Mathematica Pannonica
10/1 (1999), 83–92

NUMERICAL AND SYMBOLIC APPLICATIONS OF *MATHEMATICA*

László **Szili**

Department of Numerical Analysis, Eötvös L. University, H–1518 Budapest, Pf. 32, Hungary

János **Tóth**

Department of Computer Science, Agricultural University, H-2103 Gödöllő, Páter K. u. 1, Hungary

Dedicated to Professor Ferenc Schipp on his 60th birthday

Received: October 1998

MSC 1991: 34 D 20, 93 D 20

Keywords: Temperature control, stability of ordinary differential equations, stable polynomials, *Mathematica*.

Abstract: Two nontrivial applications of the mathematical program package *Mathematica* is shown in the field of ordinary differential equations. The first one is a control problem and the second one is related to the stability of polynomials.

1. Introduction

Several areas of human activity required intensive calculations even in the antiquity. Methods to make the calculations easier included the use of tables containing helpful intermediate results (from Babylon to Napier), the construction of mechanical, electric calculators and com-

The present work has been supported by the grants No. FKFP 0941/97 of the Hungarian Ministry for Education and Culture, by OTKA under grant T020497, and MKM (3371/97).

puters (from the abacus through the slide-rule until the personal computers), and the creation of procedures invented by the greatest personalities of applied mathematics (from Archimedes through Newton until Euler) which reduce the amount of the calculations and still increase the precision.

Thus it is no wonder that mathematicians (such as A. Turing, J. von Neumann, J. G. Kemeny or L. Kalmár) played an important role in the early period of computer science (until about 1950). The development starting from the fifties however removed mathematicians from the computer because of different reasons: unkept promises, neglect of the demands of users (including mathematicians), unplanned design of the tools, and first of all the low level of accomplishment.

The only exceptions were statisticians, discrete mathematicians and part of (!) numerical mathematicians. (Computer scientists of this period obviously attribute the fault to the mathematicians.) Even in this period many chemists, physicists and economists used computers for numerical calculations.

As to symbolic calculations: it was Lady Ada Lovelace, daughter of the poet Byron who proposed in 1843 that these might be carried out by machines. It took however more than 100 years until the idea could be realized, because such a long time was needed for the theory and for the technical tools to reach the appropriate level of maturity.

A typical problem in the sixties-seventies was to calculate the derivative or the antiderivative of a function symbolically. This has been solved by many authors (using the Polish form). (Let us remark in passing that the result was used e.g. to write down the sensitivity equations and then making numerical calculations.) This was the period to write the first theorem-proving, chess and translator programs and also the creation of logical programming languages, only to mention a few from nonnumerical applications of computers.

The speed of the computers and developments in the theory made it possible from the beginning of the early eighties the creation and spread of symbolic program packages (or computer algebra systems). These new general purpose mathematical program packages (especially Axiom, Derive, Maple and *Mathematica*) opened up new prospects for teaching, applications and research of mathematics.

The aim of the present paper is to show two nontrivial applications of the mathematical program package *Mathematica*. In our first example we show how to extend its capabilities when solving a control problem.

The second example shows how to decide whether a polynomial is stable or not: an important question in the theory and applications of ordinary differential equations. Thus, both of our examples are connected to differential equations. Further examples related to this field are to be find in [7].

It is almost obvious that the use of mathematical program packages enhances the effectivity of teaching in many fields of mathematics. We are also convinced that they can be used in research and applications, too. However, in order to utilize all the capabilities of a mathematical program package one has to study them thoroughly, for a longer period of time. Our main purpose here is to present part of our experience collected in solving these kinds of tasks.

The first example shows how to keep the temperature of a given body within prescribed limits if Newton's cooling law holds and if we also have a device to heat the body whenever its temperature reaches the lower limit. In the second example we recapitulate a less-known method for checking the stability of polynomials. The method needs much less calculations than the best known method by Routh and Hurwitz. The *Mathematica* code we present is also able to handle cases when the coefficients of the investigated polynomial are symbols (i.e. not numbers). We show how to solve emerging inequalities, using a function of *Mathematica* based upon new theoretical developments (see e.g. [6]). We also use the excellent graphical capabilities of *Mathematica*.

