Efficient Solution of Equation Systems

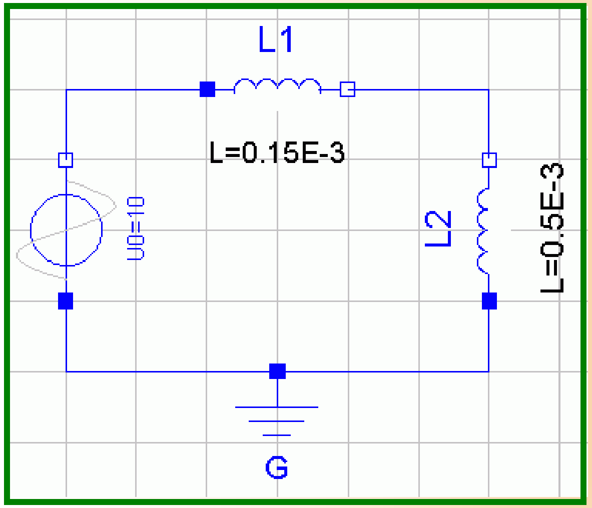
- This lecture deals with the efficient mixed symbolic/numeric solution of algebraically coupled equation systems.
- Equation systems that describe physical phenomena are almost invariably (exception: very small equation systems of dimension 2×2 or 3×3) ***sparsely populated***.
- This fact can be exploited.
- Two symbolic solution techniques: the ***tearing of equation systems*** and the ***relaxation of equation systems***, shall be presented. The aim of both techniques is to "squeeze the zeros out of the structure incidence matrix."

Table of Contents

The Tearing of Equation Systems I

- The tearing method had been demonstrated various times before. The method is explained here once more in a somewhat more formal fashion, in order to compare it to the alternate approach of the relaxation method.
- As mentioned earlier, the systematic determination of the minimal number of tearing variables is a problem of exponential complexity. Therefore, a set of heuristics have been designed that are capable of determining good sub-optimal solutions.

Tearing of Equations: An Example I



1: $u = f(t)$
2: $u - u_1 - u_2 = 0$
3: $u_1 - L_1 \cdot di_1/dt = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$ ← *Constraint equation*

1: $u = f(t)$
2: $u - u_1 - u_2 = 0$
3: $u_1 - L_1 \cdot di_1/dt = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$
7: $di_1/dt - di_2/dt = 0$

Integrator to be eliminated → di_1/dt

⇒

1: $u = f(t)$
2: $u - u_1 - u_2 = 0$
3: $u_1 - L_1 \cdot di_1 = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$
7: $di_1 - di_2/dt = 0$

Tearing of Equations: An Example II

1: $u = f(t)$
2: $u - u_1 - u_2 = 0$
3: $u_1 - L_1 \cdot di_1 = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$
7: $di_1 - di_2/dt = 0$

$\Rightarrow$

1: $u = f(t)$
2: $u - u_1 - u_2 = 0$
3: $u_1 - L_1 \cdot di_1 = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$
7: $di_1 - di_2/dt = 0$

Algebraically coupled equation system in four unknowns

Choice (u_1)

1: $u - u_1 - u_2 = 0$
2: $u_1 - L_1 \cdot di_1 = 0$
3: $u_2 - L_2 \cdot di_2/dt = 0$
4: $di_1 - di_2/dt = 0$

$\Rightarrow$

1: $u - u_1 - u_2 = 0$
2: $u_1 - L_1 \cdot di_1 = 0$
3: $u_2 - L_2 \cdot di_2/dt = 0$
4: $di_1 - di_2/dt = 0$

$\Rightarrow$

1: $u_1 = u - u_2$
2: $di_1 = u_1 / L_1$
3: $u_2 = L_2 \cdot di_2/dt$
4: $di_2/dt = di_1$

Tearing of Equations: An Example III

1: $u_1 = u - u_2$
2: $di_1 = u_1 / L_1$
3: $u_2 = L_2 \cdot di_2/dt$
4: $di_2/dt = di_1$

$\Rightarrow$

$$u_1 = u - u_2 = u - L_2 \cdot di_2/dt = u - L_2 \cdot di_1 = u - (L_2/L_1) \cdot u_1$$

