Computation of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems

Rudolf J. Lohner

University of Karlsruhe, Germany

1 Introduction

Based on classical numerical methods combined with interval arithmetic, we present methods which are able to verify the existence and compute guaranteed enclosures for the solutions of ordinary initial value problems (IVP) and boundary value problems (BVP). For IVP we use Taylor expansion, where the remainder term is enclosed by interval methods and special care is taken in adding the local error to the global error. For BVP shooting methods are used. The nonlinear systems arising here are solved by interval versions of Newton's method. The presented methods can be programmed in such a way that a user has to supply only the system of ODEs and the initial or boundary conditions in some symbolic form or as subroutines as well as a few numerical parameters. No information about any derivatives has to be supplied. Only sufficiently high smoothness of the problem is required.

In this paper we present methods for the automatic computation of guaranteed bounds on the solutions of ordinary initial and boundary value problems (IVP and BVP).

For IVP we use *stepwise Taylor* expansion where the Taylor coefficients are computed by automatic differentiation, see [19]-[23], and the remainder term is enclosed in an interval in each time step. In addition, we have to take care of the so called 'wrapping effect' which is an intrinsic difficulty in applying enclosure methods to IVP, since it potentially 'blows up' the enclosures, see [7, 10, 19, 22, 25]. We reduce this effect by suitable *coordinate transformations* and by taking intersections of different enclosures. Also, these methods can treat initial sets, e.g. interval vectors, and can bound the solution set from inside and from outside in each component, thus making possible a *true sensitivity analysis*. The enclosure method for IVP will be discussed in Section 2.

For BVP the *most important question is that of existence* of a solution. We treat this question by using a simple or multiple shooting approach. This reduces the problem to the existence of a solution of a finite dimensional nonlinear system of equations. For such systems, however, many methods are known from interval analysis which can prove the existence and compute enclosures of solutions. The methods for BVP will be discussed in Section 3.

The methods can be implemented on a computer very user friendly: only the differential equation, and initial or boundary conditions have to be furnished in some symbolic notation. The present implementation is a code for IVP called AWA and is written in PASCAL-SC, [13, 14], running on several MC68000 computers and on IBM PC compatibles. This code was also ported to FORTRAN-SC, [3], running on IBM/370 computers under VM. The present implementation also needs step sizes, orders and options for the variants of the method. Future implementations, however, will choose these parameters automatically as far as possible. Also, BVP can be treated using this code: all IVP which have to be solved in a Newton step of the shooting method are solved with AWA, then a Newton step is performed by use of one of the self-validating linear system solvers which are available in these languages.

Finally, in Section 4 we present some numerical examples which were computed using the PASCAL-SC implementation of our methods.

We assume that the reader is familiar with interval arithmetic and interval methods for linear and nonlinear systems, see [1, 15, 16, 21, 22, 24]. Also, some background in automatic differentiation, [23], is helpful. Our notation for intervals will be $[a] = [\underline{a}, \overline{a}] = \{a | \underline{a} \le a \le \overline{a}\}$, where $\underline{a}, a, \overline{a}$ may be real numbers, vectors or matrices.

Because of limited space we can only sketch the ideas of our methods. More details are contained in [18] and especially in [19].

2 Initial Value Problems

Consider the following (w.l.o.g. autonomous) initial value problem

$$\left.\begin{array}{c} u' = f(u), \quad u(t_0) = u_0, \quad u_0 \in [u_0] \subseteq D, \\ f \in C^p(D, \mathbf{R}^n), \; D \subseteq \mathbf{R}^n \text{open}, \; u = u(t) \in \mathbf{R}^n, p \ge 2, \end{array}\right\} \tag{2.1}$$

where $C^p(D, \mathbf{R}^n)$ denotes all p times differentiable functions from D into $\mathbf{R}^n$. If $[u_0]$ has positive diameter, (2.1) is a *set* of initial value problems.

Now, let $t_j = jh$, $h > 0$, $j \ge 0$, be a grid (equidistant for simplicity only), and assume that we have computed our algorithm up to the j-th node t_j. We now describe the step from t_j to t_{j+1}.

Let $\Phi = \Phi(u)$ be a one step method (of course Φ depends on f and h), and let z_{j+1} be the local discretization error (multiplied by h) in the interval $[t_j, t_{j+1}]$. Then

$$u_{j+1} = u_j + h\Phi(u_j) + z_{j+1}, \tag{2.2}$$

where $u_j = u(t_j)$ and $u_{j+1} = u(t_{j+1})$ are the *exact values* of the solution, since z_{j+1} is the *exact local error*.

