Numerical Simulation of Dynamic Systems IX

Prof. Dr. François E. Cellier
Department of Computer Science
ETH Zurich

March 26, 2013

Introduction

Given a mechanical system:

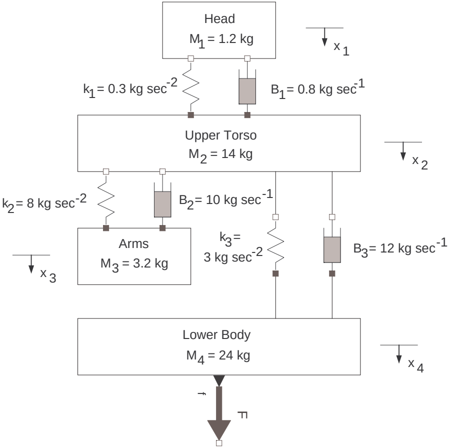


Figure: Mechanical model of a sitting human body

Introduction II

We can obtain a set of differential equations using Newton's law:

$$
\begin{aligned}
M_1 \cdot \ddot{x}_1 &= k_1 \cdot (x_2 - x_1) + B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_2 \cdot \ddot{x}_2 &= k_2 \cdot (x_3 - x_2) + B_2 \cdot (\dot{x}_3 - \dot{x}_2) + k_3 \cdot (x_4 - x_2) \\
&\quad + B_3 \cdot (\dot{x}_4 - \dot{x}_2) - k_1 \cdot (x_2 - x_1) - B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_3 \cdot \ddot{x}_3 &= -k_2 \cdot (x_3 - x_2) - B_2 \cdot (\dot{x}_3 - \dot{x}_2) \\
M_4 \cdot \ddot{x}_4 &= F - k_3 \cdot (x_4 - x_2) - B_3 \cdot (\dot{x}_4 - \dot{x}_2)
\end{aligned}
$$

Introduction II

We can obtain a set of differential equations using Newton's law:

$$
\begin{aligned}
M_1 \cdot \ddot{x}_1 &= k_1 \cdot (x_2 - x_1) + B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_2 \cdot \ddot{x}_2 &= k_2 \cdot (x_3 - x_2) + B_2 \cdot (\dot{x}_3 - \dot{x}_2) + k_3 \cdot (x_4 - x_2) \\
&\quad + B_3 \cdot (\dot{x}_4 - \dot{x}_2) - k_1 \cdot (x_2 - x_1) - B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_3 \cdot \ddot{x}_3 &= -k_2 \cdot (x_3 - x_2) - B_2 \cdot (\dot{x}_3 - \dot{x}_2) \\
M_4 \cdot \ddot{x}_4 &= F - k_3 \cdot (x_4 - x_2) - B_3 \cdot (\dot{x}_4 - \dot{x}_2)
\end{aligned}
$$

Typically, DAE models derived from mechanical systems contain second derivatives.

Introduction II

We can obtain a set of differential equations using Newton's law:

$$\begin{aligned}
M_1 \cdot \ddot{x}_1 &= k_1 \cdot (x_2 - x_1) + B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_2 \cdot \ddot{x}_2 &= k_2 \cdot (x_3 - x_2) + B_2 \cdot (\dot{x}_3 - \dot{x}_2) + k_3 \cdot (x_4 - x_2) \\
&\quad + B_3 \cdot (\dot{x}_4 - \dot{x}_2) - k_1 \cdot (x_2 - x_1) - B_1 \cdot (\dot{x}_2 - \dot{x}_1) \\
M_3 \cdot \ddot{x}_3 &= -k_2 \cdot (x_3 - x_2) - B_2 \cdot (\dot{x}_3 - \dot{x}_2) \\
M_4 \cdot \ddot{x}_4 &= F - k_3 \cdot (x_4 - x_2) - B_3 \cdot (\dot{x}_4 - \dot{x}_2)
\end{aligned}$$

Typically, DAE models derived from mechanical systems contain second derivatives.

