Mathl Comput. Modelling, Vol. 11, pp. 123–128, 1988
Printed in Great Britain

0895-7177/88 $3.00 + 0.00
Pergamon Press plc

S–SYSTEMS

EFFICIENT SOLUTION OF NONLINEAR MODELS EXPRESSED IN S-SYSTEM CANONICAL FORM

Douglas H. Irvine

Department of Microbiology and Immunology, The University of Michigan, Ann Arbor, MI 48109

Abstract. The S-system is emerging as a general canonical form for analysis of nonlinear models. Models expressed within this regularly structured system of nonlinear ordinary differential equations are obtained by applying either of two different strategies: (A) Direct derivation of an S-system utilizing the Power Law Formalism; or (B) exact recasting of an existing, well established model into S-system form. By capitalizing on the regular structure of S-systems, efficient formulas for numerical solution of this general class have been developed. For any S-system it can be shown that these formulas are more efficient than conventional multistep formulas of the same order. For implemented methods, the actual improvements in efficiency are considerably more than the minimum estimates. Preliminary tests show that time required for solution of S-systems is reduced by one or two orders of magnitude -- the relative improvement in efficiency increases with size and complexity of the problem, and with degree of accuracy required.

Keywords. Differential Equations; Numerical Solution; Recasting; Taylor Series; S-Systems.

GENERALITY OF S-SYSTEMS

S-systems originally were developed for analysis of organizationally complex systems such as cellular and molecular networks in biology (Savageau 1969a, b, 1971a,b). Each equation of an S-system describes the change in concentration of an element of the system, X_i, with time and is comprised of two terms, one representing net production or influx and the other representing net degradation or efflux. An element may be a molecule, cell, organism or some other quantifiable component of a system.

Mathematically the net production and net degradation terms for each element are products of power-law functions, giving S-systems the following form:

$$X_{i(1)} = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} \qquad i = 1...n \qquad (1)$$

where $X_{i(1)}$ is the change in X_i with time, α_i and β_i are apparent rate constants for net production and net degradation, g_{ij} and h_{ij} are apparent kinetic orders for the effect that X_j has on net production and net degradation of X_i, and n is the number of variables in the system which is equal to the number of differential equations if it's assumed for simplicity that the system is autonomous. Values for the X's are real positive numbers, for the α's and β's real nonnegative and for the g's and h's real.

S-systems have been used to represent biochemical pathways, genetic circuits, immune networks, patterns of growth and development, and ecological interactions (for references and reviews see Savageau 1971b, 1976, Irvine and Savageau 1985). In addition, many other "laws" of nature can be represented exactly as S-systems -- examples include rate laws for enzyme kinetics, special functions in physics, all of the well known growth laws, all of the well known probability distributions and the Cobb-Douglas production functions from economics (see Savageau 1979a, 1982).

Although the full extent of the class of equations that can be recast as S-systems remains to be determined, initial investigations suggest that it is extremely broad. It has been possible to recast every ordinary differential equation examined by applying the algorithm described by Savageau and Voit (1986). The resultant S-systems are equivalent representations that are no more approximate than the original equations.

Whether derived by applying the Power Law Formalism or by recasting established equations, S-systems provide a very general nonlinear formalism for analysis of organizationally complex systems (Savageau 1985, Voit and Savageau 1986). Whatever tools have been or can be developed for analysis of S-systems can be applied to a wide class of complex phenomena all unified in this standard canonical form. General mathematical methods for explicit steady-state solution, sensitivity analysis and stability analysis of S-systems have been know for several years (Savageau 1969a, b, 1979a, b, 1976). Since it also is known that a general analytical solution for the dynamic behavior of S-systems is not possible, attention has been focused on efficient methods for numerical solution. Recently it has been discovered that by capitalizing on the regular structure of S-systems, very efficient methods can be derived based on analytical continuation with Taylor series (Irvine 1986).