Now, let $z_0 = u_0$, and consider all z_k as independent variables for the moment. Then we can interpret (2.2) as the definition of a function

$$u_{j+1} = u_{j+1}(z_0, \ldots, z_{j+1}) \tag{2.3}$$

of $j+2$ independent variables $z_0, \ldots, z_{j+1}$. This function, of course, yields the value $u(t_{j+1})$ of the solution u at t_{j+1} if the variables $z_0, \ldots, z_{j+1}$ are the exact local errors. However, since we do not know these local errors exactly, we will construct intervals $[z_k]$ containing them. Then an *interval evaluation* of (2.3) over these enclosures of the local errors must yield an enclosure $[u_{j+1}]$ of the solution $u(t_{j+1})$.

In order to be able to compute such enclosures of the local errors, we have to choose our one step method Φ suitably. We require that the local errors z_k can be expressed or at least be estimated in terms of the solution u itself or derivatives of u. A method which meets this requirement is the method of Taylor expansion at the grid points. If u is expanded at t_j in a Taylor polynomial of degree $p-1$ then the local error z_{j+1} is just the remainder term of order p:

$$z_{j+1} = \frac{h^p}{p!} u^{(p)}(\tau_{j+1}), \quad \tau_{j+1} \in (t_j, t_{j+1}). \tag{2.4}$$

The p-th derivative of u can be expressed in terms of u itself by repeated differentiation of the differential equation. This can be done automatically using *automatic differentiation*. So, we could compute an enclosure of z_{j+1} if we already had an enclosure $[u^1_{j+1}]$ of u on $[t_j, t_{j+1}]$. Fortunately, this first enclosure of u does not have to be very narrow: for small h and large p the enclosure of z_{j+1} will still be rather narrow because of the factor $h^p/p!$.

Such a rough first enclosure can be obtained from the following theorem:

Theorem 1 *Let $[u^0] \subseteq D$ be an interval vector such that*

$$[u^1] := [u_j] + [0, h] f([u^0]) \quad \subseteq \quad [u^0]. \tag{2.5}$$

Then, all solutions of all initial value problems $u(t_j) \in [u_j]$ exist on the whole interval $[t_j, t_{j+1}]$, and they all do not leave $[u^1]$ on this interval.

Proof: For the very simple proof, based on Picard iteration, see [19].

If $[u^1]$ is an enclosure of u on $[t_j, t_{j+1}]$ then we can compute an enclosure $[z_{j+1}]$ of the local error by (2.4). Evaluating (2.3) or (2.2) over these intervals we get an enclosure of the solution. However, this enclosure will be far too wide in most cases. Therefore, we have to modify the method: instead of evaluating (2.3) directly, we use a *mean value form* for this evaluation.

We choose arbitrary points s_k from the intervals $[z_k]$, $k = 0, \ldots, j+1$, e.g. the midpoints, and we apply the mean value theorem to (2.2):

$$u_{j+1} = \tilde{u}_{j+1} + \sum_{k=0}^{j+1} \frac{\partial u_{j+1}(\hat{z})}{\partial z_k}(z_k - s_k) \tag{2.6}$$

where $\tilde{u}_{j+1} = u_{j+1}(s_0, \ldots, s_{j+1})$ is an approximation of $u(t_{j+1})$ with $\tilde{u}_0 = s_0 \in [u_0] = [z_0]$ and

$$\tilde{u}_{j+1} = \tilde{u}_j + h\Phi(\tilde{u}_j) + s_{j+1}. \tag{2.7}$$

The argument $\hat{z}$ in (2.6) indicates unknown intermediate vectors (which may be different in different rows). The functional matrices $\partial u_{j+1}(\ldots)/\partial z_k$ can be computed automatically. The unknown intermediate vectors do not present difficulties. Since they all must be contained in the intervals $[z_k]$, we replace $\hat{z}$ by all intervals $[z_k]$ when we evaluate (2.6). Thus the enclosing property will be maintained.

