WIS-CS-198-73
COMPUTER SCIENCES DEPARTMENT
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

Received December 11, 1973

COMPUTATION OF CONTINUOUS APPROXIMATE SOLUTIONS TO ORDINARY DIFFERENTIAL EQUATIONS BY A SIMPLIFICATION OF PICARD'S METHOD OF SUCCESSIVE SUBSTITUTIONS

by

Luis Legarreta

Computer Sciences Technical Report #198

December 1973

COMPUTATION OF CONTINUOUS APPROXIMATE SOLUTIONS TO ORDINARY DIFFERENTIAL EQUATIONS BY A SIMPLIFICATION OF PICARD'S METHOD OF SUCCESSIVE SUBSTITUTIONS

by

Luis Legarretta

ABSTRACT

The computation of continuous approximate solutions of Differential Equations has become increasingly important in order, for instance, to be able to apply error bounding techniques from Functional Analysis. An efficient procedure for computing continuous approximate solutions to initial value problems in ordinary differential equations is presented. The method is a simplification of Picard's method of successive substitutions.

INTRODUCTION.

The computation of continuous approximate solutions of differential equations has become increasingly important in order, for instance, to be able to apply error bounding techniques from Functional Analysis [7].

Picard's method of successive approximations for the solution of ordinary differential equations is one of considerable conceptual importance. However it has not been applied directly to the computer solution of such equations. The apparent necessity to process all the coefficients of a power series in the independent variable makes it seem a slow and memory consuming process. On the other hand, an analysis of the behaviour of the coefficients in the successive approximations to the solutions has shown the following property: It is not necessary to store and manipulate all the coefficients of the kth approximation in order to compute the (k+1)th approximation; the integration need be performed only on the term of degree k of the power series expansion of the equation. A simplified version of Picard's method is derived which makes use of this property.

Application of the resulting method is illustrated and techniques for error analysis are discussed. The method is then compared with a fourth order Runge-Kutta method from the point of view of actual machine computations. A final example shows how the method is useful in obtaining global error bounds on solutions to boundary value problems.

1.- DESCRIPTION OF THE METHOD.

Consider the first order differential equation

$$x'(t) = f(x,t) \tag{1.1a}$$

and the initial condition

$$x(0) = x_0 \tag{1.1b}$$

The proposed method for the approximate solution of (1.1) can be recursively defined as follows:

$$y_0 = x_0 \tag{1.2a}$$

$$y_0' = f(x_0,t) \tag{1.2b}$$

$$y_j = \int_0^t y'_{j-1}\, dt \tag{1.2c}$$

where y'_{j-1} is the term of degree j-1 in the expansion of $f(y^{(j-1)},t)$ as a power series in t and

$$y^{(j)} = y_0 + y_1 + \dots + y_j \tag{1.3}$$

Clearly the expression $y^{(j)}$ in (1.3) is a polynomial in t.
There is no difficulty in extending the method to systems of equations; so that x and f in (1.1) may be regarded as vector valued functions.

The main result of this paper is the following.

Theorem.

The polynomial $y^{(n)}(t)$ is identical to the first n+1 terms of the series representation of the solution of (1.1) computed by Picard's method of successive approximations.

The proof (given in the remainder of this section) follows by induction. First it will be shown that the constant and linear terms in (1.3) are equal to the corresponding terms computed using Picard's method.

The successive approximations to the solution of (1.1) can be computed by Picard's method from [1,2]

$$x_0(t) = x_0$$

$$x_m(t) = x_0 + \int_0^t f(x_{m-1}(s),s)\, ds \quad ; \quad 0 \le t \le T \quad ; \quad m=1,2,\dots \tag{1.4}$$

Convergence of the Picard sequence x_m to a solution x(t) of (1.1) is guaranteed if f(x,t) satisfies the Lipschitz condition

$$\max_{[0,T]} |f(x_1(t),t) - f(x_2(t),t)| \leq M \|x_1 - x_2\| \tag{1.5}$$

where x_1 and x_2 are continuously differentiable functions defined in the interval [0,T].*

The solution of (1.1) computed by application of Picard's method can be written in the form

$$x(t) = p_0 + p_1 t + p_2 t^2 + \ldots + p_n t^n + \ldots \tag{1.6}$$

The first approximation to the solution of (1.1) is:

$$x_1(t) = x_0 + \int_0^t f(x_0,s)ds \tag{1.7}$$

The expansion of $f(x_0,s)$ as a power series in s can be integrated to obtain

$$x_1(t) = x_0 + c_1 t + c_2 t^2 + \ldots \tag{1.8}$$

The constant term x_0 remains invariant in all successive approximations $x_m(t)$; m=2,3,... . The equality

$$y_0 = p_0 \tag{1.9}$$

follows from (1.6) and the definition (1.2a) of y_0.

