Appl. Math. and Comp. Sci., 1994, vol.4, No.4, 661–680

AUTOMATIC DETECTION AND CORRECTION OF COMPUTATIONAL ERRORS IN PROGRAMS

P.A.D. DE MAINE*, M.M. DE MAINE*

Two different kinds of methods that are frequently used to evaluate functions are: Regression (for single-valued parameters) and Iteration (for multi-valued parameters). Comprehensive tests have established that both regression and iteration methods can yield unexpected false answers because their criteria are necessary but not sufficient conditions. Working algorithms that automatically detect and correct computational errors are described. For regression methods, in the Maximum Tolerance (MAXTOL) algorithm, the traditional question "Do these data describe this equation?" is replaced by "Do these data describe this equation within user stipulated limits of reliability for the raw data?" For iteration methods absolute measures for computational accuracy, determined from a general form of the Law of Conservation of Mass and Energy, have led to the development of the fully Automatic Error Detection and Corrective Action (EDCA) algorithm.

1. Introduction

Key problems associated with numerical evaluations of equations are the determination of computational accuracy and the correction of computational errors. The objective of this paper is to describe and demonstrate two algorithms that determine computational accuracy, use user-supplied "limits of reliability" or "desired computational accuracy" to detect errors and then automatically correct them.

Two different kinds of methods that are used to evaluate equations are:

(i) Regression methods in which only single-valued parameters are calculated, generally from values for multi-valued parameters by graphical or statistical curve-fitting methods.

(ii) Iteration methods in which values for multi-valued parameters are generated from boundary conditions. Typical examples are the use of the well known procedures by Hamming-Kutta-Runge (for evaluating systems of differential equations) and Newton-Raphson (for evaluating systems of nonlinear equations), and the use of iterative procedures to solve nonlinear equations for both single-valued and multi-valued parameters.

* Computer Science & Engineering Department, Auburn University, Auburn AL 36849, USA

1.1. Regression Methods

Regression methods have recently been considered in (de Maine and de Maine, 1992a). With conventional procedures the answers are ultimately computed from the variances. They include measures for the "goodness of fit", calculated values for single-valued parameters and their reliabilities. Extensive experiments using machine generated data with controlled variances (de Maine, 1978a) have established that:

(a) The three different kinds of variance (normal statistical fluctuations, wrong values, and unsuspected curvature) can affect the "goodness of fit" criteria in different and unpredictable ways (de Maine, 1978b; de Maine *et al.*, 1978).

(b) Different transforms of the fitting equation yield different results (de Maine, 1965) (e.g. improve a fit by plotting Y versus X instead of Y/X versus $1/X$). Thus virtually any desired result can be confirmed by selecting an appropriate transformation.

(c) Interactive graphical methods based on variants of the method of residuals (Thisted, 1988) appear to ignore Sillen's results (Sillen, 1962) and can therefore yield incorrect answers. Sillen's results, which demonstrated that graphical and statistical curve-fitting methods yield virtually identical answers, have been independently confirmed in our laboratory (de Maine, 1965; de Maine and Seawright, 1963a).

The experiments just cited also established that the ambiguity inherent in conventional graphical and statistical curve-fitting methods occurs because the "goodness of fit" criteria are essentially average measures that are not invariant to transformations of the trial equation. Such transformations can affect the data and its variances in opposing ways.

1.2. Iteration Methods

Methods for estimating absolute computational accuracies have not been reported. Relative computational accuracies may be determined by the following two types of method that use relative measures like the rate of convergence.

(1) In convergence methods for calculating multi-valued parameters results are deemed acceptable if their values are not significantly altered in successive iterations.

(2) Sensitivity methods, normally used in iterative procedures that compute values for both single-valued and multi-valued parameters, are also used in classical iteration procedures to calculate values for multi-valued parameters. Calculated values for single-valued parameters are accepted if changes in their initial values do not significantly affect the final calculated values for any of them. Results are also accepted if small changes in the calculated value(s) for the single-valued parameter(s) do not significantly alter the calculated value(s) for any multi-valued parameter.

While both types of methods can detect chaotic or unstable conditions like those that occur at or near points of discontinuity, neither can by themselves (i) instigate or recommend corrective actions, or (ii) correctly determine the absolute computational

accuracy for any calculated parameter. That occurs because the relative measures used are necessary, but not sufficient, and therefore spurious results are often reported (de Maine and de Maine, 1992b). Sensitivity methods are particularly prone to yield incorrect answers because the effect of even small changes in the distribution or values of variances are not understood. An example is the result reported for the iterative solution of a simple system of nonlinear equations that is in error by more than 1000% (de Maine *et al.*, 1957). That particular system of equations has three possible foci (correct, false and oscillating), determined by the actual distribution of variances (de Maine and Seawright, 1963b). The reported result is a false focus.

The key problem of determining absolute computational accuracies has been solved with the realization that a general form of the physical law of conservation of mass and energy applies to mathematics. The result is the Error Detection and Corrective Action, EDCA, algorithm that employs a user-stipulated parameter to detect computational errors, automatically instigates corrective actions, and reports the absolute computational accuracy for every calculated value.

2. Maximum Tolerance Method (Regression Analysis)

Terms associated with this method are:

The values for Observables are the data that will be used to compute values for the variables in the trial equation.

The Maximum Tolerances are the user-stipulated maximum uncertainties for every value of every observable.

The Error Bounds are the maximum uncertainties for every value for every variable. They are calculated from the observables and their maximum tolerances.

A Domain contains the values for the variables that were used to successfully solve the trial equation. If there is unsuspected curvature the data are partitioned into overlapping domains that are used separately to solve the trial equation.

A Rejected Data-Point is one whose deviation exceeds its error bound. Illegal rejects result from unsuspected curvature. They lead to the partitioning of the data and their eventual acceptance in one or more of the overlapping domains. Legal rejects are not accepted in any domain.

The Maximum Error is the maximum possible error in the calculated value for a parameter in a domain. It is computed by deforming the accepted data with their calculated deviations.

