AN ABSTRACT OF THE THESIS OF

James Frank Lathrop for the Master of Science in Mathematics
(Name) (Degree) (Major)

Date thesis is presented May 7, 1963

Title STABILITY OF NUMERICAL INTEGRATION OF ORDINARY DIFFERENTIAL EQUATIONS

Abstract approved Redacted for privacy
(Major professor)

The thesis discusses stability of procedures based on linear computing formulas for numerical integration of an ordinary first-order differential equation. The theorems are proved: (1) If the procedure is asymptotically stable it is stable for small positive step size if the Lipschitz number is negative; (2) Relative stability always exists if asymptotic stability does; (3) If the Lipschitz constant is positive, there is an integration procedure based on a linear computing formula of order one, which is, however, not asymptotically stable. An algorithm for the general case is included, written in the Algol 60 language.

STABILITY OF NUMERICAL INTEGRATION OF ORDINARY DIFFERENTIAL EQUATIONS

by

JAMES FRANK LATHROP

A THESIS

submitted to

OREGON STATE UNIVERSITY

in partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

June 1963

APPROVED:

Redacted for privacy

Professor of Mathematics

In Charge of Major

Redacted for privacy

Head of Department of Mathematics

Redacted for privacy

Dean of Graduate School

Date thesis is presented May 7, 1963

Typed by Carol Baker

TABLE OF CONTENTS

STABILITY OF NUMERICAL INTEGRATION OF ORDINARY DIFFERENTIAL EQUATIONS

CHAPTER I

RESTRICTIONS ON THE METHOD AND STABILITY

A common method of solving numerically an ordinary differential equation of the form

A) $y'(x) = f(x, y)$

B) $y(x_0) = y_0$,

is to calculate the approximate values of the function by a procedure based on the formula

$$y_{n+1} = \sum_{i=0}^{P} a_i y_{n-i} + h \sum_{i=0}^{Q} b_i y'_{n-i} + E,$$

where

$$y_k = y(x_0 + k \cdot h)$$
$$y'_k = y'(x_0 + k \cdot h)$$

and E is the error. The order of the formula is the highest degree polynomial for which E is zero in the formula. In the following I have attempted to show that under the normal stability requirements there is no formula that is stable for all differential equations, but if the Lipschitz constant of the function $f(x, y)$ is negative at each x then a formula can be determined to give a stable solution.

The formula

1) $$y_{n+1} = \sum_{i=0}^{P} a_i y_{n-i} + h \sum_{i=0}^{Q} b_i y'_{n-i}$$

is usually required to be exact for polynomials of degree "0" and "1", which gives the following two conditions on the coefficients:

2) $$\sum_{i=0}^{P} a_i = 1$$

3) $$1 + n - \sum_{i=0}^{P} a_i (n-i) = \sum_{i=0}^{Q} b_i$$

The stability of this method of solution requires that the zeros of the polynomial

4a) $$G(\beta) = \beta^{n+1} - \sum_{i=0}^{P} a_i \beta^{n-i} - Lh \sum_{i=0}^{Q} b_i \beta^{n-i},$$

where L = Lipschitz constant of $f(x, y)$, lie in the unit circle $|\beta| < 1$ with the exception of a zero of multiplicity no greater than 1 at $\beta = 1$.

To insure the latter we may require that

4b) $$\frac{dG}{d\beta}\Big|_{\beta=1} \neq 0.$$

Now

5) $$\frac{dG}{d\beta} = (n+1)\beta^n - \sum_{i=0}^{P} a_i (n-i)\beta^{n-1-i} - hL \sum_{i=0}^{Q} b_i (n-i)\beta^{n-1-i}$$

and

5a) $$\frac{dG}{d\beta}\Big|_{\beta=1} = (n+1) - \sum_{i=0}^{P} a_i(n-i) - hL\sum_{i=0}^{Q} b_i(n-i).$$

Since we are concerned with solutions near $h = 0$ and we have continuity of (4a) in a neighborhood of $h = 0$, then (4b) and (5a) give us, setting $h = 0$,

5b) $$n + 1 - \sum_{i=0}^{P} a_i(n-i) \neq 0.$$

And from (3) this is equivalent to

6) $$\sum_{i=0}^{Q} b_i \neq 0.$$

CHAPTER II

LIMITATIONS OF STABILITY

If we have a method of solution which satisfies the stability criteria at $h = 0$ with one zero at $\beta = 1$ and all the remaining zeros lying within the unit circle, we need a guarantee that the zero at $\beta = 1$ does not move outside the unit circle as h increases from $h = 0$.

