

UDC 517.9:
62-503:
629.7

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-249

一般化 Newton-Raphson 法の計算機容量問題に
対する改良最適化アルゴリズム

志 甫 徹

1971 年 10 月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

既 刊 報 告

- TR-230 アナログトルク平衡方式による浮動振子型加速度計の研究
Study on the Analog Torque Rebalance Floated Pendulum Type Accelerometers for Guidance and Control Applications 1971年1月 大月正男, 鈴木孝雄
円居繁治
- TR-231 高温タービンノズル翼二次元翼列試験
(第1報 対流冷却翼の実験結果について)
Two-Dimensional Cascade Test of an Air-Cooled Turbine Nozzle
(Part I On the Experimental Results of a Convection-Cooled Blade) 1971年1月 吉田豊明, 高原北雄
能瀬弘幸, 井上重雄
三村富嗣雄, 臼井 弘
- TR-232 高温タービンノズル翼二次元翼列試験
(第2報 対流冷却翼の数値計算およびアナログ法による翼表面温度分布について)
Two-Dimensional Cascade Test of an Air-Cooled Turbine Nozzle
(Part II On the Temperature Distributions of a Convection-Cooled Blade by Numerical Calculation and Analogue Simulation Test) 1971年1月 吉田豊明, 高原北雄
能瀬弘幸, 井上重雄
三村富嗣雄, 臼井 弘
- TR-233 PSD 法の大气乱れに対する航空機強度設計への応用とその問題点
Studies on PSD Method to Aircraft Structural Design for Atmospheric Turbulence 1971年1月 竹内和之, 山根皓三郎
- TR-234 Green 関数をラプラス方程式に適用した解法による冷却タービン翼の温度分布の計算
A Calculation of Temperature Distribution with Applying Green Function to Two-Dimensional Laplace's Equation 1971年1月 西村英明
- TR-235 自動着陸予備実験
(1) 電波高度計の性能試験
Preliminary Experiments for Automatic Landing
(1) On the Performance Tests of Radio Altimeters 1971年4月 樋口一雄, 堀川勇壮
森 幹彦, 小川 鑞
真柳光美, 渡辺 顕
名越孝行
- TR-236T Small-Strain Deformations Superposed on Finite Deformations of Highly Elastic Incompressible Materials
—Part I Constitutive Equations— Jun. 1971 Tastuzo KOGA
- TR-237 自由飛行模型 FFM-10 による縦の動安定特性の飛しょう実験
Free Flight Tests on Longitudinal Dynamics Characteristics of FFM-10 Model 1971年4月 河崎俊夫, 埴 武敏
斎藤秀夫, 高島一明
河本 巖
- TR-238 リフトジェットエンジン JR-100H の動特性について
Dynamic Characteristic of Lift Jet Engine JR 100H 1971年5月 西尾健二, 杉山七契
大畑敏美, 遠藤征紀
越 沼 威
- TR-239 昇華アブレーションの直接解法
A Direct Calculation of Sublimating Ablation 1971年6月 久保田弘敏
- TR-240T A Method for the Calculation of Lifting Potential Flow Problems
—Part I Theoretical Basis— Jul. 1971 Masao EBIHARA
- TR-241 YS-11A-500/600 型主翼疲れ試験
(I セーフライフ試験荷重と試験方式)
Full-Scale Fatigue Test of YS-11A-500/600 Turboprop Transport Wing
(I Safe-Lie Fatigue Test Loads and Test Method) 1971年7月 竹内和之, 野原利雄
朝田洋雄

一般化 Newton-Raphson 法の計算機容量問題に 対する改良最適化アルゴリズム*

志 甫 徹**

Modified Optimization Algorithm for Computer Storage Problems in a Generalized Newton-Raphson Method

by Toru SHIHO

A modified algorithm for generalized Newton-Raphson method (or quasi-linearization) which essentially eliminates the necessity of computer memory storage is proposed in this paper.

Optimization problems in modern control theory can be reduced to non-linear two-point boundary value problems. The generalized Newton-Raphson method is a powerful means for solving such problems. However, in case of complicated problems with increased system order, such as trajectory optimization problems of a space vehicle, this method requires large computer storage in order to solve the simultaneous differential equations and to evaluate the convergence of the solution.

In this paper, a generalized Newton-Raphson algorithm is reformed by introducing the concepts of quasi-reference and transition matrix, and by trading the memory storage for computation time. This reformation eliminates most of the computer memory storage and also improves the computational speed.

1. 序 論

工学の分野の問題で、境界値問題に帰着できるものは非常に多い。宇宙工学の分野に例をとってみると、軌道最適化、軌道決定、軌道推定、最適制御、システム同定、多段ロケットの最適質量配分の問題などと、列挙していくと数多くの実例が見出される。これらは最適化問題と総称されるが、一般に、最適化問題は、微分方程式の境界値問題を解くことに帰着できることが知られている¹⁾。

現在までに、最適化問題の解法として、古典的な変分法、その拡張一般化と考えられる Pontryagin の最大原理、Bellman の最適原理に基づく動的計画法 (Dynamic Programming)、Bryson や Kelly らの最急降下法 (Steepest Descent Method または Gradient Method)、Bellman や Kalaba らの Quasilinearization 法 (Generalized Newton-Raphson Method ともいう)、

Breakwell, Speyer, Bryson らの二次変分法 (Second Variation Method) など、そのほかにも多くの手法が提案されている。これらの概要に関しては、例えば、参考文献 (5, 6) を参照されたい。

解析的な解法としての変分法および最大原理は、きわめて簡単な問題であれば解が求められるが、問題が少し複雑になると解析的には解けなくなり、数値解法によらざるを得なくなる。したがって、宇宙工学の分野におけるような状態変数の多い複雑な問題では、定式化は最大原理あるいは変分法に依るとしても、解を得るためには数値的解法に頼らざるを得ない。そこで、電子計算機の発達とあいまって、数値的な諸手法が脚光をあびるようになってきた訳であるが、それぞれ一長一短があって、いずれとも優劣がつけがたい。動的計画法 (DP) は、最適原理自体は定式化の段階では非常に有効な武器であるが、数値解法としてみれば、莫大な計算機容量を必要とするので、実際に必要とされる複雑な問題は解けない場合が多い。最急降下法は、最適解に近づくにつれて、傾斜 (Gradient) がゼ

* 昭和 46 年 9 月 2 日受付

** 計測部

ロに近づき収束が悪くなって最終的な解を得るのに時間がかかるという難点がある。また、二次変分法 (Second Variation Method) は、所要記憶容量は少ないといわれているが、二次変分まで用いるので、ベクトル方程式の場合には計算が多少煩雑になってくる難点がある。

なお、二次変分法は最急降下法より収束が速いといわれ、最急降下法で最適解に近づいてから二次変分法に切換えて収束を速めるのに使われているようである。一方、Quasilinearization 法あるいは一般化 Newton-Raphson 法と呼ばれる方法は、二次収束性を持つ、きわめて収束性の良い方法であるが、問題によっては第 0 近似値の選び方によって局所的極値に収束してしまう場合があり得る。

一般に、最適化問題の数値的解法には、直接法 (Direct Method) と間接法 (Indirect Method) の二つの接近の仕方がある。直接法は、与えられた方程式と所期最終条件から出発して、評価関数を最小または最大にするように逐次近似によって修正して行く方法であって、最急降下法は直接法の一例である。言い換えると、逐次近似の各段階での解は、与えられた拘束条件は満たしているが、最適の条件は満たしていないので、評価関数を最適にするように操作可能な変数を修正して最適解に収束させて行く方法である。一方、間接法は、最適値を与えるための必要条件、すなわち、最大原理という Hamilton 正準方程式、あるいは、変分法という Euler-Lagrange 方程式から出発して、境界条件を満たすように逐次近似によって修正して行って最適解を求める方法である。この定式化を採ると、最適性の必要条件と境界条件が、一般に非線型二点境界値問題を構成する。したがって、解法の眼目は任意に仮定した初期条件が、残りの規定された終端境界条件を満たすようになるまで、システムティックに修正して行く Logic にある。本報告で扱う一般化 Newton-Raphson 法は、間接法の典型である。

一般化 Newton-Raphson 法 (Quasilinearization 法* ともいう。以下、GNR 法あるいは QL 法と略記する) は、二点境界値問題を初期値問題として解く方法で、逐次近似によって近似初期値問題の解を二点境界値問題の解に収束させるものである。これは代数方程式の根を数値的に求める Newton-Raphson 法の拡張であり、その歴史的背景を探れば、1948 年に Kantrovich⁷⁾

