Computed Solution Curves for Differential Equations

by John H. Mathews
California State University, Fullerton
Fullerton, CA 92634

John H. Mathews is currently a professor of mathematics at California State University, Fullerton. He received a Ph.D. from Michigan State University under the direction of Peter Lappan in 1969. He has taught at CSUF for twenty years and is the author of two books: Numerical Methods for Computer Science, Engineering and Mathematics, Prentice-Hall, Inc., Englewood Cliffs [1987] and Complex Variables for Mathematics and Engineering, (2 cd ed.) William C. Brown, Inc. [1988]. He is interested in how numerical methods can be introduced into the calculus curriculum.

An important problem in numerical analysis is to compute approximate solutions of the differential equation.

$$y'(x) = f(x,y). \quad (1)$$

Under modest (and well known) assumptions on $f$, the "general solution" of (1) consists of an infinite family of functions, each of which may be distinguished by selection of an initial point $(a,y(a))$. Starting from this initial point, numerical methods attempt to approximate the solution $y = y(x)$ on some specified interval $[a,b]$.

Continuity of $f(x,y)$ does not ensure the continuity of $y(x)$. Suppose that $y(x)$ has an infinite discontinuity at $x = c$, that is $\lim_{x \to c} |f(x)| = \infty$. Then the reciprocal $Y(x) = 1/y(x)$ tends to zero as $x \to c$, and $Y(x)$ will have a removable singularity at $x = c$ provided that we define $Y(c) = 0$. Therefore, we can use the change of variable

$$Y(x) = 1/y(x). \quad (2)$$

Now differentiate each side of (2) and get

$$Y'(x) = - y'(x)/[y(x)]^2 = - [Y(x)]^2 y'(x).$$

Then substitute $y'(x)$ from (1) and obtain:

$$Y'(x) = - Y^2 f(x,1/Y). \quad (3)$$

Differential equation (3) is equivalent to (1) in this sense: Given a neighborhood $N$ of $x_0$ and a number $y_0 \neq 0$, equation (1) has a solution with $y(x_0) = y_0$ and $y(x) \neq 0$ for all $x$ in $N$ if and only if equation (3) has a solution with $Y(x_0) = 1/y_0$ and $Y(x) \neq 0$. We call equation (3) the companion differential equation and write it as

$$Y'(x) = g(x,Y). \quad (4)$$

Numerical methods "track" a specific solution curve through the starting point $(x_0,y_0)$. The success of using (4) for tracking the solution $y(x)$ near a singularity is the fact that $|y(x)| \to \infty$ as $x \to c$ if and only if $Y(x) \to 0$ as $x \to c$. A numerical solution $Y(x)$ to (4) can be computed over a small interval containing $c$, then (2) is used to determine a solution curve for (1) that lies on both sides of the vertical asymptote $x = c$. 

30
A procedure such as the Euler-Heun method, uses a fixed step size $h = \frac{(b-a)}{M}$ and for each $x_j = a + jh$ an approximation $y_j$ to $y(x_j)$ is computed (for $1 \leq j \leq M$). This computed value $y_j$ depends on $y_{j-1}$ and the slope function $f$. If $|y(x)| \to \infty$ as $x \to c$ then the numerical method fails to follow the true solution accurately because of the inherent numerical instability of computing a "rise" as the product of a very large slope and very small "run" (a computation which magnifies the errors present in the value $y_{j-1}$). One way to reduce this error is to select a bound $B$ and change computational strategy as soon as a value $f_L = f(x_L, y_L)$ is computed for which $|f_L| > B$, that is, as soon as the possibility of a singularity is "sensed." In particular, we may use $(x_L, 1/y_L)$ as an initial value with (4) to track the reciprocal $Y$, which will not suffer from the difficulties created by steep slopes.

Thus, the following strategy can be employed to extend any single-step numerical method (See Mathews, 1987, Chapter 9, for a discussion of single-step methods). We use equation (1) and the initial value $y(a) = y_0$ to compute the sequence of points

\[
\{(x_j, y_j)\}_{j=1}^{L} \text{ where } |f_j| \leq B \text{ for } j = 0, \ldots, L-1 \text{ and } |f_L| > B.
\]

Then we use (4) with $Y(x_L) = 1/y_L$ to compute the sequence of points

\[
\{(x_j, Y_j)\}_{j=L}^{N} \text{ where } |f_j| > B \text{ for } j = L, L+1, \ldots, N-1 \text{ and } |f_N| \leq B.
\]

Continue in a similar fashion and alternate between formula (1) and formula (4) until $j = M$.

The decision process, for the "extended" Euler-Heun method is:

IF $|f_j| \leq B$ THEN

perform one Euler-Heun step using

$y' = f(x, y)$ to compute $y_{j+1}$

ELSE

set $Y_{j+1} = 1/y_{j+1}$ and perform one Euler-Heun step using $Y' = g(x, Y)$ to compute $Y_{j+1}$ and then set $y_{j+1} = 1/Y_{j+1}$.