CONVENTIONAL FORMULAS FOR NUMERICAL SOLUTION

Subroutines for solution of ordinary differential equations are among the most widely used programs in mathematical software libraries (Rice 1983). Typically these routines numerically integrate a system of equations by piecing together a local estimate for a truncated Taylor series of the solution. Although most of these methods are general and work for virtually any functional form, including S-systems, no one method is best for every class of equations. The method that is most efficient depends on the problem solved. For a given class of equations the best method can be found only by comparing numerous existing methods or by developing specialized techniques that take advantage of structure inherent in the class.

Most conventional methods are very convenient to use since they apply to ordinary differential equations of virtually any functional form. However, because they must solve equations with many different forms, they do not take advantage of any regular structure within a given class. This means that efficiency is traded for convenience. When a problem must be solved only once, this usually is prudent, since developing a method to capitalize on structure may require more time than ultimately would be saved. On the other hand whenever a given problem must be solved repeatedly (e.g. for different error tolerances, initial conditions, time intervals or parameter values), or an entire class of problems must be solved and can be written in a standard canonical form, substantial savings are realized by developing methods that capitalize on structure within the problem. As shown in the next section, the regular structure of S-systems allows both efficient and accurate extraction of information about solutions.

CAPITALIZING ON THE S-SYSTEM FORM

Although one could apply available general-purpose algorithms for numerical solution of Eq. (1) directly, the efficiency of solution is improved by noting the special character of the nonlinear functions and simplifying them by logarithmic transformation (Savageau 1970, 1976). Even greater improvements in efficiency can be achieved by abandoning available algorithms and developing new algorithms that take advantage of the special character of S-systems throughout. The derivation of such an algorithm based on analytical continuation with Taylor series in logarithmic coordinates is outlined below.

A truncated Taylor series for solution of an S-system in logarithmic space can be written as follows:

$$Y_i(t+h) = Y_i(t) + \sum_{m=1}^{p} [\, Y_i^{(m)}(t) / m \,]\, h^m \qquad i = 1 \ldots n \qquad (2)$$

where Y_i is the logarithm of X_i, t is time, h is the increment in time to the next solution point, p is the order of the truncated expansion, and $Y_i^{(m)}(t)$ is the m^{th} derivative of Y_i evaluated at time t and divided by (m-1) factorial. Formulas for each derivative of Y_i with respect to time must be obtained to evaluate Eq. (2) exactly.

The first derivative of Y_i with respect to time can be obtained simply by dividing both sides of Eq. (1) by X_i:

$$\begin{aligned} Y_i^{(1)} &= X_i^{(1)} / X_i \\ &= \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}'} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}'} \\ &= \alpha_i \exp\Big(\sum_{j=1}^{n} g_{ij}' Y_j \Big) - \beta_i \exp\Big(\sum_{j=1}^{n} h_{ij}' Y_j \Big) \qquad (3) \\ &= A_i^{(1)} - B_i^{(1)} \end{aligned}$$

where $g_{ij}' = g_{ij} - \delta_{ij}$ and $h_{ij}' = h_{ij} - \delta_{ij}$, with $\delta_{ij} = 0$ for i not equal to j and $\delta_{ij} = 1$ for i equal to j.

Because α_i, β_i, g_{ij}' and h_{ij}' are all constants, Eq. (3) can be differentiated easily to obtain the second derivative of Y_i:

$$\begin{aligned} Y_i^{(2)} &= A_i^{(1)} \Big[\sum_{j=1}^{n} g_{ij}' Y_j^{(1)} \Big] - B_i^{(1)} \Big[\sum_{j=1}^{n} h_{ij}' Y_j^{(1)} \Big] \\ &= A_i^{(1)} G_i^{(1)} - B_i^{(1)} H_i^{(1)} \qquad (4) \\ &= A_i^{(2)} - B_i^{(2)} \end{aligned}$$

