C20

NUMERICAL TEST SETS FOR ECOSIMPRO

Author: Luis Ordóñez Inda
Empresarios Agrupados A.I.E.
C/ Magallanes 3, 28015 Madrid
E-mail; loi@empre.es

Abstract

A series of numerical tests has been carried out for EcosimPro in which we have tested the precision of the integrator, the reliability of the events detector and the possibility of resolving some special cases. This paper gives a detailed description of the most interesting tests:
- Car suspension model
- Electrical circuit of an amplifier
- System of simple equations for the detection of events
- Celestial mechanics problem

Key words:

DASSL: Method of integration, developed by Linda Petzold [1] and the default integrator used by EcosimPro. It returns the values of the variables as a function of time based on the following input formula:

$f(y,y',t) = 0$ (1)

RADAU5: Integration algorithm used in the majority of the reference solutions [2]. It returns the values of the variables as a function of time based on the following input formula:

$Ay' = f(y,t)$ (2)

where A is a matrix that may be unique.

High Index Wizard: EcosimPro assistant to help resolve high index problems. The function requires that variables be introduced which, due to the links, will no longer be regarded as state variables. We are always offered a default solution.

1 INTRODUCTION

The purpose of this work is to verify that the numerical results obtained in EcosimPro are good. To this end, we have selected numerical test sets which will help us to assess the precision of these results.

The set is comprised of fourteen tests which can be divided into three categories. The first includes purely numerical tests which are basically aimed at determining EcosimPro's precision to the nearest number of decimal places. In the second category we have three special tests by way of which we will try to find a way to integrate the equations; this could be by adding special functions which, as we will see in example 3.1, in principle allow the integration of unique cases, or by resolving high index problems which in principle do not admit direct integration. The third category includes a series of tests whose major difficulty lies in the detection of events. The following is a list of the tests carried out:

NUMERICAL TESTS: This category includes the following tests:
N1: A typical predator-prey problem, in which we have fifteen species interacting.
N2: Three chemical models, one which deals with atmospheric contamination, another which analyses the reactions of the interior of a power plant, and a third which represents the manufacture of a compound at industrial level.
N3: A celestial mechanics problem.
N4: A problem taken from solid state physics, in which we analyse the dynamics of defects on the walls of a lithium sample, where the differential equations are analogous to those of a predator-prey model.

SPECIAL TESTS: This second category consists of:
ES1: Two mechanical problems. These tests involve not so much numerical difficulty as high index problems.
ES2: A unique electrical problem which we have only been able to solve by adding a special function. This will be addressed in detail together with one of the previous problems.

DETECTION OF EVENTS: In the third and last category we have:
EV1: A constrained pendulum.
EV2: A model of collision between spheres.
EV3: An electrical model with circuit breakers.

EV4: A simple system of equations with sudden changes in data.

2 NUMERICAL TESTS

These are the most precise tests from the point of view of the results obtained. As indicated previously, these tests are basically systems of non-linear differential equations of the first order. EcosimPro has no problem integrating them and it is simply a question of determining, to the nearest number of decimal places, the reliability of the result obtained. The comparison is made with the results of RADAU5 [2]. The Pleiades test will be dealt with separately because celestial mechanics problems are less exact than the others. For the chemical tests, Figure 1 shows the evolution of the error made with respect to the reference, calculated as the average of the specific quadratic dispersion in each of the variables based on the tolerance value used in EcosimPro. In other words:

$$\bar{\sigma} = \frac{1}{N}\sum_{i}\left(\frac{y_i - y_{i,ref}}{y_{i,ref}}\right)^2 \qquad (3)$$