が Banach 空間における作用素方程式 (operator equation) の求根法として Newton-Raphson 法を一般化したのが発端であるといわれている。1953年に Warga⁹⁾ が Kantrovich の一般化の特殊例として一階常微分方程式の初期値問題を解いた。一方、1949年に Hestines⁸⁾ が境界値問題に対する Algorithm を最初に提案して、これを Differential Variation と名づけた。Bellman や Kalaba^{10,11)} らは、最適化問題と関連づけて、さらに発展させて Quasilinearization 法の名の下に、最適化問題の解法として、また常微分方程式の二点境界値問題の解法として知られるようになった。理論的には、Kalaba¹¹⁾ が二階常微分方程式の二点境界値問題に対し、関数の単調性 (Monotonicity) と凸性 (Convexity) に基づいて収束性を証明した。さらに、Kenneth と McGill¹³⁾ が、縮小写像 (Contraction mapping) を用いて N 元二階常微分方程式の境界値問題に対する収束性の十分条件を発表している。

成書としては、1965年に Bellman と Kalaba の共著になるものが出版された³⁾。それまでの研究が集大成されている標準的なテキストである。この本には、偏微分方程式への応用にも一章がさかされている。その後、Lee⁴⁾ の著書も出版された。この本は化学プロセスへの応用に主眼が置かれ、プロセスのパラメータ推定の問題にも触れている。

宇宙工学への応用は 1960 年代に入ってから現われてきた。Bellman らが¹²⁾ 軌道決定問題を QL 法と最小自乗法とを組み合わせることにより多点境界値問題として定式化できることを示唆した。Long¹⁶⁾ は二点境界値問題として QL 法を実際的な軌道決定問題に応用して有効性を示した。Bellman の多点境界値問題を著者が^{27,28)} さらに発展させて誤差を含んだ観測値によって与えられる関数型の多点境界条件に拡張して、逐次近似による非線型フィルタを構成し軌道推定問題を解いた。最終時間自由問題に対する拡張としては、McGill と Kenneth が¹⁴⁾、低推力軌道移行に例をとり、VTA (Variable Time of Arrival) 問題を解いた。これは最終時間を微小ずつ変化させて行く方法である。Long¹⁵⁾ は、さらに一般化して、時間も状態変数の中に繰り入れて解くことを提案している。

McGill¹⁷⁾ は、状態変数に制限のある問題に拡張し、さらに、Kenneth と McGill^{17,19)} が、制御変数にも制限のある問題に拡張した。

Van Dine, Fimple, Edelbaum¹⁸⁾ らは、GNR 法と差分法とを組み合わせ、不安定な方程式の場合の難点を除去することを試みた。

* Bellman, Kalaba らの影響下にある一派は Quasilinearization の名称を好んで用いている。ただし、内容は全く同じものである。

1967年頃から Lewallen, Tapley らの一連の論文が公表された²⁰⁻²⁴⁾。彼らは、QL 法と他手法との比較を行ない、QL 法の有効性を数値的に示した^{20,21)}。Lewallen は、^{22,23,24)} 終端条件を状態変数と時間の関数で与えられる関数型終端条件に拡張し、VTA 問題を解いた。また、逐次近似の各段階で計算により与えられる修正量を、そのまま適用すると計算誤差の蓄積などの影響によって、解が振動して収束しない場合があるので、計算修正値を割引いて実行するという Percentage Correction と名づける方法を考案している。

以上で、GNR 法 (または QL 法) の位置づけ、歴史的背景、解かれるべき問題の特質に応じた GNR 法の拡張について概観した。本報告では、最適化法の一つであり、二点境界値問題の数値解法の一つである GNR 法をとりあげて、所要計算機記憶容量と計算時間に関しての改良の結果について報告するものである。GNR 法をとりあげた理由は、二乗のオーダーで収束して行くという二次収束性を有し、きわめて短時間に解が収束する点と、逐次近似計算の Logic が比較的簡単な点にある。GNR 法では、逐次近似計算の各ステップで前回の近似値を必要とするので、各グリッドごとにストアしておかなければならない。状態変数の数が多い複雑な問題では計算機記憶容量が問題点となる。電子計算機を使用する場合に、ある意味において、所要記憶容量と計算時間との間に互換性があるといえる。すなわち、あるデータを記憶させておく代りに、そのデータを発生させるような計算を付加することによって、計算時間は増えるが記憶容量を減らすことが可能である。いずれにせよ、電子計算機には記憶容量と計算時間 (計算速度といい換えてもよい) の二大拘束条件があるので、使う立場からは Algorithm の工夫が問題点としてクローズ・アップされてくる。

最適化法に限って言えば、主な方法・原理は出揃った感があり、最近はその拡張および改良が行なわれ、研究は新しい方法・原理の開発から Algorithm の開発へと中心が移っている傾向にある。本報告は複雑なシステムの最適化問題、例えば、多段ロケットの軌道最適化の問題等を解く際に障害となる計算機容量の問題と、それに関連した計算時間の問題を、GNR 法の Algorithm の改良という観点からまとめたものである。

本報告の主題に関する文献はきわめて少ない。1961年に Bellman²⁾が、前回の近似値を記憶させる代りに記憶方程式 (Memory equation) を連立させて解くこ

とを示唆した。1966年に著者²⁷⁾が、軌道推定問題に GNR 法を応用する際に、原方程式を Bellman の線型近似式の代りに記憶方程式として良い結果を得ている。その後さらに検討を加え、改良を進めた結果が、本報告の内容である。この方法の Algorithm のみを取上げて公表するのは遅くなったが、すでに発表されている研究^{30,31)}において、計算手法として使用されて成果を挙げている。

なお、1966年に田辺²⁶⁾によって提案された逐次近似誘導理論は、繰り返し計算によって近似軌道を逐次修正して行き最適な誘導経路に収束させ、必要速度修正量を決定する方法であるが、この方法を一般化すれば、GNR 法の改良の積み重ねという回り道をとらずに、本報告の Algorithm に帰着できるものである。また、Baird³²⁾は、原方程式を記憶方程式として利用することにより記憶容量を減少させたことを報告している。このアイディアは、筆者がすでに試みた²⁷⁾ものであるが、Baird はスカラー微分方程式の場合について、収束性の証明を行なっている。

2. 記号

- a : 初期境界条件の規定値ベクトル (k 次元)
- a^i : 初期境界条件の規定された要素
- b : 終端境界条件の規定値ベクトル ($n-k$ 次元)
- b^j : 終端境界条件の規定された要素
- B : 終端点における残差ベクトル ($n-k$ 次元)
- c_R : 定数ベクトル (n 次元)
- c_R^* : c_R の未定要素から成る部分ベクトル ($n-k$ 次元)
- $H_R(t)$: 斉次解 ($n \times n$ 行列)
- H_{11}^* : H_R の要素から成る $(n-k) \times k$ 行列
- H_{22}^* : H_R の要素から成る $(n-k) \times (n-k)$ 行列
- h_{ij} : 斉次解行の ij 要素
- I : 単位行列
- K : 積分刻み数
- k : 初期条件で規定される要素の数
- M_1 : 連立微分方程式を解くための所要記憶容量
- M_2 : 収束判定のための所要記憶容量
- N, n : 状態変数ベクトルの次元 (要素の数)
- $p_R(t)$: 特殊解ベクトル (n 次元)
- T_H : 斉次解を求めるに要する積分時間
- T_N : 非線型微分方程式を解くに要する積分時間
- T_P : 特殊解を求めるに要する積分時間
- T_{integ} : 連立微分方程式を解く要する総積分時間

T_{M2} : 収束判定用記憶容量を計算で代替したために要する積分時間

t : 時間 $t_0 \leq t \leq t_f$

t_0 : 初期時間

t_f : 終端時間

$X(t)$: 状態変数ベクトル (n 次元)

$X_R(t)$: 一般解ベクトル (n 次元)

$X_{qR}(t)$: 準基準解ベクトル (n 次元)

α : 二次収束性の定数 ($\alpha > 1$)

$\Delta X(t)$: 相続く近似解の差, $\Delta X = X_R - X_{R-1}$

$\Delta X_R(t)$: 準基準解からのずれ, $\Delta X_R = X - X_{qR}$

$\delta X(t)$: $\Delta X_R(t)$ の部分ベクトル ($n-k$ 次元)

$\Phi_R(t, t_0)$: 遷移行列 ($n \times n$ 行列)