In general, we can obtain models of the form:

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, t)$$

Introduction III

Although it is always possible to convert such models to state-space form by augmenting the state vector by the velocity vector $\mathbf{v} = \dot{\mathbf{x}}$:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{f}(\mathbf{x}, \mathbf{v}, \mathbf{u}, t) \end{aligned}$$

this may not necessarily be desirable.

Introduction III

Although it is always possible to convert such models to state-space form by augmenting the state vector by the velocity vector $\mathbf{v} = \dot{\mathbf{x}}$:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{f}(\mathbf{x}, \mathbf{v}, \mathbf{u}, t) \end{aligned}$$

this may not necessarily be desirable.

It may be worthwhile to investigate whether we could find numerical ODE solvers that can deal with second-derivative models directly. This is the purpose of today's presentation.

Introduction IV

We can reformulate the human body model in a matrix vector form using the *partial state vector* $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$:

$$\mathbf{M} \cdot \ddot{\mathbf{x}} + \mathbf{C} \cdot \dot{\mathbf{x}} + \mathbf{K} \cdot \mathbf{x} = \mathbf{f}$$

where:

$$\mathbf{M} = \begin{pmatrix} M_1 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 \\ 0 & 0 & M_3 & 0 \\ 0 & 0 & 0 & M_4 \end{pmatrix} \quad ; \quad \mathbf{C} = \begin{pmatrix} B_1 & -B_1 & 0 & 0 \\ -B_1 & (B_1 + B_2 + B_3) & 0 & 0 \\ 0 & -B_2 & B_2 & 0 \\ 0 & -B_3 & 0 & B_3 \end{pmatrix}$$

$$\mathbf{K} = \begin{pmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & (k_1 + k_2 + k_3) & 0 & 0 \\ 0 & -k_2 & k_2 & 0 \\ 0 & -k_3 & 0 & k_3 \end{pmatrix} \quad ; \quad \mathbf{f} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ F \end{pmatrix}$$

$\mathbf{M}$ is the *mass matrix*, $\mathbf{C}$ is the *damping matrix*, $\mathbf{K}$ is the *stiffness matrix*, and $\mathbf{f}$ is the *vector of (generalized) forces*.

Introduction V

The mass matrix turned out to be a diagonal matrix in this example, but this is only true, because no rotational motions were considered in the given example. Generally, this will not be the case.

Assuming that the mass matrix is non-singular, i.e., there are as many mechanical degrees of freedom in the system as were formulated into second-order differential equations, i.e., there are no *structural singularities* in the model, the model can be solved for the highest derivatives:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x} + \mathbf{B} \cdot \dot{\mathbf{x}} + \mathbf{u}$$

where:

$$\begin{aligned} \mathbf{A} &= \sqrt{-\mathbf{M}^{-1} \cdot \mathbf{K}} \\ \mathbf{B} &= -\mathbf{M}^{-1} \cdot \mathbf{C} \\ \mathbf{u} &= \mathbf{M}^{-1} \cdot \mathbf{f} \end{aligned}$$

Introduction VI

Of special interest is the case of the *conservative (i.e., friction-less) systems* with the second-derivative form:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x} + \mathbf{u}$$

and especially, we may want to look at homogeneous, conservative, linear systems with the second-derivative model:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x}$$

Velocity-free Models

We shall define a *velocity-free model* as one that satisfies, in the linear case, the differential vector equation:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x} + \mathbf{u}$$

and, in the non-linear case, the differential vector equation:

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

Velocity-free Models

We shall define a *velocity-free model* as one that satisfies, in the linear case, the differential vector equation:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x} + \mathbf{u}$$

and, in the non-linear case, the differential vector equation:

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

Every conservative system leads to a velocity-free second-derivative model. Yet, not every velocity-free second-derivative model is conservative.

Velocity-free Models II

Given a linear, time-invariant, homogeneous state-space model of the form:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

Depending on the eigenvalues of $\mathbf{A}$, the system is either damped or undamped, stable or unstable.

