Arbitrary Accuracy with Variable Precision Arithmetic

Fritz Krückeberg

Gesellschaft für Mathematik und Datenverarbeitung mbh (GMD)

Summary

For the calculation of interval-solutions Y including the true solution y of a given problem we need not only that $y \epsilon Y$ holds. Furthermore we are interested in the value of $span(Y)$. So we should get that for an a priori and arbitrarily given bound $\varepsilon > 0$ the calculation yields that the error remains below ε or that $span(Y) < \varepsilon$. It is possible to realize $span(Y) < \varepsilon$ for arbitrary $\varepsilon > 0$ by using an interval-arithmetic with variable word length within a three-layered methodology, including validation/verification of the solution. The three-layered methodology consists of

- Computer algebra procedures,
- the numerical algorithm,
- an interval arithmetic with variable and controllable word length.

Examples are given in the domain of linear equations and ordinary differential equations (initial value problems).

1. General Aspects

For the calculation of interval-solutions Y including the true solution y of a given problem we need not only that

$$y \epsilon Y$$

holds. Furthermore we are interested in the value of $span(Y)$. So we should get that for an a priori and arbitrarily given bound $\varepsilon > 0$ the calculation yields that the error remains below ε or that

$$span(Y) < \varepsilon. \tag{1}$$

It is possible to realize (1) for arbitrary $\varepsilon > 0$ by using a combination of some methods that are described in this paper.
If we realize (1), a next level of Interval Mathematics is reached: in addition to the inclusion relation $y \epsilon Y$ the 'quality' ε can be fulfilled in a way, that the result is better (but not too much better) than ε. This means that the amount of computation can be restricted in dependency on the quality ε which is wanted. But how such flexibility may be reached?

2. Interval-arithemetic with variable word length

The condition (1) needs flexibility in two directions: to be accurate enough if necessary and to be rough enough if possible. By introducing an interval-arithmetic with variable word length this flexibility can be arrived.

For this reason we have defined a special interval-arithmetic with variable word length on a decimal base $b = 10^4$ with a fixed point architecture. Principally it would be possible to use a floating point arithmetic with variable mantissa length. The restriction to a fixed point arithmetic with variable word length is not a principal restriction.
The used interval-arithmetic was realized by FORTRAN subroutines. These subroutines leads to the four basic operations $+, -, *, /$. The result of each operation leads to a new fixed point expression, but with a longer word length if needed (to save all digits of the result). If the word length is growing more than needed an interval rounding can be executed to an arbitrary number of digits (depending on the precision needed).

For these operations some additional parameters are stored at the end of the fixed point word. The parameters contain the number of leading digits on the left side of the decimal point and the number of digits needed after the decimal point. Figure 1 explains the architecture of such a fixed point word:

	...	c_2	c_1	c_0	.	c_{-1}	c_{-2}	c_{-3}	c_{-4}	...	m	$\pm n$

Figure 1

c_i are digits with $-(10^4-1) \leq c \leq +(10^4-1)$,
each digit must be interpreted as an integer (with a sign ±).

$m =$ the number of significant digits after the decimal point,

$\pm n =$ the number of significant digits on the left side of the decimal point
(if $n < 0$, n denotes the number of zeros after the decimal point).

The minimal value z of such a fixed point word is then defined by

$$z = \sum_{i=-m}^{n-1} c_i \, 10^{4i}$$

To restrict the amount of computation the values of m and n are limited by

$$\begin{array}{rcccl} 0 & \leq & m & \leq & 20 \\ -19 & \leq & n & \leq & 20 \end{array}$$

but the limits are not fixed, they can be varied without changing the subroutines.

The interval–arithmetic with variable word length uses a pair of such words (with variable parameters m and n). In consistency with this variable base subroutines for a set of elementary functions are written (sine, cosine, natural logarithm, exponential function). For example it is possible to calculate

$$Y := sin\, Z \qquad (Y, Z \; are \; intervals) \tag{2}$$

in such a way that

$$span(Y) < \varepsilon$$

for an arbitrarily given $\varepsilon > 0$ if $span(Z)$ is small enough to realize (2). The amount of computation time to calculate such 'variable' subroutines as sine depends on the value of ε. If ε is smaller the calculation time is longer. It is not easy to write such subroutines but they are needed to get the flexibility that is wanted.

