

First Order ODEs: Mathematica and Symbolic-Numerical Methods

Ciro D'Apice, Giuliano Gargiulo, and Manzo Rosanna

University of Salerno, Department of Information Engineering and Applied
Mathematics, via Ponte don Melillo,
84084 Fisciano (SA), Italy
{dapice, gargiulo, manzo}@diima.unisa.it

Abstract. The use of information technology in addition to traditional lectures affords a means to develop student intuition and curiosity, reaching in the same time a deep knowledge of the subject of study. The aim of this work is to show the didactic use of a Computer Algebra System to illustrate and compare different symbolic-numerical methods for solving first order ordinary differential equations (ODEs). In particular, we apply, relate and compare the built-in functions of Mathematica, the method of integration by series, the Picard process and the linearization method in solving some first order ODEs. This approach allows students not only to master the basic methods for solving ODEs, but also to be naturally led to theoretical deepening of such areas as power series, stability and convergence theory, elements of functional analysis or the local-global relationship via linearization.

1 Symbolic Methods in Solving ODEs

1.1 Introduction

Consider a differential equation of the first order in normal form

$$y' = f[x, y] \tag{1}$$

and one of its solutions, that is a function $u: J \rightarrow \mathbb{R}$, where J is an interval which satisfies :

$$u'[x] = f[x, u[x]] \quad \forall x \in J. \tag{2}$$

If f is continuous (C^0), the right-hand member of (2) is continuous, since u is C^1 (continuously differentiable). If f is more smooth, for example, C^1 the right-hand member of (2) (being composition of C^1 functions) is C^1 and therefore u is C^2 . For induction it can be easily proved that, if f is C^k (k -time continuously differentiable), u is $k+1$ -time continuously differentiable, if f is infinitely continuously differentiable, also u is infinitely continuously differentiable.

In these cases, the derivatives at x_0 of a Cauchy problem solution

$$\begin{aligned} y' &= f[x, y] \\ y[x_0] &= y_0 \end{aligned} \tag{3}$$

can be computed starting from the derivatives of f at (x_0, y_0) . In fact, obviously it turns out that

$$\begin{aligned} u[x_0] &= y_0 \\ u'[x_0] &= f[x_0, y_0] \end{aligned} \tag{4}$$

Moreover, computing derivatives of (2) - it is possible if f is enough regular - we obtain:

```

eq = u'[x] == f[x, u[x]];
relat = {x -> x0, u[x0] -> y0, u'[x0] -> f[x0, y0]};
newrelat = D[eq, x] /. relat
relat = Join[relat, {u'[x0] -> newrelat[[2]]}];

u''[x0] == f[x0, y0] f^{(0,1)}[x0, y0] + f^{(1,0)}[x0, y0]

newrelat = D[eq, {x, 2}] /. relat
relat = Join[relat, {u''[x0] -> newrelat[[2]]}];

u^{(3)}[x0] == f^{(0,1)}[x0, y0] (f[x0, y0] f^{(0,1)}[x0, y0] + f^{(1,0)}[x0, y0]) +
f[x0, y0] f^{(1,1)}[x0, y0] +
f[x0, y0] (f[x0, y0] f^{(0,2)}[x0, y0] + f^{(1,1)}[x0, y0]) + f^{(2,0)}[x0, y0]
    
```

It is clear that if we go on in this way all the derivatives of u at x_0 can be computed (when the derivatives of f exist and are known).

1.2 ODE Integration by Series

Alternatively - assuming for simplicity that $x_0=0$ - we suppose that f admits Taylor development respect to y near x , for all x in a neighbourhood of x_0 :

```

uSer = SeriesData[x, 0, Table[a[k], {k, 0, 3}], 0, 4] /. a[0] -> y0
y0 + a[1] x + a[2] x^2 + a[3] x^3 + O[x]^4
    
```

So also u has a Taylor development:

```

fSer = SeriesData[y, y0, Table[b[k, x], {k, 0, 3}], 0, 3]
b[0, x] + b[1, x] (y - y0) + b[2, x] (y - y0)^2 + O[y - y0]^3
    
```

We require that the composite of the two series (i.e., the series of the composite function $f[x, u[x]]$) is equal to the derived series of u (i.e., the series of derivatives) ...

```

eqcoeffSer = LogicalExpand[(CoefficientList[#1, x] &)] /@ eqSer]
b[0, 0] == a[1] && a[1] b[1, 0] + b^{(0,1)}[0, 0] == 2 a[2] &&
a[2] b[1, 0] + a[1]^2 b[2, 0] +
a[1] b^{(0,1)}[1, 0] + 1/2 b^{(0,2)}[0, 0] == 3 a[3]
    
```

```
Solve[eqcoeffSer, Table[a[k], {k, 3}]] // Flatten
{a[3] -> 1/6 (b[0, 0] b[1, 0]^2 + 2 b[0, 0]^2 b[2, 0] +
  b[1, 0] b^(0,1)[0, 0] + 2 b[0, 0] b^(0,1)[1, 0] + b^(0,2)[0, 0]),
 a[2] -> 1/2 (b[0, 0] b[1, 0] + b^(0,1)[0, 0]), a[1] -> b[0, 0]}
```