where

$$G_i^{(1)} = \sum_{j=1}^{n} g_{ij}' Y_j^{(1)}$$

$$H_i^{(1)} = \sum_{j=1}^{n} h_{ij}' Y_j^{(1)}$$

This process of differentiation can be repeated indefinitely, and each of the higher derivatives (m = 2, 3, 4 ...) of Y_i with respect to time can be calculated recursively. However, to improve the efficiency of calculating coefficients for the corresponding Taylor series, the m^{th} derivative of Y_i divided by (m-1) factorial is calculated instead of the derivative alone. These coefficients can be written as $Y_i^{(m)}$ and can be calculated follows:

$$\begin{aligned} Y_i^{(m)} &= A_i^{(m)} - B_i^{(m)} \\ A_i^{(m)} &= \Big[\sum_{k=1}^{m-1} A_i^{(m-k)} G_i^{(k)} \Big] / (m-1) \qquad (5) \\ B_i^{(m)} &= \Big[\sum_{k=1}^{m-1} B_i^{(m-k)} H_i^{(k)} \Big] / (m-1) \end{aligned}$$

where

$$G_i^{(k)} = \sum_{j=1}^{n} g_{ij}' Y_j^{(k)}$$

$$H_i^{(k)} = \sum_{j=1}^{n} h_{ij}' Y_j^{(k)}$$

By applying the above equations, the m^{th} derivative of Y_i divided by (m-1) factorial can be obtained recursively from previously calculated and stored values for the lower-order quantities in Eq. (5). This process is very efficient as can be seen by comparison with conventional methods.

EFFICIENCY OF TAYLOR-SERIES AND RUNGE-KUTTA FORMULAS

How does numerical solution by the Taylor-series method compare with popular general-purpose methods? For comparison the subroutine RKF45 (Fehlberg 1969, Shampine and Watts 1977) is used because it is a well known method that has been compared with many other Runge-Kutta, predictor-corrector and extrapolation methods (Hull *et al.* 1972, Shampine *et al.* 1976). Even though RKF45 is not always the most efficient among these methods, reported differences are small compared with those between the Taylor-series method and RKF45.

The efficiency of any method for numerical solution of ordinary differential equations is a complex function of several factors, including the number of derivative evaluations, the cost of evaluating derivatives, the cost of estimating the local error, the cost of selecting a step size, the cost of advancing the solution and the cost of reporting solution points. The problem dependence of these factors is difficult to express, and the relative contribution of each factor usually cannot be determined in general.

This section focuses on the cost of evaluating derivatives, estimating the local error and advancing the solution for each method. These costs, which can be enumerated explicitly, are expressed as the number of operations required for each step of a solution by a fifth-order Runge-Kutta method and as the number of operations required for calculation of a fifth-order truncated Taylor series by Eqs. (2), (3) and (5). By determining these irreducible costs the minimum difference between Runge-Kutta and Taylor-series methods can be estimated, independent of the other more problem-dependent advantages of the Taylor-series method. The additional advantages, which prove to make substantial improvements in efficiency over the minimum estimate, must be examined by comparing the performance of implemented methods in the next section.

Calculation of first derivatives for a system of n differential equations like Eq. (1) requires $2n^2$ exponentiations, $2n^2$ multiplications and n additions. When the same system is transformed into logarithmic coordinates, as in Eq. (3), 2n exponentiations, $2n^2+2n$ multiplications and $2n^2-n$ additions are required. This logarithmic transformation reduces the number of exponentiations by 2n(n-1) but increases the number of multiplications by 2n and the number of additions by 2n(n-1). Roughly speaking this is a trade of 2n(n-1) exponentiations for additions. Because exponentiation typically requires from 2 to 30 times more computer time than addition, solution of an S-system in logarithmic coordinates is more efficient than solution in Cartesian coordinates. Hence, in subsequent comparisons solution of Eq. (3) is considered rather than solution of Eq. (1).

