

UDC 617.92:
518.61

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-353

常微分方程式のためのソフトウェアパッケージ

村田正秋

1978年6月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

常微分方程式のためのソフトウェアパッケージ*

村 田 正 秋**

Software Packages for the Numerical Solution of Ordinary Differential Equations

By Masaaki MURATA

Recently the most effective and efficient software packages for the numerical solution of ordinary differential equations have been developed, which appear to use either modifications of the well-known Adams predictor-corrector methods, or the relatively new methods based on Richardson extrapolation.

In order to compare the relative efficiency among these packages, we have tested two Adams methods due to Hindmarsh-Byrne, Shampine-Gordon, and an extrapolation method due to Bulirsh-Stoer as well as a number of the Runge-Kutta methods with stepsize control.

According to criteria involving the number of function evaluations, the accuracy, and the total execution time, the best general-purpose method is one due to Bulirsh-Stoer. This method can produce the highest degree of accuracy of these tested, while using step-sizes which are much larger than those required by the other methods. However, when function evaluations are relatively expensive, the implementation of the Shampine-Gordon code of variable-order variable-stepsize Adams formula is best. The frequently used RK4 (5) (Fehlberg) is also efficient when function evaluations are less expensive and tolerance is not very stringent, while RK7(8) (Fehlberg) is the best Runge-Kutta for pretty stringent tolerance. EPISODE by Hindmarsh-Byrne contains options for stiff problems that are not included in the other programs.

All the software packages discussed here were converted compatible with FACOM 230-75 system. The users are strongly encouraged to properly use these highly sophisticated integrators, depending on the characteristics of the problems to be integrated.

1 まえがき

軌道力学、軌道推定、最適制御など工学上の多くの問題が常微分方程式の初期値または境界値問題として記述されるため、それらを数値的に解くための信頼できるソフトウェアが研究のために不可欠の道具になる。当所計算センターのSSLにおいては常微分方程式に対する倍精度計算用サブルーチンとして次数4のRunge-Kutta-Gill(キザミ固定および自動可変)のコードSRKG2Dおよび多段公式による次数4のHamming(キザミ自動可変)のコードHAMP CDの2つが用意されている¹⁾。しかしながら

がら、これらのコードは容易にアクセスできて一定の結果が得られるというものの、今日の世界的レベルからみれば旧態依然と言っても過言ではない。特に研究報告に際して、どんな積分演算子によってその結果を得たかを明記して場合によっては第三者の追試にも耐えなければならぬ最近の現状では出揃確實でしかも性能的に今日最良とされる積分演算子を使用することが絶対必要ではないかと思われる。

常微分方程式の数値解法(アルゴリズム)については相当に古くから研究されてきたが、電子計算機を駆使して実際に数値解を得るためのソフトウェアに関して重要な発展が見られたのは比較的近年である。そこで本資料ではまず常微分方程式に対するソフトウェアパッケージの現状をサーベイしている。サーベイにあたってはF.T.Krogh

* 昭和53年3月3日受付

** 計測部

(Jet Propulsion Laboratory), A.C.Hindmarsh (Lawrence Livermore Laboratory), およびR.E.Mckenzie (UT) の協力を得て、現在一定の評価を得ているソフトウェアパッケージの殆んどすべてを網羅することができた。アルゴリズムのサーベイはそれ自体重要で興味があるが本資料の目的ではないことを注意しておく。また、ソフトウェアパッケージの数学的背景についても引用文献に詳しいので必要最小限にとどめた。次節において、常微分方程式に対するソフトウェアパッケージの現状をサーベイし、第3節で性能評価法、テスト問題を説明し、第4節で性能評価の結果を、第5節あとがきを述べている。さらに本資料で使用したソフトウェアは今日使用し得る最良のものと思われる所以当所ユーザーのために使用方法等を附録で説明した。

2 ソフトウェアパッケージの現状

1970年初め前後して、Gear^{2),3)}, Krogh⁴⁾ そして Shampine, Gordon⁵⁾ は Adams - Bashforth - Moulton の多段公式に基づく巧妙なプログラムを発表した。

Shampine, Gordon のコードは普通ODEと称されている。次数がそれぞれq, q+1(ただし $q < 13$)の予測子-修正子(predictor-corrector)法で、non-stiffシステム*に対して有効である。他方Gearのコードはnon-stiffシステムのみならず、stiffシステムに対しても有効な、同次数qの予測子-修正子法である(表1参照)。

さらに1973年から Hindmarsh, Byrne^{6),7)} は前記 Gearのコードを拡張することを試み、1975年GEARそして1977年EPISODE**を完成した。EPISODEでは陰伏型(implicit)のAdamsまたはStiffシステムに対する後退微分公式(Backward Differentiation Formula, BDF)を使用している。即ち

$$\begin{aligned} \mathbf{y}_n &= \sum_{j=1}^{K_1} a_{nj} \mathbf{y}_{n-j} + h_n \sum_{j=0}^{K_2} b_{nj} \dot{\mathbf{y}}_{n-j} \\ h_n &= t_n - t_{n-1}, \quad b_{n0} > 0 \end{aligned} \quad (*)$$

ここで、 $K_1 = 1$, $K_2 = q - 1$ とすると次数qの陰伏型 Adams法となり、 $K_1 = q$, $K_2 = 0$ とすると次数qのBDF法となる。式(*)は一般に \mathbf{y}_n に関する非線形代数方程式であり、各ステップで \mathbf{y}_n に関して解かねばならない。上式を変形して

$$\begin{aligned} \mathbf{G}(\mathbf{y}_n) &\triangleq \mathbf{y}_n - h_n b_{n0} \mathbf{f}(\mathbf{y}_n, t_n) \\ &- \left\{ \sum_{j=1}^{K_1} a_{nj} \mathbf{y}_{n-j} + \sum_{j=1}^{K_2} b_{nj} \dot{\mathbf{y}}_{n-j} \right\} \end{aligned} \quad (**)$$

とおこう。式(*)の解は、式(**)で定義される \mathbf{G} の零、 \mathbf{y}_n を求めることと同等であり(G-Stops), EPI-SODEでは、Gauss-Siedel法またはNewton-Raphson法によっている。Newton-Raphson法の適用にあたっては \mathbf{G} の \mathbf{y}_n に関する偏導関数の値を評価しなければならぬが、それは一般にコストが高くつくので数値微分による置換(numerical partials)をオプションとして持つ。

いずれのコードも、可動オーダー(variable-order)可動ステップ(variable-step)で、1ステップ当たりの(重みつき)局所打切り誤差を任意に指定した局所許容誤差(local error tolerance)内におさえるものであり(従って大域的に計算精度を保証するものでないことに注意)，さらにセルフスターティングである。従来の多段公式法、例えばHamming法¹⁾、では出発値(初期値)の他に出発点近傍における数個の解を通常同次数のRunge-Kutta法のごとき別のスキームで前以って計算しておかねばならなかったから、セルフスターティングと言うことは著しい改良である。

Richardson外挿法(extrapolation)の概念は、Rombergによる数値積分への応用(Romberg積分)によって一躍注目されたが、常微分方程式の数値解法への応用としては1966年の Bulirsh, Stoer⁸⁾の仕事が最初である。この論文で外挿関数として多項式よりも有理関数の方が優れていることも示した。彼らの作製した有理型外挿公式(最大次数6)によるALGOLコードは、FOX⁹⁾によりFORTRANコードへ翻訳された。この方法の特徴はステップサイズを非常に大きくとれて、しかも他の手法に較べて最も高精度計算の達成できることである。

Runge-Kutta法は今日最も良く知られ、使用頻度の高い解法であるが、それについては Fehlberg^{10),11)}の一連の研究が重要であろう。打切り誤差のオーダーと使用する係数に応じて多数のスキームが提案されているが本資料では、Mckenzie¹²⁾(テキサス大学)の製作したRK2(4)(T.E.Hull), RK4(5)(Fehlberg)およびRK7

*一般に常微分方程式 $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ の Jacobian $\partial \mathbf{f}(t, \mathbf{x}) / \partial \mathbf{x}$ の条件指数が大、例えば 10^5 、なるとき stiff であると言う。

** Effective Package for the Integration of Systems of Ordinary Differential Equations を意味する。

(8) (Fehlberg) の各コードをとりあげる。ここで例えれば RK7(8) は次数（正確度とも言う）7, かつ h をステップサイズとする時 $O(h^8)$ の項つまり打切り誤差の先頭項に關して誤差評価することを示す。また以下では単に RK7 と書く場合は固定ステップサイズの Runge-Kutta 法を表わすものとしよう。Fehlberg¹⁰⁾ は RK8(9) およびそれより高次のスキームの数値実験を行なった結果、RK7(8) と比較して著しい性能向上は見られなかったため、RK7(8) が実用上限界で最適なスキームに近いと報告している。固定ステップサイズの Runge-Kutta スキームの場合、最適なステップサイズを決めるのが極めて難かしい。通常打切り誤差のオーダーと所要精度から h を適当に小さく決めているが、多くの場合必要以上に小さくとり過ぎているようである。その結果積分ステップ数が増加し計算時間が長くなるばかりでなく、丸めの誤差 (round off) の影響も無視できなくなる***。それゆえに、解くべき常微分方程式の特性に応じて何らかの根拠ある方策で自動的にステップサイズを変化させて、許容誤差範囲内でできる限り大きなステップサイズのとれることが望ましい。