Obviously, the coefficients of the series can be expressed in terms of the derivatives of u and f , so this approach is equivalent to the previous...

```
Series[f[x, y], {x, 0, 2}]
f[0, y] + f^(1,0)[0, y] x + 1/2 f^(2,0)[0, y] x^2 + O[x]^3
Series[u[x], {x, 0, 2}]
u[0] + u'[0] x + 1/2 u''[0] x^2 + O[x]^3
```

1.3 Picard Method and Linearization

In the Picard iterative process, we search the fixed point of a transformation among abstract spaces (Banach spaces). We can expect that if we approximate this transformation with a simpler one (for example a linear one) the fixed point we find in this way is on one side the approximation of the searched fixed point (solution of the ODE) and moreover on the other side it is simpler to compute. The transformation is:

$$T[y][x] = y_0 + \int_{x_0}^x f[t, y[t]] dt \tag{5}$$

The best local linear approximation is the differential that in our case can be written in the following form:

$$\begin{aligned}
 T'[y, \varphi][x] &= \int_{x_0}^x \partial_y f[t, y[t]] \varphi(t) dt \\
 dT'[y, \varphi][x] &= T'[y][x] + T'[y, \varphi - y][x] = \\
 &= y_0 + \int_{x_0}^x f[t, y[t]] dt + \int_{x_0}^x \partial_y f[t, y[t]] (\varphi[t] - y[t]) dt
 \end{aligned}
 \tag{6}$$

to be seen as a function of φ , with y fixed). It is easy to see that the fixed point of dT (computed in y) is a solution of the following problem:

$$\varphi[x] = y_0 + \int_{x_0}^x f[t, y[t]] dt + \int_{x_0}^x \partial_y f[t, y[t]] (\varphi(t) - y[t]) dt$$

The latter is a linear (non homogeneous) ODE and therefore Mathematica is able to solve it symbolically. The solution of (6) or (7) is an approximation of the solution of the initial ODE.

$$\begin{aligned}
 \varphi[x_0] &= y_0 \\
 \varphi'[x] &= f[x, y[x]] + \partial_x f[x, y[x]] (\varphi[x] - y[x])
 \end{aligned}
 \tag{7}$$

It can be used as an approximation or as initial point (near, and so with fast convergence rate) in the Picard process applied to the original ODE. Also, it can be used as an abstract version of Newton-Raphson method. In particular, the search of the fixed points of the transformation (5) is equivalent to the search of the fixed points of the associated Newton-Raphson transformation:

$$NR(T)[y] = (y - T[y]) - (Id - T'[y])^{-1}(y - T[y]) \tag{8}$$

where Id is the identical transformation and $(\cdot)^{-1}$ is the transformation inverse. Since the inversion corresponds to the solution of a linear ODE we can apply again, even if only in part, the power of symbolic calculus of Mathematica to NR(T). We remark that analogous results are valid if T' and dT are computed with a fixed approximation $y=y_1$, instead of a variable one.

2 The Particular Case $y' = y^2$

Dsolve

```
solution =
y[x] /. Flatten[DSolve[{y'[x] == y[x]^2, y[0] == 1}, y[x], x]]
-----
1
1 - x
```

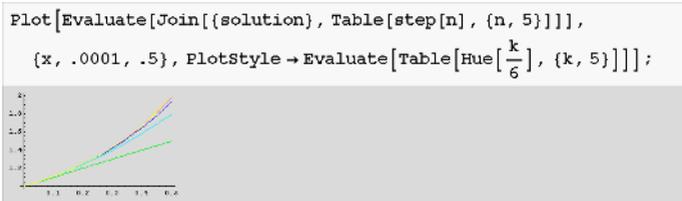
Picard method

Let us define the following integral function. We can observe that as it follows from the definition, when we give in input only the function H, it assumes for default 0 as initial point and x as final point.

```
H[function_, variable_: x, intialpoint_: 0] :=
Module[{y},
y[variable] /.
Flatten[DSolve[{y'[variable] == function^2, y[intialpoint] == 1},
y[variable], variable]]]
Clear[step]; step[1] = Nest[H, 1, 1]
1 + x
Table[step[n] = Nest[H, 1, n], {n, 2, 5}]
{
1/3 (3 + 3 x + 3 x^2 + x^3),
1/63 (63 + 63 x + 63 x^2 + 63 x^3 + 42 x^4 + 21 x^5 + 7 x^6 + x^7),
1/238140 (238140 + 238140 x + 238140 x^2 + 238140 x^3 + 238140 x^4 +
206388 x^5 + 158760 x^6 + 109620 x^7 + 67095 x^8 + 36120 x^9 +
16632 x^10 + 6300 x^11 + 1890 x^12 + 420 x^13 + 60 x^14 + 4 x^15), ...}
```

```
Normal[Series[ $\frac{1}{1-x}$ , {x, 0, 5}]]
1 + x + x2 + x3 + x4 + x5
Normal[Series[step[5], {x, 0, 5}]]
1 + x + x2 + x3 + x4 + x5
```

By comparison of the result obtained with Picard iterative process and the exact solution we can see that the difference between the two terms decreases. We plot the approximations of the solutions at the first 5 steps and the exact solution to evaluate their difference.