Evaluation of equation (1.2c) with j=1 shows that

$$y_1 = c_1 t \tag{1.10}$$

by direct comparison with (1.7) and (1.8).

An interesting property is discovered in the expansion of functions with polynomials as arguments. The following lemma will be used to show the invariance of the linear term $c_1 t$ of (1.8) in all the elements x_m ; m=2,3,... of the Picard sequence.

* See section 3 for discussion of an extension to functions which do not satisfy (1.5)

Lemma 1.

Suppose the function $f(z(t),t)$ can be expanded as a power series in t, then the term of degree k of the argument

$$z(t) = z_0 + z_1 t^1 + z_2 t^2 + \dots + z_k t^k + \dots \tag{1.11}$$

of the function contributes only to the values of terms of equal or higher degree in the resulting expansion of the function as a power series in t.

The property is easily verified for the elementary algebraic operations with polynomials as arguments [3,6]. Given the polynomials:

$$\begin{aligned} u &= \sum_i u_i t^i \\ v &= \sum_i v_i t^i \qquad ; \quad i = 0,1,2,\dots \\ w &= \sum_i w_i t^i \end{aligned} \tag{1.12}$$

the addition $w = u + v$ can be expressed as:

$$w_k = u_k + v_k \qquad ; \quad k = 0,1,2,\dots \tag{1.13}$$

the product $w = u * v$ is given by:

$$w_k = \sum_{i=0}^{k} u_i * v_{k-i} \qquad ; \quad k = 0,1,2,\dots \tag{1.14}$$

and finally the quotient $w = u / v$ can be computed from:

$$\begin{aligned} w_0 &= u_0 / v_0 \\ w_k &= \frac{u_k - \sum_{i=1}^{k} v_i * w_{k-i}}{v_0} \qquad ; \quad k = 1,2,\dots \end{aligned} \tag{1.15}$$

Equations (1.13), (1.14) and (1.15) show the stated property.

It follows from Lemma 1 and the invariance of the constant term x_0 that

$$c_1 = p_1 \tag{1.16}$$

From (1.9), (1.10) and (1.16) it is easily seen that the equality

$$y_j = p_j t^j \tag{1.17}$$

is satisfied for $j = 0,1$.

Assume (1.17) holds for $j = 0,1,2,\ldots,n-1$. Substitution of the solution $x(t)$ in equation (1.4) followed by the expansion of $f(x(s),s)$ as a power series in s results in

$$x(t) = x_0 + \int_0^t (d_0 + d_1 s + \ldots + d_{n-1}s^{n-1} + d_n s^n + \ldots)\, ds \qquad (1.18)$$

which after integration becomes

$$x(t) = p_0 + p_1 t + \ldots + p_n t^n + \ldots \qquad (1.19)$$

where

$$p_0 = x_0$$
$$p_k = \frac{d_{k-1}}{k} \quad ; \quad k = 1,2,\ldots \qquad (1.20)$$

It is important to note that Lemma 1 guarantees that the constant d_{n-1} in (1.18) depends only on the coefficients p_i ; $i \leq n-1$ of $x(t)$ and the particular form of $f(x,t)$.

The n-1 iterations carried out using equations (1.2) result in the expression

$$y^{(n-1)}(t) = y_0 + y_1 + \ldots + y_{n-1} \qquad (1.21)$$

where the terms y_i satisfy

$$y_i = p_i t^i \quad ; \quad i = 0,1,2,\ldots,n-1 \qquad (1.22)$$

by the inductive assumption. The variable y'_{n-1} is, by definition, the term of degree n-1 in the expansion of $f(y^{(n-1)}(t),t)$ as a power series in t. Lemma 1 and the inductive assumption (1.22) guarantee

$$y'_{n-1} = d_{n-1} t^{n-1} \qquad (1.23)$$

Substitution of (1.23) in (1.2c) yields

$$y_n = \frac{d_{n-1}}{n} t^n \qquad (1.24)$$

The desired result

$$y_n = p_n t^n \qquad (1.25)$$

follows by direct comparison of (1.24) and (1.20).

This completes the proof of the theorem.