A fifth-order Runge-Kutta method requires a minimum of six first-derivative evaluations for each step (Butcher 1965). Table I shows the minimum number of operations required to calculate first derivatives using Eq. (3) at six different sets of values of Y_i. One of these sets always is provided as initial values of Y_i, or as the last solution point, but the other five sets are intermediate values that must be calculated. Incrementing Y_i to these intermediate values, estimating the error and then incrementing Y_i to the next solution point, requires an additional 24n multiplications and 19n additions for each step. The total number of arithmetic operations increases as a quadratic in n. Table I also shows the number of operations required to evaluate the first through fifth derivatives for each step of a fifth-order Taylor-series method. With this method the last term of the truncated Taylor series provides the error estimate, so no additional operations are required, and incrementing Y_i to the next solution point requires only 5n multiplications and 5n additions. Once again, the total number of arithmetic operations increases as a quadratic in n.

Table I. Comparison of Fifth-Order Methods using Runge-Kutta and Taylor-Series Formulas [a]

Formula	exp	mult	add
RKF45 [b]			
$Y_i^{(1)}$	12n	$12n^2+12n$	$12n^2-6n$
error & step	0	24n	19n
total	12n	$12n^2+36n$	$12n^2+13n$
TAYLOR SERIES [c]			
$Y_i^{(1)}$	2n	$2n^2+2n$	$2n^2-n$
$Y_i^{(2)}$	0	$2n^2+3n$	$2n^2-n$
$Y_i^{(3)}$	0	$2n^2+7n$	$2n^2+n$
$Y_i^{(4)}$	0	$2n^2+9n$	$2n^2+3n$
$Y_i^{(5)}$	0	$2n^2+11n$	$2n^2+5n$
error & step	0	5n	5n
total	2n	$10n^2+37n$	$10n^2+12n$

[a] Based on the minimum number of exponentiations (exp), multiplications (mult) and additions (add) required for one step by each method.
[b] Requires six evaluations of the first derivative.
[c] Requires one evaluation of the first through fifth derivatives.

An estimate of the minimum savings for the Taylor-series method is obtained by subtracting the total number of operations listed in Table I for the Taylor-series method from that for the fifth-order Runge-Kutta method:

$$S(n) = 10n\,C_e + (2n^2 - n)\,C_m + (2n^2 + n)\,C_a \qquad (6)$$

where S(n) is the savings for the Taylor-series method in units of computer time for a system of n differential equations, with C_e, C_m and C_a equal to the unit cost of exponentiation, multiplication and addition. Because n, C_e, C_m and C_a are all greater than zero, some savings always is realized by applying the fifth-order Taylor-series method instead of the fifth-order Runge-Kutta method. Absolute savings increase commensurate with total cost, as a quadratic in n. As shown in Table II, similar conclusions hold for second-order through fourth-order methods except that absolute savings increase linearly with n.

Table II. Minimum Number of Operations Saved per Step by Taylor-Series Method [a]

order	exp	mult	add
1 [b]	0	0	0
2	+ 2n	0	+ n
3	+ 4n	- 2n [c]	+ n
4	+ 6n	- 8n [d]	- 3n
5	+ 10n	$+ 2n^2-n$	$+ 2n^2+n$

[a] The first-order through fourth-order Runge-Kutta methods used for comparison are fixed-step-length methods that do not estimate the error. The fifth-order Runge-Kutta method used for comparison is RKF45.
[b] Both Runge-Kutta and Taylor-series methods are identical to Euler's method when the order is one.
[c] The savings due to elimination of 4n exponentiations alone is greater than the cost of 2n more multiplications.
[d] Typically the cost for exponentiation is at least twice that for multiplication and addition, so the savings due to elimination of 6n exponentiations is greater than the cost of 8n more multiplications and 3n more additions.

These results show that calculation of derivatives for a truncated Taylor series by applying Eqs. (2), (3) and (5) always is more efficient than a Runge-Kutta method of the same order. With the fifth-order Taylor-series method the lower bound for the relative savings is 20%, as can be seen by dividing the absolute savings listed in Table II by the total cost listed in Table I.