2. A control problem

Let us consider the following simple control problem. One has to keep the temperatura of a body within prescribed limits in such a way that heating is switched on if the temperature reaches the lower limit and it is switched off if the temperature reaches the upper limit.

Suppose the process of cooling is described by Newton's cooling law, according to which the temperature T as a function of t obeys the differential equation

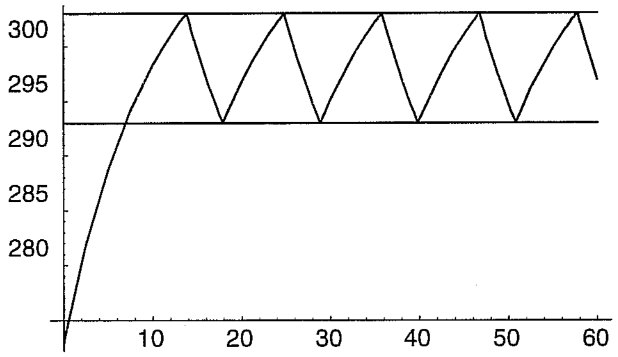
$$T'(t) = -0.1(T(t) - 273).$$

(The values of parameters have arbitrarily been chosen.) Heating goes on if the temperature is either below the lower limit or if it is between the limits and it has been working.

Let us write down the final solution and let us comment it below.

```
NDSolve[{T'[t] == -0.1*(T[t] - 273.)  + Which[
   T[t] < 293., h = 4., T[t] > 303., h = 0.,
   293.  < T[t] < 303.  && h == 4., 4.,
   293.  < T[t] < 303.  && h == 0., 0.],
   T[0.]  == 273.}, T[t], {t, 0., 60.},
   MaxSteps->Infinity, MaxStepSize->0.01]
{{T[t] -> InterpolatingFunction[{0., 60.}, <>][t]}}

Plot[Evaluate[{293., 303., T[t] /.  %}], {t, 0., 60.}]
```



It is worth remarking that the C code of the function NDSolve[ ] is as long as 500 pages. The evaluation process can be modified by an appropriate choice of the options. The help of the program gives possible values of these options and how to use them. If this is not enough to solve our problem then it is worth studying the papers [2 – 4]. Here we only emphasize some possibilities .

Multiple thousands of digits of the approximate solutions can be calculated.

Using the option Method one can choose between well-known methods or their parameters can be fine tuned. The default value of the function NDSolve[ ] is a multistep predictor-corrector method modified by a variable step size.

A special problem in numerical solution of differential equations is posed by stiff equations, in which components of the solution change at very different rates. The procedure NDSolve[ ] automatically detects stiff equations and solves them with a stiff Gear method, which maintains stability without restricting stepsize.

NDSolve[ ] also handles the problem of equations with singularities.

The above example also illustrates a further interesting capability of `NDSolve[ ]`. This makes it possible to solve control problems so simply if we use *Mathematica.* (We propose the reader to think over how a usual differential equation solver routine should be modified in order to solve a problem of the above kind.) If needs a table can be constructed from the values of the approximate solutions.

This above control problem and its generalizations can be applied in many fields. E.g. one should like to keep the concentration level of a drug in the blood between limits. It is easy to find other related problems from chemical technology or production.

As a first generalization we may consider a nonlinear equation instead of the Newton law. Another direction opens if one considers nonconstant limits: firstly, the limits may be explicitly defined function, secondly they can be solutions to other ordinary differential equations. A further generalization is obtained if one chooses a system of differential equations and requires that the trajectory be kept in a prescribed (possibly time-dependent) set. One can also call this a pursuit problem.

3. Stability of equilibrium points

In many cases the problem of the stability of the equilibrium point 0 of the autonomous differential equation

$$\dot{x} = f \circ x \tag{1}$$

given by a function $f : \mathbb{R}^N \to \mathbb{R}^N$, $f(0) = 0$ is reduced to the investigation of the stability of the zero solution of the homogeneous linear differential equation with constant coefficients

$$\dot{y} = f'(0)y, \tag{2}$$

which is known as the first approximation of (1). For example we recall the following result: If the zero solution of (2) is asymptotically stable then the equilibrium point 0 of (1) is also asymptotically stable.