Velocity-free Models II

Given a linear, time-invariant, homogeneous state-space model of the form:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

Depending on the eigenvalues of $\mathbf{A}$, the system is either damped or undamped, stable or unstable.

We can differentiate the state-space model, leading to:

$$\ddot{\mathbf{x}} = \mathbf{A} \cdot \dot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x}$$

Velocity-free Models II

Given a linear, time-invariant, homogeneous state-space model of the form:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

Depending on the eigenvalues of $\mathbf{A}$, the system is either damped or undamped, stable or unstable.

We can differentiate the state-space model, leading to:

$$\ddot{\mathbf{x}} = \mathbf{A} \cdot \dot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x}$$

Any linear, time-invariant, homogeneous state-space model can also be written in the form of a velocity-free second-derivative model, irrespective of where its eigenvalues are located. Yet, a conservative linear system has its eigenvalues spread up and down along the imaginary axis of the complex plane.

Velocity-free Models III

How can the special structure of a velocity-free second-derivative model be exploited by a simulation algorithm?

Velocity-free Models III

How can the special structure of a velocity-free second-derivative model be exploited by a simulation algorithm?

We start by developing the solution vector at time $(t+h)$ into a Taylor series around time t:

$$\mathbf{x_{k+1}} = \mathbf{x_k} + h \cdot \dot{\mathbf{x}}_\mathbf{k} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_\mathbf{k} + \frac{h^3}{6} \cdot \mathbf{x_k}^{(iii)} + \frac{h^4}{24} \cdot \mathbf{x_k}^{(iv)} + \dots$$

We also need to develop the solution vector at time $(t-h)$ into a Taylor series around time t:

$$\mathbf{x_{k-1}} = \mathbf{x_k} - h \cdot \dot{\mathbf{x}}_\mathbf{k} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_\mathbf{k} - \frac{h^3}{6} \cdot \mathbf{x_k}^{(iii)} + \frac{h^4}{24} \cdot \mathbf{x_k}^{(iv)} \mp \dots$$

Velocity-free Models III

How can the special structure of a velocity-free second-derivative model be exploited by a simulation algorithm?

We start by developing the solution vector at time $(t+h)$ into a Taylor series around time t:

$$\mathbf{x_{k+1}} = \mathbf{x_k} + h \cdot \dot{\mathbf{x}}_\mathbf{k} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_\mathbf{k} + \frac{h^3}{6} \cdot \mathbf{x_k}^{(iii)} + \frac{h^4}{24} \cdot \mathbf{x_k}^{(iv)} + \ldots$$

We also need to develop the solution vector at time $(t-h)$ into a Taylor series around time t:

$$\mathbf{x_{k-1}} = \mathbf{x_k} - h \cdot \dot{\mathbf{x}}_\mathbf{k} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_\mathbf{k} - \frac{h^3}{6} \cdot \mathbf{x_k}^{(iii)} + \frac{h^4}{24} \cdot \mathbf{x_k}^{(iv)} \mp \ldots$$

Adding these two equations together, we obtain:

$$\mathbf{x_{k+1}} + \mathbf{x_{k-1}} = 2 \cdot \mathbf{x_k} + h^2 \cdot \ddot{\mathbf{x}}_\mathbf{k} + \frac{h^4}{12} \cdot \mathbf{x_k}^{(iv)} + \ldots$$

Velocity-free Models IV

Truncated after the quadratic term:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + h^2 \cdot \ddot{\mathbf{x}}_\mathbf{k}$$

Velocity-free Models IV

Truncated after the quadratic term:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + h^2 \cdot \ddot{\mathbf{x}}_\mathbf{k}$$

We just found a *3^{rd}–order accurate explicit linear multi-step method* that makes use of the second derivative directly. In some references, the method is referred to as *Godunov's method (GE3)*.

Velocity-free Models IV

Truncated after the quadratic term:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + h^2 \cdot \ddot{\mathbf{x}}_\mathbf{k}$$

We just found a 3^{rd}*–order accurate explicit linear multi-step method* that makes use of the second derivative directly. In some references, the method is referred to as *Godunov's method (GE3)*.