PERFORMANCE OF IMPLEMENTED METHODS

A computer algorithm based on the formulas presented in the preceding sections has been implemented (Irvine 1986). This algorithm is referred to as ESSYNS, for *E*valuation and *S*imulation of *Syn*ergistic *S*ystems. Times required for solution of S-systems by ESSYNS and by RKF45 have been compared for many different problems. The focus in this section is on comparisons demonstrating the influence of accuracy (i.e. error tolerance) and problem size (i.e. the number of differential equations) on efficiency. The combination of these two types of results allow very general conclusions to be made about the advantages of the implemented Taylor-series method.

The influence of error tolerance on efficiency is difficult to state explicitly. This influence has been examined by solving problems for a wide range of error tolerances. For example, Table III shows the time required for solution of an S-system representing an immune network with n=3 (Irvine and Savageau 1985) by RKF45 and also by ESSYNS. These times are listed as a function of the negative base ten logarithm of the local error tolerance. Column R/E gives the ratio of these times. For this problem and range of error tolerance (10^{-2} to 10^{-12}) ESSYNS is between 3 and 45 times faster than RKF45. These results are representative of numerous other problems examined -- ESSYNS always is at least twice as fast as RKF45 and for stringent error tolerances can be more than two orders of magnitude faster.

Table III. Influence of Local Error Tolerance on Efficiency *

-log e	RKF45	ESSYNS	R/E
2	11.53	3.02	3.82
4	15.04	4.56	3.30
6	29.66	6.48	4.58
8	74.45	8.90	8.37
10	222.56	11.70	19.02
12	671.47	14.99	44.79

* Seconds required for solution of a representative S-system by RKF45 and by ESSYNS as a function of stringency of error tolerance. Stringency of error tolerance is the negative base ten logarithm of the local error tolerance (-log e). Column R/E represents the ratio of times for RKF45 and ESSYNS.

Although the minimum savings for one step by ESSYNS compared with one step by RKF45 was stated in Eq. (6) explicitly as a function of dimension of the system, this expression is a gross underestimate of the actual savings for ESSYNS. One major reason for this underestimation is that RKF45 typically requires more than one set of derivative evaluations for each step, whereas ESSYNS always takes one step for each set of derivative evaluations (Irvine 1986).

Numerous benchmark tests for systems of different sizes have been run in an attempt to determine the net influence of the dimension of the system, n, on the efficiency of ESSYNS compared with RKF45. The dimension of a system usually effects not only the size of the iterative loops within a given algorithm but also the stability, stiffness and rates of change for the problem. Due to these multiple effects it can be difficult to identify changes in efficiency that are strictly due to a change in scale of the problem. This confusion is avoided by examining a special set of problems in which the differential equation for each element of the system is independent of the other elements. With such a system, stiffness, stability and rates of change all are independent of dimension, and only the scale of the problem changes as n changes. For this reason systems of the following form have been selected:

$$X_i^{(1)} = \alpha_i - \beta_i X_i^{h_{ii}} \qquad i = 1...n \qquad (7)$$

where $\alpha_i=1$, $\beta_i=1$ and $h_{ii}=1.1$. Eq. (7) represents an uncoupled nonlinear system of dimension n.

Table IV gives times for RKF45 and for ESSYNS to solve the S-system given by Eq. (7). These times are listed as a function of the dimension of the system, n. The local error tolerance is 10^{-6} throughout. Column R/E gives the ratio of times. With an error tolerance of 10^{-6}, ESSYNS always is approximately 3 times faster than RKF45. With error tolerances of 10^{-3}, 10^{-9} and 10^{-12}, ESSYNS is 6, 8 and 27 times faster (results not shown). Because RKF45 is restricted to a step no larger than the report interval, even when a larger step could be taken without exceeding error tolerance, RKF45 is not any faster at an error tolerance of 10^{-3} than at 10^{-6}. But because ESSYNS can fill in points by interpolation, it is never bound by the report interval and is faster at an error tolerance of 10^{-3} than at 10^{-6}. This explains why the ratio of times for solutions by RKF45 and ESSYNS is greater at 10^{-3} than at 10^{-6}.

