Numerical Simulation of Dynamic Systems: Hw10 - Problem

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

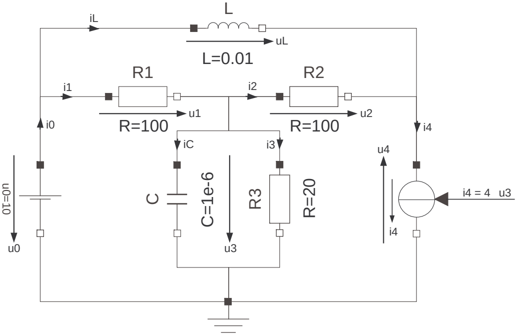
May 7, 2013

[H8.2] Inlining Radau IIA(3)

Given the electrical circuit:

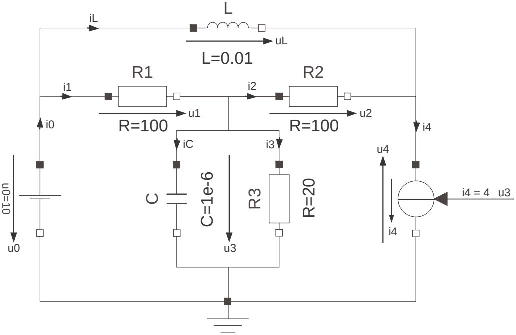
[H8.2] Inlining Radau IIA(3)

Given the electrical circuit:



[H8.2] Inlining Radau IIA(3)

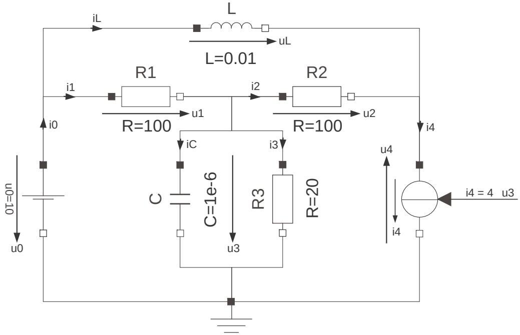
Given the electrical circuit:



- The circuit contains a constant voltage source, u_0, and a dependent current source, i_4, that depends on the voltage across the capacitor, C, and the resistor, R_3.

[H8.2] Inlining Radau IIA(3)

Given the electrical circuit:



- The circuit contains a constant voltage source, u_0, and a dependent current source, i_4, that depends on the voltage across the capacitor, C, and the resistor, R_3.
- Write down the element equations for the seven circuit elements. Since the voltage u_3 is common to two circuit elements, these equations contain 13 rather than 14 unknowns. Add the voltage equations for the three meshes and the current equations for three of the four nodes.

[H8.2] Inlining Radau IIA(3) II

- We wish to inline the fixed-step 3^{rd}-order accurate Radau IIA algorithm. Draw the structure digraph of the inlined equation system, which now consists of 30 equations in 30 unknowns, and causalize it using the tearing method.

[H8.2] Inlining Radau IIA(3) II

- We wish to inline the fixed-step 3^{rd}-order accurate Radau IIA algorithm. Draw the structure digraph of the inlined equation system, which now consists of 30 equations in 30 unknowns, and causalize it using the tearing method.

- Simulate the inlined difference equation system across 50 μsec with zero initial conditions on both the capacitor and the inductor. Choose a step size of $h = 0.5\ \mu sec$. Use algebraic differentiation for the computation of the Hessian.

[H8.2] Inlining Radau IIA(3) II

- We wish to inline the fixed-step 3^{rd}-order accurate Radau IIA algorithm. Draw the structure digraph of the inlined equation system, which now consists of 30 equations in 30 unknowns, and causalize it using the tearing method.

- Simulate the inlined difference equation system across 50 μsec with zero initial conditions on both the capacitor and the inductor. Choose a step size of $h = 0.5\ \mu sec$. Use algebraic differentiation for the computation of the Hessian.

- Plot the voltage u_3 and the current i_C on two separate subplots as functions of time.