The maximum tolerance method is an asymptotic procedure that uses the error bounds to reject unacceptable data and to detect unsuspected curvature, which leads to partitioning into overlapping domains and no illegally rejected data-points.

2.1. Calculation of Error Bounds

Values for the variables and their error bounds are mechanically computed from the values for observables, O_i, and their maximum tolerances, $dO_i > 0$. For illustrative

purposes suppose that four observables define the variables w, x, y and z thus: $w = \log(O_1 - O_2)$; $x = O_4$; $y = O_1/O_3$; and $z = O_4 - O_1$.

There are two different methods for computing error bounds (designated EB_x). In both methods the signs for the maximum tolerances are selected to yield the maximum values for the error bounds. In the *Tolerance* method:

$$EB_w = \log(O_1 + dO_1 - O_2 + dO_2) - \log(O_1 - O_2)$$

$$EB_x = dO_4$$

$$EB_y = (O_1 + dO_1)/(O_3 - dO_3) - O_1/O_3$$

$$EB_z = dO_4 + dO_1$$

The *Derivative* method, which can only be used for very small maximum tolerances, yields:

$$EB_w = (dO_1 + dO_2)/(O_1 - O_2)$$

$$EB_x = dO_4$$

$$EB_y = dO_1/O_3 + O_1 dO_3/O_3^2$$

$$EB_z = dO_4 + dO_1$$

2.2. Overview of the Maximum Tolerance, MAXTOL, Method

The MAXTOL method (de Maine and de Maine, 1992a), also called the CURFIT method (de Maine, 1978b; de Maine *et al.*, 1978), uses the following three algorithms:

(a) A central feature of the complex Self-Judgement Principle (SJP) (de Maine and Seawright, 1963b) are twelve automatic or semi-automatic mechanisms for reducing bias and eliminating erroneous data. The SJP transposes the data points (to eliminate bias), then uses the conventional method of least squares to compute the median curve and the error bounds to determine which data points are to be rejected. Rejected data points are those with a deviation greater than the corresponding error bound.

(b) The Discard Rules Procedure, DRP, is best described as an environmental analysis of each rejected data point and its three different closest neighbors to determine if the rejection is legal. A legal rejection is due to normal random statistical fluctuations.

(c) The Error Expansion Procedure, EEP, is used as a last resort to increase the error bounds in those cases where the maximum tolerances are too small.

In the Maximum Tolerance method the data points that are computed from the values for the observables are ordered with the first independent variable increasing and then they are arbitrarily partitioned into a maximum of ten consecutive sectors, each with a minimum of eight distinctly different data points. At this point the

domain for the trial equation contains all the data points that were not eliminated by the SJP. The following steps are executed.

STEP 1. The SJP is used to determine which data points in the designated domain are rejected. The median curve is computed by the conventional method of least-squares and then the error bounds are used to determine which data points are rejected.

STEP 2. The rejected data points are examined by the DRP to determine if there are illegal rejects (i.e. those due to unsuspected curvature). If there are no illegal rejects go to Step 3. Otherwise Step 4 is executed.

STEP 3. Use the calculated deviations of all accepted data points to compute the maximum error in the median value for every parameter. The maximum errors are computed by deforming the deviations for accepted values for each variable in turn to the worst configuration. If all data points have been processed go to Step 6. Otherwise redefine the domain and execute Step 1.

STEP 4. In this step there is at least one illegally rejected data point. If the domain cannot be decreased (i.e. there is only one sector) go to Step 5. If there is more than one sector, the domain is redefined by omitting either a half, if an end sector is to be omitted, or a full sector from the left or right. This assures that the domains will overlap by at least half a sector. Go to Step 1.

STEP 5. For domains with one sector the EEP, which increases the error bounds, is invoked, and Step 1 is executed.

STEP 6. The regression analysis has been completed and the maximum tolerance procedure is terminated.

2.3. Criteria of Fit

The explicit "goodness of fit" criteria in their order of significance are:

(A) The number of domains and the range of every variable in every domain.

(B) The calculated value for every parameter and its associated maximum error in every domain.

(C) The number of rejected data points and the maximum tolerances for every value for every observable.

A perfect fit occurs for the specified ranges only if there is one domain, all maximum errors are vanishingly small, and there are no rejected data points.

The advantages offered by the maximum tolerance method include:

(1) The ambiguity and uncertainty that are an inherent part of the conventional curve-fitting methods are avoided.

(2) Fits of different sets of data to the same equation or the same set of data to different equations with the same or different dimensions, can be directly and unambiguously compared.

3. Error Detection and Corrective Action, EDCA, Algorithm

In conventional iteration procedures well known relative mathematical measures such as the rate of convergence and division by zero or very small numbers are used to detect instability. If instability is detected, alternative procedures must be used. The problems associated with this approach for non trivial systems of equations are:

(i) Conventional mathematical tests are necessary but not sufficient. While terminal failures (like division by zero) do establish that instability has occurred they do not by themselves indicate which alternative procedures should be used.

(ii) When terminal failures do not occur, it is nearly impossible to determine the validity of answers, and instability cannot be detected.

Clearly the usefulness of conventional iteration procedures can be greatly enhanced with: (1) absolute measures of instability; and (2) automatic methods for improving computational accuracy.

Absolute measures are described next, and the general rules for designing EDCA algorithms are given in Subsection 3.2. The particular version that is being used for applications of the common Newton-Raphson and Hamming-Kutta-Runge methods is discussed in Subsection 3.3 and demonstrated in Section 4.

3.1. Absolute Measures of Instability

Absolute measures can be devised with a general form of the Law of Conservation of Mass/Energy applied to all systems of equations, regardless of their physical meaning or interpretation. The physical form of this law, Mass Cannot be Created or Destroyed, asserts that equations must be balanced and that all specific reaction constants (single-valued parameters) and concentrations (multi-valued parameters) of reactants must have positive real values. The general form or mathematical variant is that all equations must be balanced, so parameters can have any real or complex values.