3. The 3 Levels of computations

By introducing an interval–arithmetic with variable word length only the lowest level I of a concept of (see Figure 2) computation is realized. We have to control the variable word length in a dynamic way during the computation time by the numerical methods that are used. So the numerical methods should contain control parameters for the control of word length. A control parameter of the given method may be for example the step size h of a numerical integration method. If the step size h is smaller the word length should be longer. The numerical methods are placed at the level II of our 3–level diagram. Computer–algebra–systems belongs to the level III. The interconnection of all three levels is described in the following diagram (Figure 2):

Now it is possible to realize (1) for given problems belonging to some problem classes: Computer–algebra–systems are used for formal computation to avoid rounding errors as long as possible or to find out well conditioned domains for the application of numerical methods at level II. Between the numerical methods and the basic operations at level I exist a control feedback to determine the word length (dynamically) or to determine the number of iterations (or other parameters) at level II. In a similiar way there will be a control feedback between level II and level I (the number of higher order derivatives of a given function at level I may be connected with the number of iterations that are needed at level II).
By using such a 3–level–control structure several examples for different problem classes are calculated by the author and some cooperating persons [Demmel, Leisen, Pascoletti].

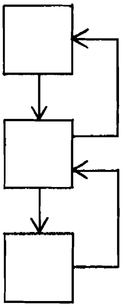
THE 3 LEVELS OF WORK	PROCEDURES	PROCESS-CONTROL	
LEVEL OF SYMBOLIC COMPUTATION	COMPUTER-ALGEBRA-SYSTEMS AND REDUMA		VERIFICATION OF RESULTS
LEVEL OF NUMERICAL METHODS	NUMERICAL METHODS		
LEVEL OF ARITHMETIC OPERATIONS	DYNAMIC INTERVAL ARITHMETIC ARITHMETIC WITH VARIABLE WORD LENGTH		

Figure 2

4. Examples

4.1 Initial value problems for ordinary differential equations

The initial value problem

$$y' = f(x,y), \qquad x \in [\underline{a}, \overline{a}] \in II(I\!R), \quad y \in I\!R^n$$

$$y(\underline{a}) = s, \qquad s \in I\!R^n$$

is to be solved for all values x with

$$\underline{a} \leq x \leq \overline{a}$$

and the true solution

$$y(x)$$

should be included in a stepwise interval polynomial $Y(x)$ so that

$$y(x) \in Y(x) \qquad \textit{for all } x \textit{ with } \quad \underline{a} \leq x \leq \overline{a} \tag{3}$$

and the condition

$$span(Y(x)) < \varepsilon \tag{4}$$

holds for an arbitrarily a priori given $\varepsilon > 0$.
We now give a general description of our method [see Leisen, Krückeberg]. To construct a solution for this problem the functions $f(x,y)$ are to be described in the form of a formal expression similar to expressions that are used in Computer–algebra. Then it is possible to calculate the derivatives of the functions $f(x,y)$ in a formal way by recursive methods that are typical for Computer–algebra routines.

The formal calculation of derivatives is a typical process within level III of Figure 2. The stepwise integration starts with

$$x_0 = \underline{a}$$

and leads to a sequence of points

$$x_0,\ x_1,\ x_2,\ \ldots$$

with a variable step size h_i. If it is possible to include the solution $y(x_{i-1})$ in an interval vector$[u_{i-1}]$ then we have to realize the integration from x_{i-1} to x_i so that the calculated interval vektor $[u_i]$ includes $y(x_i)$. The integration step from x_{i-1} to x_i is shown in Figure 3.