Formula (2.6) is of the form

$$u_{j+1} = \tilde{u}_{j+1} + r_{j+1} \tag{2.8}$$

where u_{j+1} is an approximation and r_{j+1} is the corresponding global error. The quality of the enclosure now will depend strongly on the way in which the sum for the global error r_{j+1} is evaluated in interval arithmetic. A proper choice of this evaluation is essential for obtaining tight bounds on the solution. We sketch some possibilites for evaluating enclosures of r_{j+1}, for more details see [18, 19]. In the following $[A_j] = I + \Phi'([u_j])$.

Evaluation 1 ('interval vector'): This is the simplest method.

$$[r_0] := [z_0] - s_0, \quad r_{j+1} \in [r_{j+1}] := [A_j][r_j] + [z_{j+1}] - s_{j+1} \tag{2.9}$$

We just compute r_{j+1} as an interval vector. This method is fast and simple but gives in many cases bad bounds, e.g. for oscillating solutions the bounds blow up very rapidly.

The next two evaluations use coordinate transformations and differ only in the choice of the basis matrix B_{j+1} in (2.10).

$$\left.\begin{array}{c} r_0 \in B_0[\hat{r}_0], \quad B_0 := I, \quad [\hat{r}_0] := [z_0] - s_0, \\ r_{j+1} \in B_{j+1}[\hat{r}_{j+1}], \\ [\hat{r}_{j+1}] := (B_{j+1}^{-1}[A_j]B_j)[\hat{r}_j] + B_{j+1}^{-1}([z_{j+1}] - s_{j+1}) \end{array}\right\} \tag{2.10}$$

Evaluation 2 ('parallelepiped'): Choose $B_{j+1} \in [A_j]B_j$ for $j \geq 0$, in (2.10), e.g. the midpoint matrix. This yields a parallelepiped as enclosing set for r_{j+1}. For oscillating solutions this method often yields good bounds, but in most other cases it breaks down since the method needs B_{j+1}^{-1} and often this matrix becomes ill conditioned as the interval of integration becomes larger. This is the case e.g. for all linear systems with constant coefficients whose eigenvalues have different real parts.

Evaluation 3 ('QR-decomposition'): Choose $B_{j+1} := Q_{j+1}$ where $\tilde{B}_{j+1} \approx Q_{j+1}R_{j+1}$ and $\tilde{B}_{j+1} \in [A_j]B_j$. This linear transformation encloses r_{j+1} in a rotated rectangle. The transformation matrix B_{j+1} is always (almost) orthogonal. An additional pivoting strategy is used to reorder the columns of $\tilde{B}_{j+1}$ according to their Euclidian length prior to the QR factorization. This method is usually superior to evaluation 1 or 2. It does not break down because of an ill conditioned matrix as in evaluation 2. However, in special cases evaluation 1 or 2 may still be better.

Evaluation 4 ('inner enclosure'): This evaluation is meaningful if the initial value $[u_0]$ is an interval whose diameter is considerably larger than the diameters of the local error intervals $[z_k]$. Here u_{j+1} is rewritten in the form

$$u_{j+1} = \tilde{u}_{j+1} + C_{j+1}(z_0 - s_0) + \tilde{r}_{j+1} \tag{2.11}$$

and $\tilde{r}_{j+1}$ is computed by one of the evaluations 1 - 3. The matrix C_{j+1} is chosen as in evaluation 2, but no inverse of it is needed here. With this evaluation it is possible to compute inner and outer sets for each component of the solution set. This allows a true sensitivity analysis. See [6, 18, 19].

The different evaluations can also be combined by taking the intersection of their respective results and these intersections in turn can be used to improve the enclosures in each individual evaluation method. This way of forming intersections is much more effective than just computing the evaluations independently of each other without any interaction. For example, taking intersections of evaluations 1 and 3 turned out to be the best method in most of the examples which were computed with this algorithm.

Finally, we mention that it is very easy in this method to increase the accuracy of the enclosures to be as high as desired. Since the solution is represented as approximation plus global error, the easiest way of improving the enclosures is to compute the approximation with higher accuracy. Even if multiple precision arithmetic is used to compute $\tilde{u}_{j+1}$ we still can use single precision arithmetic in the computation of r_{j+1} without making the result worse.

3 Boundary Value Problems

Consider the nonlinear boundary value problem

$$\left.\begin{array}{c} u' = f(u), \quad r(u(a), u(b)) = 0, \quad u = u(t) \in \mathbf{R}^n, \quad a < b, \\ f \in C^p(D, \mathbf{R}^n), p \geq 2, D \subseteq \mathbf{R}^n \text{open}, r : D \times D \to \mathbf{R}^n \text{smooth.} \end{array}\right\} \quad (3.1)$$

We want a method which can prove the existence and compute an enclosure of a solution of (3.1). This can be achieved using shooting methods and interval analysis.