Applications of the general form of this law are used to compute the maximum computational error in every calculated value for every parameter. Key definitions are:

The Maximum Computational Accuracy is the maximum computational error in a run. A maximum computational accuracy of 0.00000026% means all the computed results are correct to at least 0.00000026%.

The Maximum Computational Accuracy for a Data-Point is the Maximum Computational Accuracy for the computed values for all variables in that data-point.

The ACCURACY System Parameter is the user stipulated maximum value for the computational accuracy. If this is exceeded by the maximum computational accuracy for any data point corrective actions are automatically invoked until either a solution is found or the computation is terminated. The user is informed about the automatic procedures invoked to improve the maximum computational accuracy or, if the computation is terminated because the limits of computational accuracy were reached, the specific tasks that must be performed before the run is resubmitted.

Suppose that values for the variables A, B, C and D are to be computed for $t = 0, 1, 2, ..., N$ from the boundary conditions (the values for k_1 and K_1 and initial values for all variables: A_0, B_0, C_0 and D_0) for the family of equations: $-\mathrm{d}A/\mathrm{d}t = -\mathrm{d}B/\mathrm{d}t = \mathrm{d}C/\mathrm{d}t = k_1AB$; $K_1 = D/C^2$, which in the physical sciences is described thus: $A + B \rightarrow C, k_1$; $2C \leftrightarrow D, K_1$.

For each variable in turn:

1. Recompute the value for the variable. For example, the recomputed value for variable B is: $NB_t = A_0 + B_0 + C_0 + 2D_0 - A_t - C_t - 2D_t$.
2. Calculate the computational accuracy for the value of the variable. For example, the computational accuracy for B_t is the absolute value of $100(NB_t - B_t)/B_t$.
3. Calculate the computational accuracy in K_1, CAK_{1X}, with the new value for the variable. For example, if NC_t is the recomputed value for C_t then CAK_{1C} is the absolute value of $100(NK_{1C} - K_1)/K_1$ where $NK_{1C} = D_t/NC_t^2$.

It should be noted that: (i) recomputation of a variable, like NB_t or NC_t, is independent of the meaning of the equation, and (ii) recomputation of a single-valued parameter, like NK_{1C}, is determined by the type of equation.

3.2. Methodology for Designing EDCA Algorithms

EDCA algorithms depend on the iteration method(s) that are used. The general methodology that was used to develop the highly successful EDCA algorithm (Subsection 3.3) is described next. It requires the availability of implemented forms for both the absolute measures of instability and the iteration procedure.

The steps are as follows:

STEP 1. Devise tests for detecting mathematical instability and measuring computational accuracy for the iteration procedure. Tests for mathematical instability (i.e. division by zero or failure to converge) are normally an integral part of most iteration procedures.

STEP 2. Use the absolute measures of instability to determine the parameters that control mathematical instability and the range (or scan limits) for each. Values for a parameter that lie outside its scan limits cause failures or require excessive computation time. Controlling parameters can be deduced from experiments that measure the effect of change in their values on the relative tests for mathematical instability.

STEP 3. Determine the causal relationships between the values for the controlling parameters and the kind of test.

STEP 4. Use the causal relationships obtained in Step 3 to deduce actions that will improve the computational accuracy.

STEP 5. Implement the EDCA algorithm. Information obtained in Steps 3 and 4 is used thus:

(i) If possible automate those actions that improve the computational accuracy and notify the user of the remedial actions.

(ii) If automation is not possible notify the user that instability has been detected and suggest (if possible) remedial actions.

3.3. Implemented Version of the EDCA Algorithm

The general form of the law of conservation of mass and energy is independent of the iteration method that is used. The first step towards realizing an EDCA algorithm is to implement a general form of the law for use in determining the computational accuracy in calculated values for variables. The second step is to design and implement EDCA algorithms for every different iteration procedure.

3.3.1. General Form of the Law of Conservation of Mass and Energy

In chemistry the equation for a reaction: $aA + bB = cC + dD$, which can represent either a rate reaction that describes a family of first order ordinary equations or a equilibrium reaction that describes a non-linear equation, stipulates that a units of A and b units of B combine to form c units of C and d units of D. The stoichiometric constants or coefficients (lower case letters) and concentrations of entities (upper case letters) must be positive real numbers. In this simple case, with values for (i) all stoichiometric constants, (ii) all variables in an initial state, and (iii) one of the variables in a second state, values for the remaining three variables in the second state can be calculated. For example, if A_0 and A_1 designate the first and second states of A, then: $B_1 = (A_1 - A_0)b/a - B_0$; $C_1 = (A_1 - A_0)c/a - C_0$; and $D_1 = (A_1 - A_0)d/a - D_0$.

For the general form of the law of conservation of mass and energy the parameters can be negative, positive or complex. However, values for variables are computed in exactly the same way.

For any system of equations and adequate computer resources, the implemented general form of the law (de Maine, 1980b) can be used to calculate the value for any multi-valued parameter in a designated state from the values for all the parameters in another state and values for selected parameters in the designated state. The equations can represent complex continuous or discontinuous functions with derivatives and/or integrals. Moreover, the multi-valued parameters can themselves represent complex functions and the coefficients may be described by combinations of Dirac Delta Functions and factorials. Here the computational techniques are illustrated with some simple examples. The subscripts o and 1 designate the initial and target states respectively.