It is well known that the the equilibrium point 0 of (2) is asymptotically stable if and only if all eigenvalues of the $N \times N$ matrix $f'(0)$ lie in the open left half of the complex plane, i.e. all the roots of the characteristic polynomial of the matrix $f'(0)$ have negative real parts. In this case the polynomial is said to be *stable.* In the complementary case the polynomial is said to be *unstable.*

An obvious approach to the problem is to calculate the roots of the polynomial and check if the stability condition holds or not. This is by far the least effective method — especially in the case of symbolic coefficients. Therefore, the genuine problem is: How to check the stability condition only using the coefficients but without finding the roots to the polynomial?

It turned out that it is not a simple problem to find such procedures. The problem emerged at the end of the last century in connection with the centrifugal regulator. Many people were involved in this area, the most well known ones being Maxwell, Hermite, Routh, Stodola, Hurwitz, Liapunov, Liénard, Chippart and Mikhajlov.

If one has an easy to use and efficient tool for drawing functions on a computer like *Mathematica*, one can try to represent the graph of the function using the `Plot[ ]` procedure. But there are some problems, first, we may not get the exact values of the roots graphically, and, second, the `Plot[ ]` function does not show the complex roots.

In 1851, Hermite found a criterion of geometric character for the stability of polynomials. It rediscovered and popularized among engineers by Mikhailov in 1937.

We shall only consider polynomials with real coefficients, i.e. polynomials of the form

$$p(z) := a_N z^N + a_{N-1} z^{N-1} + \cdots + a_1 z + a_0 \qquad (z \in \mathbb{C}), \tag{3}$$

where N is a fixed positive integer and $a_n \in \mathbb{R}$ $(n = 0, 1, \cdots, N)$.

Let us start from the fact that polynomial (3) is an analytical function on the complex plane. As a consequence of Cauchy's theorem it is completely determined by its values taken on a single line of the complex plane. Let this line be the imaginary axis. Then the range Γ_p of the map

$$\mathbb{R} \ni t \mapsto p(it) \in \mathbb{C}$$

is the image of the imaginary axis. This curve is usually called as the *amplitude-phase characteristic* or *Mikhailov's hodograph* of the polynomial p. Certain geometric properties of Γ_p are strongly connected with the stability of the underlying polynomial.

The following statement is said to be the *Hermite–Mikhailov criterion* or the *amplitude-phase criterion* of stability (see [5, p. 37]).

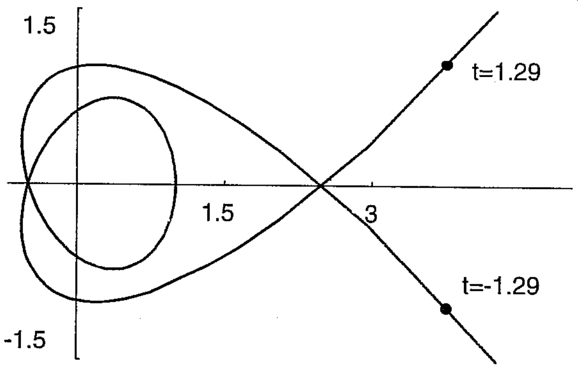
Theorem 1. *The polynomial p is stable if and only if the curve Γ_p does not cross the origin and turns around the origine in positive direction (or: anticlockwise) at an angle of $N\pi$ while its parameter t changes from $-\infty$ to $+\infty$.*

The amplitude-phase characteristic of the polynomial

```
p1[z_] := 4z^5 + 4z^4 + 8z^3 + 5z^2 + 3z + 1
```

can be drawn using *Mathematica* in the following way

```
ParametricPlot[{Re[p1[I*t]], Im[p1[I*t]]}, {t, -6, 6},
    PlotRange->{-1.5, 1.5}, Ticks->{{1.5, 3}, {-1.5, 1.5}},
    Prolog->{Text["t=1.29", {4.5, 1}],
             Text["t=-1.29", {4.5, -1}],
    PointSize[0.015],
    Point[{Re[p1[1.29*I]], Im[p1[1.29*I]]}],
    Point[{Re[p1[-1.29*I]], Im[p1[-1.29*I]]}]}]
```



The segment of the curve suggests that the criterion is fulfilled.

