Systems of first-order ODEs

5/5/2011 L

As we previously discussed, it is possible to consider a type of differential equations with several unknown functions $y_1(t), y_2(t), \dots, y_n(t)$ (where each $y_i : [t_0, +\infty) \to \mathbb{R}$) and n total given equations:

$$
\begin{aligned}
y_1'(t) &= f_1(t, y_1, y_2, \dots, y_n) \\
y_2'(t) &= f_2(t, y_1, y_2, \dots, y_n) \\
&\;\;\vdots \\
y_n'(t) &= f_n(t, y_1, y_2, \dots, y_n)
\end{aligned}
$$

The methods for approximating solutions to individual ODEs carry over naturally to systems of ODEs.

For example

$$
\begin{aligned}
y_1'(t) &= -2y_1 \sin(y_2) &&= f_1(t, y_1, y_2) \\
y_2'(t) &= -y_1^2 y_2 &&= f_2(t, y_1, y_2)
\end{aligned}
$$

Let us apply the Forward Euler method. 5/3/11

We will use the notation $y_i^{(j)} = y_i(t_j)$

Thus:

$$y_1^{(n+1)} = y_1^{(n)} + \Delta t f_1(t_n, y_1^{(n)}, y_2^{(n)}) = y_1^{(n)} - \Delta t\, y_1^{(n)} \sin(y_2^{(n)}$$

$$y_2^{(n+1)} = y_2^{(n)} + \Delta t f_2(t_n, y_1^{(n)}, y_2^{(n)}) = y_2^{(n)} - [y_1^{(n)}]^2 y_2^{(n)}$$

Thus we proceed as follows:

$$\left.\begin{matrix} y_1^{(0)} \\ y_2^{(0)} \end{matrix}\right\} \rightarrow \left.\begin{matrix} y_1^{(1)} \\ y_2^{(1)} \end{matrix}\right\} \rightarrow \left.\begin{matrix} y_1^{(2)} \\ y_2^{(2)} \end{matrix}\right\} \rightarrow \cdots$$

Another example ... let's apply <u>Backward Euler</u> to

$$\begin{aligned} p'(t) &= -c_1 p(t) && \Big(= f(t,p,q,r) \\ q'(t) &= c_1 p(t) - c_2 q(t) && \ \ = g(t,p,q,r) \\ r'(t) &= c_2 q(t) && \ \ = h(t,p,q,r)\Big) \end{aligned}$$

$$\left(\text{or: } \begin{pmatrix} p' \\ q' \\ r' \end{pmatrix} = \begin{pmatrix} -c_1 & 0 & 0 \\ c_1 & -c_2 & 0 \\ 0 & c_2 & 0 \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \right)$$

Backward Euler gives:

$$p^{(n+1)} = p^{(n)} + \Delta t\, f(t, p^{(n+1)}, q^{(n+1)}, r^{(n+1)})$$

$$= p^{(n)} - \Delta t\, c_1 p^{(n+1)} \qquad (1)$$

$$q^{(n+1)} = q^{(n)} + \Delta t\, (c_1 p^{(n+1)} - c_2 q^{(n+1)}) \qquad (2)$$

$$r^{(n+1)} = r^{(n)} + \Delta t\, c_2 q^{(n+1)} \qquad (3)$$

$$\left.\begin{aligned}
(1) &\Rightarrow (1 + \Delta t\, c_1) p^{(n+1)} = p^{(n)} \\
(2) &\Rightarrow -\Delta t\, c_1 p^{(n+1)} + (1 + \Delta t\, c_2) q^{(n+1)} = q^{(n)} \\
(3) &\Rightarrow -\Delta t\, c_2 q^{(n+1)} + r^{(n+1)} = r^{(n)}
\end{aligned}\right\} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} 1+\Delta t c_1 & 0 & 0 \\ -\Delta t c_1 & 1+\Delta t c_2 & 0 \\ 0 & -\Delta t c_2 & 1 \end{pmatrix} \begin{pmatrix} p^{(n+1)} \\ q^{(n+1)} \\ r^{(n+1)} \end{pmatrix} = \begin{pmatrix} p^{(n)} \\ q^{(n)} \\ r^{(n)} \end{pmatrix}$$

Ok, what about stability and accuracy, though? 5/5/11

→ More complicated analysis!

A. Stability of the solutions to the system of ODE's.

Here, asymptotic stability again means that if

$y_1, y_2, \dots, y_n$ AND $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$

are 2 different solutions to the system, then

$$\lim_{t\to\infty} \left| y_i(t) - \hat{y}_i(t) \right| = 0 \quad \forall\, i = 1, 2, \dots, n$$

Making a reliable determination requires tools we don't have in CS412 ... but we can do so in special cases:

* Consider the system

$$y_1' = f_1(t, y_1, \dots, y_n)$$
$$\vdots$$
$$y_n' = f_n(t, y_1, \dots, y_n)$$

Define the Jacobian matrix as

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_n} \\ \frac{\partial f_2}{\partial y_1} & & \vdots \\ \vdots & & \\ \frac{\partial f_n}{\partial y_1} & \cdots & \frac{\partial f_n}{\partial y_n} \end{pmatrix}$$

5/5/11

- If J is symmetric and negative definite (for all t) (i.e. $\underline{x}^T J \underline{x} < 0 \quad \forall \underline{x} \neq 0$) then the solutions are asymptotically stable.