Example 1. Compute the value for A_1 from B_1 for the following equation?

$$aA = bB + cC + dD$$

$$A_1 = a(B_1 - B_0)/b + A_0$$

Example 2. Compute the value for A_1 from B_1 and D_1 for the following system of equations?

$$aA + ... = bB + ...$$

$$a'A + ... = dD + ...$$

$$A_1 = a(B_1 - B_0)/b + a'(D_1 - D_0)/d + A_0$$

Example 3. Compute the value for B_1 from A_1 and D_1 for the following system of equations?

$$aA + ... = bB + ...$$

$$b'B + ... = dD + ...$$

$$B_1 = b(A_1 - A_0)/a - b'(D_1 - D_0)/d + B_0$$

Example 4. Compute the value for B_1 from A_1, D_1 and E_1 for the following systems of equations?

$$aA + ... = bB + ...$$

$$b'B + ... = dD + ...$$

$$d'D + ... = eE + ...$$

$$B_1 = b(A_1 - A_0)/a - b'\Big(D_1 - D_0 + d'(E_1 - E_0)/e\Big)/d + B_0$$

Example 5. Compute the value for A_1 from B_1, D_1 and E_1 for the following systems of equations?

$$aA + ... = bB + ...$$

$$a'A + ... = dD + ...$$

$$d'D + ... = eE + ...$$

$$A_1 = a(B_1 - B_0)/b + a'\Big(D_1 - D_0 + d'(E_1 - E_0)/e\Big)/d + A_0$$

The number of variables whose values are required in the second state is determined by the interdependence of the equations.

The general form of this algorithm is used to recompute the variable that has the lowest value. Space limitations prevent the inclusion of the detailed algorithm. However, the FORTRAN coded versions of the two key routines, called CACONC and CAMELD, have been described elsewhere (de Maine, 1980b). Copies can be obtained from the authors. Here it is sufficient to note that the primary task was to devise a book-keeping method for selecting that combination of values for parameters in the object state which simplifies the calculation of the value for the designated parameter.

The absolute measure for computational reliability is the percentile difference between the computed and the recomputed values for a parameter. If it exceeds the user stipulated value for the ACCURACY parameter then the computed value is not acceptable and corrective actions are invoked.

3.3.2. Controlling Parameters

The general purpose FRANS system, which operates in the prediction and computational modes (de Maine and de Maine, 1990), uses the well known Newton-Raphson and Hamming-Kutta-Runge iteration methods to so lve physical problems. There are eight tests for mathematical instability and computational accuracy (see Tab. 1). The first four are conventional mathematical tests, the fifth is used to indicate a successful restart in the Hamming-Kutta-Runge method, FAILED=6 results from the application of the physical form of the law of conservation of mass and energy, and the last two (FAILED=7 and 8) arise from the general form of the law of conservation of mass and energy.

Eleven parameters control computational accuracy (see Tab. 2). FUZZ is associated exclusively with the prediction mode. The remaining ten are associated with the Newton-Raphson and Hamming-Kutta-Runge iteration methods. All except EQSTEP, FUZZ, ITMAX, MINSTEP and WORDSIZE are dynamically variable, which means that they can be altered without terminating computations.

Exhaustive empirical studies with MINPOS and MAXNEG equal MINSTEP and $-1000 \times$ MINSTEP respectively have established the following.

1. Generally acceptable values for the five controlling parameters that cannot be dynamically adjusted (EQSTEP, FUZZ, ITMAX, MINSTEP and WORDSIZE). The first four are system parameters that are altered by the user when notified by the FRANS system to do so. WORDSIZE is determined by the machine architecture.
2. Generally acceptable values for the six dynamically adjustable parameters (ACCURACY, EQTOL, MINPOS, MAXNEG, TRUNCATE and SMALLEST). ACCURACY, EQTOL, TRUNCATE and SMALLEST are system parameters that can also be set by users. However, they are normally changed by the system itself.
3. The ranges of acceptable values for ACCURACY, EQTOL and TRUNCATE.
4. There is a causal relation between the kind of test and values for ACCURACY, EQTOL and TRUNCATE. Remedial actions for failures in tests are either to increase (Test 3) or decrease (all other tests) EQTOL. If EQTOL cannot be changed (i.e. it has a terminal value in its range) it is set to its default value and then TRUNCATE is changed. If TRUNCATE cannot be changed then it is set to its default value and ACCURACY is increased. If ACCURACY cannot be changed (i.e. it is the maximum value), the computational accuracy of the machine that is being used has been exceeded and the iteration is terminated with appropriate advisory messages.

The values assigned for the controlling parameters are shown in Tab. 3.

3.3.3. The Implemented EDCA Algorithm

For the implemented form of the EDCA algorithm the following definitions are required.

Tab. 1. Reliability tests for the Newton-Raphson and Hamming-Kutta-Runge iteration methods. FAILED is the error code. The first four are conventional mathematical tests. Tests 6, 7 and 8 are variants of the general form of the Law of Conservation of Mass and Energy. Test 5 indicates a successful restart of the Hamming-Kutta-Runge procedure.

FAILED	TEST
1	Failed the first Hamming-Kutta-Runge test (more than 1000 iterations were required).
2	Failed the second Hamming-Kutta-Runge test (convergence did not occur).
3	Failed the first Newton-Raphson test (convergence did not occur).
4	Failed the second Newton-Raphson test (terminal error occurred).
5	Successful recalculation of the starting value for a variable by the Hamming-Kutta-Runge method.
6	Failed because a negative value for a variable was calculated.
7	Failed because the maximum computational accuracy for the recalculated value for a variable exceeded the user-stipulated ACCURACY system parameter.
8	Failed because the maximum computational accuracy in the calculated value for a equilibrium constant exceeds the user-stipulated ACCURACY parameter.

Tab. 2. Parameters that control the computational accuracy in the FRANS system. FUZZ is used only in the prediction mode. The other parameters are used in the iteration mode. WORDSIZE is determined by the machine architecture. EQSTEP, FUZZ, ITMAX, MINSTEP and WORDSIZE are not dynamically adjustable.

Name	Description of Role of Parameter
ACCURACY	User-stipulated ACCURACY system parameter.
EQSTEP	Smallest equilibrium step for Newton-Raphson.
EQTOL	Controlling parameter for Newton-Raphson.
FUZZ	Used in the prediction mode to determine which variables are computable.
ITMAX	Maximum number of iterations for Newton-Raphson
MAXNEG	−MAXNEG is the largest negative value recognized for a computed parameter.
MINPOS	Smallest positive value recognized for a computed parameter.
MINSTEP	Smallest step in Hamming-Kutta-Runge.
SMALLEST	Smallest absolute difference recognized in the time scale for Hamming-Kutta-Runge.
TRUNCATE	Controlling parameter for Hamming-Kutta-Runge.
WORDSIZE	Number of bits in a double word.