$\Phi_{\mu\nu}$: 遷移行列の要素から成る $(n-k) \times (n-k)$ 行列

φ_{ij} : 遷移行列の ij 要素

ρ : 所望収束精度

$J(X)$: Jacobian

$(\dot{\quad})$: 時間に関する一階微分

$(\quad)^T$: ベクトルの転置

添字

ν_i : ベクトルの要素の順序を示す添字

μ_j : ベクトルの要素の順序を示す添字

q : 準基準 (Quasi-reference) を示す添字

R : 逐次近似計算回数

H : 斉次解

P : 特殊解

N : 非線型微分方程式

3. 一般化 Newton-Raphson 法の概要

本節では, GNR 法, あるいは QL 法と呼ばれる最適化手法の概要を述べる。方程式は, 現代制御理論の流儀にしたがって, 一階ベクトル常微分方程式で表示する。高階の常微分方程式は, 状態変数ベクトルの概念を導入することにより, すべてこの形に変換できることが知られている。

3.1 問題の設定——二点境界値問題

最適制御問題に限らず, 一般に最適化問題は非線型常微分方程式の二点境界値問題に帰着できる^{1,2,5,6,29)}。したがって, 問題は次のように定式化することができる。

一般に非線型なベクトル常微分方程式*

$$\frac{dX}{dt} = f(X, t), \quad t_0 \leq t \leq t_f \quad (3.1)$$

を, 境界条件

$$X^{\nu_i}(t_0) = a^{\nu_i}, \quad i=1, 2, \dots, k \quad (3.2)$$

$$X^{\mu_j}(t_f) = b^{\mu_j}, \quad j=1, 2, \dots, n-k \quad (3.3)$$

$$0 \leq k \leq n$$

のもとで解く。ここで, X は n 次元の状態変数ベクトル, t は時間を表わす。また, 境界条件は初期値において k 個の要素を規定し, 終端値において残りの $n-k$ 個の要素を規定するものとする。 ν_i, μ_j は整数で, 状態変数ベクトルの, それぞれ ν_i 番目, μ_j 番目の要素を示す添字であり, 次の関係がある。

$$1 \leq \nu_1 < \nu_2 < \dots < \nu_k \leq n$$

$$1 \leq \mu_1 < \mu_2 < \dots < \mu_{n-k} \leq n$$

ν_i と μ_j で重複 ($\nu_i = \mu_j$) するものがあるとしてもさしつかえない。 a^{ν_i}, b^{μ_j} は, それぞれ初期値および終端値の規定された要素の値を示す。

境界条件として値が規定されていない要素は, 同様にして,

$$\begin{cases} X^{\nu_i}(t_0), & i=k+1, \dots, n \\ X^{\mu_j}(t_f), & j=n-k+1, \dots, n \end{cases}$$

$$1 \leq \nu_{k+1} < \dots < \nu_n \leq n$$

$$1 \leq \mu_{n-k+1} < \dots < \mu_n \leq n$$

と表わすことができる。ここで, $(\nu_1, \nu_2, \dots, \nu_k, \nu_{k+1}, \dots, \nu_n)$ および $(\mu_1, \mu_2, \dots, \mu_{n-k}, \mu_{n-k+1}, \dots, \mu_n)$ は, 大きさの順に小さい方から並べ換えると $(1, 2, \dots, n)$ となるものである。

3.2 準線型化方程式

GNR 法は, 一般に非線型な常微分方程式 (3.1) を境界条件 (3.2), (3.3) のもとで解く代りに, (3.1) 式を線型化した微分方程式を逐次近似的に解く方法である。線型化微分方程式は次のように与えられる。

$$\dot{X}_R = f(X_{R-1}, t) + J(X_{R-1})(X_R - X_{R-1}) \quad (3.4)$$

ここで,

$$J(X_{R-1}) \triangleq \left[\frac{\partial f}{\partial X} \right]_{X=X_{R-1}} : \text{Jacobian 行列}$$

R : 逐次近似の回数 (Iteration number)

であり, $\dot{\quad}$ は時間微分を表わす。

* この微分方程式は, 例えば, Pontryagin の最大原理を用いて最適制御問題を定式化した場合の Hamilton の正準方程式

$$\begin{cases} \frac{dx}{dt} = \frac{\partial \mathcal{H}(x, \lambda, u)}{\partial \lambda} \\ \frac{d\lambda}{dt} = -\frac{\partial \mathcal{H}(x, \lambda, u)}{\partial x} \end{cases}$$

を一つのベクトル方程式にまとめたのである。ここで, x は変数, λ は補助変数, \mathcal{H} はハミルトニアンである。(文献 1, 2, 5 参照)

(3.4) 式の解は、一般に、斉次解と特殊解の組み合わせで、次のように表わされる。

$$X_R(t) = H_R(t)c_R + p_R(t) \quad (3.5)$$

ここで、

$X_R(t)$: 一般解 (n 次元ベクトル)

$H_R(t)$: 斉次解 ($n \times n$ 行列)

$p_R(t)$: 特殊解 (n 次元ベクトル)

c_R : 定数ベクトル (n 次元ベクトル)

$H_R(t), p_R(t)$ は、それぞれ次式から計算される。

$$\dot{H}_R(t) = J(X_{R-1})H_R(t), \quad (3.6)$$

$$H_R(t_0) = I \quad (\text{単位行列})$$

$$\dot{p}_R(t) = f(X_{R-1}, t) + J(X_{R-1})(p_R(t) - X_{R-1}(t)), \quad (3.7)$$

$$p_R(t_0) = 0$$

c_R は、解が求まった時には、 $X(t_0)$ に一致する定数ベクトルであり、その成分のうち k 個の要素は境界条件として規定されている。したがって残りの $n-k$ 個の要素は第 0 次近似として適当に仮定しなければならない。(3.6), (3.7) 式を解く際に、前回の近似値が必要であるが、第 0 次近似 $X(t_0)$ に関しては、適当な近似値を与えるものとする。例えば、

$$X_0(t) = g(X_0(t_0), t) \quad (3.8)$$

のような近似式で発生させることが考えられる。

なお、(3.4) 式は (3.1) 式を第 $R-1$ 次近似値に関して線型化したもので、前回の近似値を値を基準として用いて線型化するところに準線型化(Quasilinearization) という名称の起源がある。

3.3 GNR 法の逐次近似修正

GNR 法では、非線型微分方程式の境値問題を一連の線型分方程式の初期値問題によって近似し、終端条件を満たすように繰り返し計算によって修正して行く方法をとる。計算の手順を以下に述べる。

- (1) まず、第 0 次近似の初期値 $X_0(t_0)$ を、初期境界条件として規定されている k 個の要素は与えられた値をそのまま用い、残り $n-k$ 個の未規定要素には適当な推定値を与えて設定する。
- (2) 設定した初期値 $X_0(t_0)$ を用いて、(3.8) 式により第 0 次近似値を求める。
- (3) 斉次解と特殊解を、(3.6), (3.7) 式を $t=t_0$ から $t=t_f$ まで積分して求める。
- (4) 斉次解と特殊解を組み合わせで作った一般解は、一般には終端条件を満足しない。この不一致は、定数ベクトル c_R (初期値 $X(t_0)$ と一致する) のうちの未規定要素の推定値に起因する。そこで、逆に終端条件を合わせるように、定数ベクトルの

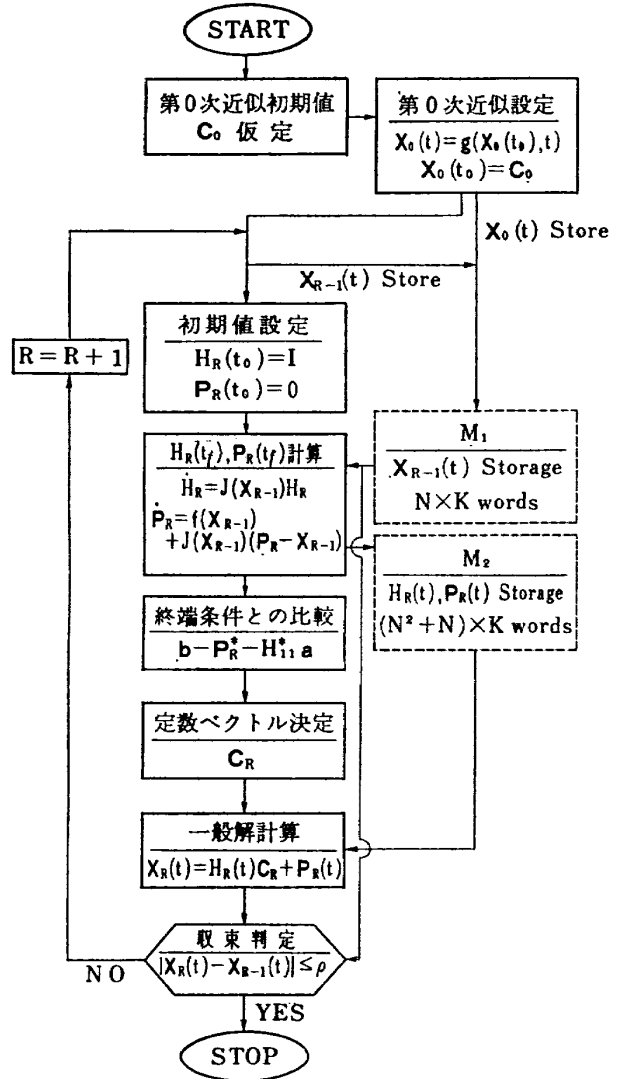


図 1 GNR 法の計算手順

未規定要素を、連立一次方程式の解として求める。

- (5) 一般解を (3.5) 式によって求める。
- (6) 得られた値と前回の近似値とを比較して、その差がある許容範囲内に収まるまで、(3) 以下の計算手順を繰り返す。