Plugging in the homogeneous linear second-derivative model:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + (\mathbf{A} \cdot h)^2 \cdot \mathbf{x_k}$$

Velocity-free Models IV

Truncated after the quadratic term:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + h^2 \cdot \ddot{\mathbf{x}}_\mathbf{k}$$

We just found a *3rd–order accurate explicit linear multi-step method* that makes use of the second derivative directly. In some references, the method is referred to as *Godunov's method (GE3)*.

Plugging in the homogeneous linear second-derivative model:

$$\mathbf{x_{k+1}} = 2 \cdot \mathbf{x_k} - \mathbf{x_{k-1}} + (\mathbf{A} \cdot h)^2 \cdot \mathbf{x_k}$$

Let:

$$\xi_\mathbf{k} = \begin{pmatrix} \mathbf{x_{k-1}} \\ \mathbf{x_k} \end{pmatrix}$$

Then:

$$\xi_\mathbf{k+1} \approx \mathbf{F} \cdot \xi_\mathbf{k} \quad ; \quad \mathbf{F} = \begin{pmatrix} \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} \\ -\mathbf{I}^{(n)} & [2 \cdot \mathbf{I}^{(n)} + (\mathbf{A} \cdot h)^2] \end{pmatrix}$$

Stability and Damping of GE3

Before we attempt to draw a stability domain of GE3, we shall draw the linear damping plot:

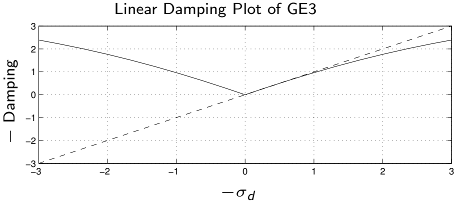


Figure: Linear damping plot of GE3 algorithm

Stability and Damping of GE3

Before we attempt to draw a stability domain of GE3, we shall draw the linear damping plot:

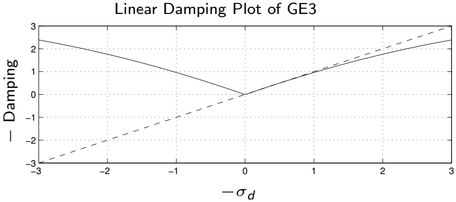


Figure: Linear damping plot of GE3 algorithm

How very disappointing! The scheme is unstable in the left half plane!

Stability and Damping of GE3

Before we attempt to draw a stability domain of GE3, we shall draw the linear damping plot:

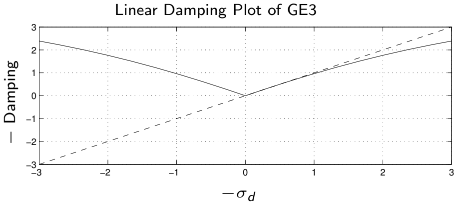


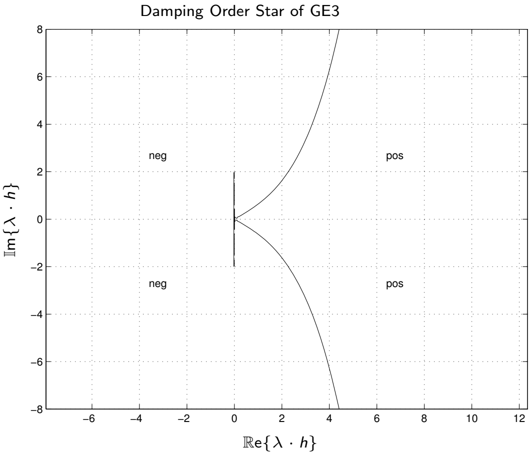
Figure: Linear damping plot of GE3 algorithm

How very disappointing! The scheme is unstable in the left half plane!

The result should not surprise us too much. Since the **F**-matrix is an *even function* in $\mathbf{A} \cdot h$, the damping properties must be symmetric to the imaginary axis. Thus there cannot exist an asymptotic region around the origin, as we would expect of any well-behaved integration algorithm.