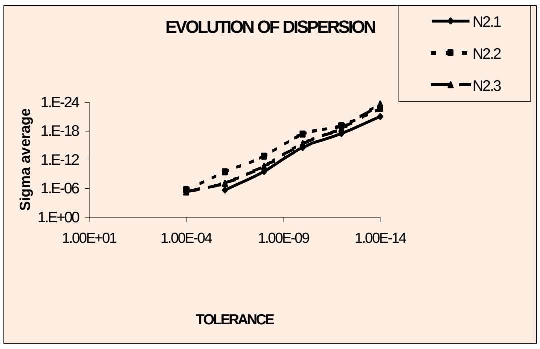

Figure 1

In the figure, we can clearly see that the error follows an evolution which is similar for the three cases and that it decreases with the tolerance used by DASSL, which can be modified with the variables **ABS_ERROR** and **REL_ERROR.**

The precision reached by EcosimPro can be calculated from Figure 1. For example, for a tolerance of 10^{-14}, in the worst case we have a precision of $\sigma = 10^{-20}$, whereby the relative error is in the tenth significant figure.

3 SPECIAL TESTS

These tests are characterised by the difficulty of the system of equations, whether this be due to the existence of singularities, or because they are high index problems. In any case the precision obtained is less, although this is because very low tolerance values are not admitted.

What also characterises these tests is the time it takes for integration. Compared to the duration of the numerical tests, integration takes a very long time.

These last two statements can be explained by the appearance of algebraic loops which make iteration necessary and therefore greatly lengthens the integration time. Figure 2 shows a graph which is analogous to Figure 1 for these tests.

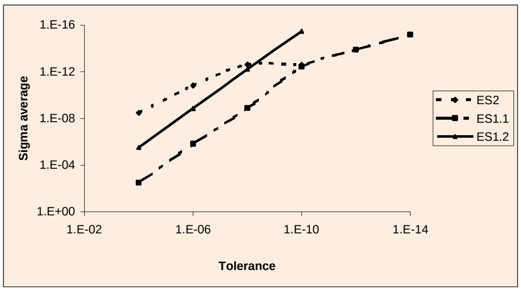

Figure 2

If we compare the two graphs, we can see that the error is slightly greater in these tests than it was in the numerical tests, except in the case of Pleiades. In this case we have a precision that reaches the eighth significant figure.

3.1 TRANSISTOR

This test integrates the equations that originate from a circuit like the one shown in Figure 3.

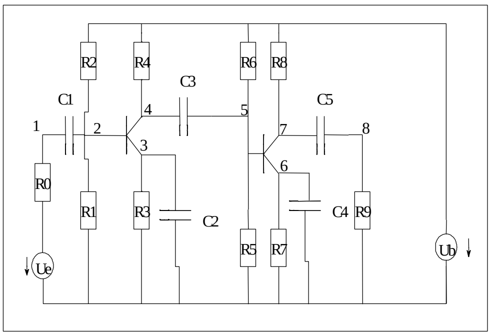

Figure 3. Equivalent transistor circuit

The system of equations that we obtain is unique; that is , as expressed in this formula:

$$M \cdot \frac{dy}{dt} = f(y) \qquad (4)$$

Where M is an 8x8 matrix with a range of five, and “y” is a vector of eight components. Or, in more concrete terms:

$$M = \begin{pmatrix} -\mathbf{C}_1 & \mathbf{C}_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{C}_1 & -\mathbf{C}_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{C}_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{C}_3 & \mathbf{C}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_3 & -\mathbf{C}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mathbf{C}_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{C}_5 & \mathbf{C}_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{C}_{51} & -\mathbf{C}_5 \end{pmatrix}$$

The problem is ready to be integrated by RADAU5. DASSL can integrate it, but EcosimPro will detect a system of linear equations and, for convenience, it will try to invert the matrix, which it is obviously not capable of doing. It is therefore necessary to make some modification so that EcosimPro will not detect that the system is linear. In this case, it is sufficient to define a DIF function which simply subtracts the two arguments. The syntax in EL is as follows:

```
FUNCTION REAL DIF (
        IN REAL V1,
        IN REAL V2)
BODY
        RETURN V1-V2
END FUNCTION
```

We can now write the equations of the first box; for example, in the following way:

DIF(y_2,y_1) = f_1(y, t) (5)
DIF(y_1,y_2) = f_2(y, t) (6)

This is not the general procedure for resolving these types of systems, but we can always do something similar, making adaptations to each particular case.