Tab. 3. Values assigned for the Controlling Parameters for the Newton-Raphson and Hamming-Kutta-Runge iteration methods. FUZZ, ITMAX and WORDSIZE are not dynamically variable. MAXNEG and MINPOS have been arbitrarily set equal to 1000 x MINSTEP and MINSTEP respectively.

Name	Minimum	Default	Maximum
ACCURACY	10^{-7}	10.0	10.0
EQSTEP		10^{-31}	
EQTOL	0.1	10^{-6}	0.1
FUZZ		10^{-12}	
ITMAX		10.	
MAXNEG		10^{-12}	
MINPOS		10^{-15}	
MINSTEP		10^{-15}	
SMALLEST		10^{-9}	
TRUNCATE	0	0	14
WORDSIZE	64 Bits	64 Bits	64 Bits

RTFLAG is the number of time dependent equations.

NP_b is the number of the first data-point correctly computed with the current values for EQTOL, TRUNCATE and ACCURACY.

NP_c is the number of the current data-point.

FAILED (the error code) is 0, 1, 2, 3, 4, 5, 6, 7 or 8. 0 and 5 indicate successful computations. 3 specifies that the EQTOL is to be increased or TRUNCATE decreased. All other values for FAILED specify that EQTOL is to be decreased or TRUNCATE increased.

$EQTOL_{Max}$ the maximum value for EQTOL.

$EQTOL_{Min}$ the minimum value for EQTOL.

$TRUNCATE_{Max}$ the maximum value for TRUNCATE.

$TRUNCATE_{Min}$ the minimum value for TRUNCATE.

$ACCURACY_{Max}$ the maximum value for ACCURACY.

$ACCURACY_{Min}$ the minimum value for ACCURACY.

INCREASE the number of occurrences of FAILED=3 since the last successful computation.

DECREASE the number of occurrences for FAILED=1, 2, 4, 6, 7 and 8 since the last successful computation.

ACCURACY, EQTOL, MAXNEG, MINPOS, SMALLEST and TRUNCATE are the current values for the dynamically adjustable parameters. The values for MAXNEG, MINPOS and SMALLEST have been arbitrarily set to MINSTEP, 1000 x MINSTEP and 10^{-9} respectively (see Tab. 3). DECREASE and INCREASE are used to detect oscillations between FAILED=3 and all other values for FAILED > 0 except FAILED=5.

If there is any time dependent equation present (RTFLAG > 0) and $NP_c - NP_b > 1$ then the iteration is restarted at the last successfully computed data-point, NP_{c-1}, without changing the values for any dynamically adjustable parameter. The iteration is terminated with appropriate advisory messages if: (i) all data-points are successfully computed; (ii) the allocated time has been completed; (iii) all possible values for ACCURACY, EQTOL and TRUNCATE have been scanned and the limits of computational accuracy have been exceeded; or (iv) both DECREASE and INCREASE are greater than five. A step-by-step description of the EDCA algorithm for the Newton-Raphson and Hamming-Kutta-Runge methods follows.

STEP 1: If FAILED=0 or 5 — the current data point, NP_c, was successfully processed — set INCREASE and DECREASE to zero and then go to Step 14. Otherwise go to Step 2.

STEP 2: If RTFLAG=0 there is no first order ordinary differential equation:

Set $NP_r = NP_c$ and $NP_d = 1$

If RTFLAG > 0 there is at least one first order ordinary differential equation.

(a) If less than two data points were successfully processed ($NP_c - NP_b < 2$):

Set $NP_r = NP_{c-1}$ and $NP_d = 1$

(b) If more than one data point was successfully processed ($NP_c - NP_b > 1$):

Set $NP_r = NP_{c-1}$ and $NP_d = NP_c - NP_b$

STEP 3: If $NP_d < 2$ — less than two data points were successfully processed — go to Step 4.
If $NP_d > 1$ — more than one data point was successfully processed — go to Step 13.

STEP 4: If FAILED=3 the value of EQTOL is to be increased. Go to Step 5.
If FAILED=1, 2, 4, 6, 7 or 8 the value of EQTOL is to be decreased. Go to Step 8.

STEP 5: Set INCREASE=INCREASE+1.
If both INCREASE and DECREASE are greater than 4 — an oscillating condition has been identified — terminate the run by executing Step 12. Otherwise go to Step 6.

STEP 6: EQTOL is to be increased.
If 10 × EQTOL < or = $\text{EQTOL}_{\text{Max}}$, set EQTOL = 10 × EQTOL then execute Step 13.
If 10 × EQTOL > $\text{EQTOL}_{\text{Max}}$ — the maximum value of EQTOL has been exceeded and the value of TRUNCATE is to be decreased — set EQTOL to its default value and then execute Step 7.

STEP 7: TRUNCATE is to be decreased.
If TRUNCATE−1 > or = $\text{TRUNCATE}_{\text{Min}}$, set TRUNCATE = TRUNCATE−1 then execute Step 13.

If TRUNCATE−1 < $\text{TRUNCATE}_{\text{Min}}$ — the minimum value of TRUNCATE has been exceeded and the value of ACCURACY must be increased — set TRUNCATE equal to its default value and then execute Step 11.

STEP 8: Set DECREASE = DECREASE+1.
If both INCREASE and DECREASE are greater than 4 — an oscillating condition has been identified — terminate the run by executing Step 12. Otherwise go to Step 9.

STEP 9: EQTOL is to be decreased.
If EQTOL/10 > or = $\text{EQTOL}_{\text{Min}}$ set EQTOL = EQTOL/10 and then execute Step 13.
If EQTOL/10 < $\text{EQTOL}_{\text{Min}}$ — the minimum value of EQTOL has been exceeded and the value of TRUNCATE is to be increased — set EQTOL to its default value and then execute Step 10.