2.- APPLICATIONS.

The computation of the solution to ordinary differential equations using the described method requires considerably less effort than direct use of Picard's method. In the case of linear equations with constant coefficients, only the term of degree k of the argument contributes to the value of the term of equal degree in the expansion of the function as a power series. Therefore, only one evaluation of the function is needed in every iteration. It follows from equations (1.14) and (1.15) that nonlinear operations require a greater number of machine operations, as well as the storage of the polynomial representation of the operands. The power series expansion of transcendental functions is achieved using Taylor's series. Since the arguments of the functions are polynomials, the process is very inefficient. It is often possible, however, to avoid the use of transcendental functions by expressing them as solutions to auxiliary rational differential equations [6, chapter 11].

In the following examples, the solutions will be carried out using the direct application of Picard's method and the proposed simplification.

Consider the linear differential equation

$$x' = 2x \tag{2.1a}$$

subject to the initial condition

$$x(0) = 1 \tag{2.1b}$$

Simplification of Picard's method.	Picard's method.
$y_0 = 1$	$x_0 = 1$
$y_1 = \int_0^t 2\,dt$	$x_1(t) = 1 + \int_0^t 2\,ds$
$y_1 = 2t$	$x_1(t) = 1 + 2t$
$y_2 = \int_0^t 4t\,dt$	$x_2(t) = 1 + \int_0^t (2 + 4s)\,ds$
$y_2 = 2t^2$	$x_2(t) = 1 + 2t + 2t^2$

$$y_3 = \int_0^t 4t^2\,dt \qquad x_3(t) = 1 + \int_0^t (2 + 4s + 4s^2)\,ds$$

$$y_3 = \frac{4}{3}t^3 \qquad x_3(t) = 1 + 2t + 2t^2 + \frac{4}{3}t^3$$

The result

$$x_3(t) = \sum_{i=0}^{3} y_i$$

can be verified by inspection.

In order to illustrate a nonlinear example, consider the equation

$$x' = x^2 \tag{2.2a}$$

and the initial condition

$$x(0) = 1 \tag{2.2b}$$

Simplification of Picard's method.

$$y_0 = 1$$

$$y_1 = \int_0^t dt$$

$$y_1 = t$$

$$y_2 = \int_0^t 2t\,dt$$

$$y_2 = t^2$$

$$y_3 = \int_0^t 3t^2\,dt$$

$$y_3 = t^3$$

Picard's method.

$$x_0 = 1$$

$$x_1(t) = 1 + \int_0^t ds$$

$$x_1(t) = 1 + t$$

$$x_2(t) = 1 + \int_0^t (1 + 2s + s^2)\,ds$$

$$x_2(t) = 1 + t + t^2 + \frac{1}{3}t^3$$

$$x_3(t) = 1 + \int_0^t (1+2s+3s^2+\frac{8}{3}s^3+\frac{5}{3}s^4+\frac{2}{3}s^5+\frac{1}{9}s^6)\,ds$$

$$x_3(t) = 1 + t + t^2 + t^3 + \frac{2}{3}t^4 + \frac{1}{3}t^5 + \frac{1}{9}t^6 + \frac{1}{63}t^7$$

The solution computed using the simplification of Picard's method

$$y^{(3)}(t) = \sum_{i=0}^{3} y_i$$

is identical to the sum of the terms of degree less than four in the solution computed by direct application of Picard's method.

3.- ACCURACY OF NUMERICAL RESULTS.

The polynomial

$$y^{(n)}(t) = \sum_{i=0}^{n} p_i t^i \tag{3.1}$$

computed using equations (1.2) is a truncation of the infinite series

$$x(t) = \sum_{i=0}^{\infty} p_i t^i \tag{3.2}$$

which represents the exact solution of equation (1.1) in the interval

$$0 \leqslant t \leqslant T \tag{3.3}$$

with T defined in (1.5).

Bounds on the magnitude of the error introduced by truncation of the series (3.2) will now be discussed.

