R gave me a DAE underneath the Linden tree

Most of my life has been concerned with understanding how to solve ordinary differential equations numerically. Just when I think I have got somewhere, I find that some new aspect of the problem has become important and I have to start at the beginning again. So it was when "stiff" differential equations were recognised as constituting a distinct class of numerical problems. I thought I knew something about how to solve the easier non-stiff problems; all one needed was a stable and consistent numerical approximation and this can be turned into an effective algorithm. Most of the traditional numerical schemes are generalizations of the famous Euler method in which the solution at a time value x_n is found by adding to the approximation at x_{n-1} the value of an approximation to the derivative, evaluated at x_{n-1}, and multiplied by $x_n - x_{n-1}$. If the differential equation is $y'(x) = f(x, y(x))$, and y_n is the approximation to $y(x_n)$, then we can write, for the Euler method, $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$, where $h = x_n - x_{n-1}$. In the special case of the linear problem, $y'(x) = \lambda y(x)$, the recursion for the approximations would be $y_n = (1 + \lambda h)y_{n-1}$.

Sometimes, buried in a complicated differential equation system, are components which behave in this fashion. Sometimes, when λ is a (possibly complex) number with negative real part, we are supposed to be approximating a negative exponential so that the effect of these components should be only transitory. However, if λ is outside a disc with centre -1 and radius 1, powers of $1 + h\lambda$ are unbounded and the computed result will be spoilt by the effect of these terms.

In such a situation, the problem will be stiff and it needs to be solved by alternative methods, such as the implicit Euler method $y_n = y_{n-1} + hf(x_n, y_n)$. For this method, the factor $1 + \lambda h$, which was the source of trouble for the forward Euler method, is now replaced by $(1 - h\lambda)^{-1}$, and no harm is done by λ being negative and of large magnitude.

I do not remember when I first became aware of differential algebraic equations, but I know when this crucial event occurred in the life of my colleague and friend Roswitha März of the Humboldt University in Berlin. I was attending a conference in the former West Germany in 1981 when someone asked me if I would like to visit East Berlin, because this person could arrange it for me. With some trepidation, I agreed to fit in with the arrangements that had to be made, and a few days later I first made the acquaintance of Roswitha. I had in my hand a number of reprints of papers by various people and some of these were on the relatively new subject of numerical methods for differential algebraic equations. I gave these to Roswitha in case she could make use of them and she certainly could. She soon became the leader of one of the most productive and insightful research groups working on this subject and I have always been proud of my small role in her introduction to this subject.

Over the last 6 or 7 years we have held a series of "ANODE" (Auckland Numerical Ordinary Differential Equations) workshops and what will almost certainly been the last of these has just finished. It was a delight to welcome Roswitha März, as an invited speaker, at this meeting. In fact it was a triple benefit because Roswitha came with two colleagues, René Lamour and Caren Tischendorf, who are outstanding research workers in their own right.

As an easy introduction to the subject of numerical differential algebraic equations, I will quote an example problem presented by Caren. This consists of the coupled system

(1) $\quad (\lambda - 1)y'(x) + \lambda x z'(x) = 0,$ $\qquad$ (2) $\quad (\lambda - 1)y(x) + (\lambda x - 1)z(x) = 0.$

This is a differential algebraic equation because it contains a differential equation together with an algebraic constraint. It is said to be of "index 1" because a single differentiation of (2) and a rearrangement converts it to the differential equation system

(3) $\quad (\lambda - 1)y'(x) + \lambda^2 x z(x) = 0,$ $\qquad$ (4) $\quad z'(x) = \lambda z(x).$

If the problem in its original formulation is solved by a natural extension of the implicit Euler method it is found that the z values satisfy the recursion $z_n = (1 + \lambda h)z_{n-1}$. It is unfortunate that this is the same recursion that would apply to (4) being solved by the explicit, rather than the implicit Euler method, and is disadvantageous if λ negative with a large magnitude.

One of the main thrusts of the Humboldt group led by Roswitha März, is that this sort of anomalous behaviour is avoided if the problem is formulated differently. What they call "numerically qualified" would cast this example problem in the form

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} \lambda - 1 & \lambda x \end{bmatrix} \begin{bmatrix} y(x) \\ z(x) \end{bmatrix} \right)' + \begin{bmatrix} 0 & \lambda \\ 1 - \lambda & 1 - \lambda x \end{bmatrix} \begin{bmatrix} y(x) \\ z(x) \end{bmatrix} = 0.$$

The crucial detail concerning the two matrices $A = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $D = \begin{bmatrix} \lambda - 1 & \lambda x \end{bmatrix}$, is that $\operatorname{im}(D)$ is constant and that $\ker(A) \oplus \operatorname{im}(D) = \mathbb{R}$. An implementation of the implicit Euler method using this formulation would propagate only values of $D \begin{bmatrix} y(x) & z(x) \end{bmatrix}^T$, and use the algebraic constraint to evaluate the individual components y_n and z_n.

During an eight month visit to Auckland, Steffen Schulz, a postgraduate student at Humboldt, wrote the Auckland Mathematics report number 497: "Four lectures on Differential-Algebraic Equations". This is a good introduction to the subject and to some of its literature.

John Butcher, `butcher@math.auckland.ac.nz`