ENDIF

When (4) is used for numerical computations, the formula for $g(x, Y)$ must be simplified in advance so that "0/0" or "∞/∞" computational problems do not occur. In particular, $-Y^2 f(x, 1/Y)$ should not be used in a computer program.

Example 1. We use the extended Euler-Heun method with the step size of $h = 0.005$ to find a numerical approximation to the solution of $y' = 1 + y^2$ with $y(0) = 0$ over the interval $[0, 2]$. The companion differential equation is $Y' = -Y^2(1 + 1/Y^2) = -1 - Y^2 = g(x, Y)$. The value $B$ is arbitrary, and experimentation was used to find that $B = 3$ gave excellent results. Selected values of $y$ are presented in Table 1 and compared with the true solution $y(x_j) = \tan(x_j)$. Figure 1 shows a graph of the numerical solution. The method behaves well at $x = 1.56$ and $x = 1.58$, which are on opposite sides of the singularity at $x = \pi/2$. 

31
TABLE I. Comparison of the approximate solution $y_j$ to $y' = 1 + y^2$ and $y = \tan(x)$ over the interval $[0, 2]$.

<table>
<thead>
<tr>
<th>$x_j$</th>
<th>The Extended Euler-Heun Solution $y_j$</th>
<th>Error in the Approximation $E_j = \tan(x_j) - y_j$</th>
<th>Relative error $E_j / \tan(x_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.422795</td>
<td>-0.000002</td>
<td>-0.000004</td>
</tr>
<tr>
<td>0.8</td>
<td>1.029640</td>
<td>-0.000001</td>
<td>-0.000001</td>
</tr>
<tr>
<td>1.2</td>
<td>2.572138</td>
<td>0.000014</td>
<td>0.000005</td>
</tr>
<tr>
<td>1.55</td>
<td>48.077308</td>
<td>0.001174</td>
<td>0.000024</td>
</tr>
<tr>
<td>1.56</td>
<td>92.616495</td>
<td>0.004001</td>
<td>0.000043</td>
</tr>
<tr>
<td>1.57</td>
<td>1255.096100</td>
<td>0.669491</td>
<td>0.000533</td>
</tr>
<tr>
<td>1.58</td>
<td>-108.653727</td>
<td>0.004523</td>
<td>-0.000042</td>
</tr>
<tr>
<td>1.59</td>
<td>-52.067896</td>
<td>0.000926</td>
<td>-0.000018</td>
</tr>
<tr>
<td>1.6</td>
<td>-34.232884</td>
<td>0.000351</td>
<td>-0.000010</td>
</tr>
<tr>
<td>1.8</td>
<td>-4.286252</td>
<td>-0.000010</td>
<td>0.000002</td>
</tr>
<tr>
<td>2.0</td>
<td>-2.185034</td>
<td>-0.000006</td>
<td>0.000003</td>
</tr>
</tbody>
</table>

Fig. 1. The graph of the solution to $y' = 1 + y^2$, $y(0) = 0$ using the extended Euler-Heun method.

The following is a sample program written in Turbo Pascal for the extended Euler-Heun method. Notice the functions $f(x, Y)$ and $g(x, Y)$ must be given and that $g(x, Y)$ must be simplified so that small values of $Y$ near 0 will not cause an exponent overflow.

The AMATYC Review  Fall 1989  Volume 11, Number 1 (Part 1)
PROCEDURE EXTENDED_EULER_METHOD;
CONST Big=3;
FUNCTION F(X,Y:REAL) : REAL;
BEGIN F:=1+Y*Y; END;
FUNCTION G(X,Y:REAL) : REAL;
BEGIN G:=-1-Y*Y; END;
PROCEDURE Euler_Small;
BEGIN
  YJ:=Y[J];
  M1:=F(X[J],YJ);
  P1:=YJ+H*M1;
  X[J+1]:=A+H*(J+1);
  M2:=F(X[J+1],P1);
  Y[J+1]:=YJ+H*(M1+M2)/2;
{Corrector}
END;
PROCEDURE Euler_Large;
BEGIN
  YJ:=1/Y[J];
  M1:=G(X[J],YJ);
  P1:=YJ+H*M1;
  X[J+1]:=A+H*(J+1);
  M2:=G(X[J+1],P1);
  Y[J+1]:=YJ+H*(M1+M2)/2;
{Corrector}
END;
BEGIN
  H:=(B-A)/M;
  X[0]:=A;
  Y[0]:=Y0;
  FOR J := 0 TO M-1 DO
    BEGIN
      IF ABS(F(X[J],Y[J])) <= Big THEN Euler_Small
      ELSE Euler_Large;
    END;
END;
References