By the same reasoning used in the proof of the theorem it can be shown that the element x_m of the Picard sequence generated with equations (1.4) can be written in the form

$$x_m(t) = \sum_{i=0}^{m} p_i t^i + \sum_{i=m+1}^{\infty} q_i t^i \tag{3.4}$$

where the q_i depend on the number of iterations m, the particular form of the function f(x,t) and the initial condition x(0). From the derivation of the error bound [2] *.

$$\|x(t) - x_m(t)\| \leqslant \left(e^{MT} - \sum_{k=0}^{m-1} \frac{M^k T^k}{k!}\right) \|x_1(t) - x_0\| \tag{3.5}$$

and the particular form (3.4) of $x_m(t)$, it is not difficult to show that the expression

$$\|x(t) - y^{(m)}(t)\| \leqslant \left(e^{MT} - \sum_{k=0}^{m-1} \frac{M^k T^k}{k!}\right) \|x_1(t) - x_0\| + \left\| \sum_{i=m+1}^{\infty} q_i t^i \right\| \tag{3.6}$$

constitutes a bound on the error of the truncated series (3.1) computed using equations (1.2).

* The symbol $\| \ \|$ denotes the norm

$$\|z(t)\| = \max_{t \in [0,T]} |z(t)|$$

The inequality (3.6) suggests two techniques to reduce the magnitude of the error: a) Computation of a polynomial of degree higher than m. b) Reduction of the length of the interval (3.4).

The degree of the computed polynomial (3.2) is limited by the hardware of the machine used, since the coefficients of higher powers of t tend to cause underflow or overflow in the computer. The question of efficiency becomes the limiting factor when dealing with nonlinear equations. The number of machine operations required to compute the nonlinear terms grows with the square of the degree of the resulting polynomial.

Clearly, the error term (3.6) in the approximate solution $y^{(m)}(t)$ vanishes as $T \to 0$. Let the computed polynomial $y^{(m)}(t)$ represent the solution within the interval

$$0 \leqslant t \leqslant h \tag{3.7}$$

where $h = T/N$. The error in the approximate solution $y^{(m)}_{[0]}(t)$* can be made arbitrarily small within the interval (3.7) by choosing a large enough value of N. The value $y^{(m)}_{[0]}(h)$ can now be calculated and used as the initial value to compute a new polynomial $y^{(m)}_{[1]}(t)$ using equations (1.2) with $y(0) = y^{(m)}_{[0]}(h)$. The polynomial $y^{(m)}_{[1]}(s)$; $0 \leqslant s \leqslant h$ represents the solution of (1.1) in t he interval $h \leqslant t \leqslant 2h$. The procedure can be repeated N times to obtain the solution for the complete interval (3.3). The procedure described is usually called analytic continuation. It can also be used in some cases to extend the solution for values of the independent variable outside the interval (3.3). The approximate solution y(t) within the complete interval (3.3) is given as a set of N polynomials of degree m as follows:

$$y(t) = y^{(m)}_{[k]}(s) \quad ; \quad 0 \leqslant s \leqslant h \quad ; \quad kh \leqslant t \leqslant (k+1)h \quad ; \quad k = 0,1,\ldots,N-1 \tag{3.8}$$

as shown in figure 1.

The error bound given by equation (3.6) may be extended to the case that the Lipschitz condition (1.5) is satisfied for x_1 and x_2 contained

* The subscript within brackets represents the sub-interval in which the polynomial represents the solution of the equation.

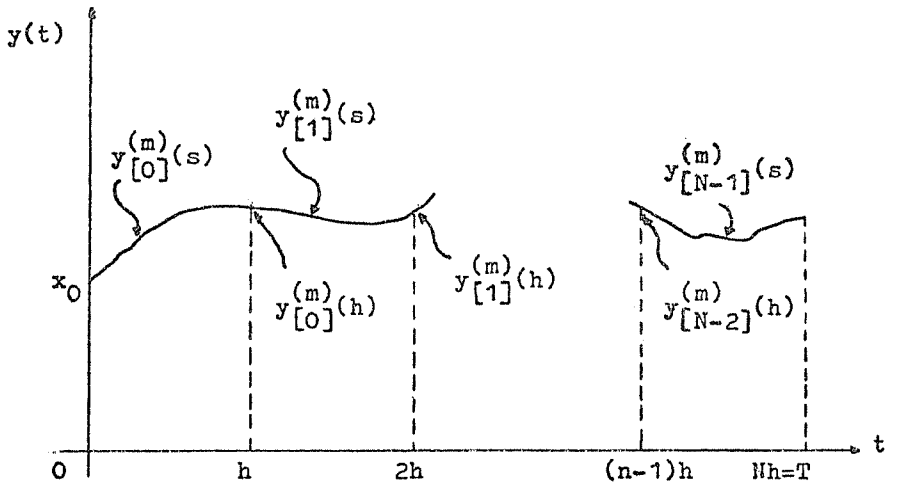

Figure 1.

in a ball

$$u(x_0,r) = \{x : \| x - x_0 \| \leq r\} \tag{3.9}$$

In this case the Lipschitz constant M depends on r and T. Convergence of the iterations (1.2) to the solution of (1.1) is then guaranteed and the error bound (3.6) holds if there exist values of T and r for which the inequality

$$r \geq e^{MT} \| x_1(t) - x_0 \| \tag{3.10}$$

is satisfied.