We are thus led to conjecture that the exact solution is $1+x+x^2+x^3+x^4+x^5+\dots$, i.e.

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

Linearization

```
T[function_, variable : x, initialpoint_ : 0] :=
```

$$\int_{\text{initialpoint}}^{\text{variable}} \text{function}^2 \, d\text{variable} + 1$$

```
L[function_] :=
```

```
y[x] /.

```

```
Flatten[DSolve[{y' [x] - 2 y[x] == D[function, x],
y[0] == (function /. x -> 0)}, y[x], x]]
```

```
P[f_] := f - L[f - T[f]]
```

```
Timing[p1 = P[1]]
```

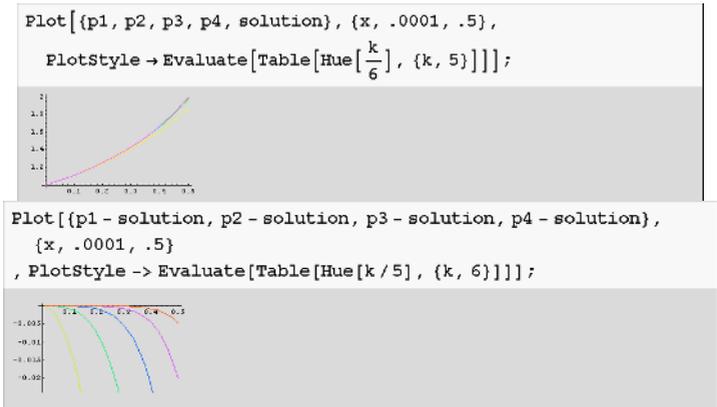
```
{0.04 Second,  $1 + \frac{1}{2} (-1 - e^{2x})$ }
```

```
Timing[p2 = P[P[1]]]
```

```
{0.23 Second,  $1 + \frac{1}{2} (-1 + e^{2x}) + \frac{1}{3} (-1 - e^{4x} - 4 e^{2x} x)$ }
```

```
Timing[p3=P[p3]]...
```

```
Timing[p4=P[p4]]...
```



By comparison of the plots it turns out that the last method seems to approximate the solutions with a greater rate than the Picard process.

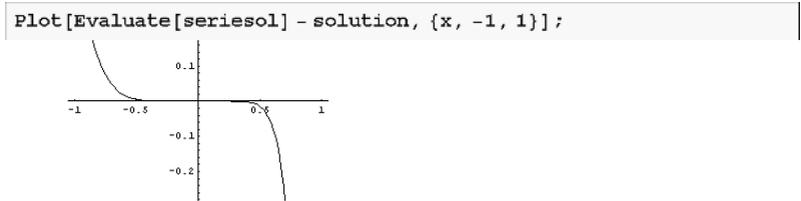
Integration by series

```
y[x_] := SeriesData[x, 0, Table[a[i], {i, 0, 6}]]
y[x]
a[0] + a[1] x + a[2] x2 + a[3] x3 + a[4] x4 + a[5] x5 + a[6] x6 + O[x]7
```

```
syssODE = LogicalExpand[Dx y[x] == y[x]2]
a[0]2 - a[1] == 0 && 2 a[0] a[1] - 2 a[2] == 0 &&
a[1]2 + 2 a[0] a[2] - 3 a[3] == 0 &&
2 a[1] a[2] + 2 a[0] a[3] - 4 a[4] == 0 &&
a[2]2 + 2 a[1] a[3] + 2 a[0] a[4] - 5 a[5] == 0 &&
2 a[2] a[3] + 2 a[1] a[4] + 2 a[0] a[5] - 6 a[6] == 0
```

```
coeff = Flatten[Solve[{syssODE, a[0] == 1}, Table[a[i], {i, 0, 6}]]]
{a[6] > 1, a[5] > 1, a[4] > 1, a[3] > 1, a[2] > 1, a[1] > 1, a[0] > 1}
```

Now we plot the difference between the solution approximated by "series" seriesol=Normal[y[x]] /. coeff (1+x²+x³+x⁴+x⁵+x⁶) and the symbolic solution .



References

1. Albano, G., D'Apice, C., Tomasiello, S.: Simulating Harmonic Oscillator and electrical circuits: a didactical proposal. *ijMEST*, Vol.33, No 2, (2002) 157-185.
2. D'Apice, C., De Simone, T., Manzo, R., Tibullo, V.: MOSFET: a Virtual Laboratory with *Mathematica*. Proceedings ICTMT6 Volos, Greece, (2003) 252-257.
3. Kamps F.,Lockheed M.: Iterative Solution of Highly Nonlinear Differential Equations Using *Mathematica* WorldWide Mathematica Conf. (1998).
4. Mezzino, M.: Discovering Solutions to Ordinary Differential Equations. WorldWide Mathematica Conf. (1998).
5. Olness, F.: Integrating Mathematica in the Undergraduate Science Curriculum: Teaching Computer Literacy with Mathematica. WorldWide *Mathematica* Conf. (1998).