First we consider simple shooting. We replace the boundary value problem (3.1) by the initial value problem

$$u' = f(u), \quad a < t < b, \quad u(a) = s \in D. \quad (3.2)$$

We denote its solution as $u(t; s)$ to indicate the dependence on the initial value s.

Now, it is well known and obvious that the solutions of the boundary value problem (3.1) are exactly those solutions of the initial value problem (3.2) which satisfy the finite dimensional system of equations

$$F(s) := r(s, u(b; s)) = 0. \quad (3.3)$$

For multiple shooting we consider the initial value problems

$$u_i' = f(u_i), \quad t_i \leq t \leq t_{i+1}, \quad u_i(t_i) = s_i \in D, \quad i = 0, \ldots, m-1 \quad (3.4)$$

on a grid $t_i := a + i(b-a)/m$. We denote its solutions as $u_i(t; s_i)$. The solutions of the boundary value problem (3.1) now are exactly those solutions of (3.4) for which the s_i are solutions of

$$F(s) := \begin{pmatrix} u_0(t_1; s_0) - s_1 \\ \vdots \\ u_{m-1}(t_m; s_{m-1}) - s_m \\ r(s_0, s_m) \end{pmatrix} = 0 \quad (3.5)$$

where, now, $s \doteq (s_0, \ldots, s_m)$ is a block vector in D^{m+1}.

Thus, in both cases, simple or multiple shooting, we have reduced the question of existence to the question of existence of a solution to a finite dimensional system. For this kind of problems, however, there are many methods available from interval analysis which are able to automatically verify existence and compute enclosures of the solutions, see e.g. [1, 11, 16, 21, 22].

All initial value problems (3.2) or (3.4), which have to be integrated in the evaluation of $F(s)$, must be solved using the methods of Section 2 in order to get enclosures. Also, if we are using an interval Newton method,

we have to enclose the functional matrix $\partial F/\partial s$ of F in (3.3) or (3.5). This means that we also have to compute enclosures for the fundamental matrices of the variational equations of (3.2) or (3.4). All this, however, can be done fully automatically so that a user of an implementation of this method will not have to compute any derivatives. He only has to supply the differential equation, the boundary conditions and some starting values for the Newton iteration.

We omit further details, since they are straightforward once the Newton method is written down for this problem, see [18, 19] and also [9, 27].

4 Numerical Examples

All numerical examples presented in this section were computed with PASCAL-SC programs on an ATARI-ST with a 13 digit decimal arithmetic. We only present a small portion of the examples which have been computed successfully. More examples such as periodic solutions of the linear and a nonlinear Mathieu's equation and the nonlinear pendulum equation, period and amplitude of Van der Pol's equation, Trösch's boundary value problem and others are contained in [18, 19].

4.1 Linear System with Constant Coefficients

This example, due to Bachmann, [2] shows nicely the difference of the enclosures computed with different methods from Section 2. The problem is

$$\begin{array}{rclcrcl} u_1' & = & u_1 - 2u_2, & & u_1(0) & = & 1, \\ u_2' & = & 3u_1 - 4u_2, & & u_2(0) & = & -1. \end{array} \quad (4.1)$$

and has the solution $u_1(t) = 5e^{-t} - 4e^{-2t}$, $u_2(t) = 5e^{-t} - 6e^{-2t}$.

The table below shows the results for the different evaluation methods from Section 2, where 1∩2 means that intersections were taken from method 1 and 2. The step size was $h = 0.1$ and the order $p = 14$ in all cases. The table shows the maximal diameter of the enclosures of both components for selected time steps. Obviously QR-decomposition is far superior in this example.

t	1	2	3	1∩2
10	2.2E-05	5.9E-10	6.6E-14	5.9E-10
20	7.8E+01	6.8E-10	6.7E-18	6.8E-10
27	—	4.0E+28	8.3E-21	8.6E-07
40	—	—	2.8E-26	3.0E+02
100	—	—	6.2E-52	—
150	—	—	1.8E-73	—
200	—	—	4.7E-95	—
250	—	—	8.0E-98	—