Th. 1 can also be translated into the language of algebra (see e.g. [5, p. 48] and [8], Th. 2).

Another necessary and sufficient condition for the stability problem may be given by means of continued fractions.

It is known that some real rational functions can be written in the following m-terminate continued fraction form:

$$(4)\qquad \cfrac{b_1}{1+\cfrac{b_2 z}{1+\cfrac{b_3 z}{\begin{matrix}\vdots\\ 1+b_m z\end{matrix}}}}$$

where b_k $(k = 1, 2, \cdots, m)$ are appropriate real numbers. It is obvious that there exist rational functions for which such a representation does not exist. Consider for example those which have the zero value at the point $z = 0$.

Now suppose that the rational function

$$(5)\qquad r(z) := \frac{c_0 + c_1 z + \cdots + c_s z^s}{d_0 + d_1 z + \cdots + d_t z^t}$$

has a representation of the form (4). Extend the definition of the coefficients c_k and d_k in (5) in the following way:

$$(6) \qquad c_n := 0 \quad (\text{if } n > s) \qquad \text{and} \qquad d_n := 0 \quad (\text{if } n > t)$$

It may be proved ([1, p. 547]) that in this case the numbers b_k in (4) satisfy the following recursive relations:

$$\begin{aligned} c_n^{(-1)} &:= d_n, \qquad c_n^{(0)} := c_n && (n = 0, 1, 2, \cdots), \\ (7) \quad b_{k+1} &= \frac{c_0^{(k)}}{c_0^{(k-1)}} && (k = 0, 1, 2, \cdots) \\ c_n^{(k+1)} &= b_{k+1} c_{n+1}^{(k-1)} - c_{n+1}^{(k)} && (k = 0, 1, 2, \cdots, \ n = 0, 1, 2, \cdots). \end{aligned}$$

The computation of b_{k+1} (which in this manner is a variant of the original Routh algorithm) starts from the first two rows of $c_n^{(-1)}$ and $c_n^{(0)}$ (the coefficients of the denominator and of the numerator) and then proceeds as indicated in the following table:

$$\begin{tikzcd}
 & c_0^{(-1)} \arrow[dl] & c_1^{(-1)} \arrow[ddl] & c_2^{(-1)} \arrow[ddl] & c_3^{(-1)} \arrow[ddl] & \cdots \\
b_1 & c_0^{(0)} \arrow[l] \arrow[dl] & c_1^{(0)} \arrow[dl] \arrow[ddl] & c_2^{(0)} \arrow[dl] \arrow[ddl] & c_3^{(0)} \arrow[dl] \arrow[ddl] & \cdots \\
b_2 & c_0^{(1)} \arrow[l] \arrow[dl] & c_1^{(1)} \arrow[dl] & c_2^{(1)} \arrow[dl] & c_3^{(1)} \arrow[dl] & \cdots \\
b_3 & c_0^{(2)} \arrow[l] & c_1^{(2)} & c_2^{(2)} & c_3^{(2)} & \cdots \\
 & & & \cdots & &
\end{tikzcd}$$

From Ths. 12.7b, 12.7a and 12.6c of [1] we obtain the continued fraction criterion:

Theorem 2. *The polynomial* (3) *is stable if and only if the rational function*

$$(8) \qquad \frac{a_1 + a_3 z + a_5 z^2 + \cdots}{a_0 + a_2 z + a_4 z^2 + \cdots} = \frac{\sum_{k=0}^{[(N-1)/2]} a_{2k+1} z^k}{\sum_{k=0}^{[N/2]} a_{2k} z^k}$$