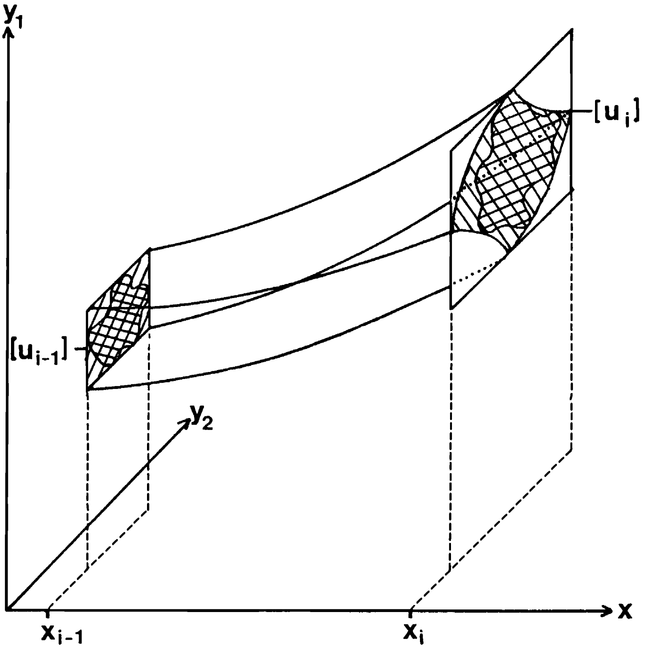


Figure 3

To held condition (4) it is necessary to control the step size h. But h is only one parameter to contol the precision. Annother parameter is the degree s of the highest derivative of $f(x,y)$ that is used in the taylor evaluation for the integration steps. But the parameters h and s are not sufficient to fulfill condition (4). We also need a variable word length (that means we have to control m). If we use a control procedure for all three parameters

h step size

s degree of derivation

m word length of interval–arithmetic

then it is possible to fulfill condition (4). Many examples are calculated [Krückeberg, Leisen] with good results.

Examples

For a linear system of differential equations

$$\begin{aligned} y_1' &= \tfrac{3}{2}y_1 - y_2 - \tfrac{1}{2}y_3, & y_1(0) &= 2 \\ y_2' &= -\tfrac{1}{2}y_1 + 2y_2 + \tfrac{1}{2}y_3, & y_2(0) &= -6 \\ y_3' &= \tfrac{1}{2}y_1 + y_2 + \tfrac{5}{2}y_3, & y_3(0) &= 0 \end{aligned}$$

and two different conditions (4) with

$$\varepsilon_1 = 10^{-50}$$

and

$$\varepsilon_2 = 10^{-70}$$

the integration was executed. Figure 4 shows the results. For the $\varepsilon - axis$ a logarithmical scale is used.

$$|\tilde{y} - y|$$

denotes the deviation from the true solution.

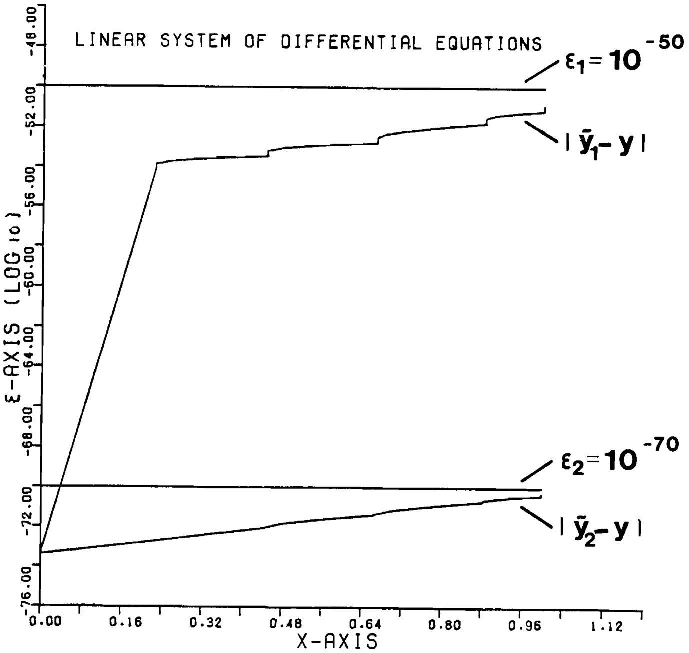


Figure 4

Figure 4 shows the possibility to control the integration in such a way that condition (4) is fulfilled. The values of $\varepsilon_1, \varepsilon_2$ for practical use are extremely small, but it should be explained that variable word length arithmetic is powerful enough to fulfill such conditions. The method was also tested for several nonlinear systems of differential equations. Further work is to be done for other problems in the domain of differential equations, specially for boundary value problems.