4.2 Restricted Three Body Problem

The equations of the well known restricted three body problem, [4, 26], are

$$\left.\begin{aligned} u_1' &= u_3 \\ u_2' &= u_4 \\ u_3' &= u_1 + 2u_4 - \frac{\lambda(u_1+\mu)}{((u_1+\mu)^2+u_2^2)^{3/2}} - \frac{\mu(u_1-\lambda)}{((u_1-\lambda)^2+u_2^2)^{3/2}} \\ u_4' &= u_2 - 2u_3 - \frac{\lambda u_2}{((u_1+\mu)^2+u_2^2)^{3/2}} - \frac{\mu u_2}{((u_1-\lambda)^2+u_2^2)^{3/2}} \\ \mu &= 1/82.45,\ \lambda = 1-\mu, \\ u_1(0) &= 1.2,\ u_2(0) = 0,\ u_3(0) = 0,\ u_4(0) = -1.04935750983, \end{aligned}\right\} \quad (4.2)$$

where the initial conditions belong (approximately) to a periodic orbit of period $T = 6.192169331396$ (approx.). Here, too, only integration using QR-decomposition yields enclosures for one complete revolution of the orbit. All other methods from Section 2 break down earlier.

The integration with step sizes $0.00025 \le h \le 0.05$ and orders $12 \le p \le 14$ results in more than 5 correct digits at $t = T$. When the same problem is computed with $0.0002 \le h \le 0.04$ and $p = 22$, the approximation $\tilde{u}_{j+1}$ is computed in double precision, and the global error r_{j+1} is computed in single precision, then we get more than 11 correct digits after one complete revolution.

4.3 Stiff, Linear, Homogeneous Boundary Value Problem

The stiff, linear, homogeneous boundary value problem

$$u'' - u' - 110u = 0, \quad u(0) = u(10) = 1 \quad (4.3)$$

cannot be treated with simple shooting, since perturbations of the exact initial condition $u(0) = 1$ by ε cause changes of $\approx 10^{37}\varepsilon$ at $t = 10$. Therefore we used multiple shooting with $h = 0.005$, $p = 16$ and $m = 20$ subintervals. We also changed the direction of the integration on the subintervals which produced a very interesting effect. We get:

$$\text{integrations from left to right :} \begin{cases} \text{6 correct digits near } t \approx 0, \\ \text{10 correct digits near } t \approx 10, \end{cases}$$

$$\text{integrations from right to left :} \begin{cases} \text{10 correct digits near } t \approx 0, \\ \text{6 correct digits near } t \approx 10. \end{cases}$$

Taking the *intersection* of both enclosures we get *without any further computation*:

10 correct digits near $t \approx 0$ and near $t \approx 10$ and
8 correct digits near $t \approx 5$!

If we had computed only approximations then at the best we could have taken the mean values of both approximations instead of the intersection, thus obtaining only 6 correct digits near the boundaries.

4.4 Nonlinear Eigenvalue Problem in Lubrication Theory

In [12] the following nonlinear eigenvalue problem from lubrication theory is treated

$$\varepsilon u' = \frac{\pi}{2}(\sin^2\frac{\pi}{2}t - \lambda\frac{\sin^4\frac{\pi}{2}t}{u}), \quad u(-1) = u(1) = 1. \quad (4.4)$$

For different values of $\varepsilon > 0$ we seek λ such that this boundary value problem has a solution.

The table below shows the enclosures for λ and the approximations $\tilde{\lambda}$ (from [12]) of different solution pairs ε, λ. Obviously, the approximations from [12] are quite good for small ε but they are worse for larger values of ε. Now, it is interesting to see, that in [12] from error considerations and alternative computation it was expected that the approximations were good for larger values of ε and worse for smaller values of ε which is just the opposite of the actual behaviour.

ε	$\lambda \in$	$\tilde{\lambda}$ from [12]
2	1.32406570616^{87}_{62}	1.3078921
1	$1.29790652874^{494}_{034}$	1.2932779
0.2	1.06465545^{1041}_{0518}	1.06465511
0.1	1.01865676^{807}_{750}	1.01865686
0.01	1.00019984^{309}_{228}	1.00019988

This shows that even careful error investigations may be misleading if only approximations have been computed and if the error estimation is done approximately, too.

References

[1] Alefeld, G., Herzberger, J. (1983). *Introduction to Interval Computations*, Academic Press, New York.