3.2 CAR SUSPENSION

This test uses a simple model for the suspension of a car, as shown in Figure 4.

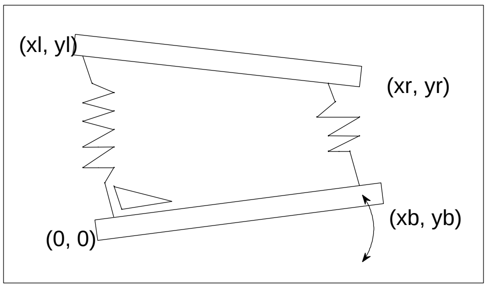

Figure 4. Car suspension model

The equations of motion, with the corresponding constraints, are as follows:

$$\frac{\varepsilon^2 M}{2}\frac{d^2\overline{x_l}}{dt^2} = (l_0 - |\overline{x_l}|)\frac{\overline{x_l}}{|\overline{x_l}|} + \lambda_1\overline{x_b} + 2\lambda_2(\overline{x_l} - \overline{x_r}) - \frac{\varepsilon^2 M}{2}\begin{pmatrix}0\\1\end{pmatrix} \qquad (7)$$

$$\frac{\varepsilon^2 M}{2}\frac{d^2\overline{x_r}}{dt^2} = (l_0 - |\overline{x_r} - \overline{x_b}|)\frac{\overline{x_r} - \overline{x_b}}{|\overline{x_r} - \overline{x_b}|} - 2\lambda_2(\overline{x_l} - \overline{x_r}) - \frac{\varepsilon^2 M}{2}\begin{pmatrix}0\\1\end{pmatrix} \qquad (8)$$

$$x_b^2 + y_b^2 = 1 \qquad (9)$$

$$(x_l - x_r)^2 + (y_l - y_r)^2 = 1 \qquad (10)$$

$$x_l x_b + y_l y_b = 0 \qquad (11)$$

Equations (7)-(11) are of the second order. The advantage of EcosimPro is that it admits equations of the second order, and it will reduce the order so that DASSL can integrate them. RADAU5 on the other hand, like many other integrators, requires that the reduction of order be performed beforehand.

This system presents a high index problem in which EcosimPro will give us help to solve the equations. The test is easily passed observing the physics of the problem. The method consists in detecting the degrees of freedom of the problem and finding variables which represent them.

Now it only remains to introduce the remaining variables to the *High Index Wizard*, which EcosimPro will no longer regard as state variables. In the case in question, looking at Figure 4 it is easy to see that the problem has no more than two degrees of freedom. Three angles determine the position of the two rods and the stretching of the two springs: the two angles formed by the rods with the horizontal axis, and the angle formed by the right spring with the vertical axis; the left spring necessarily forms a right angle with the bottom rod. One of the angles determines the excitation produced by the road, which we will assume to be sinusoidal to simulate the bumps, which means we effectively have two angles. Afterwards, the variables y_l and y_r can be selected as real state variables. This selection has to be made carefully because it is essential to know the good state variables to resolve these types of problems. Another way to resolve the problem is by studying the form of equations (7) - (11). This procedure is carried out by the tool itself, but it does not always obtain the result, especially when the equations are complicated because their form is not sufficiently indicative.

4 DETECTION OF EVENTS

These tests are characterised not so much by the precision of the final results, but by the exactitude with which they measure the time in which the events take place. The clearest example is of a simple system of two differential equations in which the values of the data change, depending on whether the system is found to be in one state or the other:

$y_1' = c_1\,(y_2 + c_2 - y_1)$ (12)
$y_2' = c_3\,(c_4 - y_2)$ (13)

The values of the constants c1 and c_3 are fixed:

$c_1 = 2.7E6;\ c_3 = 3.5651205$

The values of c_2 and c_4 depend on the state in which the system is found:

State1: $c_2 = 0.4;\ c_4 = 5.5$
State2: $c_2 = -0.3;\ c_4 = 2.73$

Both states give an exponential solution for both variables, but in state 1 the solution for y_1 is increasing and in state 2 the solution is decreasing. When y_1 reaches a certain higher elevation, the state changes. Now the solution is decreasing and the change of state takes place when y_1 reaches a minimum elevation. The evolution can be seen in Figure 5.