Table IV. Influence of Dimension of Problem on Efficiency *

n	RKF45	ESSYNS	R/E
2	9.56	3.24	2.95
4	17.91	5.93	3.02
6	26.25	8.62	3.05
8	34.55	11.26	3.07
10	42.89	13.89	3.09
12	51.25	16.53	3.10
14	59.54	19.28	3.09
16	67.88	21.86	3.11
18	76.23	24.55	3.11
20	84.53	27.24	3.10

* Seconds required for solution of a representative S-system by RKF45 and by ESSYNS as a function of the number of equations. These results were obtained by solving S-systems for uncoupled nonlinear systems with n equations. $\alpha_i=\beta_i=1$, $h_{ii}=1.1$, and all other parameters equal zero -- values for each element of the system initially were set equal to two, and solution proceeded from an initial time of 0 to a final time of 10 with points reported every 0.1 unit. The local error tolerance is 10^{-6} throughout. Column R/E gives the ratio of times for RKF45 and ESSYNS.

In the above comparisons, a sparse matrix method always is used to calculate first derivatives of S-systems regardless of the method used, ESSYNS or RKF45. The linear increase in time for solution by each method as a function of the number of equations (Table IV) demonstrates that both routines utilize the sparse method effectively. Without this method the increase in time would be quadratic in n. We chose to implement RKF45 with the sparse matrix method for S-systems so only irreducible differences between ESSYNS and RKF45 would be observed or, in other words, so only the advantage of techniques which cannot be implemented with RKF45 are displayed.

The advantage of the sparse matrix method for S-systems has been examined separately by solving problems with no g's or h's equal to zero. This allows us to draw more general conclusions about the efficiency of ESSYNS relative to conventional methods. To minimize the influence of differences in stability, stiffness or rates of change due to variation in dimension of the problem, equations that are virtually identical to those in Eq. (7) have been chosen. The only difference is that every g or h that is zero in Eq. (7) is set equal to 10^{-15}. Solutions for these problems do not differ appreciably from solutions for the original problem, but the time for solution of the second set increases as a quadratic in n since the problem no longer is sparse. Table V gives results for a tolerance of 10^{-6}. The poorer performance of each method in this Table compared with Table IV is attributable to loss of the sparse matrix method. The ratio of times for solution by RKF45 and by ESSYNS is given in column R/E. For the problems with nonsparse matrices and an error tolerance of 10^{-6}, ESSYNS is 3 times faster than RKF45 when n=2 and 5 times faster when n=20. Unlike with the sparse method, the relative efficiency of ESSYNS now increases with dimension of the problem. This result, together with the previous results for problems with sparse matrices and for problems with a wide range of error tolerance, supports the conclusion that the lower bound for the efficiency of ESSYNS relative to RKF45 is two, and that the advantage increases with size and complexity of the problem and with degree of accuracy required.

Table V. Influence of Dimension of Problem on Efficiency Under Nonsparse Conditions *

n	RKF45	ESSYNS	R/E
2	10.98	3.52	3.12
4	26.31	7.69	3.42
6	47.40	12.52	3.79
8	74.21	18.23	4.07
10	106.78	24.82	4.30
12	145.11	32.29	4.49
14	189.16	40.64	4.65
16	239.03	49.82	4.80
18	294.51	59.92	4.92
20	355.81	70.80	5.03

* Seconds required for solution of a representative nonsparse S-system (i.e. no parameters equal zero) by RKF45 and by ESSYNS as a function of the number of equations. These results were obtained by solving S-systems representing weakly coupled nonlinear systems with n equations. $\alpha_i=\beta_i=1$, $h_{ij}=1.1$, and all other parameters equal 10^{-15} -- other details are as outlined in Table IV.

Because many equations can be recast as S-systems, ESSYNS can be used to solve a wide variety of ordinary differential equations. There is, however, a cost associated with recasting equations as S-systems. Whenever this cost does not exceed the efficiency gained by solving with ESSYNS, solution of the problem in S-system form will be more efficient than conventional methods. An example is given here to demonstrate that substantial savings can be realized by recasting equations as S-systems and then solving with ESSYNS.