Runge-Kutta 法に対するステップサイズ制御としては附録 1 に解説するところの D.G. Hull¹³⁾ の手法が最も合理的で優れているようである。

表 1 に今日最良とされるソフトウェアパッケージの概要をまとめておく。表 1 の全コードおよび RK2(4) を FACOM 230-75 用に変換し、種々の問題に適用することにより計算コスト、信頼性等に関する資料を得た。Gear の原コード³⁾ は最大次数 7 (Adams) に制限されているので、以下では性能評価の対象から除外し EPI SODE を考慮している。

3 性能評価

評価規準

常微分方程式の数値解法およびソフトウェアの数は莫大なものがあり、そのために各プログラムを具体的な問題に適用してコスト、信頼性（精度）、安定性等の性能を比較してみる必要があった (Krogh⁴⁾, Hull 他¹⁵⁾, Shampine, Gordon⁵⁾ および Byrne, Hindmarsh 他⁷⁾。性能評価を合理的に行なうためには、評価規準とテスト問題が明確に規定されていなければならない。本資料では評価規準として、(1) 計算精度、(2) 導関数評価の回数、および

(3) 総計算時間、を考える。計算精度は、次式で定義される規格化された最大誤差 NME (normalized maximum error) によって評価する。

$$NME \triangleq \max_j \frac{\| \mathbf{e}(t_j) \|_{\infty}}{\tau}$$

ここで、 $\mathbf{e}(t_j)$ は指定された時点 t_j ($j = 1, 2, \dots$) における計算値 $\mathbf{x}_c(t_j)$ の真値 $\mathbf{x}(t_j)$ に対する絶対誤差（または相対誤差）ベクトル、 τ は許容誤差とする。また $\| \mathbf{e}(t_j) \|_{\infty}$ はマクシマムノルム、つまり

$$\begin{aligned} \| \mathbf{e}(t_j) \|_{\infty} &= \max_k \{ |e_k| \}, \quad k = 1, \dots, n \\ \mathbf{e} &= (e_1 \ e_2 \ \dots \ e_n)^T \end{aligned}$$

を表わす。導関数評価とは、解くべき常微分方程式を一般に

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x} &\triangleq (x_1 \ \dots \ x_n)^T, \quad \mathbf{f} \triangleq (f_1 \ \dots \ f_n)^T \end{aligned}$$

（ T は転置作用素）と書く時、右辺の関数 $\mathbf{f}(t, \mathbf{x})$ を計算することである。導関数評価の回数はスキームに固有で、使用計算機およびプログラミング技術に依存しないと考えられ、しかも総計算時間に占める導関数評価のための時間の割合が一般に極めて高いことからも、計算精度と並ぶ重要な規準と考えられる。一方、総計算時間とかその他所要記憶容量の多少等は計算機およびプログラミング技術の巧拙に依存するので普遍的尺度とは言い難い。しかし同一機種で各コードの性能を評価する場合にはユーザーにとって興味ある尺度になろう。

テスト問題

計算機で実行し得るテスト問題の数には限度があり、性能評価に影響するところ大であるから、問題の選択には慎重でなければならぬ。数値実験結果を正確に解釈するために物理的あるいは数学的に意味のある問題で常微分方程式の充分に広いクラスを含むことが望ましい。また計算精度をチェックするために真の解が既知であるか、あるいはエネルギー積分とか周期性のように解の満足すべき、ある種の関係式が利用できることも必要な条件である。Krogh⁴⁾ は豊富な経験に基づいてテスト問題の標準化に対する試案

*** Brower, Clemence¹⁴⁾によれば、丸めの誤差は n ステップ後には $\log_{10} (.1124 n^{3/2})$ 術まで成長し得る。例えば $n = 200$ とすると

$$\log_{10} [.1124 (200)^{3/2}] \sim 2.5$$

であるから、有効数字 6 術の計算に対しては少くとも 6 + 2.5 ~ 9 術の計算を演算器 (accumulator)において保持しなければならない。

を発表している。本資料ではKrogh, Hull他¹⁵⁾から以下に述べる9例について数値実験を試み、その結果から性能比較を行なう。

問題 1

$$\frac{dx}{dt} = -x, \quad x(0) = 1, \quad t \in [0, 100]$$

解 $x(t) = e^{-t}$ 。 $e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差、および e^{-t_j} に関する相対誤差、 $j = 1, 2, \dots$

問題 2

$$\frac{dx}{dt} = +x, \quad x(0) = 1, \quad t \in [0, 100]$$

解 $x(t) = e^t$ 。 $e(t_j) = j$ ステップ目の時刻 t_j における相対誤差、 $j = 1, 2, \dots$

問題 3

$$\frac{dx}{dt} = t(1-x) + (1-t)e^{-t}, \quad x(0) = 1, \\ t \in [0, 100]$$

解 $x(t) = e^{-t^2/2} - e^{-t} + 1$ 。 $e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差、 $j = 1, 2, \dots$

問題 4

$$\frac{dx}{dt} = \begin{cases} \frac{2}{3}t^{-1/3} & \text{if } t \neq 0, \quad t \in [-1, 1] \\ 0 & \text{if } t = 0 \end{cases}$$

$$, \quad x(-1) = 1, \quad t \in [-1, 1]$$

解 $x(t) = t^{2/3}$ 。 $e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差、 $j = 1, 2, \dots$

問題 5

$$\frac{dy}{dt} = Uz - UBUy, \quad y = (y_1, y_2, y_3, y_4)^T, \\ y(0) = (-1, -1, -1, -1)^T, \quad t \in [0, 10]$$

ここで

$$z = (w_1^2, w_2^2, w_3^2, w_4^2)^T$$

$$w = (w_1, w_2, w_3, w_4)^T = Uy$$

$$U = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

$$B = \begin{pmatrix} \beta_1 & 0 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ 0 & 0 & 0 & \beta_4 \end{pmatrix}$$

解

$$y(t) = U \begin{bmatrix} \beta_1 / (1 - (1 + \beta_1)e^{\beta_1 t}) \\ \beta_2 / (1 - (1 + \beta_2)e^{\beta_2 t}) \\ \beta_3 / (1 - (1 + \beta_3)e^{\beta_3 t}) \\ \beta_4 / (1 - (1 + \beta_4)e^{\beta_4 t}) \end{bmatrix}$$

$e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差、 $j = 1, 2, \dots$

問題 6

$$\ddot{y} + 11\dot{y} + 10y = 0, \quad y(0) = 1, \quad \dot{y}(0) = -1, \\ t \in [0, 100]$$

解 $y(t) = e^{-t}$ 。 $e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差、 $j = 1, 2, \dots$

問題 7 (2体問題)

$$x'' = -x/r^3, \quad x(0) = 1 - \varepsilon, \quad x'(0) = 0 \\ y'' = -y/r^3, \quad y(0) = 0, \quad y'(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$$

$$r = (x^2 + y^2)^{1/2}, \quad t \in [0, 4\pi] \text{ (2 period)}$$

解

$$x = \cos u - \varepsilon, \quad x' = -\sin u / (1 - \varepsilon \cos u) \\ y = \sqrt{1 - \varepsilon^2} \sin u, \quad y' = \sqrt{1 - \varepsilon^2} \cos u / (1 - \varepsilon \cos u)$$

但し、 $t = u - \varepsilon \sin u$ (Keplerの方程式)、 ε = 離心率。 $e(t_j) = j$ ステップ目の時刻 t_j における絶対誤差。

問題 8 (制限3体問題)

$$y_1'' = 2y_2' + y_1 - \mu(y_1 + \mu)/r_1^3 - \mu(y_1 - \mu)/r_2^3,$$

$$y_2'' = -2y_1' + y_2 - \mu'y_2/r_1^3 - \mu y_2/r_2^3$$

$$y_1(0) = 1, 2, \quad y_1'(0) = 0, \quad y_2(0) = 0,$$

$$y_2'(0) = -1.04935, 75098, 30319, 90726 \dots$$

$$t \in [0, T_p]$$

ここで

$$r_1 = [(y_1 + \mu)^2 + y_2^2]^{1/2}$$

$$r_2 = [(y_1 - \mu)^2 + y_2^2]^{1/2}$$

$$\mu = 1/82.45, \quad \mu' = 1 - \mu.$$

$e(T_p) = 1$ 周期後の絶対誤差。各ステップで Jacobi 積分 J の絶対誤差をチェック。

問題 9 (Euler の運動方程式)

$$\mathbf{y}' = \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} = \begin{pmatrix} y_2 y_3 \\ -y_1 y_3 \\ -k^2 y_1 y_2 \end{pmatrix}, \quad \mathbf{y}^{(0)} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix},$$

$$k^2 = 0.51 \quad t \in [0, 28K]$$

解

$$\mathbf{y}(t) = \begin{bmatrix} \text{sn}(t; k) \\ \text{cn}(t; k) \\ \text{dn}(t; k) \end{bmatrix}.$$

$e(t_j)$ = 時刻 jK における絶対誤差, $j = 1, 2, \dots, 28$.