Stability and Damping of GE3 II

To gain a better understanding of the damping properties of the algorithm, let us plot the damping order star.



Stability and Damping of GE3 III

Interesting is the line segment stretching from $-2j$ to $+2j$ along the imaginary axis. Evidently, there is zero damping along this line segment, which is exactly, what it should be. To verify the results, let us plot the linear damping properties once more, but this time along the imaginary rather than the real axis.

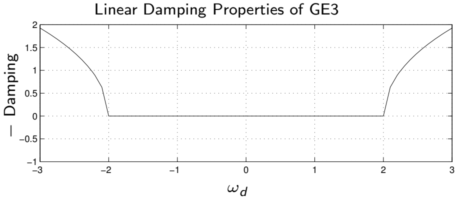


Figure: Linear damping properties of GE3 along imaginary axis

Stability and Damping of GE3 III

Interesting is the line segment stretching from $-2j$ to $+2j$ along the imaginary axis. Evidently, there is zero damping along this line segment, which is exactly, what it should be. To verify the results, let us plot the linear damping properties once more, but this time along the imaginary rather than the real axis.

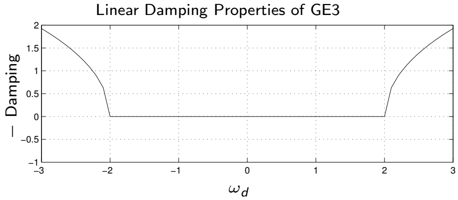


Figure: Linear damping properties of GE3 along imaginary axis

The GE3 algorithm is only useful for strictly conservative systems.

Stability and Damping of GE3 III

Interesting is the line segment stretching from $-2j$ to $+2j$ along the imaginary axis. Evidently, there is zero damping along this line segment, which is exactly, what it should be. To verify the results, let us plot the linear damping properties once more, but this time along the imaginary rather than the real axis.

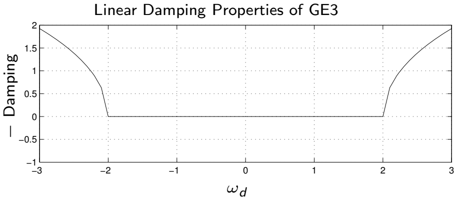


Figure: Linear damping properties of GE3 along imaginary axis

The GE3 algorithm is only useful for strictly conservative systems.

In order to obtain marginally stable results, the largest absolute eigenvalue multiplied by the step size must be smaller than or equal to 2:

$$|\lambda|_{\max} \cdot h \leq 2$$

Stability and Damping of GE3 IV

Let us now plot the linear frequency properties of GE3 along the imaginary axis.

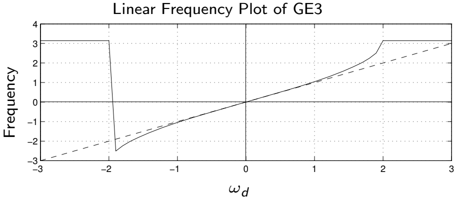


Figure: Linear frequency plot of GE3

Stability and Damping of GE3 IV

Let us now plot the linear frequency properties of GE3 along the imaginary axis.

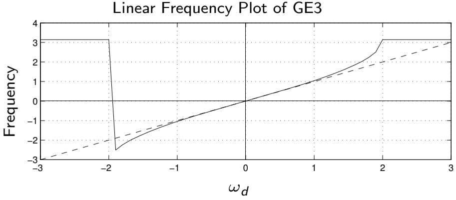


Figure: Linear frequency plot of GE3

The algorithm produces results that are decently accurate for:

$$|\lambda|_{\max} \cdot h \leq 1$$

Stability and Damping of GE3 V

We can draw the following conclusions:

Stability and Damping of GE3 V

We can draw the following conclusions:

- Since the **GE3 algorithm** is numerically unstable everywhere in the left-half complex plane, the method cannot be used for the simulation of damped mechanical systems.