定数ベクトル c_R の修正法を、数式を用いて具体的に説明すると、次のようになる。

第 R 次逐次近似の終端値は次のような関係式を満足しなければならない。

$$X_R(t_f) = H_R(t_f)c_R + p_R(t_f) \quad (3.9)$$

および

$$c_R^i \equiv X^{vi}(t_0) = a^i, \quad i=1, 2, \dots, k \quad (3.10)$$

$$X_R^{uj}(t_f) \equiv X^{uj}(t_f) = b^j, \quad j=1, 2, \dots, n-k \quad (3.11)$$

ここで、

$X_R(t_f)$: $n-k$ 要素は境界条件により規定され

る。すなわち (3.11) 式。

c_R : k 要素は境界条件により, (3.10) 式のように規定され, 残りの $n-k$ 要素は未知。

$H_R(t_f)$: (3.6) 式の解として与えられる。

$p_R(t_f)$: (3.7) 式の解として与えられる。

(3.9) 式を各要素を用いて表わすと,

$$\begin{pmatrix} X_R^1(t_f) \\ X_R^{\mu j}(t_f) \\ X_R^n(t_f) \end{pmatrix} = \begin{pmatrix} h_{11} & \cdots & h_{1\nu_1} & \cdots & h_{1n} \\ \vdots & & \vdots & & \vdots \\ h_{\mu j 1} & \cdots & h_{\mu j \nu_i} & \cdots & h_{\mu j n} \\ \vdots & & \vdots & & \vdots \\ h_{n1} & \cdots & h_{n\nu_1} & \cdots & h_{nn} \end{pmatrix} \begin{pmatrix} c_R^1 \\ \vdots \\ c_R^{\nu_i} \\ \vdots \\ c_R^n \end{pmatrix} + \begin{pmatrix} p_R^1(t_f) \\ \vdots \\ p_R^{\mu j}(t_f) \\ \vdots \\ p_R^n(t_f) \end{pmatrix} \quad (3.12)$$

必要な部分け取り出すと,

$$\begin{pmatrix} X_R^{\mu 1}(t_f) \\ \vdots \\ X_R^{\mu n-k}(t_f) \end{pmatrix} = \begin{pmatrix} h_{\mu 1 \nu_1} & \cdots & h_{\mu 1 \nu_i} & \cdots & h_{\mu 1 n} \\ \vdots & & \vdots & & \vdots \\ h_{\mu n-k \nu_1} & \cdots & h_{\mu n-k \nu_i} & \cdots & h_{\mu n-k n} \end{pmatrix} \begin{pmatrix} c_R^1 \\ \vdots \\ c_R^{\nu_i} \\ \vdots \\ c_R^n \end{pmatrix} + \begin{pmatrix} p_R^{\mu 1}(t_f) \\ \vdots \\ p_R^{\mu n-k}(t_f) \end{pmatrix} \quad (3.13)$$

(2.13) 式の右辺を書き直すと,

$$\begin{pmatrix} X_R^{\mu 1}(t_f) \\ \vdots \\ X_R^{\mu n-k}(t_f) \end{pmatrix} = \begin{pmatrix} h_{\mu 1 \nu_1} & \cdots & h_{\mu 1 \nu_k} \\ \vdots & & \vdots \\ h_{\mu n-k \nu_1} & \cdots & h_{\mu n-k \nu_k} \end{pmatrix} \begin{pmatrix} c_R^{\nu_1} \\ \vdots \\ c_R^{\nu_k} \end{pmatrix} + \begin{pmatrix} h_{\mu 1 \nu_{k+1}} & \cdots & h_{\mu 1 \nu_n} \\ \vdots & & \vdots \\ h_{\mu n-k \nu_{k+1}} & \cdots & h_{\mu n-k \nu_n} \end{pmatrix} \begin{pmatrix} c_R^{\nu_{k+1}} \\ \vdots \\ c_R^{\nu_n} \end{pmatrix} + \begin{pmatrix} p_R^{\mu 1}(t_f) \\ \vdots \\ p_R^{\mu n-k}(t_f) \end{pmatrix} \quad (3.14)$$

ここで, 次のような量を定義する。

$$\left\{ \begin{array}{l} H_{11}^* \triangleq \begin{pmatrix} h_{\mu 1 \nu_1} & \cdots & h_{\mu 1 \nu_k} \\ \vdots & & \vdots \\ h_{\mu n-k \nu_1} & \cdots & h_{\mu n-k \nu_k} \end{pmatrix} : (n-k) \times k \text{ 行列} \\ H_{12}^* \triangleq \begin{pmatrix} h_{\mu 1 \nu_{k+1}} & \cdots & h_{\mu 1 \nu_n} \\ \vdots & & \vdots \\ h_{\mu n-k \nu_{k+1}} & \cdots & h_{\mu n-k \nu_n} \end{pmatrix} : \begin{matrix} (n-k) \\ \times (n-k) \\ \text{行列} \end{matrix} \\ c_R^* \triangleq \begin{pmatrix} c_R^{\nu_{k+1}} \\ \vdots \\ c_R^{\nu_n} \end{pmatrix}, \quad p_R^* \triangleq \begin{pmatrix} p_R^{\mu 1}(t_f) \\ \vdots \\ p_R^{\mu n-k}(t_f) \end{pmatrix} \\ a \triangleq \begin{pmatrix} a^{\nu_1} \\ \vdots \\ a^{\nu_k} \end{pmatrix}, \quad b \triangleq \begin{pmatrix} b^{\mu 1} \\ \vdots \\ b^{\mu n-k} \end{pmatrix} \end{array} \right.$$

上の定義式と (3.10), (3.11), 式を用いると, (3.14) 式は次のように書ける。

$$b = H_{11}^* a + H_{12}^* c_R^* + p_R^* \quad (3.16)$$

従って, 定数ベクトル c_R の未知要素は,

$$c_R^* = H_{12}^{*-1} [(b - p_R^*) - H_{11}^* a] \quad (3.17)$$

として求められる。ここで, H_{12}^* は齊次解 $H_R(t_f)$ の小行列で, 逆行列を持ち, ill-condition でないことが必要である。

未規定要素の部分ベクトル c_R^* が求められれば, 定数ベクトル c_R が決定され, (3.5) 式から一般解 $X_R(t)$ が計算できる。 $X_R(t)$ が計算されたならば, 前回の逐次近似値と比較値 $X_{R-1}(t)$ して, 停止条件

$$|X_R(t) - X_{R-1}(t)| \leq \rho \quad (3.18)$$

を満たすまで逐次近似計算を繰り返す。ここで, ρ は要求収束精度である。

以上に説明した GNR 法の計算手順のフロー・チャートを図 1 に示す。

3.4 GNR 法の特長および問題点

GNR 法の特長は, 比較的簡単なアルゴリズムときわめて良好な収束性にある。特にその収束性は最適解に近づくにつれて加速度的によくなって行く。これは最急降下法が最適解に近づくにしたがって傾斜 (Gradient) がゼロに近づき収束性が劣化し最終的な解に到達するのに時間がかかるのと対照的である。GNR 法の収束の仕方は, 一般に二次収束性 (Quadratic Convergence) と呼ばれるものである^{8,11)}。すなわち, X を真の解とすれば,

$$|X_R(t) - X(t)| < \alpha |X_{R-1}(t) - X(t)|^2, \quad \alpha > 1 \quad (3.19)$$