[H8.3] Step-size Control for Radau IIA(3)

We wish to augment the solution to problem [H8.2] by adding a step-size control algorithm.

[H8.3] Step-size Control for Radau IIA(3)

We wish to augment the solution to problem [H8.2] by adding a step-size control algorithm.

- Use:

$$\mathbf{x}_{\mathbf{k+1}}^{\mathbf{blended}} = -\frac{1}{13}\,\mathbf{x}_{\mathbf{k-1}} + \frac{2}{13}\,\mathbf{x}_{\mathbf{k-\frac{2}{3}}} + \frac{14}{13}\,\mathbf{x}_{\mathbf{k}} - \frac{2}{13}\,\mathbf{x}_{\mathbf{k+\frac{1}{3}}} + \frac{11h}{13}\,\dot{\mathbf{x}}_{\mathbf{k+\frac{1}{3}}} + \frac{3h}{13}\,\dot{\mathbf{x}}_{\mathbf{k+1}}$$

as the embedding method for the purpose of error estimation, and use Fehlberg's step-size control algorithm:

$$h_{\mathrm{new}} = \sqrt[5]{\frac{tol_{\mathrm{rel}} \cdot \max(|x_1|, |x_2|, \delta)}{|x_1 - x_2|}} \cdot h_{\mathrm{old}}$$

for the computation of the next step size. Of course, the formula needs to be slightly modified, since it assumes the error estimate to be 5^{th}-order accurate, whereas in our algorithm, it is only 3^{rd}-order accurate. Remember that the step size can never be modified two steps in a row.

[H8.3] Step-size Control for Radau IIA(3) II

- Simulate the inlined difference equation system across 50 μsec with zero initial conditions on both the capacitor and the inductor.

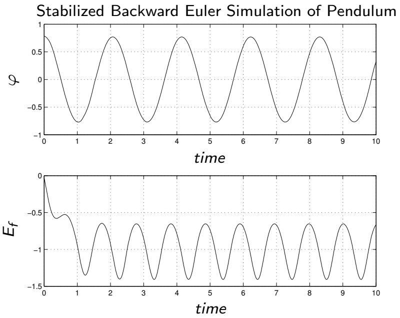
[H8.3] Step-size Control for Radau IIA(3) II

- Simulate the inlined difference equation system across 50 μsec with zero initial conditions on both the capacitor and the inductor.

- Plot the voltage u_3, the current i_C, and the step size h on three separate subplots as functions of time.

[H8.7] Stabilized BE Simulation of Overdetermined DAE System

In class (Presentation XX), I showed you how to stabilize the inlined BE simulation of the pendulum using an additional constraint equation. We obtained the following stabilized trajectories:



[H8.7] Stabilized BE Simulation of Overdetermined DAE System II

On purpose, I haven't shown you the details of how these trajectories have been derived. In particular, I didn't provide you with a formula for when to end the Newton iteration. Since the linear system is now only solved in a least square sense, you can no longer test for $\|\mathcal{F}\|$ having decreased to a small value. The way I did it was to compute the norm of $\mathcal{F}$ and save that value between iterations. I then tested, whether the norm of $\mathcal{F}$ has no longer decreased significantly from one iteration to the next:

while abs($\|\mathcal{F}^{\ell}\| - \|\mathcal{F}^{\ell-1}\|$) $< 1.0e - 6$,
    perform iteration
end,

[H8.7] Stabilized BE Simulation of Overdetermined DAE System II

On purpose, I haven't shown you the details of how these trajectories have been derived. In particular, I didn't provide you with a formula for when to end the Newton iteration. Since the linear system is now only solved in a least square sense, you can no longer test for $\|\mathcal{F}\|$ having decreased to a small value. The way I did it was to compute the norm of $\mathcal{F}$ and save that value between iterations. I then tested, whether the norm of $\mathcal{F}$ has no longer decreased significantly from one iteration to the next:

```
while abs(||F^ℓ|| − ||F^(ℓ−1)||) < 1.0e − 6,
    perform iteration
end,
```

Reproduce the graph showing the trajectories.