Consider a system of equations describing an unbranched metabolic pathway. If this pathway is assumed to have an initial reaction, V_0, modulated by endproduct inhibition and nine intermediate reactions, V_i, governed by Michaelis-Menten kinetics, the equations are as follows:

$$V_0 = Vm_0 X_0 K_0^{-1} K_I / (K_I + X_0 K_0^{-1} K_I + X_9)$$

$$V_i = Vm_i X_i / (X_i + K_i) \qquad i = 1...9$$

$$X_i^{(1)} = V_{i-1} - V_i \qquad (8)$$

where Vm_0, X_0, K_0, K_I, Vm_i and K_i are constants. If the following variables are introduced

$$X_{10} = (K_I + X_0 K_0^{-1} K_I + X_9)$$

$$X_j = (X_{j-10} + K_{j-10}) \qquad j = 11...19$$

Eq. (8) can be recast as the following S-system:

$$X_1^{(1)} = Vm_0 X_0 K_0^{-1} K_I X_{10}^{-1} - Vm_1 X_1 X_{11}^{-1}$$

$$X_i^{(1)} = Vm_{i-1} X_{i-1} X_{i+9}^{-1} - Vm_i X_i X_{i+10}^{-1} \qquad i = 2...9$$

$$X_{10}^{(1)} = X_9^{(1)} \qquad (9)$$

$$X_j^{(1)} = X_{j-10}^{(1)} \qquad j = 11...19$$

Since the number of dependent variables increases from 9 to 19, one might expect that solution of Eq. (9) would be less efficient than solution of Eq. (8). However, because Eq. (9) is an S-system it can be solved with ESSYNS, and the cost associated with recasting is more than recouped, as is shown in Table VI. The column labeled RKF45D represents the time for direct solution of Eq. (8) by RKF45 as a function of the negative base ten logarithm of error tolerance, the column labeled RKF45S represents the time for solution of Eq. (9) by RKF45, and the column labeled ESSYNS represents the time for solution of Eq. (9) by ESSYNS. The column labeled COST gives the cost of recasting, which is the ratio of times for RKF45S and RKF45D; and the column labeled BENEFIT gives the savings for the recast system solved with ESSYNS, which is the ratio of times for RKF45S and ESSYNS. For every error tolerance the cost of recasting is approximately two fold, but the benefit with ESSYNS always is greater. The greatest improvement in efficiency is seen with an error tolerance of 10^{-12} -- solution of the recast equations with ESSYNS is 10 times more efficient than direct solution of the original equations with RKF45. This example demonstrates that the cost of recasting can be small compared with the efficiency of solution gained by using ESSYNS, especially at stringent error tolerances.

Table VI. Cost and Benefit Associated with Solving Equations by Recasting into Equivalent S-System Form *

-log e	RKF45S	RKF45D	ESSYNS	COST	BENEFIT
2	67.40	29.44	22.90	2.29	2.94
4	78.10	34.94	30.09	2.24	2.60
6	149.45	65.08	48.22	2.30	3.10
8	374.59	157.25	71.90	2.38	5.21
10	1098.57	461.43	101.45	2.38	10.83
12	3350.84	1385.94	137.47	2.42	24.38

* Times for solution of a nonlinear system of ordinary differential equations comprised of rational functions as described in the text. Solutions were obtained by three different methods: (1) Solution of the equivalent S-system form of equations with RKF45 (RKF45S), (2) Solution of original equations directly with RKF45 (RKF45D) and (3) solution of S-system form with ESSYNS. The column labeled COST is the ratio of times for RKF45S and RKF45D, and the column labeled BENEFIT is the ratio of times for RKF45S and ESSYNS.

Because solution of S-systems with ESSYNS is at least twice as fast as solution with RKF45, we expect that when recasting does not more than double the number of operations for the recast system, solution in S-form with ESSYNS will be faster than direct solution with RKF45. For such problems we expect that ESSYNS will be one to two orders of magnitude faster at stringent error tolerances. Even for recast problems which more than double the number of operations, ESSYNS still will be faster when high accuracy is required.

