Rationale for Guaranteed ODE Defect Control

Robert Corless
George Corliss

CRPC-TR91250
December 1991

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Appears in *The Proceedings of SCAN '91: International Symposium on Computer Arithmetic and Scientific Computing* (Oldenburg, Germany; October 1-4, 1991), from *IMACS Manuals on Computing and Applied Mathematics,* ed.: J. Herzberger. Also available as *Argonne Technical Memorandum MCS-P273-1191,* from the Argonne National Laboratory, Mathematics and Computer Science Division.

Rationale for Guaranteed ODE Defect Control*

Robert M. Corless[e] and George F. Corliss[i]

[e]Dept. Applied Mathematics, University of Western Ontario, London, Ontario, CANADA

[i]Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL 60439-4801 USA. On leave from Department of Mathematics, Marquette University, Milwaukee, WI 53233 USA

Abstract

We introduce a modification of existing algorithms that allows easier analysis of numerical solutions of ordinary differential equations. We relax the requirement that the specified problem be solved, and instead solve a "nearby" problem exactly, in Wilkinson's tradition of backward error analysis. The precise meaning of "nearby" is left to the user. This inexpensive algorithm sublimates the well-known difficulties associated with the propagation of accumulated error and avoids the difficulty of exponential growth of inclusion widths associated with interval techniques. No claim is made for the accuracy with which the specified problem is solved. It is shown that often no such claim is necessary.

1 Problem Addressed

The idea of guaranteed defect control is applicable to large classes of operator equations. The concept of "defect," or "residual," was used by Krückeberg [10] in the solution of partial differential equations. We restrict our consideration in this paper to initial value problems in ordinary differential equations,

$$\frac{dx}{dt} = f(x,t), \quad x(t_0) = x_0, \tag{1}$$

where $x \in R^n$. We refer to Equation (1) as the *specified problem* [11]. We discuss a new error control strategy based on computing an interval enclosure of the defect for such problems.

Enright and others [7, 8, 9] have examined the idea of "defect control" as an error control strategy in this context and found it practical. They use point methods and *estimates* for the defect. We show here that it is practical and more satisfactory to use interval methods to *bound* the defect.

*In Proceedings of SCAN 1991: International Symposium on Computer Arithmetic and Scientific Computing (Oldenburg, October 1 – 4, 1991), Herzberger, J., (ed.), IMACS Annals on Computing and Applied Mathematics. Supported in part by the National Science Foundation grant No. CCR–8802429, by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, and through NSF Cooperative Agreement No. CCR-8809615.

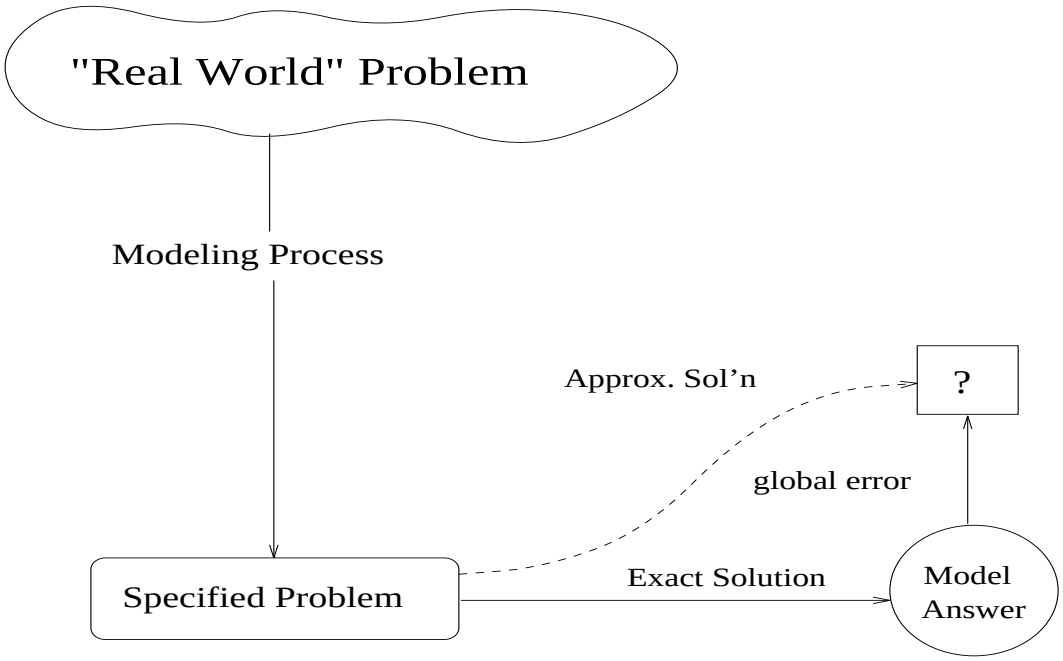

Figure 1: Modeling based on a specified problem.