問題 1 ~ 3 は簡単なスカラーの常微分方程式で、計算精度および安定性のチェックに使われる。もっとも、安定性に関する検討は本資料では行わないが、一般的に計算誤差がある許容範囲をはじめて超える時間 t が大きいほど、スキームは安定であると言える（例えば Lapidus, Seinfeld¹⁶）。問題 4 は、積分可能な特異点（integrable singularity）の例、問題 5, 6 は“Stiff” システムの例であり、スキームはこれらの場合にも適用できることが望ましい。問題 5について、Krogh⁴ の注意は興味深い。 \mathbf{y} に関する \mathbf{y}' の Jacobian の固有値は、 $t=0$ で $-(\beta_i + 2)$, $t \rightarrow +\infty$ で $-|\beta_i|$ に近づく。 β_i の大きさに大差がある時一般に方程式は Stiff であると言われる。この場合許容誤差を $\min(|\beta_i|)$ より大きく指定すると有界でない解の発生する可能性がある。例えばいま

$$y' = -\beta y + y^2$$

と考えてみよう（問題 5 はこの一般化である）。この方程式の解は、 $0 < \beta < y$ または $\beta < 0 < y$ ならば有界でない。 $y > 0$, $y' > 0$ から、 y が増えると y' も単調増大するからである。したがって、平衡点 ($\beta > 0$ で $= 0$, $\beta < 0$ で $= \beta$) 近くの解は、 $|\beta|$ より大なる解の増大に対して有界でない解を与える。局所的打切り誤差が許容誤差を超えないという制約下で任意のステップサイズを許すスキームに対しては、 $|\beta|$ より大なる許容誤差を指定して計算することは危険である。もっとも實際上は許容誤差を $|\beta|$ よりかなり大きく指定しても有界な解が得られるようである。問題 5 では次の値を使用した。

$$\beta_1 = 6, \beta_2 = 1, \beta_3 = -0.1, \beta_4 = 10^{-5}.$$

問題 7 ~ 9 は軌道力学で良く知られた方程式である。まず問題 7 は、2体問題に対する Newton の運動方程式で解軌道は周期 2π の梢円である。離心率 ϵ として、0, 0.3,

0.6, 0.9 の 4通りの場合について計算する（図 1）。

Kepler 方程式は、Newton-Raphson 法により 10^{-16} の精度で計算する。問題 8 は、地球一月 - 宇宙船に関する制限 3 体問題であるが、その定義および定式化については Szebehely¹⁷ が権威ある文献である。解の陽型な（explicit）表式はないが、先の初期条件のもとで周期が

$$T_p = 6.19216 \ 93313 \ 19639 \ 70674 \dots$$

の閉軌道であること、および次の Jacobi 積分 J (y_1 , y_2 , y_1' , y_2') の成立したこと、が知られている；任意の解軌道に沿って、

$$J(y_1 \ y_2 \ y_1' \ y_2') = \frac{1}{2} (y_1'^2 + y_2'^2 - y_1^2 - y_2^2) - \frac{\mu}{r_1} - \frac{\mu}{r_2}$$

は定数で、その値は初期値から一意に決まる。

問題 9 は、外力の無い剛体に対する Euler の運動方程式で厳密解は Jacobi の梢円関数によって表現され、周期 $4K$ を持つことが知られている。ここで、

$$K = 1.86264 \ 08023 \ 32738 \ 55203 \\ 02812 \ 20579 \dots$$

4 数値実験

計算は当所計算センターの FACOM 230-75 (FORTRANH, OPT = 2) により倍精度で実行した。初期ステップサイズは Bulirsh-Stoer に対して 10^{-2} 、他はすべて 10^{-5} とした。* 表 2 ~ 11において、

FCN CALLS = 導関数の評価回数、

NO OF STEPS = 総ステップ数、

$$\text{NORMALIZED MAX. ERROR} = \max_j \frac{\|e(t_j)\|_\infty}{\tau}$$

TOTAL (EXECUTION) TIME = マシンタイム (ミリ秒)、および計算実行中に発生した異常事態に対して各コードの打ち出したメッセージを示した。なお左肩の 3 数字は許容誤差 τ を表わす。尚表 2 ~ 10において各スキームに対して最上段が $\tau = 10^{-3}$ 、中段が $\tau = 10^{-6}$ 、下段が $\tau = 10^{-9}$ に対応する計算結果を示している。また例えば 2.48 (-4) は、 2.48×10^{-4} を意味するものとする。

(1) Runge-Kutta 法は一般に低次数のスキームほど指定した τ に対して数桁以上精度は高まるが導関数評価の回数、したがって計算時間は飛躍的に増大する。そこで、

* RK 法に対して 10^{-5} は小さ過ぎる。実際他の計算例によれば積分時間々隔または軌道周期の 50 ~ 100 等方位で同精度の結果が得られる。また ODE に対しては初期ステップサイズは不要である（附録 2）。

導関数評価が低コストならば τ の比較的広範囲な値に対しで RK 4(5), 導関数評価が比較的低コストでかつ高精度で長時間の積分が必要な場合, またはパラメータの値を変えて繰り返し積分が必要な場合に対して RK 7(8)が適切である。RK 7(8)をスカラ一方程式に適用するのはおおげさ過ぎるが, コスト・信頼性の見地からは RK 2(4), RK 4(5)にはるかに優っていることは注目して良い。同次の Runge-Kutta に対する一般的なステップサイズ制御のできるサブルーチンの方が高信頼かつ高能率である。

附録 2 に示すように, 3 つの RK コードは ABSERR = RELERR = 0.D0 と指定することにより固定ステップサイズ h の積分を行なう。その際, 重要な注意として RK 2(4), RK 4(5) および RK 7(8) は夫々 RK 3, RK 5, および RK 8 またはそれより若干高次の RK 法に対応する。その理由は RK 法に対する D.G. Hull のステップサイズ制御の手法(附録 1)から明らかのように, 打切り誤差の先頭項を評価するために常に数次, 高次項まで考慮しているからである。実際表 1-1 によると, 1 ステップ当たりの導関数評価の回数は夫々 4, 6, 13 であり, また $\epsilon = 0.0$ に対して, 絶対誤差(NME を $\tau = 1$ として計算)は概ね, 夫々 $O(h^4)$, $O(h^6)$, $O(h^{12})$ になっている。さて, 表 1-1 は問題 7(2 体問題)において種々の離心率 ϵ に対して, 固定ステップサイズ h_1, h_2, h_3 の RK 法を適用した計算例である。ここで $h_1 = 2\pi/30$ (1 軌道周期の 30 分等分) $h_2 = 2\pi/60$, および $h_3 = 2\pi/90$ とする。TOTAL TIME は各ステップで Kepler の方程式を解く時間だけ各場合で若干異なる。表 1-1 は, $\epsilon = 0.0$ に対応する実行時間を示す。表 8 と表 1-1 の結果をこのまま比較することはあまり公平とは言えない。実際, 表 8 の結果は既に注意したように初期ステップサイズ $h_0 = 10^{-5}$ として得られたが, これは小さ過ぎて, 例えば表 8-1 の総ステップ数から推定して $2\pi/50 \sim 2\pi/100$ 位で充分のようである。 h_0 が過少(または過大)であると, 各ステップで最適なんを決める(図 3 参照)ために余分な計算時間がかかることになる。この理由から表 8 における TOTAL TIME はもう少し割り引いてよい。いずれにせよ, 次の 2 点は明らかに指摘してよいと思う。まず ϵ が 0 に近づく, つまり軌道が円形に近づく程, ステップサイズ制御は指定された許容誤差 τ に対して定まる殆んど最適な等分割を与える。 ϵ が 1 に近づく, つまり軌道が偏平な橢円となるにつれ, 長軸の端点において軌道変化が著しく(図 1), この近傍ではステップサイズを充分小さくとり, 逆に短軸付近ではかなり大きくとって良いことが推測できる。この場合, 表 8-3, 8-4, および 1-1 から明らかのようにステップサイズ制御つきの RK 法については τ をかなり小さく指定

することにより, 計算時間および精度の点で実用的な解を得られる。しかし高精度な解, つまり NME の値が高くても 10 位の解を得るのは至難である。固定ステップサイズの RK 法は使用した h_1, h_2, h_3 に対しては, ϵ が小なる場合殆んど同等の結果を与えるが, ϵ が 1 に近づくと精度が極めて悪い。

(2) 問題 8 に関する各スキームの計算過程(ただし $\tau_{ABS} = 10^{-7}$)を示した図 2 からも明らかのように, Richardson の外挿公式の特徴はステップサイズを大きくとれて, しかも表 2 ～ 10 における成功例は Bulirsh-Stoer コードが最も高精度計算を達成できることを示している。計算精度は RK 2(4) と同位かそれ以上である。ただし導関数評価の回数が例えば RK 7(8) の約 2 ～ 3 倍かかるので, それが比較的低コストでしかも高精度計算の要求される問題に向いている。導関数評価について今迄低成本, 比較的低成本, 等々かなり曖昧に使ってきた。Hull 他¹⁵⁾ は, 導関数 f の 1 要素 f_j の計算につき演算回数 25 をその境の目安としている。この点に関する研究はまだ皆無に近く, 上記文献が唯一のものであろう。

(3) FCN CALLS のみに限れば, Shampine-Gordon が最少であるにも拘わらず, TOTAL TIME が他のスキームより長いことは, 内部オーバーヘッドタイム(マシンタイムと導関数評価のための時間との差), つまりサブルーチン固有の処理時間の大きいことを意味する。これは導関数評価が複雑で高コストな問題に対して, このスキームが適切であることを示している。

(4) 表 6, 7 は, Stiff なシステムに形式的に適用した結果である。EPISODE #21 は BDF 法で Newton-Raphson 法(解析的偏導函数)を使って解いているが, 性能的にこれと匹敵するのはわずかに RK 4(5)のみである。