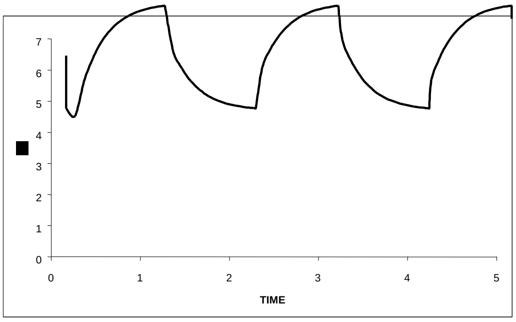

Figure 5

The difficulty of this test lies in the fact that the last discontinuity is produced almost at the end of the integration time, and a badly located event can lead to the final values of y_1 being very different from those of the reference. The problem has an analytical solution [3]:

$y(5) \approx 5.369$ (14)
$time \approx\ 4.999999646$ (15)

Table 1 shows the time taken to locate the last discontinuity based on the tolerance.

Table 1. Discontinuity time

Tolerance	Time(s)
1e-6	4.9999996619500
1e-8	4.9999996464032
1e-10	4.9999996462670
1e-12	4.9999996462643
1e-14	4.9999996462705

The final values of variable y_1 are shown in Table 2.

Table 2. Final values of y_1

Tolerance	$y_1(5)$
1e-6	5.3809963055899
1e-8	5.3694451321339
1e-10	5.3693460753435
1e-12	5.3693441563171
1e-14	5.3693486655019

As previously indicated, a relative error of the order of 10^{-8} in determining the time in which the last event takes place, gives rise in the final result to a relative error of the order of 10^{-2}.

In any case, very precise results are obtained, and the correct value is reached with a tolerance of 1e-8, which is relatively high.

It is worth noting that the tolerance to which we refer in this test is not the same as the one we spoke about earlier. In the other tests, the tolerance mentioned is that which DASSL asks us for, which has been fixed at a value of 10^{-13} for this problem, but in this case the tolerance is that of the statement WHEN, which is the one that intervenes in the detection of events. The default value of this tolerance is fixed at 10^{-6}, but it can be changed in the component.

The following is a code that was written to resolve the problem:

```
COMPONENT Two_state
DATA
  REAL C1 = 2.7e6
  REAL C3 = 3.5651205
  REAL y1_o = 5.8
  REAL y2_o = 2.5
  REAL C21 = 0.4
  REAL C41 = 5.5
  REAL C22 = -0.3
  REAL C42 = 2.73
DECLS
  REAL y1, y2
  REAL C2, C4
INIT
  C2 = C21
  C4 = C41
DISCRETE
   WHEN ( y1 == y1_o) TOL 1e-8 THEN
      C2 = C22
      C4 = C42
   END WHEN
   WHEN ( y1 == y2_o) TOL 1e-8 THEN
      C2 = C21
      C4 = C41
   END WHEN
CONTINUOUS
  y1' = C1 * (y2 + C2 - y1)
  y2' = C3 * (C4 - y2)
END COMPONENT
```

The remaining tests of this type give qualitatively good results, but the references are not as exact and they do not facilitate a thorough comparison.