We combine Enright's defect control idea with techniques for computing an enclosure for the range of a function. We outline the algorithm for using guaranteed enclosures of the defect to control the step-size selection for the numerical solution of ODEs. A more detailed description of the algorithm with numerical examples is in preparation [5]. Here, we concentrate on understanding what the "answer" we compute actually means. Specifically, we are concerned not with the accuracy of the solution computed, but rather with the validity of the model of the physical problem. The conventional view of modeling as the formulation and solution of the specified problem is depicted in Figure 1. The defect control approach is illustrated in Figure 2. It asks whether the nearby problem, for which the exact solution is known, is a sufficiently accurate model of the physical problem.

2 Defect

To define the defect for a system of ODEs $\dot{x} = f(x,t)$, we need a continuous representation of our computed solution $\hat{x}$. Since other applications also require a continuous representation of the solution, many current codes for the numerical solution of ODEs already provide one. We define the defect as

$$\delta(t) := \frac{d\hat{x}}{dt} - f(\hat{x},t). \tag{2}$$

Clearly, our computed solution $\hat{x}$ is an *exact* solution to the related problem

$$\begin{aligned} \frac{dx}{dt} &= f(x,t) + \delta(t) \\ &= f(x,t) + \varepsilon v(t), \end{aligned}$$

where $\varepsilon \geq ||\delta(t)||$ and $||v(t)|| \leq 1$. Later we shall use ε to mean the input tolerance specified by the user.

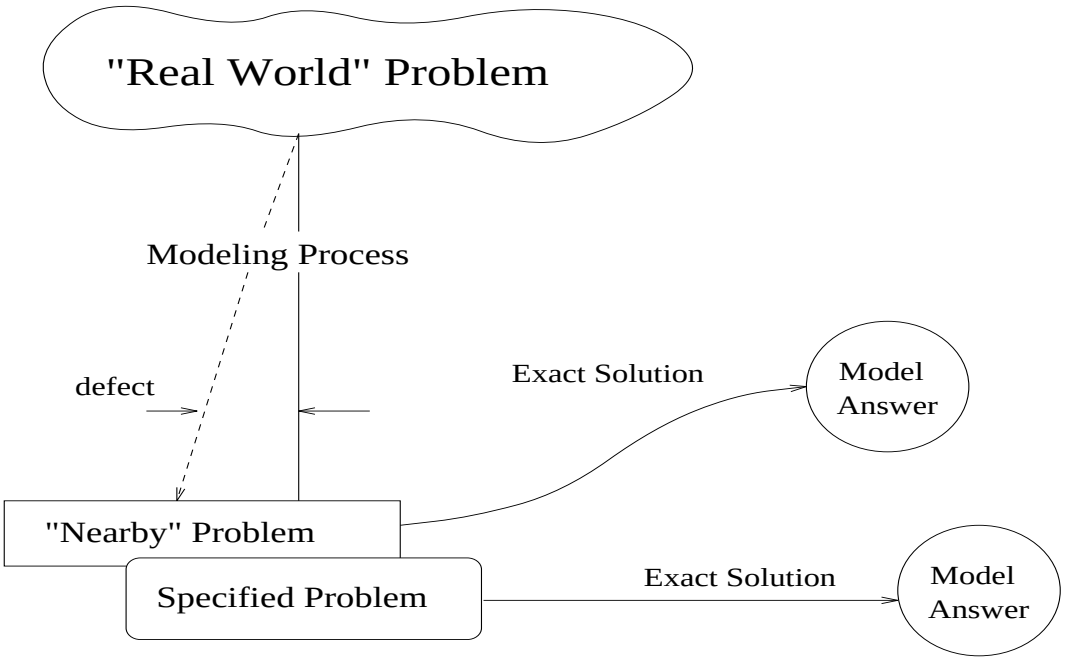

Figure 2: Modeling based a nearby problem.