A novel approach to numerical solution of ordinary differential equations has been presented. The advantages are high efficiency and wide-spread applicability. The approach is particularly well suited for solution of large, complex systems requiring high accuracy. The advantages of the Taylor-series method presented here are offset somewhat by a computational cost associated with recasting into S-system canonical form and, at least in the current implementation, by a limitation to nonstiff and moderately stiff systems. These limitations are expected to diminish as systematic methods are developed to identify and recast into S-systems that minimize cost, and as methods are incorporated to detect and treat stiffness.

REFERENCES

Butcher, J. C. (1965), On the Attainable Order of Runge-Kutta Methods, Math. Comp., 19, p. 408.

Fehlberg, E. (1969), Low Order Classical Runge-Kutta Formulas with Step-Size Control and Their Application to Some Heat Transfer Problems, NASA Tech. Rep. R-315, Huntsville, Ala.

Hull, T. E., W. H. Enright, B. M. Fellen and A. E. Sedgwick (1972), Comparing Numerical Methods for Ordinary Differential Equations, SIAM J. Numer. Anal., 9, p. 603.

Irvine, D. H., and M. A. Savageau (1985), Network Regulation of the Immune Response: Alternative Control Points for Suppressor Modulation of Effector Lymphocytes, J. Immunol., 134, p. 2100.

Irvine, D. H. (1986), *S-System Analysis of Organizationally Complex Systems: Network Regulation of the Immune Response*, Thesis Dissertation, The University of Michigan, Ann Arbor, MI.

Rice, J. R. (1983), *Numerical Methods, Software, and Analysis*, McGraw-Hill, New York, p. 265.

Savageau, M. A. (1969a), Biochemical Systems Analysis I. Some Mathematical Properties of the Rate Law for the Component Enzymatic Reactions, J. Theor. Biol. 25, p. 365.

Savageau, M. A. (1969b), Biochemical Systems Analysis II. The Steady-State Solutions for an *n*-Pool System using a Power-Law Approximation, J. Theor. Biol. 25, p. 370.

Savageau, M. A. (1970), Biochemical Systems Analysis III. Dynamic Solutions using a Power-Law Approximation, J. Theor. Biol. 26, p. 215.

Savageau, M. A. (1971a), Parameter Sensitivity as a Criterion for Evaluating and Comparing the Performance of Biochemical Systems, Nature 229, p. 542.

Savageau, M. A. (1971b), Concepts Relating the Behavior of Biochemical Systems to Their Underlying Molecular Properties, Arch. Biochem. Biophys. 145, p. 612.

Savageau, M. A. (1976), *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*, Addison-Wesley, Reading, Mass.

Savageau, M. A. (1979a), Growth of Complex Systems Can Be Related to the Properties of Their Underlying Determinants, Proc. Nat. Acad. Sci. USA 76, p. 5413.

Savageau, M. A. (1979b), Allometric Morphogenesis of Complex Systems: Derivation of the Basic Equations from First Principles, Proc. Nat. Acad. Sci. USA 76, p. 6023.

Savageau, M. A. (1982), A Suprasystem of Probability Distributions, Biometrical J. 24, p. 323.

Savageau, M. A. (1985), Mathematics of Organizationally Complex Systems, Biomed. Biochim. Acta. 44, p. 839.

Savageau, M. A., and E. O. Voit. (1986), Recasting Nonlinear Differential Equations as S-systems: A Canonical Nonlinear Form, (submitted to SIAM J. Math. Anal.)

Shampine, L. F., and H. A. Watts (1977), Subroutine RKF45 published in *Computer Methods for Mathematical Computation*, Prentice-Hall, Englewood Cliffs, NJ, p. 129.

Shampine, L. F., H. A. Watts and S. M. Davenport (1976), Solving Nonstiff Ordinary Differential Equations - The State of the Art, SIAM Review, 18, p. 376.

Voit, E. O., and M. A. Savageau (1986), Equivalence between S-Systems and Volterra-Systems, Math. Biosci., 78, p. 47.