(the Hurwitz alternant of the polynomial p) can be represented by such an N-terminating continued fraction of the form (4)*, in which every number* b_k $(k = 1, 2, \cdots, n)$ *is positive.*

A possible code of this criterion follows.

```
CFCriterion[polyvalue_, variable_] :=
   Module[{cf = CoefficientList[polyvalue, variable],
          deg, cfde, cfnu,b, sQ , temp = { }, m = 1},
      deg = Length[cf] - 1;
      sQ = !FreeQ[NumberQ /@ cf, False];
      cfnu = Table[cf[[k]], {k, 2, deg + 1, 2}];
      cfde = Table[cf[[k]], {k, 1, deg + 1, 2}];
      While[((First[cfnu] > 0 && m < deg + 1) ||
             (sQ && m < deg +1)),
           b[m] = First[cfnu]/First[cfde];
         If[Length[cfnu] < Length[cfde],
            cfnu = Join[cfnu, {0}]
           ];
         temp = cfnu;
         cfnu = Drop[b[m] cfde - cfnu, 1];
         cfde = temp;
         m = m + 1
         ];
      If[m < deg, "Unstable", If[!sQ, "Stable",
           Together[Simplify[Table[b[k], {k, 1, m-1}]]]]]
       ]
```

The above procedure uses the symbolic capabilities of *Mathematica*. If the coefficients have exact numerical values (e.g. $1/2, \sqrt{2}, \pi, \log 2$) then we obtain the exact result:

```
CFCriterion[p1[z], z]

Stable
```

If there are parameters involved in the coefficients of the polynomial then the answer still contains relevant information. Let us consider the following examples

```
p2[z_] := z^4 + 4*a*z^3 + 6*a*z^2 + 4*a*z + 1
CFCriterion[p2[z], z]

                -2 + 6 a      1
{4 a, -1 + 6 a, --------, --------}
                -1 + 6 a  -1 + 6 a
```

As it can be seen, in this case those expressions are provided the positivity of which has to be checked. We can use the `InequalitySolve` function of the `Algebra'InequalitySolve'` package to solution the corresponding system of inequalities:

```
<<Algebra'InequalitySolve'
InequalitySolve[4*a<0 && -1+6*a>0 &&
          (2-6*a)/(-1+6*a) && 1/(-1+6*a)>0, a]
    1
a > -
    3
```

Thus the polynomial p2 is stable if and only if $a > 1/3$.

The theoretical problem of solving systems of (polynomial) inequalities in one or more unknowns is more complicated (see e.g. [6]). In this case we can use the `Algebra'AlgebraicInequalities'` package of *Mathematica*.

References

[1] HENRICI, P.: Applied and computational complex analysis, Vol. II., John Wiley & Sons, New York, London, Sydney, Toronto, 1977.

[2] KEIPER, J.: Numerical computation I, in: Selected tutorial notes, Wolfram Research, 1993, 1–84, see also The N functions of Mathematica, Reprint from The Mathematica Conference, Boston, 1992 `http://www.mathsource.com` item number 0203–948.

[3] KEIPER, J.: Numerical computation II, in: Selected tutorial notes, Wolfram Research, 1993, 339–358, see also Mathematica numerics: controlling the effects of numerical errors in computation, Reprint from The Mathematica Conference, Boston, 1992, `http://www.wolfram.com/mathsource` item number 0203–937.

[4] PETERSEN, T.: Differential equations, *The Mathematica Journal* **1**/3 (1991), 33–35.

[5] POSTNIKOV, M.M.: Stable polynomials, Nauka, Moscow, 1981 (in Russian).

[6] STRZEBOŃSKI, A.W.: An algorithm for systems of strong polynomial inequalities, *The Mathematica Journal* 4/4 (1994), 74–77.

[7] SZILI, L. AND TÓTH, J.: Mathematics and Mathematica. Eötvös Kiadó, Budapest, 1996 (in Hungarian).

[8] TÓTH, J., SZILI, L. AND ZACHÁR, A.: Stability of polynomials, *Mathematica in Education and Research* **7**/2 (1998).