For some problems, it is more appropriate to consider the relative defect defined by

$$\frac{d\hat{x}}{dt} = f(\hat{x}, t)(1 + \delta(t)). \tag{3}$$

In either case, the defect $\delta(t)$ is simply some function of t, and we have a formula for that function. To illustrate, we consider a simple logistic equation

$$\frac{dx}{dt} = x - x^2, \quad x(0) = 1/2.$$

Let $\hat{x}(t) := \frac{1}{2} + \frac{t}{4} - \frac{t^3}{96}$ be an approximate solution on $0 \le t \le h$. The approximate solution can be generated by a number of methods. Here, we introduced a deliberate error into a 4-term Taylor series. From Equation (2) (using Maple [3]),

$$\begin{aligned} \delta(t) &:= \frac{d\hat{x}}{dt} - f(\hat{x}, t) \\ &= \frac{1}{4} - \frac{t^2}{32} - \hat{x} + \hat{x}^2 \\ &= \frac{1}{32}t^2\left(1 - \frac{1}{6}t^2 + \frac{1}{288}t^4\right). \end{aligned}$$

The defect $\delta(t)$ is a polynomial in t. There is no remaining evidence of the ODE. The function $\hat{x}(t)$ is the *true solution* of the equation

$$\frac{dx}{dt} = f(x, t) + \frac{1}{32}t^2 - \frac{1}{192}t^4 + \frac{1}{9216}t^6. \tag{4}$$

If we prefer to use the relative defect for the logistic equation, we get instead (using Mathematica [18])

$$\delta(t) := \frac{\frac{d\hat{x}}{dt}}{f(\hat{x}, t)} - 1$$

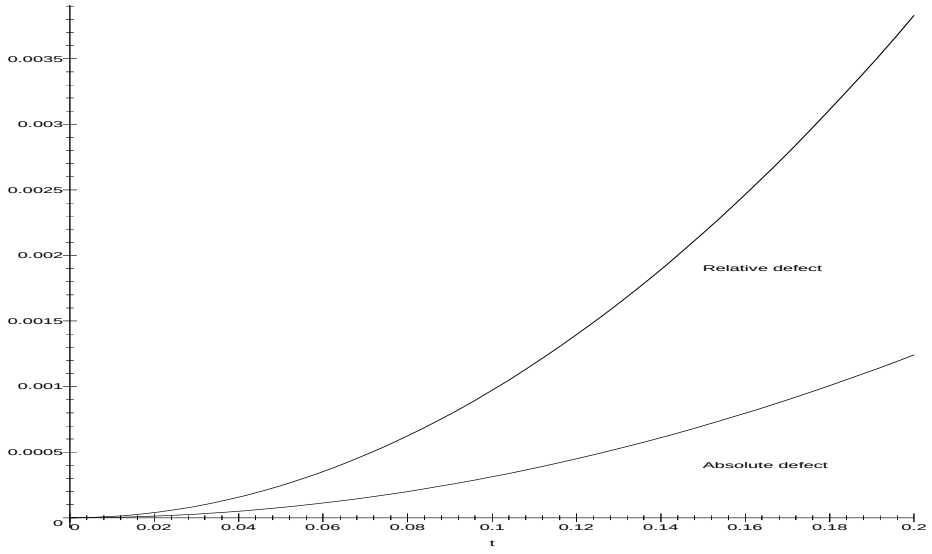

Figure 3: Absolute and relative defects for the logistic equation.

$$= \frac{t^2(t^2+48t-228)}{t^6-48t^4+576t^2-2304}.$$

With either the absolute or the relative definition of δ, we see in Figure 3 that $|\delta(t)|$ is small if t is small.

The key question to be asked is this: Is Equation (4) a sufficiently good model of the underlying scientific problem being studied? We provide insight into the answer by computing guaranteed bounds for $\delta(t)$. We can do this computation because the problem of bounding the range of a function is a very well studied problem in interval analysis (see [13] or [15]). The step-size control strategy comes from determining a step h for which we can guarantee that $||\delta(t)|| \leq \varepsilon$ for all $t \in [0,h]$, where $||\cdot||$ is some appropriate norm (usually L_∞).

The defect can often be interpreted physically, offering insight into the modeled problem. If $\delta(t)$ is small relative to the terms that were neglected in the derivation of the equations, or if it is small relative to uncertainty in parametric values, then one would expect that the equation that we have exactly solved is just as good a model as the specified problem. In this case, $\hat{x}(t)$ is just as good for practical purposes as the solution to the specified problem would have been (see Figure 2). Further, the modeler can choose the step size appropriately to control the size of $\delta(t)$ and to *guarantee* that no error larger than those already made in the modeling process will be introduced by the solution process.