5 あとがき

常微分方程式に対するソフトウェアパッケージの現状をサーベイし, 若干のテスト問題に適用することにより各ソフトウェアの特徴および性能に関する資料を得ることができた。その結果は, あらゆる問題に対してコスト・信頼性ともに最良とみなしえるソフトウェアは現時点では存在しないことを示している。強いて限定するとすれば, non-stiff システムのみならず stiff システムへも適用可能な EPISODE と言うことになろうが, 計算結果は non-stiff システムに関する性能が他コードに較べて若干見劣りすることを示している。計算精度のみに限れば, 明らかに Bulirsh-Stoer が最良である。勿論これはソフトウェアとして最良であることを意味するのではなく, アルゴリズムの特徴であると見るべきである。と言うのは, あら

かじめ指定された局所許容誤差 τ に対して、計算結果が大域的に $NME = 1 \sim 10$ を満足すればソフトウェアとして充分信頼してよいからである。とにかく導関数評価が比較的低コストで高精度計算が必要な問題に対して、この Bulirsh-Stoer の外挿法が最適であると言えよう。また、導関数評価が簡単で許容誤差があまり厳しくない問題に対して R K 4(5), 導関数評価が比較的低コストで高精度計算または繰り返し計算の必要な問題に対して R K 7(8), 導関数評価が複雑で高コストな問題に対しては可動オーダー可動ステップの ODE が適切である。そして、導関数評価のコストとしては演算回数 25 を一応境界として良いであろう。以上はいわゆる non-stiff システムに対して成立することである。stiff システムに対しても、計算時間さえ問題にしなければ、上のコードが適用できる場合が多い（勿論 stiffness 比に依存するが）。しかし、stiff システムに対しては、それに対して適切なアルゴリズムを有する EPISODE (または GEAR) を最初から使用する方が能率的である。実際許容精度を保持したままで計算コストを大巾にセイブできるからである。

従来この種の汎用ソフトウェアの開発において、(1) 計算実行中に発生する予測される異常事態に対して、それを検知し、何らかのエラーメッセージを印字すること、(2) ユーザーの問題に応じて、プログラムの部分変更が可能であること、という 2 点が軽視されていたように思われる。常微分方程式に対するソフトウェアとしては、(1) は言うに及ばず (2) も極めて重要視しなければならない。例えば、打切り誤差に関する重みは、ユーザーの問題と目的によっては、ソフトウェアにあらかじめ用意されている重みとは異なるものを使用した方が合理的な場合がある。常微分方程式に対する深い洞察および数値計算に関して豊富な経験を有するユーザーはその必要性を常々痛感しているところであろう。(2) の意味で、本資料で述べたソフトウェアは本来的にソースプログラムの形でユーザーにアクセスされなければならないと言うのが著者の強調したい点である。

本資料では触れなかったが、軌道力学等でよく現われる 2 階常微分方程式に対しては、これを 1 階常微分方程式系になおさずにそのまま直接積分する手法がある。Gauss-Jackson (第 2 積和とも言う) 法, Runge-Kutta-Nyström 法 (例えば RKN 7(8), Fehlberg¹¹⁾) が代表的であり、コードも完備していることを附言しておく。

本資料で述べたソフトウェアのすべては、当所計算センターにおいて、ソースまたはライブラリー形式でアクセスできる。軌道力学、最適制御または流体力学等々で現われる常微分方程式の高信頼かつ高能率計算のために本資料で述べた各コードを適切に選択して使用されることが期待さ

れる。

最後に Hindmarsh, Krogh 両博士との私信なくして本資料は存在しなかったことを記して、感謝の意を表したい。

参考文献

- 1) FACOM FORTRAN SSL 使用手引書, および FACOM SSL 解法解説書
- 2) C.W.Gear ; The Automatic Integration of Ordinary Differential Equations, Comm ACM., 14, pp.176-179. 1971.
- 3) C.W.Gear ; Algorithm 407, DIFSUB for Solution of Ordinary Differential Equations, Ibid., 14, pp.185-190. 1971.
- 4) F.T.Krogh ; On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations, JPL Tech. Memo. 217 (Rev.), California Institute of Technology, Pasadena, 1970.
- 5) L.F.Shampine, M.K.Gordon ; Computer Solution of Ordinary Differential Equations, The Initial Value Problem, W.H. Freeman and Company, San Francisco, 1974.
- 6) A.C.Hindmarsh, G.D.Byrne ; EPISODE - An Effective Package for the Integration of Systems of Ordinary Differential Equations, Lawrence Livermore Laboratory Report UCID-30112, Rev.11, April, 1977.
- 7) G.D.Byrne, A.C.Hindmarsh et al ; A Comparison of Two ODE Codes : GEAR and EPISODE, Computers and Chemical Engineering, vol. 1, pp.133-147, 1977.
- 8) R.Bulirsh, J.Stoer ; Numerical Treatment of Ordinary Differential Equations by Extrapolation Method, Numerish Math. 8, pp.1-13, 1966.
- 9) P.A.Fox ; DESUB : Integration of a First-order System of Ordinary Differential Equations, pp.477-507, Chapter 9 in Mathematical Software, John R.Rice ed., Academic Press, New York, 1971.
- 10) Erwin Fehlberg ; Classical Fifth-, Sixth-, Seventh-, and Eighth-order Runge-Kutta Formulas with Stepsize Control, NASA TR R-287, October, 1968.

- 11) Erwin Fehlberg : Classical Seventh-,
Sixth-, and Fifth-order Runge-Kutta-
Nyström Formulas with Stepsize Control
for General Second-order Differential
Equations, NASA TR R-432, October, 1974.
- 12) R.E.Mckenzie : Private Communication
Dept. of Aerospace Engineering & Enginee-
ring Mechanics, University of Texas at
Austin, 1976.
- 13) D.G.Hull : Lecture Note, Dept. of
Aerospace Engineering & Engineering
Mechanics, University of Texas at Austin,
1976.
- 14) D.Brower and G.M.Clemence : Methods
of Celestial Mechanics, Academic Press,
New York, 1961.
- 15) T.E.Hull,W.H.Enright,B.M.Fellen,
A.E.Sedgwick : Comparing Numerical
Methods for Ordinary Differential
Equations, SIAM J. Numer.Anal., vol.9,
no.4, December 1972, pp.603-637.
- 16) L.Lapidus,J.H.Seinfeld : Numerical
Solution of Ordinary Differential Equa-
tions, Academic Press, New York, 1971.
- 17) V.Szebehely : Theory of Orbits (The
Restricted Problem of Three Bodies),
Academic Press, New York, 1967.

附録1 Runge-Kutta 法のステップサイズ制御

Runge-Kutta 法のステップサイズ制御に関する Hull¹³⁾ の手法を R K 7(8)を例として説明する。 Fehlberg¹⁰⁾ によれば、 1 階連立常微分方程式

$$\mathbf{y}' = \mathbf{f}(X, \mathbf{y}) \quad (1)$$

にに対する R K 7(8)は次式で与えられる。

$$\mathbf{f}_0 = \mathbf{f}(X_0, \mathbf{y}_0) \quad (2)$$

$$\mathbf{f}_K = \mathbf{f}(X_0 + a_K h, \mathbf{y}_0 + h \sum_{k=0}^{K-1} \beta_{K,k} \mathbf{f}_k) \quad (3)$$

$$\mathbf{y} = \mathbf{y}_0 + h \sum_{k=0}^{10} C_k \mathbf{f}_k + O(h^8) \quad (4)$$

$$\hat{\mathbf{y}} = \mathbf{y}_0 + h \sum_{k=0}^{12} \hat{C}_k \mathbf{f}_k + O(h^9) \quad (5)$$

式(2)～(4)における未知係数 $a_K, \beta_{K,k}, C_k, \hat{C}_k$ は、式(3)および(4)が夫々 7 および 8 次の Runge-Kutta 法即ち R K 7, R K 8 となり、特に $\{a_K, \beta_{K,k}\}$ ($K=1 \sim 10$) が式(3), (4)で同一の値となるように決定される。この制限下で R K 7, R K 8 の差をとって、

$$\begin{aligned} T.E. &\triangleq \hat{\mathbf{y}} - \mathbf{y} \\ &= h \left[\sum_{k=0}^{10} (\hat{C}_k - C_k) \mathbf{f}_k + \sum_{k=11}^{12} \hat{C}_k \mathbf{f}_k \right] \quad (5) \\ &\sim O(h^8) \end{aligned}$$

を得る。式(5)は R K 7 の 1 ステップ当りの局所打切り誤差の先頭項を表わすとみなされ、それを T.E. と書いた。さらに、相対的打切り誤差 $R.T.E.$ を

$$R.T.E. = \frac{T.E.}{y_j} \quad (j = 1 \sim n)$$

にて定義する。この $R.T.E.$ の絶対値を最大にする y_j を y_r とし、式(5)により計算された T.E. に対して

$$T.E. = \xi r^8$$

を成立させる定数ベクトルを ξ とする。その時、

$$\frac{T.E.}{y_r} = \frac{\xi_r h^8}{y_r}$$

即ち

$$R.T.E. = \eta_r h^8, \quad \eta_r = \frac{\xi_r}{y_r} \quad (6)$$

が成り立つ。D.G.Hull は、式(6)によって決定せる η_r の値が次のステップでも近似的に同一であると仮定した。つまり添字 r を省略して

$$R.T.E. = \eta h^8$$

$$R.T.E. = \eta h^8$$

から

$$h^8 = \frac{R.T.E.}{R.T.E.} h^8$$

を得た。さらに $R.T.E.$ が相対許容誤差 τ になるとし

$$h^8 = \left(\frac{\tau}{R.T.E.} \right)^{1/8} h^8 \quad (7)$$

を得て、式(7)で計算された h^8 を次のステップサイズとした。但し実行上は式(5)の省略項の影響を考慮した一般式

$$h_{\text{new}} = h_{\text{old}} \left(\frac{\tau}{R.T.E.} \right)^{1/8} \cdot PCT \quad (8)$$