と表わされるような収束性を示す。GNR 法の収束性の証明に関しては, Kalaba¹¹⁾ や McGill と Kenneth¹³⁾ の文献を参照されたい。

次に GNR 法の持つ問題点について触れておく。

(1) 記憶容量の問題

DP のように記憶容量の問題が決定な障害とはならないが, GNR 法では微分方程式を解く際に前回の逐次近似値 $X_{R-1}(t)$ を積分刻みごとに記憶させておかなければならないことと, 定数ベクトル c_R が最終点まで積分が終了する時点で決定されるので, 齊次解 $H_R(t)$ と特殊解 $p_R(t)$ を積分刻みごとに保持しておかなければ収束判定ができないことから, 状態変数ベクトルの次元が大きい場合には記憶容量が問題点となる。この点に関して本研究の目的でもあるので次節で検討する。

(2) 第0次近似の問題

GNR 法では微分方程式を解く際に前回の逐次近似値 $X_{R-1}(t)$ を必要とするが、近似許算を開始する時点で第0次近似値をどのように定めるかが問題となる。おそらく (3.8) 式のような近似関数を用いて発生させるものと考えられるが、考えられた問題ごとに近似関数を工夫しなければならない上に、近似関数の選び方および第0次近似初期値の選び方が問題となる。

(3) 最適解への収束の保証の問題

上述の問題点と密接な関係がある問題である。GNR 法はきわめて収束が速い特長を持つ反面、第0次近似の選び方がまずいと、与えられた問題によっては、真の最適解に収束せず局所的最適解に収束してしまう場合がある。これに関しては与えられた問題の性質あるいは物理的解釈などによって難点を除去することは可能であるが、それでも第0次近似値の選択には細心の注意が必要であることには変りない。

3.5 記憶容量 (Memory Storage) の問題

GNR 法の計算を実行する際の計算機記憶容量の問題を検討するために、次のような設定を行なう。

(1) 状態変数ベクトル X は N 次元ベクトルとする。

(2) 積分区間 $t_0 \leq t \leq t_f$ を K に分割する。

解くべき連立微分方程式の数は、(3.6), (3.7) 式を併せて $N^2 + N$ 個となる。(3.6), (3.7) 式の微分方程式を解く際に必要な前回の逐次近似値 $X_{R-1}(t)$ を各積分刻みごとに記憶させておかなければならない。このために必要な記憶容量を M_1 とすると、

$$M_1 = N \times K \text{ (words)} \quad (3.20)$$

となる。

また、逐次近似計算の収束を判定するためにも、記憶容量を確保しておくことが必要となる。一般に、逐次近似の停止条件は (3.18) 式のように与えられる。これは積分区間の全域にわたって要求される条件であり、定数ベクトル c_R は1回の逐次近似計算が終るまでは決まらないから、積分が終るまで斉次解 H_R および特殊解 p_R を各積分刻みごとに記憶させておかなければならない。このために要記憶容量を M_2 とすると、

$$M_2 = (N^2 + N) \times K \text{ (words)} \quad (3.21)$$

必要である。

したがって、 M_1, M_2 を併せたものを M とすると、

$$M = M_1 + M_2 = (N^2 + 2N) \times K \text{ (words)} \quad (3.22)$$

の記憶容量が必要となる。状態変数の次元が小さい場合にはあまり問題にならないが、次元の大きい複雑な問題を解く場合には致命的な障害となりうる。例えば、 $N=12, K=1,000$ とした場合には、

$$M_1 = 12,000 \text{ (words)}$$

$$M_2 = 156,000 \text{ (words)}$$

となり、両方で $168k$ words 必要となる。この難点を解決することが本報告の主題であることを、ここで再び強調しておく。なお、 M_1, M_2 の他に、計算プログラムを実行するのに必要な指令の格納場所や作業番地などの容量が必要であるが、ここでは触れないことにする。

4. Bellman の改良アルゴリズム

Bellman が 1961 年に、計算機記憶容量を $N \times K$ words 占有して (3.6), (3.7) 式を解くに必要な前回の近似値 $X_{R-1}(t)$ を保持しておく代り、記憶方程式 (Memory equation) を導入して $X_{R-1}(t)$ を発生させることにより使用記憶容量を減少させることを提案した²⁵⁾。

記憶方程式としては、(3.4) 式の準線化微分方程式を利用する。したがって、Bellman の改良アルゴリズムでは、次の連立微分方程式を解くことになる。

$$\begin{cases} \dot{H}_R(t) = J(X_{R-1})H_R(t), & H_R(t_0) = I \quad (4.1) \\ \dot{p}_R(t) = f(X_{R-1}(t), t) + J(X_{R-1})(p_R(t) \\ - X_{R-1}(t)), & p_R(t_0) = 0 \quad (4.2) \end{cases}$$

$$\begin{cases} \dot{X}_l = f(X_{l-1}, t) + J(X_{l-1})(X_l - X_{l-1}), & X_l(t_0) = c_l \\ & (l=1, 2, \dots, R-1) \quad (4.3) \end{cases}$$

$$\begin{cases} X_0(t) = g(X_0(t_0), t), & X_0(t_0) = c_0 \quad (4.4) \end{cases}$$

ここで、(4.1), (4.2) 式は (3.6), (3.7) 式をそれぞれ再録したものであり、(4.3) 式が記憶方程式である。(4.4) 式は (4.3) 式を解くのに必要な第0次近似を与える近似式で、(3.8)を再録したものである。

解くべき連立微分方程式の数は全部で $N^2 + R \times N$ 個となる。 $R-1$ 個の記憶方程式の解かせるためにはその初期値として、 $c_l (l=0, 1, 2, \dots, R-1)$ を記憶させておかなければならないから、(4.1), (4.2) 式を解くのに必要な記憶容量と考えれば、

$$M_1 = N \times R \text{ (words)} \quad R < K \quad (4.5)$$

となる。逐次近似計算の回数 R は積分刻み K に比べると、はるかに小さいと考えられるから、記憶容量としては大分節約されると述べている。例えば、 $N=12, K=1,000$ とし、仮に最大の逐次近似回数を $R=30$ とすると、

原アルゴリズム : $M_1=12,000$ words

改良アルゴリズム : $M_1=360$ words

となり, M_1 に関する限りは大幅に減少する。

この改良アルゴリズムでは, $R-1$ 個の記憶方程式を増やすことによって, 所要記憶容量を $N \times (K-R)$ words 減らしたが, 反面, 所要計算時間を増加させることになった。1 回の逐次近似において, 連立微分方程式を解くのに要する時間を T_{integ} とすると, 次のように表わすことができる。

$$T_{integ} = T_H + T_P + (R-1)T_P = T_H + RT_P \quad (4.6)$$

ここで, T_H は (4.1) 式を, T_P は (4.2) 式を解くのに要する計算時間である。記憶方程式 (4.3) 式は (4.2) 式と同じ構造を持っているから, 計算時間は T_P と考えてさしつかえない。一方同様に, GNR 法の原アルゴリズムの方を考えると

$$T_{integ} = T_H + T_P \quad (4.7)$$

である。したがって, 改良アルゴリズムは, $(R-1)T_P$ の計算時間の増加によって, $N \times (K-R)$ words の所要記憶容量を減少させることを可能にしたといえる。このことは, 序論にも述べたように, 計算機を使用する際には, 計算時間と所要記憶量との間に, ある意味での互換性または等価性が在る事実を利用したと言い換えることができる。この点にアルゴリズムの工夫の意義がある訳である。

Bellman の改良アルゴリズムのフロー・チャートを図 2 に示す。図 1 の GNR 法の原アルゴリズムと対比させてみると, 右側のループが異なっていることに気がつく。

Bellman の改良アルゴリズムは, 連立微分方程式を解くために必要な記憶容量 M_1 の減少を目的としたものである。通常, 積分刻み数 K は 100 とか 500 あるいは 1,000 程度であり, 逐次近似回数 R は精々 10 とか 20 程度であって, $R \cong K$ となることは, まずありえないから, M_1 に関する限り所要記憶容量を大幅に減少させることが可能である。ただ, 逐次近似計算は停止条件を満たすまで繰り返されるものであるから, R の値をあらかじめ予測しておくことはできない。したがって, R の予想最大値に十分余裕を持たせて M_1 を確保しておかなければならない。

また, この改良アルゴリズムでは, 逐次近似計算の回数が増えるにしたがい解くべき連立方程式の数が増えるので, 1 回の逐次近似計算に要する積分時間が回を追うにしたがい等差級数的に増大して行く難点がある。すなわち, R 回の逐次近似で解が収束するとしたと

