PERIODICA POLYTECHNICA SER. CIV. ENG. VOL. 44, NO. 1, PP. 89–97 (2000)

SYMBOLIC SOLUTION OF BOUNDARY VALUE PROBLEM VIA MATHEMATICA

Béla PALÁNCZ* and György POPPER**

*Laboratory of Informatics
**Research Group for Computational Mechanics of the Hungarian Academy of Sciences
Department of Structural Mechanics
Faculty of Civil Engineering
Technical University of Budapest
H–1521 Budapest, Hungary
Fax: +36 1 463 1099, E-mail: palancz@epito.bme.hu

Received: December 20, 1998

Abstract

Symbolic computation has been applied to Runge-Kutta technique in order to solve a two-point boundary value problem. The unknown boundary values are considered as symbolic variables, therefore they will appear in a system of algebraic equations, after the integration of the ordinary differential equations. Then this algebraic equation system can be solved for the unknown initial values and substituted into the solution. Consequently, only one integration pass is enough to solve the problem instead of using an iteration technique like shooting method. This procedure is illustrated by solving the boundary value problem of the mechanical analysis of a liquid storage tank. Computations were carried out by the MATHEMATICA symbolic system.

Keywords: symbolic computation, boundary value problem, mechanical analysis, MATHEMATICA.

1. Introduction

There are many engineering models represented by ordinary differential equations with split boundary value problem. Shooting and finite difference methods [1–2], trial function expansion based on variational principle or weighted residual method as well as different types of collocation, quasi-linearization [3], and perturbation techniques [4] have been widely used for a long time to solve such problems.

Computer algebra systems like MACSYMA, REDUCE, MAPLE, MATHEMATICA, and to a certain extent other types of systems as MATLAB and MATHCAD, give possibility to carry out not only numerical but also symbolical computations. Many traditional algorithms can be improved, sometimes considerably, via embedding symbolical parts into the numerical algorithm. These hybrid techniques involving numeric as well as symbolic manipulations, provide arbitrary precision in defeating instability problems and reduce the number of iterations in general.

The application of hybrid techniques to boundary value problems was studied in [5] for the case of second and third order, linear and non-linear ordinary differential equations including eigenvalue and stiffness problems. In this paper a method proposed in [5], the so-called slope retention technique has been extended for non-autonomous, linear system of differential equations, using symbolic Runge-Kutta method.

2. Boundary Value Problem

Let us consider a linear, non-autonomous differential equation system of n variables in matrix form :

$$\frac{\partial}{\partial x} y(x) = A(x) y(x) + b(x),$$

where A is a matrix of $n \times n$ dimensions, $y(x)$ and $b(x)$ are vectors of n dimensions, and x is a scalar independent variable. In the case of a boundary value problem, the values of some dependent variables are not known at the beginning of the integration interval, at $x = x1$, but they are given at the end of this interval, at $x = x2$. The usually employed methods need subsequent integration of the system, because of their trial-error technique or they require solution of a large linear equation system, in the case of discretization methods. In this paper a new technique is suggested, which is based on the symbolic evaluation of the well known Runge-Kutta algorithm. This technique needs only one integration of the differential equation system and a solution of the linear equation system representing the boundary conditions at $x = x2$.

3. Symbolic Runge-Kutta Method

The well known fourth-order Runge-Kutta method, in our case, can be represented by the following formulas :

$$\begin{aligned}
R1_i &= A(x_i) y(x_i) + b(x_i), \\
R2_i &= A\left(x_i + \frac{h}{2}\right)\left(y(x_i) + \frac{R1_i h}{2}\right) + b\left(x_i + \frac{h}{2}\right), \\
R3_i &= A\left(x_i + \frac{h}{2}\right)\left(y(x_i) + \frac{R2_i h}{2}\right) + b\left(x_i + \frac{h}{2}\right), \\
R4_i &= A(x_i + h)(y(x_i) + R3_i h) + b(x_i + h)
\end{aligned}$$

and then the new value of $y(x)$ can be computed as :

$$y_{i+1} = y(x_i) + \frac{(R1_i + 2(R2_i + R3_i) + R4_i)h}{6}.$$

A symbolic system like MATHEMATICA, is able to carry out this algorithm not only with numbers but also with symbols. It means that the unknown elements of $y(x1)$ can be considered as unknown symbols. These symbols will appear in every evaluated y value, as well as in $y(x2)$, too. The following MATHEMATICA procedure can carry out this symbolic computation:

```
RKSymbolic[x0_,y0_,A_,b_,M_,N_,h_]:=
   Module[{R1,R2,R3,R4,y,i,j,ylist},
   y=y0;ylist={};
   For [j=1,j<=M,j++,
     ylist=Append[ylist,0];
     ylist[[j]]={{x0,y0[[j]]}};
   ];
     For[i=1,i<=N,i++,
     R1=Expand[A[x0+i h].y+b[x0+i*h]];
     R2=Expand[A[x0+i h+h/2].(y+R1 h/2)+b[x0+i h+h/2]];
     R3=Expand[A[x0+i h+h/2].(y+R2 h/2)+b[x0+i h+h/2]];
     R4=Expand[A[x0+i h +h].(y+R3 h)+b[x0+i h+h]];
     y=Expand[y+(R1+2 (R2+R3)+R4) h/6];
     For[j=1,j<=M,j++,
       ylist[[j]]=Append[ylist[[j]],{x0+i h,y[[j]]}];
     ];
   ];
   {y,ylist}
];
```

Let us consider a simple illustrative example. The differential equation is :

$$\left(\frac{\partial^2}{\partial x \partial x} y(x)\right) - \left(1 - \frac{x}{5}\right) y(x) = x.$$

The prespecified boundary values are:

$$y(1) = 2$$

and

$$y(3) = -1.$$

After introducing

$$y_1(x) = y(x)$$

and

$$y_2(x) = \frac{\partial}{\partial x} y(x),$$

the matrix form of the differential equation is:

$$\left[\frac{\partial}{\partial x} y1(x), \frac{\partial}{\partial x} y2(x)\right] = \begin{bmatrix} 0 & 1 \\ 1 - \frac{1}{5}x & 0 \end{bmatrix} [y1(x), y2(x)] + [0, x].$$

Employing MATHEMATICA's notation :

```
A[x_]:={{0,1},{1-1/5 x,0}};
b[x_]:={0,x};
x0=1;
y0={2.,s}
```

The unknown initial value is s. The order of the system $M = 2$. Let us consider the number of the integration steps as $N = 10$, so the step size is $h = 0.2$.

```
ysol=RKSymbolic[x0,y0,A,b,2,10,0.2];
```

The result is a list of list data structure containing the corresponding (x, y) pairs, where the y values depend on s.

```
ysol[[2]][[1]]
{{1,2.},{1.2,2.05533+0.200987 s},{1.4,2.22611+0.407722 s},
{1.6,2.52165+0.625515 s},
{1.8,2.95394+0.859296s}, {2.,3.53729+1.11368s},
{2.2,4.28801+1.39298 s},
{2.4,5.22402+1.70123 s},{2.6,6.36438+2.0421 s},
{2.8,7.72874+2.41888 s},{3.,9.33669+2.8343 s}}
```

Consequently, we have got a symbolic result using traditional numerical Runge-Kutta algorithm.

4. Solving Boundary Value Problem

In order to compute the proper value of the unknown initial value, s, the boundary condition can be applied at $x = 3$. In our case $y1(3) = -1$.

```
eq=ysol[[1]][[1]]==-1
9.33669+2.8343 s==-1
```

Let us solve this equation numerically, and assign the solution to the symbol s:

```
sol=Solve[eq,s]
{{s -> -3.647}}
s=s/.sol
{-3.647}
s=s[[1]]
-3.647
```

Then we get the numerical solution for the problem:

```
ysol[[2]][[1]]
{{1,2.},{1.2,1.32234},{1.4,0.739147},{1.6,0.240397},
{1.8,-0.179911}, {2.,-0.524285},{2.2,-0.792178},
{2.4,-0.980351},{2.6,-1.08317}, {2.8,-1.09291},{3.,-1.}}
```

The truncation error can be decreased by using smaller step size h, and the round off error can be controlled by the employed number of digits.

5. Mechanical Analysis of a Liquid Storage Tank

Let us consider a cylindrical liquid storage tank, where the thickness of wall/radius ratio is small enough to ensure membrane stress state, see *Fig. 1*. The four differ-

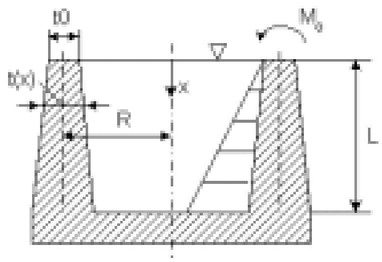

Fig. 1. Liquid storage tank

ential equations describing the deflection, $w(x)$, rotation, $\alpha(x)$, bending moment, $M_b(x)$, and transverse shear force, $F_Q(x)$ distributions along the length of the storage are the following [6]:

$$\begin{aligned}
\frac{\partial}{\partial x} w(x) &= \alpha(x), \\
\frac{\partial}{\partial x} \alpha(x) &= -\frac{M_b(x)}{No\,\delta(x)}, \\
\frac{\partial}{\partial x} M_b(x) &= F_Q(x), \\
\frac{\partial}{\partial x} F_Q(x) &= Do\,\delta(x) w(x) - \gamma x,
\end{aligned}$$

where

$$\begin{aligned}
Do &= \frac{E t_0}{R^2}, \\
\delta(x) &= \frac{t(x)}{t_0}
\end{aligned}$$

and

$$No = \frac{E t_0^3}{12(1-\nu^2)}.$$

The boundary conditions are:

$$\begin{aligned}
w(L) &= 0, \\
\alpha(L) &= 0, \\
M_b(0) &= 0, \\
F_Q(0) &= 0.
\end{aligned}$$

Let us suppose that the thickness of the wall is linear function of x, namely

$$t(x) = \frac{1 + \frac{x}{L}}{2}.$$

Introducing variables y_i, $i = 1, 2, 3, 4$:

$$\begin{aligned} y_1(x) &= w(x), \\ y_2(x) &= \alpha(x), \\ y_3(x) &= M_b(x), \\ y_4(x) &= F_Q(x), \end{aligned}$$

one may get:

$$\begin{aligned} \frac{\partial}{\partial x} y_1(x) &= y_2(x), \\ \frac{\partial}{\partial x} y_2(x) &= -\frac{8 y_3(x)}{No\left(1 + \frac{x}{L}\right)^3}, \\ \frac{\partial}{\partial x} y_3(x) &= y_4(x), \\ \frac{\partial}{\partial x} y_4(x) &= \frac{Do\left(1 + \frac{x}{L}\right) y_1(x)}{2} - \gamma x \end{aligned}$$

and

$$\begin{aligned} y_1(L) &= 0, \\ y_2(L) &= 0, \\ y_3(0) &= 0, \\ y_4(0) &= 0. \end{aligned}$$

The matrix form of the system is:

$$\left[\frac{\partial}{\partial x} y_1(x), \frac{\partial}{\partial x} y_2(x), \frac{\partial}{\partial x} y_3(x), \frac{\partial}{\partial x} y_4(x)\right] =$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -8\frac{1}{No\left(1+\frac{x}{L}\right)^3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{2} Do\left(1 + \frac{x}{L}\right) & 0 & 0 & 0 \end{bmatrix} [y_1, y_2, y_3, y_4] + [0, 0, 0, -g\rho x].$$

Let us consider the following data:

```
R=5.
L=1.5
E=4000000000.
t0=0.1
ρ =1000.
ν =1/6
g=9.81
```

then

```
A[x_] := {{0, 1, 0, 0}, {0, 0, -8/(No(1 + x/L)^3), 0},
          {0,0, 0, 1}, {D0/2(1 + x/L), 0, 0, 0}}
b[x_]:={0,0,0,-g ρ x}
x0=0
y0={s1,s2,0,0}
m=4
n=100
```

where $s1$ and $s2$ are the unknown initial values.

```
ysol=RKSymbolic[x0,y0,A,b,m,n,L/n];
```