4.2 Systems of linear equations

If a system of linear equations

$$Ax = b \tag{5}$$

is given then we ask for an interval vector X so that

$$x \epsilon X$$

and

$$span(X) < \varepsilon \tag{6}$$

is fulfilled for an arbitrarily given $\varepsilon > 0$. To solve this problem we have constructed a procedure that includes a control process for the word length. By [Demmel, Krückeberg] such a procedure was constructed and tested.

In some cases only integer solutions of (5) are of interest. Then it is sufficient to take

$$\varepsilon = 0.5$$

and to control the word length in such a way that the result is not much better than ε. The constructed algorithm uses only as much precision as needed to achieve (6).

Examples

To test our algorithm we tried to invert Hilbert matrices scaled to have integer entries. For Hilbert matrices of the order 10, 11, 12, 13 it was easy to calculate an interval vector solution X with

$$span(X) < 10^{-9}$$

in such a way that the algorithm adjust its own parameters to minimize computation time. It is obviously that the needed word length (controlled by the algorithm) depends on the condition of the matrix.
In the case of linear equation only level II and level I of Figure 2 are used. But sometimes it may be of interest to perform some formal transformations (level III) before the numerical algorithm is started.

5. Consequences

It seems to be realistic to follow the general aspects of chapter 1 and to fulfill condition (1). Then a next level of Interval Mathematics is reached and an economic principle is introduced: to minimize computation time in relation to the given $\varepsilon > 0$. The including of Computer–algebra (level III of Figure 2) seems to be necessary in many cases of analytic problems to have a better chance for controlling the numerical algorithm at level II.
Hopefully several new methods will be constructed to solve problems of applied mathematics (and of Interval Mathematics) at this level of Interval Mathematics.
The new methods will contain several parameters at the level I, II and III. So it may be problematic to realize an automatic control of all parameters by feedback loops between the levels I, II and III, so that condition (1) is fulfilled. For this reason some additional feedback is needed between the user and the computer: expert system are expected to be useful to support such a feedback between the user and the computation process.

REFERENCES

Demmel J W, Krückeberg F (1985) An Interval Algorithm for Solving of Linear Equations to Prespecified Accuracy. Computing 34: 117–129.

Krier N, Spelluci P (1975) Untersuchungen der Grenzgenauigkeit von Algorithmen zur Auflösung linearer Gleichungssysteme mit Fehlererfassung. In: Interval Mathematics, Lecture Notes in Computer Science 29: 288–297 ed. by Nickel K, Springer Verlag, Berlin Heidelberg, New York.

Krückeberg F, Leisen R (1985) Solving Initial Value Problems of Ordinary Differential Equations to Arbitrary Accuracy with Variable Precision Arithmetic. In: Proceedings of the 11th IMACS World Congress on System Simulation and Scientific Computation: 111–114 ed. by Wahlstrom B, Henriksen R, Sundby N P, Oslo, Norway.

Leisen R (1985) Zur Erzielung variabel vorgebbarer Fehlereinschließungen für gewöhnliche Differentialgleichungen mit Anfangswertmengen mittels dynamisch steuerbarer Arithmetik. Diplomarbeit, Universität Bonn.

Pascoletti K H (1982) Ein intervallanalytisches Iterationsverfahren zur Lösung von linearen Gleichungssystemen mit mehrparametriger Verfahrenssteuerung. Diplomarbeit, Universität Bonn.

Prof. Dr. Fritz Krückeberg
Gesellschaft für Mathematik und Datenverarbeitung mbh (GMD)
Postfach 1240, Schloß Birlinghoven, D–5205 St. Augustin