$\Downarrow$

$$[\, 1 + (L_2/L_1)\,] \cdot u_1 = u$$

$\Downarrow$

$$u_1 = \frac{L_1}{L_1 + L_2} \cdot u$$

$\Rightarrow$

1: $u = f(t)$
2: $u_1 = \dfrac{L_1}{L_1 + L_2} \cdot u$
3: $u_1 - L_1 \cdot di_1 = 0$
4: $u_2 - L_2 \cdot di_2/dt = 0$
5: $i - i_1 = 0$
6: $i_1 - i_2 = 0$
7: $di_1 - di_2/dt = 0$

Tearing of Equations: An Example IV

1: $u = f(t)$		1: $u = f(t)$		1: $u = f(t)$
2: $u_1 = \frac{L_1}{L_1 + L_2} \cdot u$		2: $u_1 = \frac{L_1}{L_1 + L_2} \cdot u$		2: $u_1 = \frac{L_1}{L_1 + L_2} \cdot u$
3: $u_1 - L_1 \cdot di_1 = 0$	$\Rightarrow$	3: $u_1 - L_1 \cdot di_1 = 0$	$\Rightarrow$	3: $di_1 = u_1 / L_1$
4: $u_2 - L_2 \cdot di_2/dt = 0$		4: $u_2 - L_2 \cdot di_2/dt = 0$		4: $di_2/dt = di_1$
5: $i - i_1 = 0$		5: $i - i_1 = 0$		5: $u_2 = L_2 \cdot di_2/dt$
6: $i_1 - i_2 = 0$		6: $i_1 - i_2 = 0$		6: $i_1 = i_2$
7: $di_1 - di_2/dt = 0$		7: $di_1 - di_2/dt = 0$		7: $i = i_1$

$\Rightarrow$ *Question: How complex can the symbolic expressions for the tearing variables become?*

The Tearing of Equation Systems II

- In the process of tearing an equation system, algebraic expressions for the tearing variables are being determined. This corresponds to the symbolic application of ***Cramer's Rule***.

$$A \cdot x = b \quad \Rightarrow \quad x = A^{-1} \cdot b$$

$$A^{-1} = \frac{A^{\dagger}}{|A|} \quad ; \quad (A^{\dagger})_{ij} = (-1)^{(i+j)} \cdot |A_{\neq j,i}|$$

Tearing of Equations: An Example V

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & -L_1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$

$$u_1 = \frac{\begin{vmatrix} -L_1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -L_2 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & 0 & 0 & 1 \\ 1 & -L_1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -L_2 & 1 \end{vmatrix}} \cdot u = \frac{L_1}{L_1 + L_2} \cdot u$$

The Tearing of Equation Systems III

- ***Cramer's Rule*** is of polynomial complexity. However, the computational load grows with the fourth power of the size of the equation system.
- For this reason, the symbolic determination of an expression for the tearing variables is only meaningful for relatively small systems.
- In the case of bigger equation systems, the tearing method is still attractive, but the tearing variables must then be ***numerically*** determined.

The Relaxation of Equation Systems I

- The relaxation method is a symbolic version of a ***Gauss elimination without pivoting***.
- The method is only applicable in the case of linear equation systems.
- All diagonal elements of the system matrix must be $\neq 0$.
- The number of non-vanishing matrix elements above the diagonal should be minimized.
- Unfortunately, the problem of minimizing the number of non-vanishing elements above the diagonal is again a problem of exponential complexity.
- Therefore, a set of heuristics must be found that allow to keep the number of non-vanishing matrix elements above the diagonal small, though not necessarily minimal.

Relaxing Equations: An Example I

1: $u - u_1 - u_2 = 0$
2: $u_1 - L_1 \cdot di_1 = 0$
3: $u_2 - L_2 \cdot di_2/dt = 0$
4: $di_1 - di_2/dt = 0$

$\Rightarrow$

$u_1 + u_2 = u$
$u_1 - L_1 \cdot di_1 = 0$
$di_2/dt - di_1 = 0$
$u_2 - L_2 \cdot di_2/dt = 0$

$\Downarrow$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & -L_1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The non-vanishing matrix elements above the diagonal correspond conceptually to the tearing variables of the tearing method.