Stability and Damping of GE3 V

We can draw the following conclusions:

- Since the **GE3 algorithm** is numerically unstable everywhere in the left-half complex plane, the method cannot be used for the simulation of damped mechanical systems.

- The algorithm is therefore only useful for the simulation of *linear conservation laws*, i.e., either *linear mechanical systems that are strictly conservative* or *linear hyperbolic partial differential equations (PDEs)*, i.e., the *wave equation*. We shall deal with the simulation of *distributed parameter systems* in the next chapter.

Stability and Damping of GE3 V

We can draw the following conclusions:

- Since the **GE3 algorithm** is numerically unstable everywhere in the left-half complex plane, the method cannot be used for the simulation of damped mechanical systems.

- The algorithm is therefore only useful for the simulation of *linear conservation laws*, i.e., either *linear mechanical systems that are strictly conservative* or *linear hyperbolic partial differential equations (PDEs)*, i.e., the *wave equation*. We shall deal with the simulation of *distributed parameter systems* in the next chapter.

- These results are unfortunately not very exiting, because the restrictions are too severe. How often will it happen that I wish to simulate a pure linear wave equation or a strictly conservative linear mechanical system?

Linear Velocity Models

Given:

$$\ddot{\mathbf{x}} + \mathbf{B} \cdot \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

Linear Velocity Models

Given:

$$\ddot{\mathbf{x}} + \mathbf{B} \cdot \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

We apply the following variable transformation:

$$\xi = \exp\left(\frac{\mathbf{B} \cdot t}{2}\right) \cdot \mathbf{x}$$

Therefore:

$$\begin{aligned}
\mathbf{x} &= \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \xi \\
\dot{\mathbf{x}} &= -\frac{\mathbf{B}}{2} \cdot \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \xi + \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \dot{\xi} \\
\ddot{\mathbf{x}} &= \frac{\mathbf{B}^2}{4} \cdot \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \xi - \mathbf{B} \cdot \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \dot{\xi} + \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right) \cdot \ddot{\xi}
\end{aligned}$$

Linear Velocity Models II

With the abbreviation:

$$\mathbf{E} = \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right)$$

we find:

$$\begin{aligned}
\mathbf{x} &= \mathbf{E} \cdot \xi \\
\dot{\mathbf{x}} &= -\frac{\mathbf{B}}{2} \cdot \mathbf{E} \cdot \xi + \mathbf{E} \cdot \dot{\xi} \\
\ddot{\mathbf{x}} &= \frac{\mathbf{B}^2}{4} \cdot \mathbf{E} \cdot \xi - \mathbf{B} \cdot \mathbf{E} \cdot \dot{\xi} + \mathbf{E} \cdot \ddot{\xi}
\end{aligned}$$

Linear Velocity Models II

With the abbreviation:

$$\mathbf{E} = \exp\left(\frac{-\mathbf{B} \cdot t}{2}\right)$$

we find:

$$\begin{aligned}
\mathbf{x} &= \mathbf{E} \cdot \xi \\
\dot{\mathbf{x}} &= -\frac{\mathbf{B}}{2} \cdot \mathbf{E} \cdot \xi + \mathbf{E} \cdot \dot{\xi} \\
\ddot{\mathbf{x}} &= \frac{\mathbf{B}^2}{4} \cdot \mathbf{E} \cdot \xi - \mathbf{B} \cdot \mathbf{E} \cdot \dot{\xi} + \mathbf{E} \cdot \ddot{\xi}
\end{aligned}$$

Plugging these expressions into the *linear-velocity second-derivative model*:

$$\ddot{\xi} = \mathbf{E}^{-1} \cdot \frac{\mathbf{B}^2}{4} \cdot \mathbf{E} \cdot \xi + \mathbf{E}^{-1} \cdot \mathbf{f}(\mathbf{E} \cdot \xi, \mathbf{u}, t)$$

we convert this model to a *velocity-free second-derivative model*.