For the particular case where the Lipschitz constant $M = \theta < 1$ the error bound

$$\| x(t) - y^{(m)}(t) \| \leq \frac{\theta}{1-\theta} \| x_1(t) - x_0 \| + \| \sum_{i=m+1}^{\infty} q_i t^i \| \tag{3.11}$$

is derived by direct application of the Contraction Mapping Principle [2].

4.- CONCLUSIONS.

The method presented in its analytic continuation form may be thought of as a variable order method for the solution of initial value problems in ordinary differential equations. The order of the method is given by the degree of the polynomial which represents the solution over one step and it can be specified by a parameter in the computer program [5].

The most important advantage of the method lies in the fact that it is an efficient procedure to compute continuous approximate solutions to initial value problems in ordinary differential equations. The analytic continuation scheme described in section 3 represents the solution in the form of piecewise polynomial functions of any desired degree. This allows the possibility of a stable differentiation algorithm. The kth derivative of the m degree polynomial representing the solution yields a polynomial of degree m-k. The degree is limited only by the hardware restrictions of the computer.

A digital-analog simulator program named SAS using the described method has been implemented in the Burroughs B5500, B6500 and Univac 1108 computers [3,4,5]. Table 1 shows the results of the comparison between SAS and the fourth order Runge-Kutta method described in [9,page 367] for the following problems:

a) $y'' = -200y' - 10^{-6}y$
$y(0) = 10 \quad ; \quad y'(0) = 0$

b) $y'' = -y + e^{t}$
$y(0) = 0 \quad ; \quad y'(0) = 0$

c) $y'' = -0.025y^{2} + 9.81$
$y(0) = 0 \quad ; \quad y'(0) = 0$

The maximum error was determined by comparison with the analytic solutions evaluated using double precision arithmetic. The programs are written in NUALGOL [10] and were run on the Univac 1108 computer.

	Problem a)		Problem b)		Problem c)	
	SAS	Runge	SAS	Runge	SAS	Runge
Maximum error.	10^{-6}	10^{-4}	10^{-6}	10^{-5}	10^{-4}	10^{-4}
CPU time millisec.	137.4	498.4	13.5	64.4	312.6	90.2
Step size.	.004	.0001	1.5	1/15	1.0	0.2
No. of steps.	5	200	1	15	8	40
Degree of polynomials.	20	—	11	—	10	—

Table 1.

The CPU times shown in the table do not include the time invested in the evaluation of the polynomials for the values of the independent variable needed for comparison. The results indicate the method's efficiency in the solution of linear differential equations.

A comparison of the third derivatives of the functions computed by SAS and the double precision evaluation of the analytic third derivatives is summarized in table 2. The step size, number of steps and degree of the polynomials are those shown in table 1.

Problem	a)	b)	c)
Maximum error in computed y'''	1400.+	10^{-3}	10^{-5}
CPU time millisec.	164	20.2	395.4

+ The solution is of order 10^{8}.
The relative error is less than 10^{-5}.

Table 2.

The method has been used successfully to compute the approximate solutions of the 'intermediate' linear boundary value problems resulting from the application of Newton's method [2,8] for the solution of certain nonlinear boundary value problems.

The application of Newton's method in Banach spaces [2,8] to find an approximate solution of the nonlinear boundary value problem

$$x''(t) = f(f(x,t)$$
$$x(0) = x(1) = 0 \qquad (4.1)$$

requires the solution of the sequence of linear boundary value problems given by

$$x''_{k+1}(t) = f_x(x_k(t),t)x_{k+1}(t) + f(x_k(t),t) - f(x_k(t),t)x_k(t)$$
$$x_{k+1}(0) = x_{k+1}(1) = 0 \quad ; \quad k = 0,1,2,... \qquad (4.2)$$