At a zero of $G(\beta)$ we have

$$\beta^{n+1} - \sum_{i=0}^{R} (a_i + hLb_i)\beta^{n-i} = 0,$$

where $R = \max(P, Q)$.

Assuming a_i, b_i, L to be fixed, consider β as a function of h. Then, differentiating once with respect to h we get

$$(n+1)\beta^n \frac{d\beta}{dh} = \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-i-1} \frac{d\beta}{dh} + \sum_{i=0}^{R} Lb_i\beta^{n-i}$$

or

$$\frac{d\beta}{dh}\left[(n+1)\beta^n - \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-1-i}\right] = L\sum_{i=0}^{R} b_i\beta^{n-i} .$$

Under the assumption that

$$(n+1)\beta^n - \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-1-i} \neq 0$$

we have

$$7) \qquad \frac{d\beta}{dh} = \frac{L\cdot \sum_{i=0}^{R} b_i \beta^{n-i}}{(n+1)\beta^n - \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-1-i}} .$$

And we have seen that $(n+1)\beta^n - \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-1-i} = 0$ at $h = 0$ only when we have a double zero of the polynomial $G(\beta)$. So we have the existence of $\frac{d\beta}{dh}$ in a neighborhood of $h = 0$, $\beta = 1$, and so we have

$$\left.\frac{d\beta}{dh}\right|_{\substack{h=0\\ \beta=1}} = L\cdot \frac{\sum b_i}{(n+1) - \sum_{i=0}^{R} a_i (n-i)} ,$$

which from (3) reduces to

$$8) \qquad \left.\frac{d\beta}{dh}\right|_{\substack{h=0\\ \beta=1}} = L.$$

Therefore the zero at $\beta = 1$ moves inside the unit circle only if the Lipschitz constant L is negative, and if it is positive the zero moves outside the unit circle giving an unstable solution.

If we have $L < 0$ the question arises; what happens to the other zeros as h increases from zero? If β is a continuous function of h, we can make a slight increase in h without moving these points outside the unit circle. From (7) we see that $\frac{d\beta}{dh}$ exists at $h = 0$ if

$$(n+1)\beta^{n} - \sum_{i=0}^{R} (a_i + hLb_i)(n-i)\beta^{n-1-i} \neq 0.$$

And this happens if $G(\beta)$ has no multiple zeros. So we need only require that there be no multiple zeros of $G(\beta)$ other than $\beta = 0$ since we can see from (4) that the zeros at $\beta = 0$ are not affected by a change in h.

These results can be stated in a theorem as follows:

Theorem 1: A procedure for solving the equation (A) with initial conditions (B) based on the equation

$$y_n = \sum_{i=0}^{P} a_i y_{n-i} + h \sum_{i=0}^{Q} b_i y'_{n-i} ,$$

where

$$\sum_{i=0}^{P} a_i = 1, \quad 1+n - \sum_{i=0}^{P} a_i (n-i) = \sum_{i=0}^{Q} b_i ,$$

and such that the polynomial

$$G(\beta) = \beta^{n+1} - \sum_{i=0}^{P} a_i \beta^{n-i} = 0$$

has no multiple roots other than $\beta = 0$ and all the roots lie in the unit circle with the exception of one root at $\beta = 1$, is stable for sufficiently small $h > 0$ if and only if the Lipschitz constant of $f(x, y)$ is negative.

For negative L we could then calculate for a particular method of solution a minimum value of $h \cdot L$ for which we would have a stable solution. This value, A, could be used in actual computing to limit the size of the successive integration steps and keep the solution stable.