Relaxing Equations: An Example II

Gauss elimination technique:

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - A_{ik}^{(k)} A_{kk}^{(k)^{-1}} A_{kj}^{(k)}$$

$$b_i^{(k+1)} = b_i^{(k)} - A_{ik}^{(k)} A_{kk}^{(k)^{-1}} b_k^{(k)}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & -L_1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} -L_1 & 0 & c_1 \\ 1 & -1 & 0 \\ 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_2 \\ 0 \\ 0 \end{bmatrix}$$

$c_1 = -1$
$c_2 = -u$

Relaxing Equations: An Example III

$$\begin{bmatrix} -L_1 & 0 & c_1 \\ 1 & -1 & 0 \\ 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_2 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} -1 & c_3 \\ -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_4 \\ 0 \end{bmatrix}$$

$$c_3 = c_1 / L_1$$
$$c_4 = c_2 / L_1$$

$$\begin{bmatrix} -1 & c_3 \\ -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_4 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_5 \end{bmatrix} \cdot \begin{bmatrix} u_2 \end{bmatrix} = \begin{bmatrix} c_6 \end{bmatrix}$$

$$c_5 = 1 - L_2 \cdot c_3$$
$$c_6 = - L_2 \cdot c_4$$

Relaxing Equations: An Example IV

Gauss elimination technique :

$$x_k = A_{kk}^{(k)^{-1}} \left(b_k^{(k)} - \sum_{j=k+1}^{n} A_{kj}^{(k)} x_j\right)$$

$$\left[c_5 \right] \cdot \left[u_2 \right] = \left[c_6 \right] \quad \Rightarrow \quad u_2 = c_6 \,/\, c_5$$

$$\begin{bmatrix} -1 & c_3 \\ -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_4 \\ 0 \end{bmatrix} \quad \Rightarrow \quad di_2/dt = (c_4 - c_3 \cdot u_2) \,/\, (-1)$$

$$\begin{bmatrix} -L_1 & 0 & c_1 \\ 1 & -1 & 0 \\ 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} c_2 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad di_1 = (c_2 - c_1 \cdot u_2) \,/\, (-L_1)$$

Relaxing Equations: An Example V

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & -L_1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -L_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ di_1 \\ di_2/dt \\ u_2 \end{bmatrix} = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow u_1 = u - u_2$$

$\Rightarrow$ *By now, all required equations have been found. They only need to be assembled.*

Relaxing Equations: An Example VI

1: $u - u_1 - u_2 = 0$
2: $u_1 - L_1 \cdot di_1 = 0$
3: $u_2 - L_2 \cdot di_2/dt = 0$
4: $di_1 - di_2/dt = 0$

⇒

$c_1 = -1$
$c_2 = -u$
$c_3 = c_1 / L_1$
$c_4 = c_2 / L_1$
$c_5 = 1 - L_2 \cdot c_3$
$c_6 = - L_2 \cdot c_4$
$u_2 = c_6 / c_5$
$di_2/dt = (c_4 - c_3 \cdot u_2) / (-1)$
$di_1 = (c_2 - c_1 \cdot u_2) / (-L_1)$
$u_1 = u - u_2$

⇒

$u = f(t)$
$c_1 = -1$
$c_2 = -u$
$c_3 = c_1 / L_1$
$c_4 = c_2 / L_1$
$c_5 = 1 - L_2 \cdot c_3$
$c_6 = - L_2 \cdot c_4$
$u_2 = c_6 / c_5$
$di_2/dt = (c_4 - c_3 \cdot u_2) / (-1)$
$di_1 = (c_2 - c_1 \cdot u_2) / (-L_1)$
$u_1 = u - u_2$
$i_1 = i_2$
$i = i_1$

The Relaxation of Equation Systems II

- The relaxation method can be applied symbolically to systems of slightly larger size than the tearing method, because the computational load grows more slowly.
- For some classes of applications, the relaxation method generates very elegant solutions.
- However, the relaxation method can only be applied to linear systems, and in connection with the ***numerical Newton iteration***, the tearing algorithm is usually preferred.

References

- Elmqvist H. and M. Otter (1994), "Methods for tearing systems of equations in object-oriented modeling," *Proc. European Simulation Multiconference*, Barcelona, Spain, pp. 326-332.

- Otter M., H. Elmqvist, and F.E. Cellier (1996), "Relaxing: A symbolic sparse matrix method exploiting the model structure in generating efficient simulation code," *Proc. Symp. Modelling, Analysis, and Simulation*, CESA'96, IMACS MultiConference on Computational Engineering in Systems Applications, Lille, France, vol.1, pp.1-12.