により計算される。ここで PCT は修正率 (percent correction) と呼ばれるパラメータで通常 0.7 ～ 0.95 の範囲にとる。

プログラムでは、 $R.T.E. < \tau$ の時次のステップでは式(8)により h を増加させることを試み、新しいステップとして

$$h_{\text{new}} = \min \left\{ h_{\text{max}}, h_{\text{old}} \left(\frac{\tau}{R.T.E.} \right)^{1/8} \cdot PCT \right\} \quad (9)$$

を採用し、同様に $R.T.E. > \tau$ の時

$$h_{\text{new}} = \max \left\{ h_{\text{min}}, h_{\text{old}} \left(\frac{\tau}{R.T.E.} \right)^{1/8} \cdot PCT \right\} \quad (10)$$

により新ステップを決め、 h が小さくなり過ぎぬようにする。ここで $h_{\text{max}}, h_{\text{min}}$ は夫々ステップサイズの許容最大値、許容最小値を示す。なお、 y_r が零に極めて近い値をとる場合には $R.T.E.$ が決まらないのでプログラムでは与えられた絶対(相対)許容誤差 τ_{ABS} (τ_{REL}) に対しても

$$\|T.E.\| \leq \tau_{REL} \|y\| + \tau_{ABS} \quad (11)$$

が常に満たされる様に h を制御している。図 3 にステップサイズ制御のロジックを示した。

附録2 常微分方程式を解くソフトウェア

本資料で使用された 7 つの倍精度計算用サブルーチンは当所計算機システム FACOM 230-75 の FORTRANH, OPT = 2 によりコンパイルされ、ファイル(ディスクバック)に SL 形式で登録されているので次の制御文により使用することが出来る。：

¥ LIED SL=ON, LF=K000636, SPFODEC,
LV=USER01

または

¥ APPLIED SL=ON, LF=K000636. SPFOD
EA, LV=USER01.

各サブルーチンの仮引数はかなり複雑なのでユーザーの便宜のため以下の3つの制御ルーチンを作製し上のファイルに加えた。まずRK4(5), RK7(8)およびShampine-Gordonコードは

```
CALL INTEGD(F,NEQS,X,T,TOUT,  
            REVERR,ABSERR,IFLAG,  
            WORK,DT,IMETH)
```

により呼ばれる。ここで

F	ユーザーの用意する導関数計算のサブルーチン。独立変数Tと従属変数Xが与えられた時、導関数DXを計算。Fの呼び方はCALL F(T,X,DX)。
EXTERNAL F	が必要。
NEQS	従属変数Xの次元
X(I)	従属変数(1次元アレイ)の初期値、実行後終端値に置き換えられる。
T	独立変数の初期値、実行後終端値(TOUT)に置き換えられる
TOUT	独立変数の終端値
REVERR	τ_{REL}
ABSERR	τ_{ABS}
IFLAG	出力情報パラメータ、即ち正常に終れば($T = TOUT$) IFLAG = 2, それ以外 IFLAG ≠ 2となりエラーメッセージを印字
WORK(I)	作業領域。RK4(5), RK7(8), Shampine-Gordonに応じて夫々少くとも $7 * NEQS$, $14 * NEQS$, $21 * NEQS + 100$ 必要
DT	初期ステップサイズ。Shampine-Gordonには不要
IMETH	スキームの選択、即ち = 1 Shampine-Gordon = 4 RK4(5) = 7 RK7(8)

注1. IMETH = 4, 7 即ちRunge-Kutta法に對してのみREVERR = ABSERR = 0.0 D0 を指定すれば、固定ステップサイズDTの積分が可能。

注2. INTEGDは全要素に関する打切り誤差評価に基づきステップサイズ制御をしているが、状態ベクトルのNEQS 1番からNEQS 2番目の要素に関してのみ制御するようコードを拡張することは容易である。例えば、人

工衛星の運動はある慣性座標系において

$$\dot{R} = V$$

$$\dot{V} = A$$

によって記述される。ここで R, V, A は夫々3次元位置、速度、加速度ベクトルである。この時、速度ベクトルに関してのみステップサイズ制御を行ない、結果的に位置ベクトルに関しても所要精度を達成し得る。

このような目的のために次のルーチンを用意した。

```
CALL INTEGV(F,NEQS,X,T,TOUT,REVERR,  
            R,ABSERR,WORK,DT,IMETH,NEQS1,  
            NEQS2)
```

注3. Shampine-Gordon は極めて強力なので単体としても使用することが多い。その場合の呼び方は、

```
CALL ODE(F,NEQS,X,T,TOUT,REVERR,  
         ABSERR,IFLAG,WORK,IWORK)
```

ここでWORK(I), IWORK(I)は作業領域で夫々少くとも $21 * NEQS + 100$, および5必要。DTは不要であることに注意。

Bulirsh-Stoerコードの呼び方は、

```
CALL BULSTD(F,NEQS,X,T,TOUT,EPS,  
            IFLAG,SP,WORK,DT,NERROR)
```

ここで

X(I)	従属変数(1次元アレイ)
EPS	τ_{REL} (または τ_{ABS})
SP	中間の出力指定。即ちSP間隔毎にXを印字。不要ならば必ずSP = 0.0 D0とせよ
NERROR	誤差評価規準の選択。即ち NERROR = 1, 2, 3 に応じて夫々 $ \delta_j / (X_j)_{max} < EPS$, $ \delta_j / X_j < EPS$, および $ \delta_j < EPS$, が選ばれる。ここで δ_j はある意味の局所打切り誤差を表わす

DT 初期ステップサイズ、但し
 $DT > |TOUT - T| / 2^{15}$

WORK(I) 作業領域。少くとも $40 * NEQS$ 必要

Gear コードの呼び方は、

```
CALL GEARD(F,NEQS,X,T,TOUT,EPS,  
            IFLAG,WORK,DT,MF)
```

ここで

X(8, I)	従属変数、8行NEQS列の2次元アレイの第1行目に必ずセイブせよ
---------	----------------------------------

EPS	τ_{REL} (または τ_{ABS})	$ \delta_j / X_j < EPS$, および
DT	初期ステップサイズ, 但し $20.D0 > DT > 1.D - 1.6$	$ \delta_j / (X_j)_{max} < EPS$
MF	スキームの選択, 即ち MF = 0 Adams-Basforth-Moulton MF = 1 stiffness 注3 MF = 2 stiffness 注4	スキームの選択, 即ち MF = 10, 11, 12, 13 non-stiff システム MF = 20, 21, 22, 23 stiff システム
WORK(I)	作業領域。MF = 0 の時少くとも $13 * NEQS$, それ以外は少くとも $2 * NEQS * (NEQS + 6)$ 必要	IFLAG 出力情報パラメータ。即ち正常に終れば (T = TOUT) IFLAG = 0, それ以外 IFLAG < 0 となりエラーメッセージを 印字

注3. stiff および non-stiff システムに対する多段公式法が動らく。MF = 1 およびその時に限りユーザーはサブルーチン PEDERV を用意しなければならない。これは導関数の Jacobian を計算するもので、その行列の I 行 J 列の要素は必ずしも 1 次元アレイ PW の $I + M * (J - 1)$ 番目にセイブする。PEDERV の呼び方は、

```
CALL PEDERV(T, X, PW, M)
```

注4. MF = 2 の場合、機能としては MF = 1 と全く同一であるが相違は注3で述べた Jacobian の計算をここでは数値微分で行なうにある。従って PEDERV は不要である。

上記コード GEARD は最大次数 7(Adams), 6(stiff) である。第2節で述べたように Hindmarsh, Byrne^{6),7)} は、それを発展させて GEAR, EPISODE を開発した。GEAR は現時点ではアクセスできない。EPISODE の呼び方は

```
CALL EDRIV(F, NEQS, T, DT, X, TOUT, EPS,  
NERROR, MF, IFLAG)
```

ここで

X(I)	従属変数 (1 次元アレイ !)
EPS	τ_{REL} (または τ_{ABS})
DT	初期ステップサイズ (充分小)
NERROR	誤差評価規準の選択, 即ち NERROR = 1, 2, 3 に応じて夫々 $ \delta_j < EPS$,

MF	$ \delta_j / X_j < EPS$, および スキームの選択, 即ち MF = 10, 11, 12, 13 non-stiff システム MF = 20, 21, 22, 23 stiff システム
IFLAG	出力情報パラメータ。即ち正常に終れば (T = TOUT) IFLAG = 0, それ以外 IFLAG < 0 となりエラーメッセージを 印字

ユーザーは、ソースリストの注釈を読んで各パラメータの意味を正確に把握して後使用することが肝要である。本資料で使用している全コードのソースリストは次のディスクファイルから得られる(暫定的で将来変更されよう)。