STEP 10: TRUNCATE is to be increased.
If TRUNCATE+1 < or = $\text{TRUNCATE}_{\text{Max}}$ set TRUNCATE equal to TRUNCATE+1 and then execute Step 13.
If TRUNCATE+1 > $\text{TRUNCATE}_{\text{Max}}$ — the maximum value of TRUNCATE has been exceeded and the value of ACCURACY must been increased — set TRUNCATE equal to its default value and then execute Step 11.

STEP 11: ACCURACY is to be increased.
If 1.10 x ACCURACY < or = $\text{ACCURACY}_{\text{Max}}$ set ACCURACY equal to 1.10 x ACCURACY and then execute Step 13.
If 1.10 x ACCURACY > $\text{ACCURACY}_{\text{Max}}$ — the maximum value of ACCURACY has been exceeded and the run is to be terminated — execute Step 12.

STEP 12: At this point three of the six dynamically adjustable parameters (EQTOL, TRUNCATE and ACCURACY) have been scanned without finding an acceptable solution. The run is terminated with recommendations for the restarting conditions and options. The options in order of preference are:

Option A. If FAILED=8 — Test 8 in Tab. 1 — delete all references to the designated culprit calculated constant from the model.

Option B. If FAILED=6 or 7 — Tests 6 and 7 in Tab. 1 — delete all references to the designated culprit calculated variable from the model.

Option C. If FAILED=1, 2, 3 or 4 and $NP_r = 1$ — Tests 1, 2, 3 and 4 in Tab. 1 — the probable cause is the presence of a reactant with zero concentration that is not formed in any reaction. Set the culprit initial concentration to non-zero.
In many thousands of hours of computer time a terminal condition with $NP_r > 1$ has never been observed.

Option D. Increase the system parameters MINPOS and MAXNEG one hundred fold. All computed positive values less than MINPOS and negative values greater than −MAXNEG are to be ignored.

Option E. Respond to the prompt, and increase the ranges for the EQTOL and TRUNCATE system parameters.

STEP 13: Restart the computation with data-point NP_r then execute Step 14.

STEP 14: If the last data-point has been processed terminate the run. Otherwise determine the remaining time. If it is less than five seconds set FAILED equal to 12+FAILED and then print messages about the recommended conditions for manually restarting the run. If more than five seconds remain continue with the computation of the next data point.

It is planned to extend this algorithm to include automation of changes for the remaining three dynamically adjustable parameters (MAXNEG, MINPOS and SMALLEST).

4. Applications of the EDCA Algorithm

The purpose of this section is to demonstrate with selected examples the EDCA method that is used in the FRANS system to control computational errors. The user-friendly interfaces can only be demonstrated on computers. Examples that are trivial from the mathematical viewpoint are used because of space limitations. Within the available computer resources there are no limits on the size or complexity of models. Here it should be noted that the meaning of the widely used notation for describing families of equations is as follows. $RKF1$, $RKF2$, RKB and EK are single-valued parameters. All other parameters are multi-valued. Both single- and multi-valued parameters may be functions of several parameters.

$A \rightarrow 2.1 \times B + C$, $RKF1$; means a family of first order ordinary differential equations:

$$-\mathrm{d}A/\mathrm{d}t = \mathrm{d}B/2.1\mathrm{d}t = \mathrm{d}C/\mathrm{d}t = RKF1.A$$

$C \Leftrightarrow A + D$, $RKF2$, RKB; means two families of first order differential equations:

$$-\mathrm{d}C/\mathrm{d}t = \mathrm{d}A/\mathrm{d}t = \mathrm{d}D/\mathrm{d}t = RKF2.C$$

$$-\mathrm{d}A/\mathrm{d}t = -\mathrm{d}D/\mathrm{d}t = \mathrm{d}C/\mathrm{d}t = RKB.A.B$$

$X + B \leftrightarrow C + A$, EK; means the non-linear equation:

$$EK = C.A/X.B$$

The use of the FRANS system has been described in (de Maine and de Maine, 1987; 1990; Marsili, 1990; de Maine *et al.*, 1985) and especially in (de Maine, 1980a). Here it is used to demonstrate the implemented form of the EDCA algorithm. The concentrations for all reactants are to be computed at times 0.0, 0.1, 0.5 and 1.0 seconds for the system of equations:

$$A \rightarrow 2.1 \times B + C,\ RKF1;\quad C \Leftrightarrow A + D,\ RKF2,\ RKB;\quad X + B \leftrightarrow C + A,\ EK;$$

with $RKF1 = 1$, $RKF2 = 0.001$, $RKB = 10$, $EK = 5$, and the initial concentrations for A, B, C, D and X equal to 0.1, 0, 0, 0 and 10 moles/liter respectively. $RKFx$ and RKB are the specific rate constants for the forward and backward reactions, and EK is the equilibrium constant. All times quoted in this paper were measured on a 386 based Toshiba 5200.

Tab. 4. Pertinent part of the output from FRANS for the system of equations defined in the text. The maximum computational accuracy establishes that all the calculated values for concentration of the five reactants are correct to at least 1.4327329 percent. A computational accuracy of 0.0 percent means that the calculated value is accurate to within the minimum recognized positive value, 10^{-15}.

TABULATED RESULTS OUTPUTTED BY THE SOLVER PART OF THE CRAMS SYSTEM.

4 SIGNIFICANT FIGURES ARE TO BE PRINTED FOR EACH OF THE VALUES FOR THE 5 COMPOUNDS FOR THOSE OF THE 4 DATA-POINTS THAT WERE COMPUTED.
"TRUNCATE", "EQTOL", "ACCURACY" & "MINSTEP" ARE SYSTEM PARAMETERS.
FOR DATA-POINTS 1 TO 4 "TRUNCATE" = 0, "EQTOL" = 0.1000D−05 &
"ACCURACY" = 0.1000D+02.
THE DEFAULT VALUES ARE: −"TRUNCATE" = 0, "EQTOL" = 0.1000D−05,
"ACCURACY" = 0.1000D+02 AND
"MINSTEP" = 0.1000D−14.
THE MINIMUM POSITIVE VALUE FOR A CONCENTRATION THAT WAS USED FOR CHECKING IS 0.1000D−14.
ALL NEGATIVE CONCENTRATIONS GREATER THAN 0.1000D−11 WERE SET TO ZERO.