CHAPTER III

RELATIVE STABILITY

The preceding results make the prospects of solving a differential equation with a positive Lipschitz constant $(L > 0)$ look pretty dim. However, we should investigate what happens when a root of $G(\beta) = 0$ tends outside the unit circle. The actual error at any one step is

$$9) \qquad \epsilon_n = \sum_{i=0}^{R} A_i (\beta_i)^n ,$$

where β_i is the i^{th} non-zero root of $G(\beta) = 0$ and the coefficients A_i depend on the original errors. Now if n is not too large, i. e., if the calculation does not require many steps, then a root just outside the unit circle will not cause a very large error. For example consider $\beta = 1.001$, $n \leq 20$, in this case $\beta^n \leq 1.020$. And so for small positive L and a sufficiently small step size h we would be able to integrate the differential equation for a short distance without losing much accuracy.

Another consideration might be the growth of the true solution y_n as compared with the growth of the error. Hamming claims (1, p. 198) that the solution of the differential equation tends to grow as

10) $y_n = C \cdot e^{Lhn}$.

If this were true we could require that the error growth be less than the growth of the true solution and our numerical solution would not be affected by the error.

Unfortunately, this is not necessarily true, for consider the differential equation with initial condition,

$$y' = x - y, \; y(1) = 0, \; x \geq 1 .$$

In this case $L = -1$ and the true solution is $y = x-1$ which obviously does not tend to grow as $y_n = C e^{-hn}$.

However, in the special case of $y' = Ay$, A an arbitrary constant, the solution is $y = y_0 e^{A(x-x_0)}$, and in many cases the solution is dominated by the exponential. So it is not totally unwarrented to require for "relative stability" that

11a) $|\beta|^n \leq e^{Lhn}$, for all n.

This can also be written

$$|\beta \cdot e^{-Lh}|^n \leq 1$$

or

11) $|\beta \cdot e^{-Lh}| \leq 1$.

At $h = 0$ the root $\beta = 1$ satisfies this and for real β we can write

$$-1 \leq \beta e^{-Lh} \leq 1.$$

Taking the derivative of $\beta \cdot e^{-Lh}$ with respect to h we get

$$12)\qquad \frac{d}{dh}\left[\beta \cdot e^{-Lh}\right] = \frac{d\beta}{dh} \cdot e^{-Lh} - L\beta e^{-Lh}$$

and at $h = 0$, $\beta = 1$, since $\frac{d\beta}{dh} = L$ we have

$$\frac{d}{dh}\bigg|_{\substack{h=0\\ \beta=1}} \left[\beta \cdot e^{-Lh}\right] = L \cdot 1 - L \cdot 1 = 0$$

So although we do not have absolute stability for all L, we may have relative stability for all L. To consider the growth of $\beta \cdot e^{-Lh}$ for the other roots β we can rewrite (12) as

$$\frac{d}{dh}\left[\beta \cdot e^{-Lh}\right] = e^{-Lh}\left[\frac{d\beta}{dh} - L\beta\right]$$

and using (7), (assuming no multiple roots)

$$\frac{d}{dh}\left[\beta \cdot e^{-Lh}\right] = L \cdot e^{-Lh}\left[\frac{\Sigma b_i \beta^{n-i} - (n+1)\beta^{n+1} + \Sigma(a_i + hLb_i)(n-i)\beta^{n-i}}{(n+1)\beta^{n} - \Sigma(a_i + hLb_i)(n-i)\beta^{n-1-i}}\right]$$

and at $h = 0$

$$13)\qquad \frac{d}{dh}\left[\beta \cdot e^{-Lh}\right] = L \cdot \left[\frac{-(n+1)\beta^{n+1} + \Sigma a_i (n-1)\beta^{n-i} + \Sigma b_i \beta^{n-i}}{(n+1)\beta^{n} - \Sigma a_i (n-i)\beta^{n-1-i}}\right].$$

And so if we can require that the real part in (13) be small, we can allow a slight increase in h without losing our relative stability since for the roots $|\beta| < 1$ (11) is satisfied with a strict inequality at $h = 0$. This can be stated in the following theorem.