where $f_x(x_k(t),t)$ is the partial derivative of the function with respect to x evaluated at $x_k(t)$. In order to apply Kantorovic's convergence theorem and error bound for the Newton sequence $x_k(t)$; k = 1,2,... starting from an initial guess $x_0(t)$, it is necessary to compute continuous twice differentiable approximate solutions to the sequence of linear differential equations (4.2). If problem (4.2) has a unique solution, it may be expressed as a linear combination of solutions to two related initial value problems: Aparticular solution of (4.2) and a solution to the corresponding homogeneous equation [8]. The simplification of Picard's method can be used to compute the continuous approximate solutions of the related initial value problems. The linear combination of the computed approximate solutions yields the desired continuous twice differentiable functions which represent the approximate solutions of the sequence (4.2). A modified version of the program SAS has been used to compute approximate solutions to boundary value problems of the form (4.1). A summary of the results obtained in the computation of the approximate solution to the problem

d) $$x'' = e^{-x}$$
$$x(0) = x(1) = 0$$

is shown in table 3. It is known thatproblem d) has exactly two solutions: The 'small' solution which attains a minimum value of approximately -0.14 and the 'big' solution which attains a minimum value of approximately -4.09. The columns of table 3 are labeled accordingly.

	'small' solution	'big' solution
Maximum error.	10^{-7}	10^{-2}
CPU time millisec.	600	874
No. of Newton iterations.	3	2
Step size.	0.5	0.1
No. of steps.	2	10
Degree of polynomials.	5	5

Table 3.

ACKNOWLEDGMENTS.

The author wishes to express his gratitude to Dr. Ramon E. Moore for his valuable suggestions and encouragement throughout the development of this paper. Thanks are also due to Dr. Enrique Chicurel for introducing the author to the study of Picard's method. Finally, thanks to the Consejo Nacional de Ciencia y Tecnologia of the Mexican Government for their financial support.

REFERENCES

1.- E.A. Coddington & N. Levinson, Theory of Ordinary Differential Equations, Mc. Graw Hill, 1955.

2.- L.B. Rall, Computational Solution of Nonlinear Operator Equations, John Wiley & Sons, 1969.

3.- L. Legarreta, Lenguaje Simulador Analogico-digital SAS, Mexico, 1970.

4.- E. Chicurel & L. Legarreta, SAS, Simulador Analogico en Series, Memoria Conferencia Internacional IEEE Sobre Sistemas, Redes y Computadoras, Vol. 1, Jan., 1971

5.- E. Chicurel & L. Legarreta, Manual para el uso del Simulador Analogico SAS II (Simulador Analogico en Series), Instituto de Ingenieria, UNAM, Mexico, Oct., 1971.

6.- R. E. Moore, Interval Analysis, Prentice Hall, 1966.

7.- R. E. Moore, Functional Analysis for Computers, Funktionalanalytische Methoden der Numerichen Mathematik, L.Collatz & H. Unger, ed., Birkhauser Verlag, Basel und Stattgart, 1969.

8.- T.D. Talbot, Guaranteed Error Bounds for Computed Solutions of Nonlinear Two-point Boundary Value Problems, MRC Technical Summary Report No. 875, Mathematics Research Center, University of Wisconsin, Madison, 1966.

9.- B. Carnahan, H.A. Luther & J.O Wilkes, Applied Numerical Methods, John Wiley & Sons, 1969.

10.- Univac 1100 series NUALGOL (norwegian University ALGOL) Programmer Reference Manual, Sperry Rand Corp., 1971.

BIBLIOGRAPHIC DATA SHEET	1. Report No. WIS-CS-198-73	2.	3. Recipient's Accession No.

4. Title and Subtitle

COMPUTATION OF CONTINUOUS APPROXIMATE SOLUTIONS TO ORDINARY DIFFERENTIAL EQUATIONS BY A SIMPLIFICATION OF PICARD'S METHOD OF SUCCESSIVE SUBSTITUTIONS

5. Report Date: December 1973

6.

7. Author(s): Luis Legarreta

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

COMPUTER SCIENCES DEPARTMENT
University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

The computation of continuous approximate solutions of Differential Equations has become increasingly important in order, for instance, to be able to apply error bounding techniques from Functional Analysis. An efficient procedure for computing continuous approximate solutions to initial value problems in ordinary differential equations is presented. The method is a simplification of Picard's method of successive substitutions.

17. Key Words and Document Analysis. 17a. Descriptors

Ordinary Differential Equations
Picard's Method of Successive Substitutions
Functional Analysis
Norm
Error Bound

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Available to the public	UNCLASSIFIED	18
	20. Security Class (This Page) UNCLASSIFIED	22. Price