ファイル名 : K000617.NMFODES

(VOL = USER02)

以下に使用上の若干の注意を記す。

注6. 各ルーチン (EPISODE を除く) は次の COMMON 文を持つ。

```
COMMON /CINTEG /NTEXEC, NSTEP,  
NREJT
```

このうち変数 NTEXEC, NREJT, NSTEP は夫々 TOTAL EXECUTION TIME (ミリ秒), NO OF REJECTED STEPS (RKのみ), NO OF STEPS を表わす出力パラメータである。

注7. 各ステップ毎の値を出力したい場合、そしてその時に限りユーザーはサブルーチン USER を用意する。呼び方は、CALL USER(T, X, NSTEP, NEQS)。

注8. EPISODE を除く各コードに共通して、計算の実行が正常 (T = TOUT) に終れば IFLAG = 2 となり、それ以外は何らかのエラーメッセージが印字される。

注9. ユーザーのプログラムが単精度でも積分の部分のみ倍精度計算することを試みることは意義がある。

Table 1 Summary of the Current State of the Art for ODE Solvers

Package	Inventor	Method of Numerical Integration	Feature
RK4 (5), RK7(8)	McKenzie	Runge-Kutta-Fehlberg single-step formula	suitable for non-stiff system
ODE	Shampine-Gordon	Adams-Basforth-Moulton multi-step formula	suitable for non-stiff system, predictor of order q and corrector of order q+1 where q must be less than 13
GEAR	Gear	Adams-Basforth-Moulton or stiff methods of Gear	predictor of order q and corrector of order q where q must be less than 8 or 7 for Adams or stiff methods respectively
GEAR	Hindmarsh	Implicit Adams or Backward Differentiation multi-step formula(BDF)	maximum order of the method is 12 for Adams methods and 5 for BDF methods, option for the G-stops
EPIODE	Hindmarsh-Byrne	Implicit Adams or BDF multi-step formula	same as GEAR by Hindmarsh except the strategy of the selection of step-size and order
Bulirsch-Stoer	Bulirsch-Stoer(-Fox)	Richardson extrapolation single-step formula	suitable for non-stiff system, rational extrapolation of order 6

Table 2 - 1. $y' = -y$, $t \in [0, 100]$, With an
Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	- ^a	168	39	1.87
	93	821	203	.435
	7.97	7418	1853	3.83
RK 4(5)	96	143	23	.169
	-	262	42	.163
	93	700	115	.209
RK 7(8)	-	256	16	.248
	93	309	21	.474
	-	465	33	.244
ODE	-	99	47	3.44
	94	139	69	4.79
	96	233	115	2.55
EPISODE #10	96	231	69	3.61
	96	351	115	6.96
	425	907	384	3.89
Bulirsh- Stoer	93	731	24	2.48(-4)
	97	1401	33	3.71(-7)
	396	3841	67	1.73(-9)

a 1 msec 以下

b Implicit Adams method with Gauss-Siedel
iterations

Table 2 - 2. $y' = -y$, $t \in [0, 100]$, With a
Relative Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	196	1252	313	.324
	1597	12260	3065	.0300
	15197	122320	30580	0.00298
RK 4(5)	97	455	75	.179
	197	1560	260	1.74
	696	5940	990	2.26
RK 7(8)	-	403	31	24.7
	111	819	63	12.5
	178	1820	140	9.64
ODE	NOT EXECUTABLE			
EPISODE #10	NOT EXECUTABLE			
Bulirsh- Stoer	NOT EXECUTABLE			

Table 3. $y' = y$, $t \in [0, 100]$, With a
Relative Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	195	1220	305	.275
	1696	12224	3056	.0296
	13897	122288	30572	.00298
RK 4(5)	97	307	42	26.4
	196	1416	236	34.4
	596	5796	966	2.67
RK 7(8)	68	312	24	7.51
	124	741	57	6.36
	196	1742	134	7.19
ODE	6	137	68	7.80
	188	288	143	12.2
	196	535	267	3.45
EPISODE #10	186	247	116	42.5
	328	595	279	295
	848	1709	836	30.1
Bulirsh-Stoer	96	705	23	.0319
	196	1363	30	.0108
	196	2105	39	.0193

Table 4. $y' = t(1-y) + (1-t)e^{-t}$, $t \in [0, 100]$
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	463	2458	475	6.28
	529	3189	669	28.8
	1897	11411	2747	7.53
RK 4(5)	351	2289	344	.251
	440	2463	378	.254
	496	3421	546	.245
RK 7(8)	897	4915	247	.757
	698	4922	254	.663
	894	5438	290	.586
ODE	798	1742	317	6.32
	697	1725	314	5.48
	896	1867	375	7.73
EPISODE #10	1297	5459	1572	3.27
	1596	6095	1877	4.93
	1447	5979	1797	11.5
Bulirsh-Stoer	1396	14165	177	.0329
	1997	16214	211	1.03(-5)
	6897	49667	693	1.04(-8)

Table 5. Integrable Singularity Problem,
 $t \in [-1, 1]$, With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	—	240	45	.852
	288	1413	327	.905
	2394	12711	3150	.571
RK 4(5)	96	256	26	1.55
	96	701	81	.725
RK 7(8)	NOT EXECUTABLE ^a			
	NOT EXECUTABLE ^b			
ODE	—	68	31	39.6
	195	373	173	118
	601	980	466	298
EPISODE #10	43	145	44	11.5
	271	591	189	48.7
	779	1401	523	161
Bulirsh-Stoer	NO CONVERGENCE			

a MORE THAN MAXREJ REJECTED STEPS IN
 ONE TIME T = -0.D0

b IFLAG = 2 WITH LARGE ABSOLUTE ERRORS

Table 6. $\mathbf{y}' = U\mathbf{z} - UBU\mathbf{y}$, $t \in [0, 10]$,
 With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	118	172	40	2.27
	757	1104	276	.00340
	6995	10764	2691	3.51(-4)
RK 4(5)	96	154	24	.402
	196	251	41	.246
	495	780	130	.0867
RK 7(8)	196	244	16	1.05
	195	322	22	1.13
	296	429	33	.225
ODE	83	108	52	4.71
	216	202	100	1.58
	404	340	169	2.64
EPISODE #20	196	301	90	4.47
	374	405	138	9.34
	703	709	310	24.5
Bulirsh-Stoer	495	829	21	5.25(-4)
	995	1605	33	2.32(-8)
Bulirsh-Stoer	86195	145237	1639	2.94(-5)

Table 7. $y'' + 11y' + 10y = 0, \quad t \in [0, 100],$
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO.OF STEPS	NORMALIZED MAX.ERROR
RK 2(4)	80	1053	210	1.87
	96	911	221	.435
	1096	7508	1871	1.91
RK 4(5)	96	1019	154	.170
	—	340	55	.234
	291	2072	322	.0381
RK 7(8)	596	3604	184	.0599
	396	3463	175	.134
	496	3734	194	.132
ODE	597	1240	74	.956
	596	1318	117	.562
	696	1513	216	.632
EPISODE #21	21	103	35	1.84
	168	287	92	3.81
	313	705	231	9.50
Bulirsh- Stoer	1196	9855	130	.00199
	796	10137	137	1.28(-5)
	1297	11463	157	2.83(-9)

a BDF using Newton-Raphson method with
the analytic partials

Table 8 - 1 Two-Body Problem ($\epsilon = 0.0$), 2 Periods,
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO.OF STEPS	NORMALIZED MAX.ERROR
RK 2(4)	93	312	78	.440
	796	2976	744	.0351
	7995	29656	7414	.00342
RK 4(5)	88	132	22	21.3
	103	468	78	27.6
	496	1800	300	28.2
RK 7(8)	54	194	14	19.4
	100	338	26	64.4
	198	689	53	45.4
ODE	36	73	36	61.5
	101	141	70	12.5
	255	273	136	39.1
EPISODE #10	97	139	49	276
	417	411	184	247
	731	667	287	2980
Bulirsh- Stoer	229	933	23	1.03(-4)
	596	2873	47	4.69(-8)
	56095	294147	3446	8.76(-7)

Table 8 - 2. Two-Body Problem ($\epsilon = 0.3$), 2 Periods,
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	96	337	82	.615
	596	3140	785	.0371
	10096	31288	7822	.00479
RK 4(5)	44	186	26	4.22
	48	503	83	6.85
	595	1914	319	8.05
RK 7(8)	97	269	17	39.4
	102	472	28	93.2
	290	943	55	145
ODE	84	119	58	61.4
	231	257	127	24.1
	474	451	224	74.9
EPISODE #10	189	185	73	495
	431	421	177	302
	1098	923	399	377
Bulirsh- Stoer	105	953	23	.000765
	596	2585	43	180(-8)
	87796	465822	5389	8.38(-6)

Table 8 - 3. Two-Body Problem ($\epsilon = 0.6$), 2 Periods,
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FON CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	187	520	112	1.83
	1496	4232	1058	.116
	14696	42228	10557	.0151
RK 4(5)	99	269	34	632
	201	757	107	149
	755	2406	401	186
RK 7(8)	88	332	20	343
	148	611	35	244
	386	1233	69	329
ODE	125	183	87	1040
	386	381	186	681
	688	661	327	171
EPISODE #10	210	263	86	1810
	674	657	258	2590
	1933	1443	620	2520
Bulirsh- Stoer	298	1390	25	.00484
	596	3161	51	6.21(-6)
			NO CONVERGENCE	