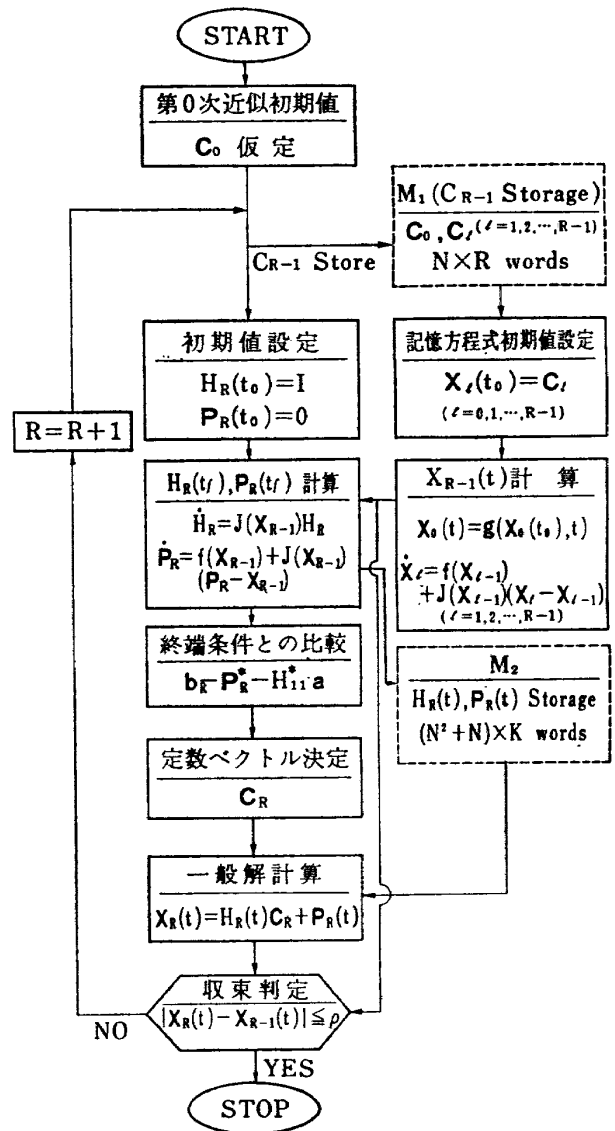


図 2 Bellman 改良アルゴリズムの計算手順

きに要する積分時間の合計は, 原アルゴリズムでは $R(T_H + T_P)$ であるのに対し, 改良アルゴリズムでは $RT_H + R(R+1)T_P/2$ となり, 逐次近似計算の回数が大きくなるにしたがい, その差は大きくなって行く。

以上は, いずれも逐次近似計算回数があらかじめはっきりと予測できない点, つまり R の不確定性に起因する問題点に言及したものであるが, M_1 の減少を目的とする点では, まず順当な改良法であろう。ただ, Bellman の改良アルゴリズムでは収束判定のために必要な記憶容量 M_2 に関しては何も述べていない。GNR 法の概説の項で述べたように, 定数ベクトル c_R は 1 回の逐次近似計算が終了するまでは決定されないで, (3.18) 式のような積分区間全域にわたって要求される収束判定のための停止条件を与えられた場合には, 斉次解 H_R , および特殊解 P_R を各種分刻み

ごとに保存しておかなければならない。したがって、この改良アルゴリズムの場合にも、 M_2 としては、

$$M_2 = (N^2 + N) \times K \text{ wrds} \quad (4.8)$$

の記憶容量が必要となる。 M_1 と比べると、 M_2 の方がはるかに多くの記憶容量を必要とする。ただ、与えられた問題によっては、積分区間全域にわたって収束を判定せずに、終端点においてのみ収束を判定すればよい場合もある。また、(3.1) 式で与えられる原方程式が、非線型ではなく、変数が多い複雑な方程式としても線型系であるならば、終端点のみで収束を判定すれば事は足りる。そこで、Bellman は、いかなる場合にも共通する問題点として、 M_1 の減少を目的としたアルゴリズムを提案したのではないかと思われる。ただ、宇宙飛行体の最適軌道計算のような問題では、軌道全域にわたって収束を判定することが要求されるので、 M_2 は必要欠くべからざるものとなることを付け加えておく。

5. 改良アルゴリズム (1) GNRM

改良アルゴリズム GNRM は、著者がすでに文献 27 において、計算アルゴリズムとして使用していたものであり、本項ではそれをさらに一般化した形で述べる。GNRM の名称は、GNR 法の Modification を意味する略語である。GNRM は、Bellman の改良アルゴリズムに二つの点で改良を加えたものである。すなわち、(1) 逐次近似回数 R による不確定性の除去と、(2) 収束判定のために必要な記憶容量 M_2 の減少との二つを改良点としている。

5.1 逐次近似回数による不確定性の除去

前回の逐次近似値 $X_{R-1}(t)$ を発生させる記憶方程式として、Bellman の記憶方程式 (4.3) の代わりに、問題として与えられる原方程式 (3.1) を用いる。したがって、計算すべき連立微分方程式は次のようになる。

$$\dot{H}_R(t) = J(X_{R-1})H_R(t), \quad H_R(t_0) = I \quad (5.1)$$

$$\dot{p}_R(t) = f(X_{R-1}, t) + J(X_{R-1})(p_R(t) - X_{R-1}(t)), \quad p_R(t_0) = 0 \quad (5.2)$$

$$\dot{X}_{R-1} = f(X_{R-1}, t), \quad X_{R-1}(t_0) = c_{R-1} \quad (5.3)$$

ここで、(5.3) 式が原方程式 (3.1) を利用した記憶方程式である。解くべき連立微分方程式の数は $N^2 + 2N$ 個となり、Bellman の改良アルゴリズムの $N^2 + RN$ 個のように逐次近似回数 R に依存することはなくなる。

GNRM では、 $X_{R-1}(t)$ を記憶させて置く代りに記憶方程式を連立させて、所要記憶容量を計算時間で代

替する点では Bellman と同じ発想であるが、次に述べるような観点から原方程式を用いることにした。

(1) 解が収束するまでに要する逐次近似回数が増大するにつれ、解くべき微分方程式の数の数が増し、それとともに記憶容量 M_1 と微分方程式の積分時間が等差級数的に増大して行くことは好ましくない。

(2) 第 0 近似を与える近似関数を、記憶方程式とは別個に扱い、与えられた問題ごとに工夫しなければならないのは、アルゴリズムの一般性という点で望ましくない。

(3) Bellman の記憶方程式の代わりに原方程式を用いれば、 $X_{R-1}(t)$ の記憶容量の問題、積分時間の不確定性の問題、および第 0 近似発生の問題が同時に解決できる。

(4) 原方程式

$$\dot{X}_{R-1} = f(X_{R-1}, t), \quad X_{R-1}(t_0) = c_{R-1}$$

の解 X_{R-1} と、Bellman の記憶方程式

$$\dot{X}_l = f(X_{l-1}, t) + J(X_{l-1})(X_l - X_{l-1})$$

$$X_l(t_0) = c_l, \quad l=1, 2, \dots, R-1$$

の解 $(X_l)_{l=R-1}$ とは一般に一致しないが、後者の方程式は本来前者の近似式であるから、解が収束するならば逐次近似計算が進むにつれて、両者の解は一致してくるはずである。

記憶方程式に原方程式 (3.1) を採用することによって、逐次近似回数による記憶容量 M_1 の不確定性は除去できる。すなわち、 $N^2 + 2N$ 個の連立微分方程式を解くために要する記憶容量 M_1 としては、前回の定数ベクトル c_{R-1} だけを保存しておけばよい。

$$M_1 = N \text{ (words)} \quad (5.4)$$

となり、Bellman の $N \times R$ words に比べると、はるかに小さくなる。

また、1 回の逐次近似計算における連立微分方程式の積分時間も R の値に依存しなくなり、

$$T_{\text{integ}} = T_H + T_P + T_N \quad (5.5)$$

と表わされる。ここで、 T_N は非線型な原方程式を解くのに要する積分時間で、 T_H , T_P は Bellman の場合と同じである。 T_N と T_P の大小関係は一般には論じられないが、大体は、 $T_N \cong T_P$ と考えてよいから、積分時間の方も減少させることができたと言える。

第 0 次近似値も、記憶方程式として導入した原方程式によって発生させることができ、Bellman の改良アルゴリズムのように、第 0 次近似を特別扱いにして、与えられた問題ごとに近似関数式を工夫する必要はなく