Linear Velocity Models III

We draw the following conclusions:

Linear Velocity Models III

We draw the following conclusions:

- Although it is possible to convert any *linear velocity model* mathematically into an equivalent *velocity-free model*, the conversion is not helpful, because the resulting velocity-free model is not strictly conservative, i.e., does not have the eigenvalues of its Jacobian located on the imaginary axis at all times.

Linear Velocity Models III

We draw the following conclusions:

- Although it is possible to convert any *linear velocity model* mathematically into an equivalent *velocity-free model*, the conversion is not helpful, because the resulting velocity-free model is not strictly conservative, i.e., does not have the eigenvalues of its Jacobian located on the imaginary axis at all times.

- Yet, we cannot simulate arbitrary velocity-free models using the **GE3 algorithm**, but only the small sub-class of *linear conservation laws*.

Linear Velocity Models III

We draw the following conclusions:

- Although it is possible to convert any *linear velocity model* mathematically into an equivalent *velocity-free model*, the conversion is not helpful, because the resulting velocity-free model is not strictly conservative, i.e., does not have the eigenvalues of its Jacobian located on the imaginary axis at all times.
- Yet, we cannot simulate arbitrary velocity-free models using the **GE3 algorithm**, but only the small sub-class of *linear conservation laws*.
- **Godunov** was justified in his approach, because he was explicitly and exclusively interested in the simulation of linear conservation laws, and for this special problem, the GE3 algorithm is ideally suited.

Conclusions

- In this presentation, we have started to look at a new class of linear multi-step algorithms, specifically designed for the simulation of second derivative systems, i.e., models that contain the second derivatives of the partial state vector explicitly in their model equations.

Conclusions

- In this presentation, we have started to look at a new class of linear multi-step algorithms, specifically designed for the simulation of second derivative systems, i.e., models that contain the second derivatives of the partial state vector explicitly in their model equations.

- Although it is always possible to convert such models to state-space form and deal with them using any of the numerical ODE solvers introduced in earlier presentations, it could possibly be advantageous to simulate the second-derivative system directly.

Conclusions

- In this presentation, we have started to look at a new class of linear multi-step algorithms, specifically designed for the simulation of second derivative systems, i.e., models that contain the second derivatives of the partial state vector explicitly in their model equations.

- Although it is always possible to convert such models to state-space form and deal with them using any of the numerical ODE solvers introduced in earlier presentations, it could possibly be advantageous to simulate the second-derivative system directly.

- Unfortunately, the **GE3 algorithm** turned out to be a disappointment, because it cannot be used to simulate systems with damping.

Conclusions

- In this presentation, we have started to look at a new class of linear multi-step algorithms, specifically designed for the simulation of second derivative systems, i.e., models that contain the second derivatives of the partial state vector explicitly in their model equations.

- Although it is always possible to convert such models to state-space form and deal with them using any of the numerical ODE solvers introduced in earlier presentations, it could possibly be advantageous to simulate the second-derivative system directly.

- Unfortunately, the **GE3 algorithm** turned out to be a disappointment, because it cannot be used to simulate systems with damping.

- The method is numerically unstable everywhere in the open left-half complex plane, because its $\mathbf{F}$-matrix turned out to be an even function in $\mathbf{A} \cdot h$.

Conclusions

- In this presentation, we have started to look at a new class of linear multi-step algorithms, specifically designed for the simulation of second derivative systems, i.e., models that contain the second derivatives of the partial state vector explicitly in their model equations.
- Although it is always possible to convert such models to state-space form and deal with them using any of the numerical ODE solvers introduced in earlier presentations, it could possibly be advantageous to simulate the second-derivative system directly.
- Unfortunately, the **GE3 algorithm** turned out to be a disappointment, because it cannot be used to simulate systems with damping.
- The method is numerically unstable everywhere in the open left-half complex plane, because its $\mathbf{F}$-matrix turned out to be an even function in $\mathbf{A} \cdot h$.
- **We'll definitely need something better.**

References

1. Beamis, Christopher Paul (1990), *Solution of Second Order Differential Equations Using the Godunov Integration Method*, MS Thesis, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ.