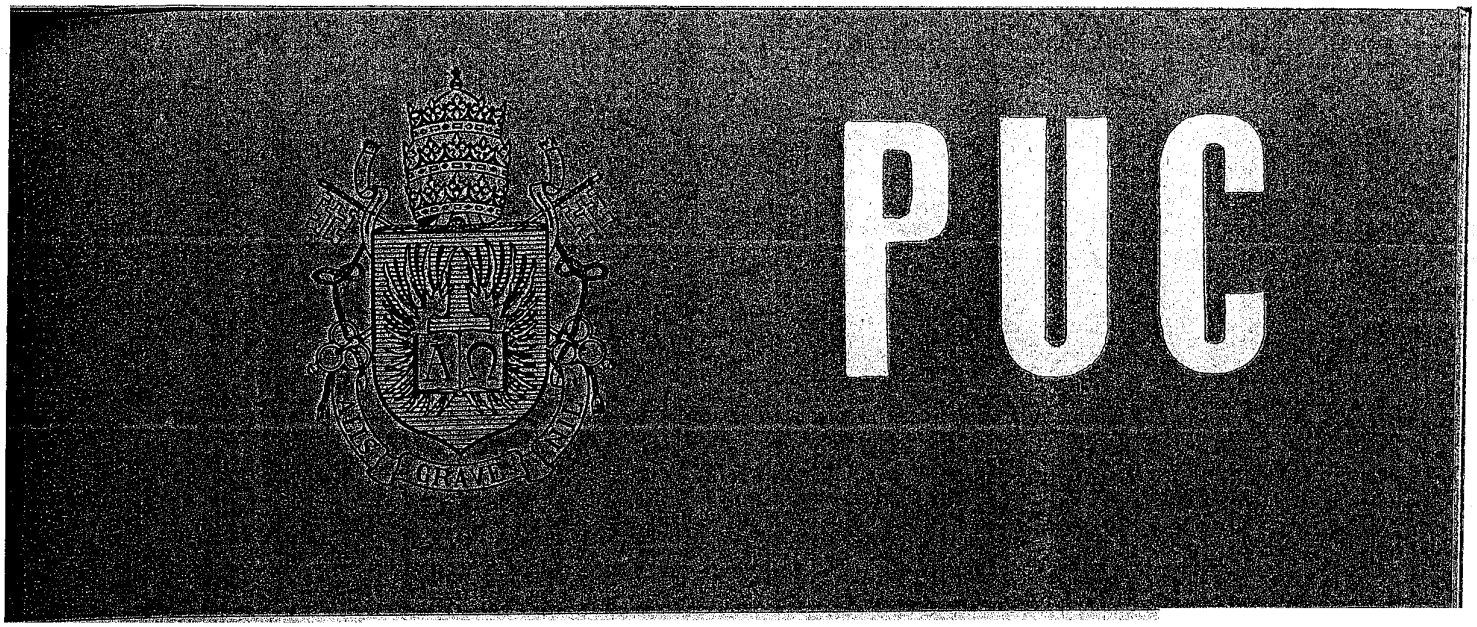

Series: Monographs in Computer Science
and Computer Applications

Nº 4/69

SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY THE TAYLOR SERIES METHOD USING FORMULA MANIPULATION

by

ANTÔNIO LUZ FURTADO

Computer Science Department - Rio Datacenter

CENTRO TECNICO CIENTIFICO
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY THE TAYLOR SERIES METHOD USING FORMULA MANIPULATION

ANTÔNIO LUZ FURTADO
COMPUTER SCIENCE DEPARTMENT
PUC - RIO DE JANEIRO

SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY THE TAYLOR SERIES METHOD USING FORMULA MANIPULATION

1. Introduction

The use of Taylor series has been avoided in computer programming on account of the necessity of obtaining high-order derivatives by hand.
Formula manipulation, as described in 3 , automates differentiation and provides mechanisms for the evaluation of resulting derivatives. This makes it practical to use Taylor series, for instance, in the solution of ordinary differential equations.

One advantage is tis generality:
Taylor's formula remains the same as we add to it as many terms as desired.
It is also very easily extended to solve equations of any order, or systems of any number of equations.
Tests were performed on the IBM - 7044 of the "Pontificia Universidade Católica do Rio de Janeiro" - Brasil.

2. The Algorithm

Taylor's formula for ordinary differential equations is:

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2!} y_n'' + \frac{h^3}{3!} y'''n + \ldots (1)$$

being given $y' = f(x,y)$ (as the differential equation), and the initial condition $y_0 = f(x_0)$. To compute (1) with terms of up to the n^{th} order we must differentiate $y'=f(x,y)$ n-1 times, with respect to x.

However the variable y will be considered as a constant, unless we use a special notation to indicate that it is a function of x.
To illustrate this notation, consider 2 :
$y' = y^2 + \frac{1}{x}$ It would be rewritten as (note the use of $f_0(y)$ instead of $f_0(x)$ to repre

sent y):

$$y'' = f_0(y)^2 + \frac{1}{x} \quad (2)$$

Differentiating (2) twice, we obtain:

$$y'' = 2 f_0(y) f_1(y') - \frac{1}{x^2} \quad (3)$$

$$y''' = 2 f_1(y')^2 + 2 f_0(y) \cdot f_2(y'') + \frac{2}{x^3} \quad (4)$$

Note that differentiation rules for these special functions are:

$$\frac{d f_0(y)}{dx} = f_1(y'); \quad \frac{d f_1(y')}{dx} = f_2(y'')$$

And their values are their own arguments:

$$f_0(y) = y; \quad f_1(y') = y' ; \quad f_2(y'') = y''$$

A computer program to implement this algorithm would take as input:

a. the differential equation, not as a function sub-program, but as data;

b. the order of the highest order derivative to be used;

c. the initial condition x_0, y_0;

d. the number of points of the solution curve to be computed;

e. the interval h between two consecutive points.

Then, successive points x_{k+1}, y_{k+1} would be obtained using formulas (2), (3), (4), (1), in this sequence.

3. An example

Equation: $y' = \frac{y^2 - 1}{xy - y}$ (example taken from [ ])

Analytic solution: $y = \sqrt{\frac{1 - (x-1)^2}{\lambda}}$

For $\lambda = -1$ we have $y(x = -1.45) = 2.6462$, as our initial values.

With the interval $h = 0.1$ we computed 50 points, using a third and then a fourth order process. Results (y_3, y_4) are listed below, together with the analytic solution (y_a).

x	y_3	y_4	y_a
-1.45	2.6462	2.6462	2.6462
-1.35	2.5539	2.5539	2.5539
-1.25	2.4622	2.4622	2.4622
-1.15	2.3712	2.3712	2.3712
-1.05	2.2809	2.2809	2.2809
-0.95	2.1915	2.1915	2.1915
-0.85	2.1030	2.1030	2.1030
-0.75	2.0156	2.0156	2.0156
-0.65	1.9294	1.9294	1.9294
-0.55	1.8446	1.8446	1.8446
-0.45	1.7614	1.7614	1.7614
-0.35	1.6800	1.6800	1.6800
-0.25	1.6008	1.6008	1.6008
-0.15	1.5240	1.5240	1.5240
-0.05	1.4500	1.4500	1.4500
0.05	1.3793	1.3793	1.3793
0.15	1.3124	1.3124	1.3124
0.25	1.2500	1.2500	1.2500
0.35	1.1927	1.1927	1.1927
0.45	1.1413	1.1413	1.1413
0.55	1.0966	1.0966	1.0966
0.65	1.0595	1.0595	1 0595
0.75	1.0308	1.0308	1.0308
0.85	1.0112	1.0112	1.0112
0.95	1.0013	1.0012	1.0012
1.05	1.0013	1.0012	1.0012
1.15	1.0114	1.0112	1.0112
1.25	1.0314	1.0308	1.0308
1.35	1.0607	1.0595	1.0595
1.45	1.0986	1.0966	1.0966
1.55	1.1441	1.1413	1.1413

1.65	1.1965	1.1928	1.1927
1.75	1.2548	1.2501	1.2500
1.85	1.3183	1.3126	1.3124
1.95	1.3863	1.3795	1.3793
2.05	1.4581	1.4502	1.4500
2.15	1.5332	1.5242	1.5240
2.25	1.6111	1.6010	1.6008
2.35	1.6915	1.6803	1.6800
2.45	1.7740	1.7617	1.7614
2.55	1.8584	1.8449	1.8446
2.65	1.9443	1.9297	1.9294
2.75	2.0316	2.0159	2.0156
2.85	2.1202	2.1033	2.1030
2.95	2.2098	2.1918	2.1915
3.05	2.3004	2.2813	2.2809
3.15	2.3918	2.3716	2.3712
3.25	2.4840	2.4627	2.4622
3.35	2.5768	2.5544	2.5539
3.45	2.6702	2.6467	2.6462

execution (IBM - 7044) - third order - 37 seconds fourth order-111 seconds.

4. References

1. Massarani,G. - "Introdução ao Cálculo Numérico" - Livro Técnico - 1967

2. Wylie,C.R. "Advanced Engineering Mathematics" - Mc Graw Hill - 1966

3. Furtado,A.L. and Freire,F. - "Formula Manipulation and the Fletcher and Powell Optimization Method" - PUC-RDC-1969.