Theorem 2: A procedure based on the equation

$$y_n = \sum_{i=0}^{P} a_i y_{n-i} + h \sum_{i=0}^{Q} b_i y'_{n-i}$$

where

$$\sum_{i=0}^{P} a_i = 1, \quad 1 + n - \sum_{i=0}^{P} a_i (n-i) = \sum_{i=0}^{Q} b_i$$

and such that the polynomial

$$\beta^{n+1} - \sum_{i=0}^{P} a_i \beta^{n-i} = 0$$

has no multiple roots except at $\beta = 0$ and all the roots lie in the unit circle with the exception of one root at $\beta = 1$, is "relatively stable" for sufficiently small $h > 0$ for any differential equation, and the error will not grow faster than the true solution for any differential equation whose solution is of the order e^{Lhn}.

CHAPTER IV

SOME FURTHER POSSIBILITIES

We can get an interesting result if we ignore condition (4b) and require that the polynomial (4a) have a zero at $\beta = 1$ for any value of h. This is equivalent to requiring that

$$G(\beta) = (\beta - 1)\ \Phi(\beta).$$

Dividing (4a) by $(\beta - 1)$ we get

$$14)\qquad \frac{G(\beta)}{\beta - 1} = \beta^{n} - (a_0 + b_0 hL - 1)\beta^{n-1} - (a_0 + a_1 + hLb_0 + hLb_1 - 1)\beta^{n-2} + \\ + \ldots + \sum_{i=0}^{R} \frac{(a_i + hLb_i) - 1}{\beta - 1} \quad .$$

And for divisibility we require that

$$\sum_{i=0}^{P} a_i + hL \sum_{i=0}^{Q} b_i - 1 = 0$$ for all h, L which gives, since

$$\sum_{i=0}^{P} a_i = 1$$ from (2)

$$hL \sum_{i=0}^{Q} b_i = 0 \text{ for all h, L or } \sum_{i=0}^{Q} b_i = 0 .$$

This means that we have a root of multiplicity at least 2 at $\beta=1$ when $h=0$. Suppose we require our first order method to have only two roots at $\beta=1$ for $h=0$. This would require that

$$\frac{d^2}{d\beta^2} G(\beta) \neq 0 \text{ at } \beta=1,\ h=0 .$$

From (5) we have

$$\frac{d}{d\beta} G(\beta) = (n+1)\beta^n - \sum_{i=0}^{P} a_i (n-i)\beta^{n-1-i} - hL\sum_{i=0}^{Q} b_i (n-i)\beta^{n-1-i} .$$

Differentiating once more with respect to β we get

15a) $$\frac{d^2}{d\beta^2} G(\beta) = (n+1)n\beta^{n-1} - \sum_{i=0}^{P} a_i (n-i)(n-1-i)\beta^{n-2-i}$$

$$-hL\Sigma b_i (n-i)(n-1-i)\beta^{n-2-i}$$

and

15) $$\left.\frac{d^2}{d\beta^2}\right|_{\substack{\beta=1\\ h=0}} G(\beta) = (n+1)n - \sum_{i=0}^{P} a_i (n-i)(n-1-i).$$

Thus we require that

16) $$(n+1)n - \sum_{i=0}^{P} a_i (n-i)(n-1-i) \neq 0.$$

We have seen that one root remains at $\beta=1$ for all values of h, so we can investigate the behavior of the root at $\beta=1$ for $h=0$ in the polynomial

$$17) \quad \frac{G(\beta)}{\beta - 1} = \beta^n - \sum_{i=0}^{R} \left(\sum_{j=0}^{i} a_j + hL \sum_{j=0}^{i} b_j - 1 \right) \beta^{n-1-i} = 0$$

with the requirements on the coefficients that

$$17a) \quad \sum_{i=0}^{P} a_i = 1,$$

$$17b) \quad 1 + n - \sum_{i=0}^{P} a_i (n-i) = \sum_{i-0}^{Q} b_i = 0,$$

and

$$17c) \quad (n+1)n - \sum_{i=0}^{P} a_i (n-i)(n-1-i) \neq 0.$$

At a solution of (17) we have

$$\beta^n - \sum_{i=0}^{R} \left(\sum_{j=0}^{i} a_j \right) \beta^{n-1-i} - hL \sum_{i=0}^{R} \left(\sum_{j=0}^{i} b_j \right) \beta^{n-1-i} + \sum_{i=0}^{R} \beta^{n-i-1} = 0.$$

