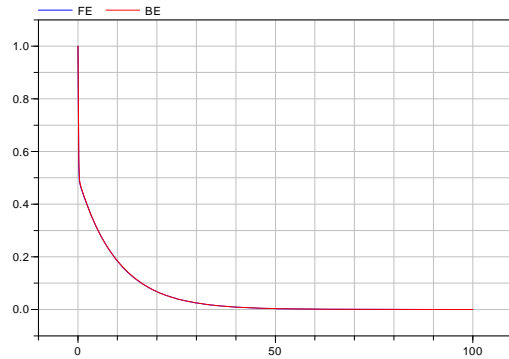
$$A = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \begin{pmatrix} -10 & 0 \\ 0 & -0.1 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$$

- The system is stable and the eigenvalues are separation by a factor 100. The system is stiff.
- We are going to simulate the system using FE and BE for different step sizes.

© Dirk Zimmer, December 2014, Slide 24

Stiff Systems: FE vs. BE

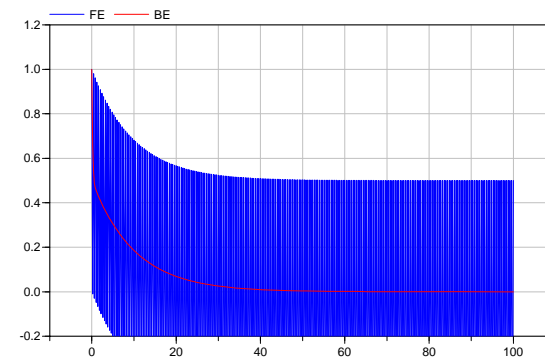
FE and BE are applied on the stiff system ($h=0.02$).



© Dirk Zimmer, December 2014, Slide 25

Stiff Systems: FE vs. BE

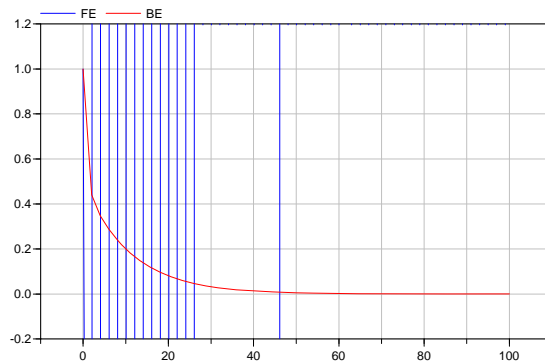
FE and BE are applied on the stiff system ($h=0.2$).



© Dirk Zimmer, December 2014, Slide 26

Stiff Systems: FE vs. BE

FE and BE are applied on the stiff system ($h=2$).



© Dirk Zimmer, December 2014, Slide 27

Questions ?