Adding a small, time-dependent forcing term $\delta(t)$ to the logistic equation is reasonable in many physical contexts modeled by the logistic equation. For example, if the logistic equation is being used to model population growth of some species, then small, time-dependent perturbations of that population are realistic. The perturbations might be due to such factors as accidental deaths or to momentary fluctuations in the birth rate caused by small changes in the food supply. To simplify the solution process, one usually ignores such fluctuations. In contrast in the defect-controlled method, it is the difference between the small physical perturbations and the small numerical perturbation that is ignored.

The idea of considering the defect is related to Wilkinson's idea of backward error analysis for linear systems [17]. It is in sharp contrast to the usual approach in interval mathematics of considering the accuracy of the solution computed for the specified problem. More details of the history of the study of the defect in the context of differential equations can be found in [4].

As noted by Enright [8], a major practical advantage of the defect-controlled approach is a separation of the concepts of any numerical instability resulting from the approximation method used and any ill-conditioning of the problem itself. If the problem is well conditioned *and* the defect is small, then $\hat{x}$ commits a small global error. However, the global error for an ill-conditioned problem may be expected to be large, even for small defects. Clearly, the model is very sensitive to the modeling errors made in deriving the specified problem, and a small global error usually is not a reasonable goal. Nevertheless, a small defect is achievable and gives much insight into the physical problem being modeled.

Chaotic systems give rise to unstable initial value problems, by definition. On the other hand, achievement of a small global error over long time integration is computationally intractable (see [1], for example). Achievement of a small defect is both possible and useful for such systems [4]. The defect-controlled approach sidesteps the bothersome question of computational chaos.

3 Defect-Controlled Algorithm

An outline of the defect-controlled algorithm is given in Listing 1. A more complete description of the algorithm is given in [5].

```
Input: t_0, t_final, x_0, ε = max ||δ(t)||
Output: Nodes t_0, t_1, ..., t_n = t_final,
        Continuous x̂ which solves dx/dt = f(x,t) + δ(t),
        Guarantee that ||δ(t)|| ≤ ε for all t ∈ [t_0, t_final].

h := Initial trial step;
t := t_0;
loop for each step k = 0, ...
    Compute x̂(t), a continuous approximate solution on t_k ≤ t ≤ t_k + h;
    Define the defect δ(t) := dx̂/dt − f(x̂,t);
    Δ := Enclosure of ||δ(t)||;          -- Only interval part.
    if Δ > ε then
        reduce h and repeat
    else if Δ << ε then
        increase h and repeat
    else
        accept step;
        t := t + h;
    end if;
end loop;
```

Listing 1. Defect-controlled algorithm

The outline of this algorithm is essentially like the outline of any modern ODE solver. The defect control functions is a part of the step-size selection strategy. In our implementation,