Differentiating once with respect to h we get

$$18a) \quad n\beta^{n-1} \frac{d\beta}{dh} - \sum_{i=0}^{R} \left(\sum_{j=0}^{i} a_j \right) (n-i-1)\beta^{n-i-2} \frac{d\beta}{dh} - L \sum_{i=0}^{R} \left(\sum_{j=0}^{i} b_j \right) \beta^{n-i-1}$$

$$- hL \sum_{i=0}^{R} \left(\sum_{j-0}^{i} b_j \right) (n-i-1)\beta^{n-i-2} \frac{d\beta}{dh} + \sum_{i=0}^{R} (n-i-1)\beta^{n-i-2} \frac{d\beta}{dh} = 0,$$

or

18b) $$\frac{d\beta}{dh}\left[n\beta^{n-1} + \sum_{i=0}^{R} (n-i-1)\beta^{n-i-2} - \sum_{i=0}^{R} \left(\sum_{j=0}^{i} a_j \right) (n-i-1)\beta^{n-i-2} - hL\sum_{i=0}^{R} \left(\sum_{j=0}^{i} b_j \right) (n-i-1)\beta^{n-i-2} \right] = L\sum_{i=0}^{R} \left(\sum_{j=0}^{i} b_j \right) \beta^{n-i-1}$$

The polynomial multiplying $\frac{d\beta}{dh}$ in (18b) is continuous and at $h=0$, $\beta=1$ it reduces to

19a) $$n + \sum_{i=0}^{R} (n-i-1) - \sum_{i=0}^{R} \left(\sum_{j=0}^{i} a_j \right) (n-i-1)$$

which equals

19b) $$n + \frac{n(n-1)}{2} - \frac{(n-R-1)(n-R-2)}{2} - \sum_{i=0}^{R} a_i \left[\frac{(n-i)(n-i-1)}{2} - \frac{(n-R-1)(n-R-2)}{2} \right],$$

and since $\sum_{i=0}^{R} a_i = 1$ the $\frac{(n-R-1)(n-R-2)}{2}$ terms drop out and we have

19c) $$\frac{n(n+1)}{2} - \frac{1}{2}\sum_{i=0}^{R} a_i (n-i)(n-i-1)$$

which (16) guarantees is not equal to zero. So we have at $h=0$ and

$\beta = 1$

20a) $$\frac{d\beta}{dh} = \frac{L\cdot\sum_{i=0}^{R}\sum_{j=0}^{i} b_j}{\frac{1}{2}\left[n(n+1) - \sum_{i=0}^{R} a_i(n-i)(n-i-1)\right.} \; .$$

Now $$\sum_{i=0}^{R}\sum_{j=0}^{i} b_j = (R+1)b_0 + Rb_1 + (R-1)b_2 + \ldots + b_R$$

$$= \sum_{i=0}^{R}(R+1-i)b_i = (R+1)\sum_{i=0}^{R} b_i - \sum_{i=0}^{R} ib_i \; ,$$

and since by (17b) $\Sigma b_i = 0$ we have

$$\sum_{i=0}^{R}\sum_{j=0}^{i} b_j = -\sum_{i=0}^{R} ib_i \; .$$

So from (20a) we can get

20b) $$\frac{d\beta}{dh} = -\frac{2L\cdot\sum_{i=0}^{R} i\,b_i}{n(n+1) - \sum_{i=0}^{R} a_i(n-i)(n-i-1)} \; ,$$

and for positive L we need only require that

21) $$\mathrm{sgn}\left[\sum_{i=0}^{R} i\cdot b\right] = \mathrm{sgn}\left[n(n+1) - \sum_{i=0}^{R} a_i(n-i)(n-i-1)\right].$$

However under these conditions we cannot have a 2nd order method of solution, for the 2nd order method would require that

22) $$(1+n)^2 - \sum_{i=0}^{R} a_i (n-i)^2 = 2\sum_{i=0}^{R} b_i (n-i)$$

which can be rewritten as

$$n\cdot(1+n)+(1+n)-\Sigma a_i (n-i)(n-i-1)-\Sigma a_i (n-i)=2n\Sigma b_i - 2\Sigma i b_i,$$

but by condition (17b) this is equivalent to

23) $$n\cdot(1+n)-\Sigma a_i (n-1)(n-i-1) = -2\Sigma b_i \cdot i \ .$$

But from (20b) this would mean $\frac{d\beta}{dh} = L$ and this is not consistent with (21).