[2] Bachmann, K.-H. (1983). Fehlereinschließung für Näherungslösungen von Systemen gewöhnlicher Differentialgleichungen, *ZAMM* **63**, 64-65.

[3] Bleher, J.H., *et al.* (1987). FORTRAN-SC: A Study of a FORTRAN Extension for Engineering/Scientific Computation with Access to ACRITH, *Computing* **39**, 93-110.

[4] Bulirsch, R., Stoer, J. (1966). Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods, *Numerische Mathematik* **8**, 1-13.

[5] Corliss, G.F., Krenz, G.S., Davis, P.H. (1988). *Bibliography on Interval Methods for the Solution of Ordinary Differential Equations*, Technical Report No. **289**, Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, Wisconsin.

[6] Cornelius, H., Lohner, R. (1984). Computing the Range of Values of Real Functions with Accuracy Higher than Second Order, *Computing* **33**, 331-347.

[7] Davey, D.P., Stewart, N.F. (1976). Guaranteed Error Bounds for the Initial Value Problem using Polytope Arithmetic, *BIT* **16**, 257-268.

[8] Eijgenraam, P. (1981). *The Solution of Initial Value Problems using Interval Arithmetic*, Stichting Mathematisch Centrum Amsterdam, Mathematical Centre Tracts **144**.

[9] Fox, L., Valenca, M.R. (1980). Some Experiments with Interval Methods for Two-Point Boundary Value Problems in Ordinary Differential Equations, *BIT* **20**, 67-82.

[10] Jackson, L.W. (1975). Interval Arithmetic Error-Bounding Algorithms, *SIAM J. Numer. Anal.*, Vol. **12**, No.2, 223-238.

[11] Kaucher, E.W., Miranker, W.L. (1984). *Self-Validating Numerics for Function Space Problems*, Academic Press, New York.

[12] Keller, H.B. (1968). *Numerical Methods for Two Point Boundary Value Problems*, Blaisdell Publishing Company, Waltham, Massachusetts.

[13] Kulisch, U. (ed.), (1987). *PASCAL-SC, A Pascal Extension for Scientific Computation, Information Manual and Floppy Disks*, Version IBM PC/AT; operating system DOS, B.G. Teubner, Stuttgart.

[14] Kulisch, U. (ed.), (1987). *PASCAL-SC, A Pascal Extension for Scientific Computation, Information Manual and Floppy Disks*, Version ATARI ST, B.G. Teubner, Stuttgart.

[15] Kulisch, U., Miranker, W.L. (1981). *Computer Arithmetic in Theory and Practice*, Academic Press, New York.

[16] Kulisch, U., Miranker, W.L., (eds.), (1983). *A New Approach to Scientific Computation*, Academic Press, New York.

[17] Lohner, R., Adams, E. (1984). Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben, *ZAMM* **64**, T295-T297.

[18] Lohner, R. (1987). Enclosing the Solutions of Ordinary Initial and Boundary Value Problems, in: Kaucher et al. (eds.), *Computerarithmetic, Scientific Computation and Programming Languages*, 255-286 B.G. Teubner, Stuttgart.

[19] Lohner, R. (1988). *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Dissertation, Institute for Applied Mathematics, University of Karlsruhe.

[20] Moore, R.E. (1965). The Automatic Analysis and Control of Error in Digital Computing Based on the Use of Interval Numbers, in: L.B Rall (ed.), *Error in Digital Computation*, Vol. **1**, 61-130, John Wiley & Sons, New York, London, Sydney.

[21] Moore, R.E. (1966). *Interval Analysis*, Prentice-Hall, Englewood Cliffs, N.J.

[22] Moore, R.E. (1979). *Methods and Applications of Interval Analysis* SIAM, Philadelphia.

[23] Rall, L.B. (1981). *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science No. **120**, Springer, Berlin.

[24] Rump, S.M. (1980). *Kleine Fehlerschranken bei Matrixproblemen*, Dissertation, Institute for Applied Mathematics, University of Karlsruhe.

[25] Stewart, N.F. (1971). A Heuristic to Reduce the Wrapping Effect in the Numerical Solution of x'=f(t,x), *BIT* **11**, 328-337.

[26] Stroud, A.H. (1974). *Numerical Quadrature and Solution of Ordinary Differential Equations*, Springer, New York, Heidelberg, Berlin.

[27] Valenca, M.R. (1985). Multiple Shooting Using Interval Analysis, *BIT* **25**,425-427.