- if J is diagonally dominant and all diagonal entries are negative, then the solutions are asymptotically stable.

e.g.
$$\left.\begin{aligned} p' &= -3p + q \\ q' &= p - 3q + r \\ r' &= q - 3r \end{aligned}\right\} \Rightarrow J = \begin{pmatrix} -3 & 1 & \\ 1 & -3 & 1 \\ & 1 & -3 \end{pmatrix}$$

diagonally dominant!
negative diagonal entries!
$\Rightarrow$ asymptotically stable solutions!

Thm If a system has asymptotically stable solutions AND the method we use is unconditionally stable for individual equations, then it will be unconditionally stable for systems, too.

Note: If the method has a stability condition for single equations, it will also have such a condition on systems; however, it is not clear what this maximum allowable Δt exactly is! $\Rightarrow$ Can try different values and reduce Δt if solutions are becoming unbounded)

Accuracy

The accuracy exhibited in individual equations carries over to systems. e.g:

Forward Euler, Backward Euler $\Rightarrow$ 1st order

Trapezoidal Rule $\Rightarrow$ 2nd order.

Systems of ODEs arise naturally from the consideration of Higher-order differential equations, e.g.

$$y^{(n)}(t) = f(t, y, y', y'', y''', \dots, y^{(n-1)})$$

e.g. $y''(t) = -5y'(t) - 4y(t)$

or. $y'''(t) = -2t\sin(yy')e^{y''}$

How about the stability of these solutions?

Once again, this can be a complicated determination...

However, there is the special case when the equation is "linear", e.g.

$$y^{(n)}(t) = c_0(t)y(t) + c_1(t)y'(t) + c_2(t)y''(t) + \dots + c_{n-1}(t)y^{(n-1)}(t).$$

In this case we employ the following criterion:

→ Replace the k-th derivative $y^{(k)}(t)$ with the k-th power of an auxiliary variable λ, i.e $y^{(k)}(t) \longleftrightarrow \lambda^k$

Thus you generate a polynomial:

$$\lambda^n = c_0(t) + c_1(t)\lambda + c_2(t)\lambda^2 + \dots + c_{n-1}(t)\lambda^{n-1}$$

→ If, for all $t \geq t_0$, this polynomial has (possibly complex) roots with negative real parts, then the ODE has asymptotically stable solutions

e.g. $y'' = -5y' - 4y$

$$\rightarrow \lambda^2 = -5\lambda - 4 \Rightarrow \lambda^2 + 5\lambda + 4 = 0 \quad \begin{cases} \lambda_1 = -1 < 0 \\ \lambda_2 = -4 < 0 \end{cases}$$

Asym. Stable!

also. $y'' = -6y' - 10y$

$$\rightarrow \lambda^2 + 6\lambda + 10 = 0 \quad \begin{cases} \lambda_1 = -3+i & Re\{\lambda_1\} < 0 \\ \lambda_2 = -3-i & Re\{\lambda_2\} < 0 \end{cases}$$

As. stable!

Solution method: Convert the higher-order ODE into a system of 1st order ones!

Methodology: Consider $y^{(n)}(t) = f(t, y, y', y'', \dots, y^{(n-1)})$

Define $y_k(t) := y^{(k)}(t) \quad k = 0, 1, \dots, n-1.$

Thus, the ODE becomes:

$$y_{n-1}'(t) = f(t, y_0, y_1, y_2, \dots, y_{n-1})$$
$$(= y^{(n)}(t))$$

Along with the $n-1$ equations:

$$y_k'(t) = y_{k+1}(t) \quad k = 0, 1, \dots, n-2$$

Putting everything together as a system:

$$\begin{aligned} y_0' &= y_1 \\ y_1' &= y_2 \\ &\vdots \\ y_{n-2}' &= y_{n-1} \\ y_{n-1}' &= f(t, y_0, y_1, \dots, y_{n-1}). \end{aligned}$$

e.g. 5/5/11

$$y'' = -6y' - 10$$

becomes

$$y_0' = y_1$$
$$y_1' = -6y_1 - y_0$$

or

$$y''' = -2t\sin(yy')e^{y''}$$

becomes

$$y_0' = y_1$$
$$y_1' = y_2$$
$$y_2' = -2t\sin(y_0y_1)e^{y_2}$$

Then we solve it with any desired method!

Final note: The initial conditions supplied for a higher order ODE $y^{(n)} = f(t, y, y', \dots, y^{(n-1)})$ are

$$\left.\begin{array}{l} y(t_0) = c_0 \\ y'(t_0) = c_1 \\ \vdots \\ y^{(n-1)}(t_0) = c_{n-1} \end{array}\right\} \text{ and converted to: } \begin{array}{l} y_0(t_0) = c_0 \\ y_1(t_0) = c_1 \\ \vdots \\ y_{n-1}(t_0) = c_{n-1} \end{array}$$