A possible solution to the problem of higher orders of accuracy may be to allow more roots at $\beta = 1$.

CHAPTER V

AN ALGORITHM FOR SOLVING AN ORDINARY DIFFERENTIAL EQUATION

```
procedure  asode (start, deriv, solu, othersolu, N, A, hmin,
                  hmax, xstart, ystart, xout, output, c1)

     value  hmin, hmax, xstart, ystart, xout, N, A;

     procedure  start, output;

     real procedure  deriv, solu, othersoly;

     real  hmin, hmax, xstart, ystart, xout, A;

     integer  N, c1;
```

comment asode solves the differential equation with initial conditions y'=deriv(x, y), y(xstart) = ystart. The input requires the following procedures: start which finds the necessary starting values for the integration, deriv which calculates the derivative at any point (x, y), solu the procedure for the actual integration which is stable for negative L, othersolu which carries out the integration when solu is not stable, and output which transfers control to some outside process. Also input are the following parameters: N - the number of back points needed by the two integrating procedures, A - a number associated with the solu such that solu is stable if $h \leq \frac{A}{L}$, hmin and hmax - the minimum and maximum desired for stepsize, xstart and ystart - the initial conditions for the differential equation, xout - the point at which control is desired to be transferred to output, and c1 to inform whether the back values of the solution are available c1=2 if they are and c1=1 if they are not;

```
begin  integer  n, sn, i;
       real  L, probh, h, ypr1, ypr2, del;
       real array  x, y, yprime [0:10];
    comment  the dimension of the arrays is arbitrary, 0 to 10 was
       used in this case because it is usually more than enough;
       x(0):=xstart
       y(0):=ystart;
       h:= hmax;
Test 1:  sn:= 1
    comment  This section finds the largest stable stepsize between
             hmin and hmax. If this can't be done, sn is set equal
             to 2 and method othersolu is used. In the case L is
             positive othersolu should make use of the discussion
             in Chapter IV;
       ypr1:= derive (x(0), y(0));
       del:= hmax·ypr1/2.;
       ypr2:= deriv(x(0), y(0) + del);
```

```
L:= (ypr2-ypr1)/del;
probh:= A/L
if  probh > h then h:=h
    else  if  probh > hmin then
          begin  h:= probh; xstart:=x(0);
                 ystart:= y(0); cl :=1;
                 go to initial end
          else begin h:= hmax; sn:=2 end

initial: if  cl=2  then go to integrate
          else start (xstart,  ystart,  deriv, x, y, yprime, h, N);
comment  If they are needed we use procedure start to find the
           first N values;
go to  Test 2;

integrate: for i=N  step -1 until 1 do
          begin  x(i):= x(i-1);
                 y(i):= y(i -1);
                 yprime(i):= yprime(i-1)  end
    x(0):= x(1) +h
    if  sn=1  then  y(0):= solu (x, y, yprime)
            else  y(0):= othersolu(x, y, yprime);
            yprime(0):= deriv(x(0), y(0));
```

```
Test 2: if  x(0) ≥  xout then output (x, y, yprime)

           else  go to Test 1;

                 end asode
```

BIBLIOGRAPHY

1. Hamming, Richard Wesley. Numerical methods for scientists and engineers. New York, McGraw-Hill, 1962. 411 p. (International series in pure and applied mathematics)

2. Hildebrand, F. B. Introduction to numerical analysis. New York, McGraw-Hill, 1956. 511 p. (International series in pure and applied mathematics)

3. Milne, William Edmund. Numerical calculus; approximations, interpolation, finite differences, numerical integration and curve fitting. Princeton, H. J., Princeton University Press, 1949. 393 p.

4. ________________. Numerical solution of differential equations. New York, Wiley, 1953. 275 p. (Applied mathematical series)

5. Naur, Peter (ed). Report on the algorithmic language ALGOL 60. Communications of the ACM 3:299-314. 1960.