Now we have a linear equation system for the unknown values

```
eq1=ysol[[1]][[1]]==0
0.00299378-8.41784 s1-1.59483 s2==0

eq2=ysol[[1]][[2]]==0
0.00656905-13.3954 s1-6.51855 s2==0
```

The solutions are

```
sol=Solve[{eq1,eq2},{s1,s2}]
{{s1 -> 0.000269739,s2 -> 0.000453442 }}
```

We assign these values to the symbolic solution

```
s={s1,s2}/.sol
{{0.000269739,0.000453442}}
```

Let us display the different curves of the mechanical analysis, evaluated with MATHEMATICA (see *Fig. 2*:

6. Conclusions

The extended form of the slope retention technique can be applied to solve linear boundary value problems for non-autonomous linear differential equation systems. To carry out this application, symbolic Runge-Kutta technique can be employed followed by numerical solution of a linear algebraic system representing the boundary conditions at the end of the integration interval.

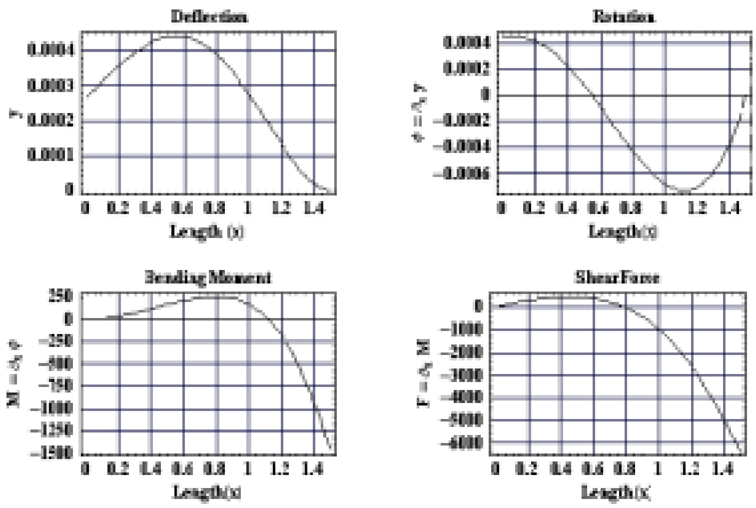

Fig. 2. Graphical representation of the mechanical analysis

Although the method needs only one integration pass, because of symbolical operations, the computation speed will slow down as compared with the pure numerical integration. In the future, the speed of symbolic calculation could be increased considerably through software and hardware developments similar to graphical operations.

Both examples demonstrated that the proposed method could be useful especially in the case of systems with many unknown initial conditions.

Acknowledgements

The first author is thankful to the *Wolfram Research* for providing access to the MATHEMATICA symbolic system.

This study was partly supported by the Hungarian Scientific Research Foundation under Grant No. OTKA T025258 and T029771.

References

[1] BIRKELAND, B. (1997) : Mathematics with Mathcad, Chartwell-Bratt Ltd., Studentlitteratur, Sweden.

[2] MEADE, D. B. et. al. (1996): The Shooting Technique for the Solution of Two-Point Boundary Value Problems, *MapleTech*, Vol. 3, No. 1, p. 85.

[3] FINLAYSON, B. A. (1980) : Nonlinear Analysis in Chemical Engineering, McGraw-Hill, New York.
[4] ZHANG, J. (1996): Symbolic Computation on Complex Polynomial Solution of Differential Equations, *Symbolic Computation*, Vol. 22, p. 345.
[5] MILLS, R. D. (1992) : Slope Retention Technique for Solving Boundary-Value Problem in Differential Equations, *Symbolic Computation*, Vol. 13, p. 59.
[6] KRAWIETZ, A. (1997): Maple V für das Ingenieurstudium, Springer, Berlin.