ALL 4 OF THE REQUESTED DATA-POINTS WERE COMPUTED.

THE AMOUNT OF TIME FOR CALCULATIONS WAS 0.2000D+01 SECONDS.

THE MAXIMUM COMPUTATIONAL ACCURACY IS 0.14327329D+01 %.
− COMPUTATIONAL ACCURACY FOR DATA-POINT 1 IS 0.00000000D+00%.
− COMPUTATIONAL ACCURACY FOR DATA-POINT 2 IS 0.14327329D+01%.
− COMPUTATIONAL ACCURACY FOR DATA-POINT 3 IS 0.21810109D−01%.
− COMPUTATIONAL ACCURACY FOR DATA-POINT 4 IS 0.22093028D−03%.
THE COMPUTATIONAL ACCURACY CAN BE IMPROVED BY DECREASING THE SYSTEM PARAMETER "ACCURACY" FROM 10.00000000 TO 0.00022083.
HOWEVER THE TIME REQUIRED MAY BE SUBSTANTIALLY INCREASED.
THE COMPUTATIONAL SPEED CANNOT BE INCREASED BY CHANGING "ACCURACY".

TIME	*A*	*B*	*C*	*D*
0.0000D+00	0.1000D+00	0.0000D+00	0.0000D+00	0.0000D+00
0.1000D+00	0.1116D+00	0.7414D−04	0.3268D−01	0.1549D−05
0.5000D+00	0.1725D+00	0.7288D−03	0.2054D+00	0.3719D−04
0.1000D+01	0.2956D+00	0.3472D−02	0.5570D+00	0.1250D−03

TIME	*X*
0.0000D+00	0.1000D+02
0.1000D+00	0.9835D+01
0.5000D+00	0.9720D+01
0.1000D+01	0.9483D+01

Table 4 contains the pertinent part of the output with the ACCURACY system parameter set to its default value, 10. Two seconds were required to achieve a maximum computational accuracy of 1.437 percent for all the values of A, B, C, D and

X that are displayed at the end of the Table. It should be noted that the maximum computational accuracy of 1.437 percent could only be determined by application of the general form of the law of conservation of mass and energy. The advisory messages include the statement that the computational accuracy can be improved by changing the system parameter from its default value, 10.0, to 0.00022083.

Tables 5 and 6 contain the advisory messages for the ACCURACY system parameter set to 0.00022083, 0.00000073 and 0.00000037. The amount of computer time increases from two seconds, for ACCURACY = 10.0, to 205 seconds for ACCURACY = 0.00000037. With ACCURACY equal to 0.00000037 the computational limits for a 64-Bit machine is exceeded and there are two remedial options. With the second option, deleting reactant X, the computer time and maximum computational accuracy are reduced to one second and 0.0 percent respectively (Tab. 6). The results displayed in Tables 4, 5 and 6 are summarized in Tab. 7.

Tab. 5. EDCA messages for the system parameter ACCURACY set to 0.00022083 and 0.00000078.

ACCURACY System Parameter 0.00022083

ALL 4 OF THE REQUESTED DATA-POINTS WERE COMPUTED.

THE AMOUNT OF TIME FOR CALCULATIONS WAS 0.5000D+01 SECONDS.

THE MAXIMUM COMPUTATIONAL ACCURACY IS 0.11435083D−05 %.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 1 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 2 IS 0.88418585D−06%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 3 IS 0.11435083D−05%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 4 IS 0.83249193D−06%.

THE COMPUTATIONAL ACCURACY CAN BE IMPROVED BY DECREASING THE SYSTEM PARAMETER "ACCURACY" FROM 0.00022083 TO 0.00000073.
HOWEVER THE TIME REQUIRED MAY BE SUBSTANTIALLY INCREASED.
THE COMPUTATIONAL SPEED CAN BE IMPROVED BY INCREASING THE SYSTEM PARAMETER "ACCURACY" FROM 0.00022083 TO 10.00000000.

ACCURACY System Parameter 0.00000078

ALL 4 OF THE REQUESTED DATA-POINTS WERE COMPUTED.

THE AMOUNT OF TIME FOR CALCULATIONS WAS 0.38D+02 SECONDS.

THE MAXIMUM COMPUTATIONAL ACCURACY IS 0.47218717D−06 %.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 1 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 2 IS 0.47218717D−06%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 3 IS 0.33973727D−07%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 4 IS 0.27417794D−07%.

THE COMPUTATIONAL ACCURACY CAN BE IMPROVED BY DECREASING THE SYSTEM PARAMETER "ACCURACY" FROM 0.00000078 TO 0.00000037.
HOWEVER THE TIME REQUIRED MAY BE SUBSTANTIALLY INCREASED.
THE COMPUTATIONAL SPEED CAN BE IMPROVED BY INCREASING THE SYSTEM PARAMETER "ACCURACY" FROM 0.00000078 TO 10.00000000.

Tab. 6. EDCA messages generated for the two runs with the system parameter ACCURACY set to 0.00000037. For the second run all references to the reactant X were deleted. All calculated values are correct to within the minimum positive value that is recognized, 10^{-15}.

ACCURACY System Parameter 0.00000037

ALL 4 OF THE REQUESTED DATA-POINTS WERE COMPUTED.

THE AMOUNT OF TIME FOR CALCULATIONS WAS 0.2050D+03 SECONDS.

THE MAXIMUM COMPUTATIONAL ACCURACY IS 0.46336507D−06 %.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 1 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 2 IS 0.46336507D−06%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 3 IS 0.52034360D−09%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 4 IS 0.85641250D−09%.