なった。

5.2 収束判定に要する記憶容量 M_2 の減少

逐次近似計算の収束判定のための停止条件が、

$$|X_R(t) - X_{R-1}(t)| \leq \rho \quad (5.6)$$

のように積分区間全域にわたって課せられている場合を考えると、GNR 法の原アルゴリズムにしる、Bellman の改良アルゴリズムにしる、繰り返し計算の誤差判定のためには、1 回の逐次近似計算が終了して定数ベクトル c_R が決定するまで、斉次解 H_R と特殊解 p_R を各積分刻みごとに保存しておかなければならない。一たん、 c_R が計算されれば、一般解 X_R は、

$$X_R(t) = H_R(t)c_R + p_R(t) \quad (5.7)$$

の関係式によって計算され、誤差比較が可能となる。したがって、収束判定に要する記憶容量 M_2 としては、

$$M_2 = (N^2 + N) \times K \text{ words}$$

必要である。

極めて簡単な問題のときはよいが、例えば、変数の数 $N=6$ 、積分刻み数 $K=1,000$ とした場合ですら、すでに $M_2=42,000$ words となる。まして、宇宙飛行体の三次元質点運動を考え、補助変数 (adjoint variables) を導入して、最適軌道問題を解くことになれば、 $N=12$ となり、 M_2 は 156k words 必要となって、計算機容量をはるかに超過してしまい、問題を解くこと自体全く不可能となる。剛体運動も含めた三次元六自由度の最適軌道問題ともなれば、変数の次元 N はさらに増えて行くので、 M_1 よりも M_2 の問題の方が決定的な障害となることは明白である。

解決策としては、 $X_{R-1}(t)$ の保存問題の処理のところで使われているように、計算機容量と計算時間との間のある意味での等価性あるいは互換性の利用をさらに押し進めればよい。すなわち、連立微分方程式の積分を、定数ベクトル c_R を決定するための積分計算 (Overall integration) と、斉次解 H_R と特殊解 p_R を記憶させる代りの積分計算 (Stepwise integration) との二段階を繰り返すことにすればよい。Stepwise の意味は、各積分刻みごとに一般解 $X_R(t)$ を計算することを含んでいる。以上のようにすれば、1 回の逐次近似計算に要する全計算時間 (積分時間はこの中に含まれる) は、当然 2 倍近くなるが、計算機容量超過によって解決不能となることは避けられる。収束判定に要する記憶容量 M_2 は、

$$M_2 = 0 \quad (5.8)$$

と全く不要になるが、その代り

$$T_{M_2} = T_H + T_P + T_N \quad (5.9)$$

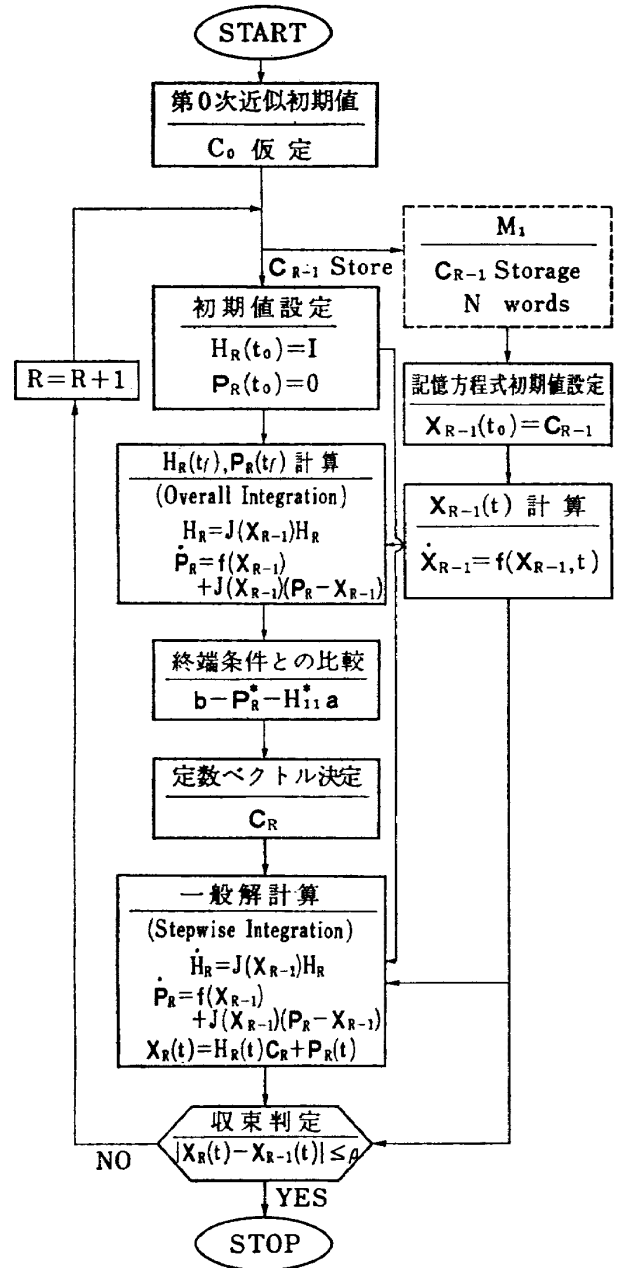


図 3 GNRM アルゴリズムの計算手順

だけ積分時間が余計に必要となる。

5.3 GNRM に関する補注

以上に述べた改良アルゴリズム GNRM の計算手順を表わすフロー・チャートを図 3 に示す。定数ベクトル c_R の決定法は、3.3 に述べた GNR 法の場合と全く同じである。

なお、GNRM とは別個に、Baird³²⁾ が原方程式を記憶方程式として用い、記憶容量を減少させたことを報告している。Baird は収束判定のための記憶容量 M_2 に関しては、なんら言及していないが、スカラー微分方程式 ($N=1$ の場合) について、解の収束性の証明を行なっている。

6. 改良アルゴリズム (2) QLMD

改良アルゴリズム QLMD は本報告の中心をなすものである。QLMD の略号は Quasilinearization Method の Modification を意味している。この段階での改良は、問題本来の立場に戻り、前節で述べた GNRM のアルゴリズムを再検討することによって進められる。

6.1 GNRM アルゴリズムの再検討

齊次解 $H_R(t)$, 特殊解 $p_R(t)$ および定数ベクトル c_R の組み合わせによって与えられる一般解 $X_R(t)$ は、与えられた問題の原方式を第 $R-1$ 次近似値を基準 (Reference) として線型化した方程式

$$\dot{X}_R = f(X_{R-1}, t) + J(X_{R-1})(X_R - X_{R-1}) \quad (6.1)$$

の解にほかならない。(6.1) 式は (3.4) 式の再録)

この式は、

$$\dot{X}_{R-1} = f(X_{R-1}, t)$$

の関係があるから、

$$\Delta X(t) \triangleq X_R(t) - X_{R-1}(t) \quad (6.2)$$

の定義式を導入すれば、次のように表わされる。

$$\Delta \dot{X}(t) = J(X_{R-1}) \Delta X(t) \quad (6.3)$$

この (6.3) 式は基準 (Reference) からのずれ (Deviation) を支配する微分方程式と考えることができる。

この式を解くのに現代制御理論では遷移行列 (Transition matrix) $\Phi(t, t_0)$ を導入して、

$$\Delta X(t) = \Phi(t, t_0) \Delta X(t_0) \quad (6.4)$$

と書き表わすことがよく行なわれる。ここで遷移行列 $\Phi(t, t_0)$ は $n \times n$ 行列で、

$$\dot{\Phi}(t, t_0) = J(X_{R-1}) \Phi(t, t_0), \quad \Phi(t_0, t_0) = I \quad (6.5)$$

の解として与えられる。通常、遷移行列は与えられた問題に固有の基準 (Reference) に基づいて計算されるが、(6.5) 式は逐次近似計算の第 $R-1$ 次近似 $X_{R-1}(t)$ を基準にとったものとして表わされている。

遷移行列を与える (6.5) 式は、齊次解 $H_R(t)$ を与える (5.1) 式と全く同型である。したがって、第 $R-1$ 次近似を基準にとった場合の遷移行列には、

$$\Phi(t, t_0) \equiv H_R(t) \quad (6.6)$$

の恒等関係がある。ここで、線型化の基準としての reference を固有のものでなく、逐次修正して行くものと考えようという Quasi-reference (準基準) の概念 (文献 27, 28) を導入すれば、逐次近似計算法自体を Quasi-reference による Successive Linearization とみなすことができる。そして、(5.1) 式の解である齊次解 $H_R(t)$ は Quasi-reference に関する遷移行列と考えることができる。