Table 8 - 4. Two-Body Problem ($\epsilon = 0.9$), 2 Periods,
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	389	907	187	571
	3188	7096	1774	5.83
	31698	70896	17724	.581
RK 4(5)	190	405	55	123
	417	1261	166	275
	1473	3702	617	1220
RK 7(8)	197	585	33	3280
	299	954	54	8060
	598	1956	108	8890
ODE	328	356	169	4730
	798	636	309	7630
	1381	1116	50	6080
EPISODE #10	405	429	129	3330
	1118	1071	385	143551
	3168	2107	865	127776
Bulirsh-Stoer	789 1394	2994 5249	36 74	18.9 4.8(-5) NO CONVERGENCE

Table 9. Restricted Problem of Three Bodies,
1 Period, With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR	$\frac{\Delta J^a}{\tau}$
RK 2(4)	287	1144	244	.0773	.0721
	2758	9280	2320	.00691	.00682
	26130	92692	23173	4.99(-5)	.000590
RK 4(5)	94	528	68	1.66	.731
	394	1586	211	2.85	2.39
	1295	4746	791	9.07	4.16
RK 7(8)	95	722	38	3.74	4.42
	295	1170	66	31.4	12.1
	795	2364	132	7.55	5.41
ODE	280	377	180	27.8	32.9
	747	814	398	35.2	20.9
	1361	1361	172	40.8	19.9
EPISODE #10	321	561	165	175	70.2
	1192	1227	449	508	92.4
	2811	2193	898	144	141
Bulirsh-Stoer	1095 1793	3895 6417	48 96	9.56(-4) 7.22(-5)	4.89(-4) 320(-7) NO CONVERGENCE

a $\max_k |J_k - J_0| / \tau$, $k = 1, 2, \dots$ where $J_k = J(t_k)$

Table 10. Euler Equation, 7 Periods,
With an Absolute Error Test

$10^{-3}/10^{-6}/10^{-9}$	TOTAL TIME	FCN CALLS	NO. OF STEPS	NORMALIZED MAX. ERROR
RK 2(4)	186	4846	277	.123
	1679	42035	2600	.0397
	1797 ^a	14812	3703	6.41(-5)
RK 4(5)	90	1606	73	115
	275	5943	241	28.9
	882	22806	948	25.7
RK 7(8)	89	2796	39	179
	278	4352	74	85.0
	483	7902	154	312
ODE	188	1812	238	9.44
	581	3948	490	10.7
	1078	6636	826	18.9
EPISODE #10	281	1787	791	159
	792	4563	1969	38.6
	2484	11597	5041	938
Bulirsh-Stoer	581	14028	119	4.19(-4)
	1183	30996	168	112(-6)
	6296 ^a	46927	588	8.45(-8)

^a only 1 period

Table 11. Two-Body Problem, 2 Periods,
With Fixed Stepsize

$h_1/h_2/h_3$	MAXIMUM $\epsilon = 0.0$	ERROR $\epsilon = 0.3$	NORM $\epsilon = 0.6$	$\epsilon = 0.9$	TOTAL TIME	FCN CALLS	NO. OF STEPS
RK 2(4)	10.7(-3)	1.92(-2)	2.94(0)	63.9	—	240	60
	5.80(-5)	5.74(-5)	3.58(-1)	33.8	131	481	120
	1.09(-5)	7.60(-4)	5.09(-2)	10.7	219	721	180
RK 4(5)	7.90(-5)	9.02(-5)	7.31(-2)	43.9	34	360	60
	2.54(-6)	2.94(-6)	9.26(-3)	9.31	194	721	120
	3.37(-7)	3.94(-7)	1.39(-3)	6.21	296	1081	180
RK 7(8)	7.95(-9)	9.61(-7)	4.27(-3)	46.0	196	780	60
	2.33(-11)	2.67(-9)	3.58(-5)	14.6	396	1561	120
	8.11(-13)	8.39(-11)	1.26(-6)	38.7	596	2341	180

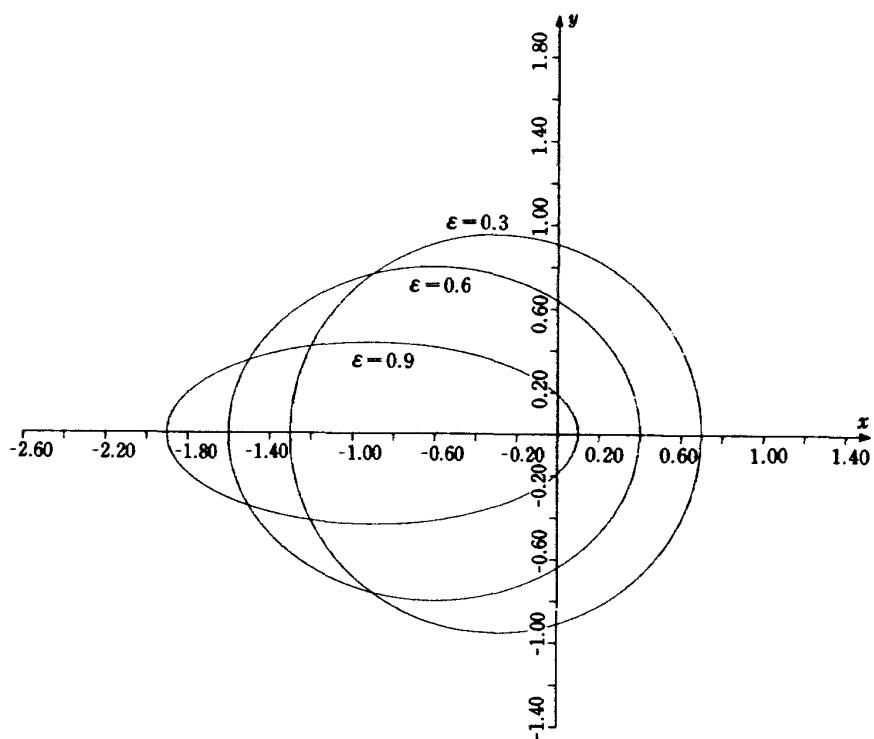


Fig 1. Newton's equations of motion for the two body problem in elliptic motion.

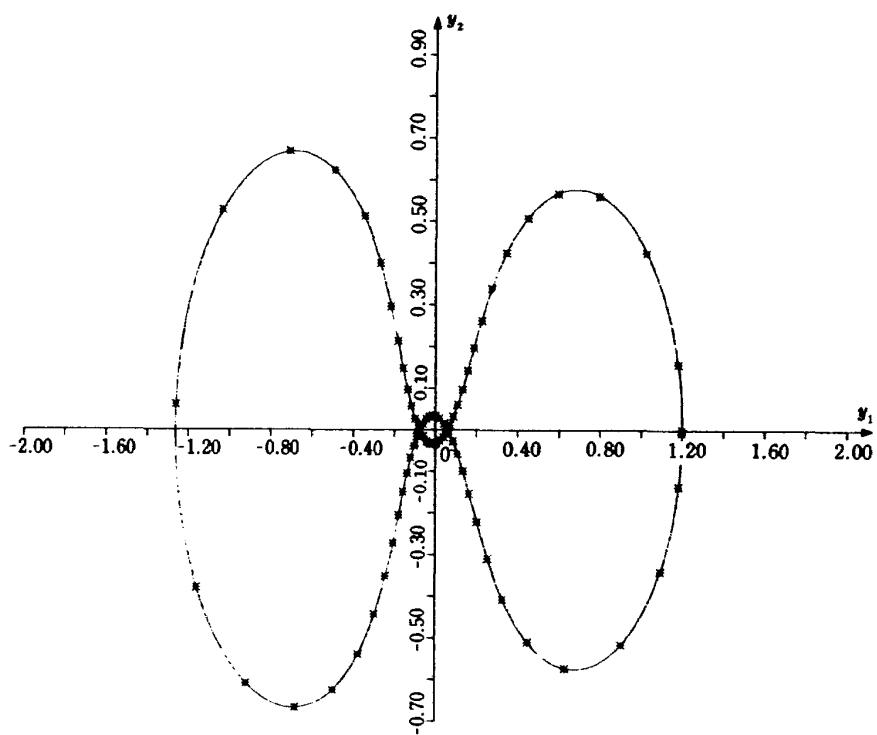


Fig 2-1 The solution of the restricted three body problem by RK7(8) (Fehlberg).