THE COMPUTATIONAL LIMITS FOR A 64-BIT WORD MACHINE HAVE BEEN EXCEEDED WITH "ACCURACY"= 0.3700D−06. THE REMEDIAL OPTIONS ARE TO CHANGE THE SCAN LIMITS FOR "TRUNCATE" AND "EQTOL" WHEN INTERROGATED BY THE FRANS SYSTEM OR:−

- DELETE THE REACTANT X AS FOLLOWS:−
  - A. SET SYSTEM PARAMETERS "TRUNCATE"=0, "EQTOL" =0.1000D−05 AND "ACCURACY" TO 0.3700D−06.
  - B. DELETE THE EQUATION(S) WITH VARIABLE X FROM THE REACTION MODEL.
  - C. DELETE THE REACTION CONSTANT(S) IN THE DELETED REACTION(S).
  - D. DELETE THE VALUE(S) OF THE DELETED VARIABLE X.
  - E. FOR INITIAL VALUES USE THOSE VALUES SHOWN FOR DATA-POINT 1 AND SET THE STARTING TIME TO 0.0000D+00 UNITS.
  - F. RESUBMIT THIS RUN WITH T = 220 SECONDS.

ACCURACY System Parameter 0.00000037 with Reactant X Deleted

ALL 4 OF THE REQUESTED DATA-POINTS WERE COMPUTED.

THE AMOUNT OF TIME FOR CALCULATIONS WAS 0.1000D+01 SECONDS.

THE MAXIMUM COMPUTATIONAL ACCURACY IS 0.00000000D+00 %.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 1 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 2 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 3 IS 0.00000000D+00%.
- COMPUTATIONAL ACCURACY FOR DATA-POINT 4 IS 0.00000000D+00%.

THE COMPUTATIONAL ACCURACY CANNOT BE INCREASED BY CHANGING "ACCURACY".

THE COMPUTATIONAL SPEED CANNOT BE INCREASED BY CHANGING "ACCURACY".

Tab. 7. Summary of the pertinent results displayed in Tables 4, 5 and 6 for the EDCA algorithm, obtained with a TOSHIBA 5200. SPA and MCA are the System Parameter ACCURACY and the Maximum Computational Accuracy respectively. In Run 4 all references to the variable X were deleted from the model, the second option given in Tab. 6. MCA = 0.00000000 means that the values for the variables A, B, C and D normally displayed (not shown in Tab. 6) are within their correct values as stipulated by the values for dynamically adjustable parameters MINPOS, MINNEG and SMALLEST.

Run	Time (secs)	SPA (%)	MCA (%)
0	2	10.0000000	1.43273300
1	5	0.00022082	0.00000114
2	38	0.00000078	0.00000047
3	205	0.00000037	0.00000047
4	1	0.00000037	0.00000000

5. Conclusions

Two methods for determining computational accuracy have been described and demonstrated. In the Maximum Tolerance procedure user-supplied estimates of the reliability of raw-data are used to eliminate the ambiguity that is an inherent part of the conventional curve-fitting methods.

In the Error Detection And Corrective Action (EDCA) algorithm, which is an application of a general form of the law of conservation of mass and energy, user stipulated computational accuracies are used in iteration procedures to detect mathematical instability and invoke corrective actions to ensure that only acceptable answers are displayed.

References

de Maine P.A.D. (1965): *The self-judgement method of curve-fitting.* — Comm. Assoc. Comp. Mach., v.8, pp.518–526.

de Maine P.A.D. (1978a): *Automatic curve-fitting I. Test methods.* — Computers and Chemistry, v.2, pp.1–6.

de Maine P.A.D. (1978b): *Empirical relationships for random self-avoiding walks on lattices.* — Computers and Chemistry, v.2, pp.53–64.

de Maine P.A.D. (1980a): *Operation Manual for the CRAMS System.* — Report No.5, Series: Automatic Systems for the Physical Sciences, Computer Science Department, The Pennsylvania State University, University Park, PA 16802.

de Maine P.A.D. (1980b): *Systems Manual for the CRAMS System.* — Report No.6, Series: Automatic Systems for the Physical Sciences, Computer Science Department, The Pennsylvania State University, University Park, PA 16802, p.125.

de Maine P.A.D. and de Maine M.M. (1987): *Automatic deductive systems I. Chemical reaction models.* — Computers and Chemistry, v.11, No.1, pp.49–65.

de Maine P.A.D. and de Maine M.M. (1990): *Computer aids for chemists.* — Anal. Chim. Acta, v.235, pp.7–26.

de Maine P.A.D. and de Maine M.M. (1992a): *A computer tool kit for chemists, Part II. Maximum tolerance procedure.* — Anal. Chim. Acta, v.256, pp.361–368.

de Maine P.A.D. and de Maine M.M. (1992b): *A computer tool kit for chemists, Part III. Error detection and corrective action procedure.* — Computers and Chem., v.16, pp.53–60.

de Maine P.A.D., de Maine M.M., Cartee B.C. and Chung W. (1985): *Computer Methods for Experiment Design and Analysis.* — Report No.7, Series: Automatic Systems for the Physical Sciences, Computer Science and Engineering Department, Auburn University, Auburn, AL 36849, p.230.

de Maine P.A.D., de Maine M.M. and Goble A.G. (1957): Trans. Faraday Soc., v.53, p.427.

de Maine P.A.D. and Seawright R.D. (1963a): *The self-judgement principle in scientific data processing.* — Ind. Eng. Chem., v.55, pp.29–32.

de Maine P.A.D. and Seawright R.D. (1963b): *Digital Computer Programs for Physical Chemistry Volume I.* — New York: The Macmillan Company (See Chapter IV, pp.123–125).

de Maine P.A.D., Springer G.K. and Mikelskas R.A. (1978): *Automatic curve-fitting II. Linear equations.* — Computers and Chemistry, v.2, pp.7–14.

Marsili M. (1990): *Computer Chemistry.* — CRC Press Inc., Chapter 5.

Sillen L.G. (1962): Acta Chem. Scand., v.16, pp.159–172.

Thisted R.A. (1988): *Elements of Statistical Computation: Numerical Computation.* — New York: Chapman and Hall.

Received: February 15, 1994
Revised: May 19, 1994