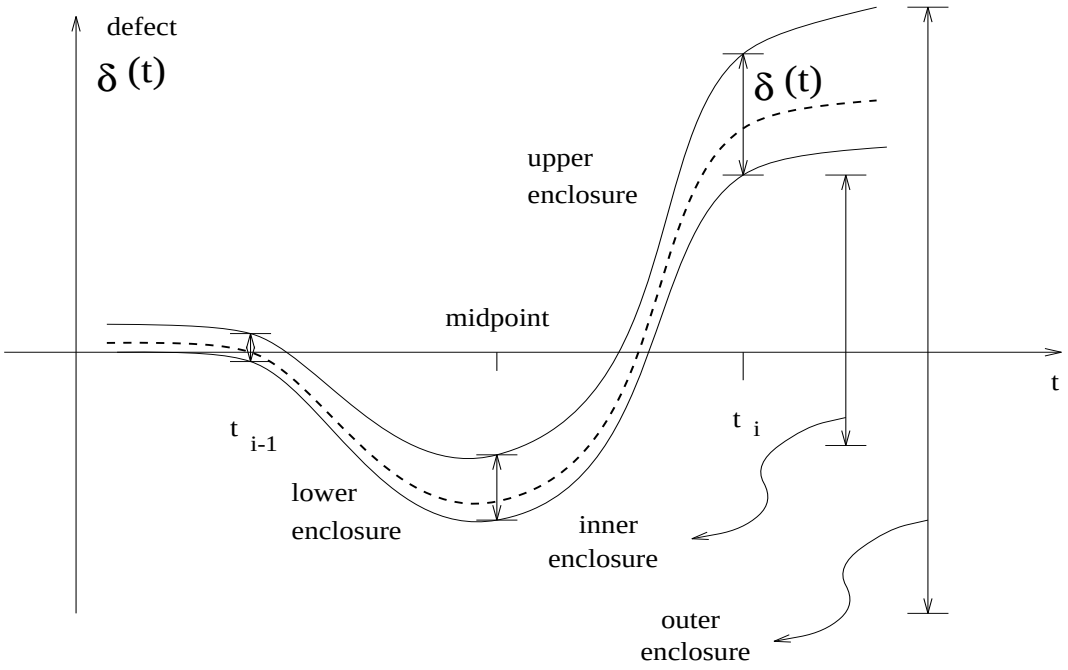

Figure 4: Inner and outer enclosures of the defect.

interval computations are restricted to computing an enclosure of $||\delta(t)||$. We use interval Taylor operators implemented in Ada [6]. These operators achieve tight bounds on the range of the defect and its derivatives by using natural interval extensions, monotonicity, concavity, mean value forms, centered forms, and Taylor forms. In the logistic equation example presented earlier, we use interval Taylor arithmetic to (in effect) evaluate the assignment statements

```
T := Taylor (0);  -- Taylor series for t at 0 = (0, 1, 0, ...)
XHAT := (1/2) + T * ((1/4) + T * (0 + T * (-1/96)));
                                                  -- Horner form
DEFECT := (1/4) + T * (0 + T * (-32)) - XHAT + XHAT * XHAT;
```

As part of the computation of a tight enclosure for $||\delta(t)||$, we evaluate $\delta(t)$ at each end, at the midpoint, and on the entire interval of the integration step. This approach allows us to compute both inner and outer enclosures to help verify the tightness of the enclosures (see Figure 4). Our operators allow integration steps 10^3 greater than naive interval arithmetic evaluation of $||\delta(t)||$.

In Figure 4, the dotted line represents the true defect $\delta(t)$. The defect-controlled algorithm in Listing 1 computes Δ, an enclosure of $\delta(t)$ in the interval $[t_{i-1}, t_i]$. In Figure 4, lower and upper enclosures of $\delta(t)$ are represented as curves. The outer enclosure Δ must contain the lower and upper enclosures at all points in the interval, but it may include some overestimation. The inner enclosure is an interval which must be interior to the range of $\delta(t)$. The set difference between the outer and the inner enclosures is an indication of the tightness with which the range of $\delta(t)$ has been enclosed.

4 Conditioning

If we compare our algorithm to other defect-controlled algorithms (see Enright [7, 8]), we see that our approach provides a *guaranteed* bound on the range of the defect, while conventional

approaches *estimate* the range by evaluating it at the final point ($\delta(t+h)$), at an intermediate point ($\delta(t+\theta h)$ with $0 < \theta < 1$), or at a sample of intermediate points ($\delta(t+\theta_i h)$ with $0 < \theta_i < 1$). By providing a guaranteed bound, we can be assured that the problem we have solved is indeed close enough to the specified problem to be of interest.

If we compare our algorithm to Lohner's interval method for solving ODEs [12], we see that our approach encloses the defect, whereas Lohner encloses the solution. With our approach, $\hat{x}$ is the exact solution to a problem whose distance from the specified problem is guaranteed to be small. Lohner [12] computes an interval that is guaranteed to enclose the exact solution to the specified problem. These are complementary approaches; each has its own domain of applicability.

Our guaranteed control of the defect and Lohner's guaranteed enclosure of the solution are connected by the condition number of the differential equation. The concept of the condition number of a differential equation is the same as the better-known concept of a condition number of a system of linear equations. The condition number is a number C for which one can make statements of the form

$$|| \text{ Error in the answer } || \leq C \cdot || \text{ Error in the problem } ||.$$