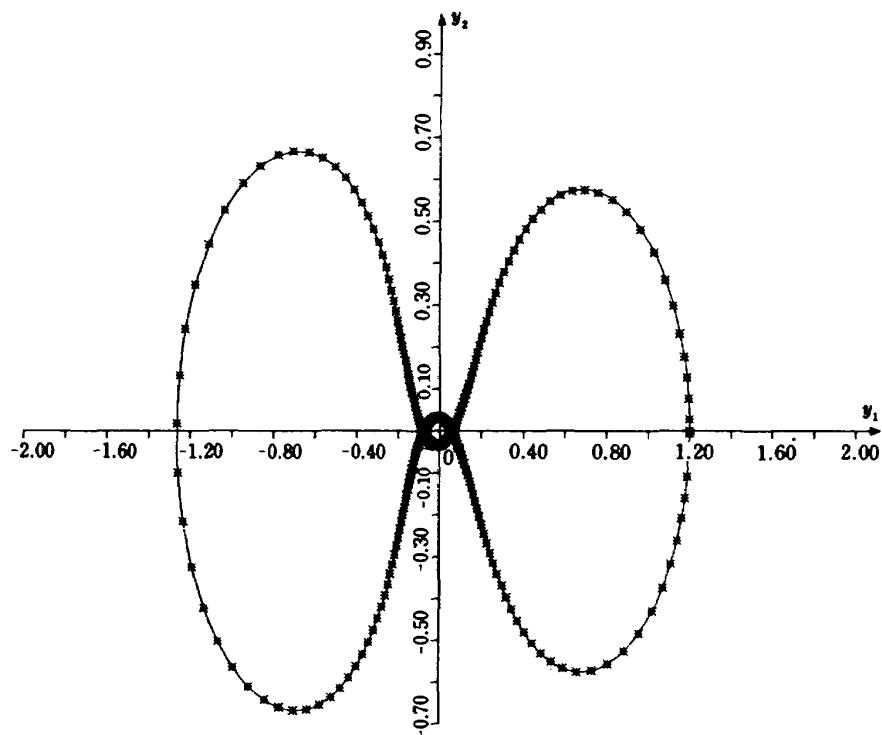


Fig 2-2 The solution of the restricted three body problem
by RK4 (5) (D.G.Hull).

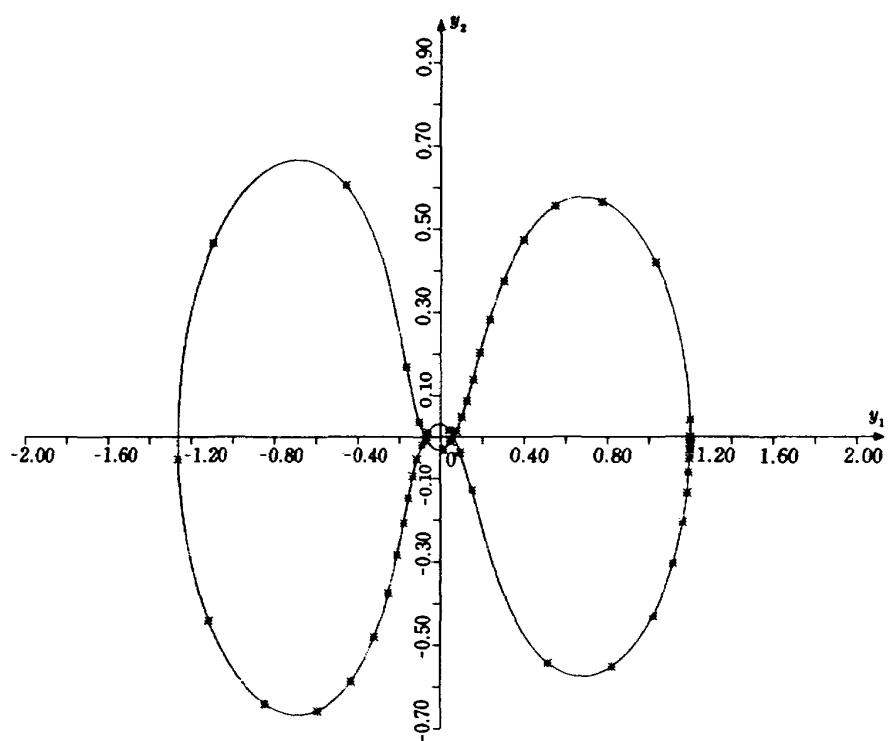


Fig 2-3 The solution of the restricted three body problem
by Bulirsh/Stoer.

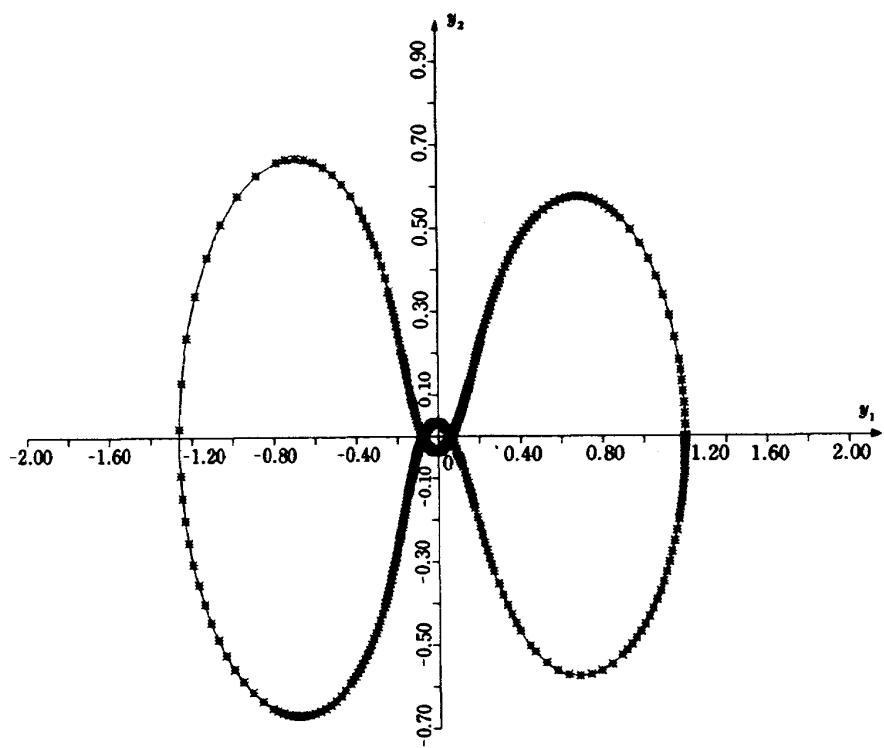


Fig.2-4 The solution of the restricted three body problem
by Shampine/Gordon.

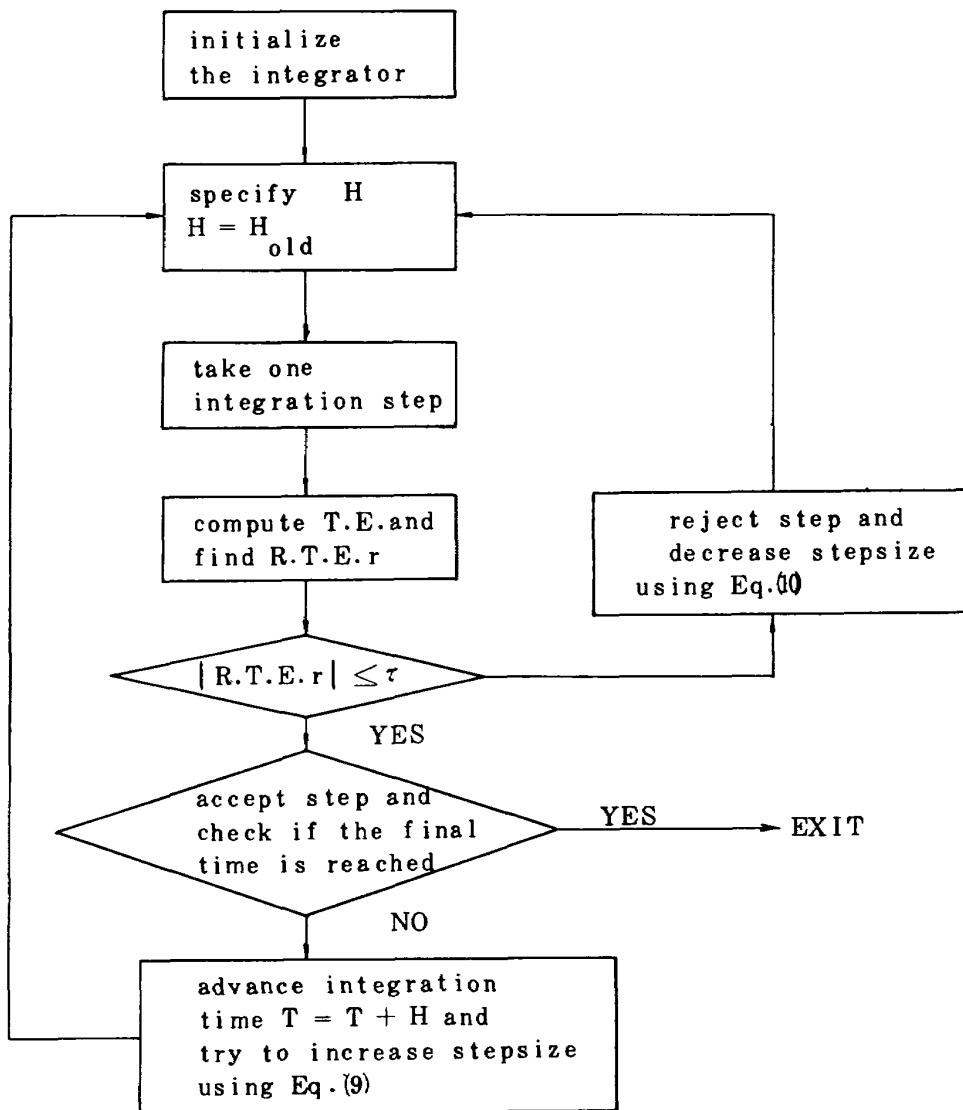


Fig.3. Stepsize Control Logic by D.G.Hull

航空宇宙技術研究所資料353号

昭和53年6月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武藏野三鷹(0422)47-5911(大代表) 182

印刷所 株式会社 東京プレス
東京都板橋区桜川 2~27~12

Printed in Japan

This document is provided by JAXA.