5 THE PLEIADES PROBLEM

One of the most interesting tests carried out involved a celestial mechanics problem reduced to one plane. It simulated the movement of seven bodies in gravitational interaction which, as is well known, causes many problems when it comes to integration. This is partly explained by the form of the equations:

$$m_i \frac{d^2 r_i}{dt^2} = \sum_{j \neq i} \frac{m_i m_j (r_i - r_j)}{\left| r_i - r_j \right|^3} \qquad i = 1, \dots, 7 \qquad (16)$$

It is again worth noting that EcosimPro will not need to reduce the order of the system of equations, unlike other integrator algorithms. In addition, it allows us to write compact equations, as we can see in the EL code for this component:

```
COMPONENT PLEIADES
DATA
   REAL  m[7] = {1, 2, 3, 4, 5, 6, 7}
DECLS
  REAL   x[7]
  REAL   y[7]
  REAL   r[7,7]
CONTINUOUS
  EXPAND_BLOCK(i IN 1,7)
    EXPAND (j IN 1,7)
      r[i,j] = ((x[i] - x[j])**2 + (y[i] - y[j])**2)**1.5
       x[i]'' = SUM(j IN 1,7 EXCEPT i ; m[j] * (x[j] - x[i]) / r[i,j])
       y[i]'' = SUM(j IN 1,7 EXCEPT i ; m[j] * (y[j] - y[i]) / r[i,j])
  END EXPAND_BLOCK
END COMPONENT
```

If there are quasi-collisions among the stars, that is, they get very close together, one small error in the calculation of the distance between them can lead to a big error in the final result, due to the appearance of the term r_{ij}^3 in the denominator of the equations (16). We can see that these collisions do effectively take place if we take a look at Figure 6, which shows the evolution of the position of two stars during the integration time:

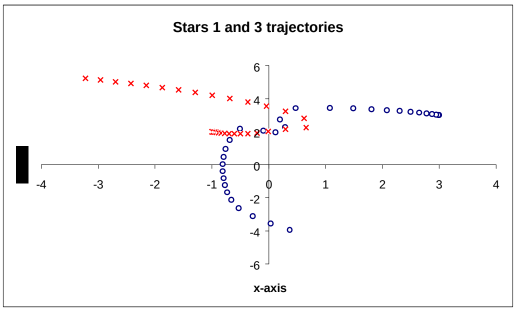

Figure 6. Movement of two stars

The stars drawing nearer in this way is what provokes the big errors which are made for high tolerances.
In any case, precision greatly improves with low tolerance values.
Figure 7 shows the evolution of the error made, calculated with equation (3), based on the tolerance introduced in DASSL.

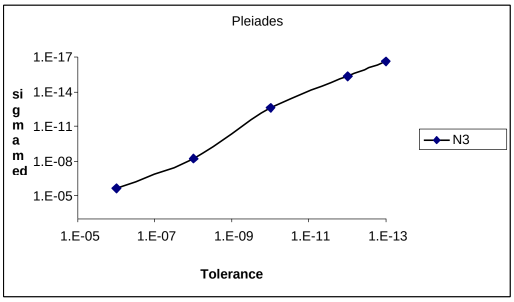

Figure 7. The Pleiades problem

As may be deducted by comparing the results with those given in Figure 1, the evolution of the error is simple, but the values are six orders of magnitude greater than the previous ones. In any case, by lowering the tolerance the precision of the results are sufficiently high that they reach the eighth significant figure.

6 CONCLUSIONS

As a conclusion, several important facts could be pointed out:
- The numerical results obtained by EcosimPro are very precise.
- The response to unique or high index problems is good in the sense that EcosimPro allows these problems to be dealt with, although to resolve them the user will need knowledge of the model.
- The detection of events is very good and does not require very low tolerances in order to obtain precise results.

7 REFERENCES

[Ref. 1] *L.R. Petzold, "A description of DASSL: A Differential/Algebraic System Solver" SAN82-8637, Sandia National Laboratories, Livermore, CA, January 1984*

[Ref. 2] *"Test set for Initial Value Problems"* www.cwi.nl/cwi/projects/IVPtestset/index.htm

[Ref. 3] *"EUROSIM Software Comparison"* http://eurosim.tuwien.ac.at/comparisons/comp_text.html