Suppose that $\hat{x}(t)$ is the exact solution to Equation (3) and that $x(t)$ is the exact solution to Equation (1). Then we have

$$x(t) = \hat{x}(t) - \varepsilon x_1(t) + O(\varepsilon^2),$$

where x_1 satisfies the first variational equation

$$\frac{dx_1}{dt} = J_f(\hat{x}(t))x_1(t) + v(t), \tag{5}$$

which has the solution

$$x_1(t) = \Psi(t)x_1(0) + \int_0^t \Psi(t) \cdot \Psi^{-1}(\tau)v(\tau)\, d\tau,$$

where $\Psi(t)$ is a fundamental solution matrix of the homogeneous version of Equation (5). Let $x_1(0) = 0$ for simplicity. Define the condition number of the differential equation to be

$$C := \int_0^t ||\Psi(t) \cdot \Psi^{-1}(\tau)||\, d\tau.$$

This condition number depends on t, while the condition number defined in [2] is the maximum of our condition number taken over the relevant domain of t. With our definition (recall that $\varepsilon = ||\delta(t)||$),

$$||x - \hat{x}|| \leq C \cdot ||\delta||, \text{ in the limit as } \varepsilon \to 0.$$

We can replace the above with a bound valid for all values of ε by starting instead with the Alexeev-Gröbner nonlinear variation of constants formula [14].

One often hesitates to use the condition number to compute global error bounds because it is hard to compute or bound C exactly, and sometimes the quantity $C||\delta||$ is overly pessimistic. In contrast, one may choose to use the condition number because it may not be too difficult to estimate C, and an estimate of C is useful in the modeling context.

5 Conclusions

For stable problems (perhaps containing interval coefficients), solution enclosures may work better than a defect-controlled approach. Similarly for Hamiltonian systems, fixed time-step, symplectic methods appear to be superior [16].

For a wide range of problems, however, the $\delta(t)$ term introduced by numerical methods can be viewed as one more in a sequence of reasonable simplifications made in the quest for an exact solution. In particular, defect-controlled methods appear to be appropriate for chaotic problems, for they avoid the difficulty of exponential growth of the error they are monitoring, namely the norm of the defect. Defect-controlled methods can yield useful results, even for chaotic problems, at a reasonable cost.

References

[1] E. Adams, *Periodic solutions: Enclosure, verification, and applications*, in Computer Arithmetic and Self-Validating Numerical Methods, C. Ullrich, ed., Notes and Reports in Mathematics in Science and Engineering, Vol. 7, Academic Press, Boston, 1990, pp. 199–246.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, N. J., 1988.

[3] B. W. Char, K. O. Geddes, G. H. Gonnet, M. B. Monagan, and S. M. Watt, *MAPLE Reference Manual*, Watcom Publications, Waterloo, Ontario, Canada, 1988.

[4] R. M. Corless, *Defect controlled numerical methods and shadowing for chaotic differential equations* (to appear).

[5] R. M. Corless and G. F. Corliss, private communication.

[6] G. F. Corliss and L. B. Rall, *Computing the range of derivatives*, IMACS Annals on Computing and Applied Mathematics, (to appear).

[7] W. H. Enright, *Analysis of error control strategies for continuous Runge-Kutta methods*, SIAM J. Numerical Analysis, 26 (1989), pp. 588–599.

[8] W. H. Enright, *A new error-control for initial value solvers*, Applied Mathematics and Computation, 31 (1989), pp. 288–301.

[9] P. M. Hanson and W. H. Enright, *Controlling the defect in existing variable-order Adams codes for initial-value problems*, ACM Trans. Math. Software, 9 (1983), pp. 71–97.

[10] F. Krückeberg, *Partial differential equations*, in Topics in Interval Analysis, E. Hansen, ed., Clarendon Press, Oxford, 1968, pp. 98–101.

[11] U. Kulisch and H. J. Stetter, *Automatic result verification*, in Scientific Computation with Automatic Result Verification, U. Kulisch and H. J. Stetter, eds., Computing Supplementum 6, Springer, Vienna, 1988, pp. 1–6.

[12] R. J. Lohner, *Enclosing the solutions of ordinary initial and boundary value problems*, in Computer Arithmetic: Scientific Computation and Programming Languages, E. W. Kaucher, U. W. Kulisch, and C. Ullrich, eds., Wiley-Teubner Series in Computer Science, Stuttgart, 1987, pp. 255–286.

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

[14] S. P. Norsett and G. Wanner, *Perturbed collocation and Runge-Kutta methods*, Numer. Math., 38 (1981), pp. 193–208.

[15] H. Ratschek and J. Rokne, *Computer Methods for the Range of Functions*, Series in Math. Appl., Ellis Horwood, Chichester, 1984.

[16] J. M. Sanz-Serna, *Recent results on symplectic Runge-Kutta and related methods,* in Proceedings of the 11th CNLS Conference, (to appear).

[17] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.

[18] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, Reading, Mass., 1988.