改良アルゴリズム GNRM では、 $X_{R-1}(t)$ を積分

刻みごとに記憶させる代りに、原方程式を用いた記憶方程式 (5.3) 式によって発生させることにしたが、ここで、もとの問題に戻って、特殊解の方程式 (5.2) と記憶方程式 (5.3) を検討してみる。(5.2) 式は原方程式を $X_{R-1}(t)$ に関して線型した近似微分方程式であり、初期値として、 $p_R(t) = 0$ を与えて解くものである。一方、(5.3) 式は記憶方程式として導入したとはいえ、初期値として、 $X_{R-1}(t_0) = c_{R-1}$ を与えて、原方程式そのものを解くものである。したがって、(5.2) 式、(5.3) 式は、近似式と原式の差および初期条件の違いはあるが、本質的には同質の方程式を解いていると考えてもよい。これは無駄な努力ではないだろうか？ それにしても初期条件の違いをどのように解釈すればよいか？

ここで齊次解、特殊解の組み合わせで与えられる一般解を再検討してみよう。一般解は

$$X_R(t) = H_R(t) c_R + p_R(t) \quad (6.7)$$

で与えられ、定数ベクトル c_R は境界条件を満たすように決定すればよい。いま、 $p_R(t_0) = 0$ と限定せずに考えてみる。(6.7) 式で、 $t = t_0$ とおくと、

$$X_R(t_0) = c_R + p_R(t_0) \quad (\because H_R(t_0) = I)$$

$$\therefore c_R = X_R(t_0) - p_R(t_0) \quad (6.8)$$

となり、 $p_R(t_0)$ は与えられた初期条件で各要素は既知であるが、 $X_R(t_0)$ の方は境界条件として規定されている k 個の要素のみ既知であるから、定数ベクトル c_R の要素のうち k 個しか決定されない。残りの $n-k$ 個の要素は最終条件から決定される。したがって、 c_R の未知の $n-k$ 個の要素を決定することは、端末条件から $X_R(t_0)$ の未知要素 $n-k$ 個を決定することに帰着できる。(6.8) 式を (6.7) 式に代入して、 $t = t_f$ とおくと、

$$X_R(t_f) = H_R(t_f) [X_R(t_0) - p(t_0)] + p_R(t_f) \quad (6.9)$$

となる。この式の左辺 $X_R(t_f)$ は $n-k$ 個の要素を終端境界条件として規定されているので、3.3 節で述べたように必要な部分のみを取り出すと、実質上は $X_R(t_0)$ の要素のうちの $n-k$ 個の未知数に関する連立一次方程式となる。これを解くことにより、 $X_R(t_0)$ の全要素が決定され、(6.8) 式によって定数ベクトル c_R が確定し、(6.7) 式によって一般解が決定される訳である。以上の議論では、必ずしも $p_R(t_0) = 0$ である必要はなく、ある確定した初期値が与えられていればよい訳である。では何故 (5.2) 式で、 $p_R(t_0) = 0$ としているのだろうか？ (6.8) 式で明らかのように $p_R(t_0) = 0$ と選べば、定数ベクトル c_R の要素のうち

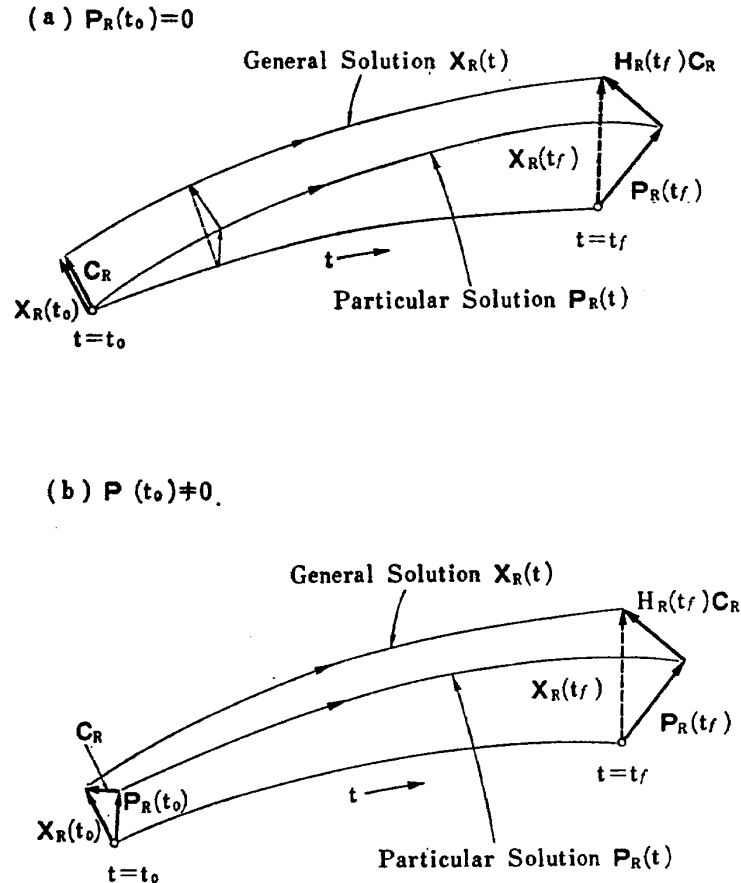


図 4 齊次解と特殊解の組合せの幾何学的解釈

の k 個は、与えられている初期境界条件そのものとして規定されるので簡単であるというに過ぎない。ただし、この簡単さも境界条件が状態変数そのものとして与えられる場合にのみいえることであり、条件が状態変数の関数を規定する型で与えられた場合には、必ずしも簡単ではない。一般に一階線型微分方程式は、齊次解と任意の特殊解の一次結合によって表現されるから、 $P_R(t_0)=0$ と選んでもよいというだけで、他の観点から見て都合が好ければ、 $P_R(t_0) \neq 0$ となる初期値を与えてもさしつかえない訳である。図4に、齊次解と特殊解の組み合わせによる一般解の幾何学的規釈を、 $P_R(t_0)=0$ と $P_R(t_0) \neq 0$ の場合について図解しておく。

なお、最近入手した Pitkin と Carpenter の論文³³⁾によれば、特殊解の初期値を $P_R(t_0)=X_{R-1}(t_0) \neq 0$ とし、齊次解 $H_R(t)$ を遷移行列として取り扱うアルゴリズムを提案している。このようにすると定数ベクトル C_R の要素のうち初期境界条件が規定されている要素に対応するものはゼロとなり、齊次解行列の要素のうち C_R のゼロ要素に対応するものは記憶させておく必要がなくなるために、所要記憶容量を減少させるこ

とができると述べている。すなわち、 $H_R(t)$ の n^2 個の要素のうち kn 個は記憶させる必要がなくなる訳である。

6.2 QLMD アルゴリズムの構成

前節において GNRM を再検討した結果、得られた結論は次の通りである。

- i) GNR 法の計算は、原理的には、第 $R-1$ 次近似を Reference と考えた摂動法 (Perturbation Method) である。
- ii) 齊次解 $H_R(t)$ は、第 $R-1$ 次近似を Reference とした場合の遷移行列 $\Phi(t, t_0)$ に相当する。
- iii) 特殊解 $P_R(t)$ の微分方程式と $X_{R-1}(t)$ を与える記憶方程式は、同質な方程式である。
- iv) 特殊解の初期値は、 $P_R(t_0)=0$ としなければならない必然性はない。

以上の結論に加えて、Quasi-reference の概念^{27,28)}と遷移行列の概念を導入して、非線型微分方程式の二点境界値問題の数値解法を、次のような方針の下に再構成してみよう。

- (1) Quasi-reference (準基準) の概念を導入することによって、結論 iii), iv) から、特殊解方程式と

記憶方程式を、準基準解 (Quasi-reference solution) $X_q(t)$ に関する方程式に統合する。

(2) 斉次解 $H_R(t)$ は、Quasi-reference に関する遷移行列に置換える。

以下、順を追って QLMD のアルゴリズムを構成して行くが、解くべき問題は 3.1 節に述べた二点境界値問題であり、問題に関する表記法は同じものを用いる。

準基準解 (Quasi-reference solution) $X_q(t)$ を考えると、第 R 次近似の一般解 $X_R(t)$ は、次のように定義できる。

$$X_R(t) \triangleq X_{qR}(t) + \Delta X_R(t) \quad (6.10)$$

ここで添字の R は第 R 回目の逐次近似を表わす。

準基準解 $X_{qR}(t)$ は、次式で与えられる。

$$\dot{X}_{qR} = f(X_{qR}, t), \quad t_0 \leq t \leq t_f \quad (6.11)$$

$$X_{qR}(t_0) = X_{qR-1}(t_0) + \Delta X_{R-1}(t_0) \quad (6.12)$$

ここで $X_q(t_0)$ の第 0 次近似は、初期境界条件として k 個の要素は規定されているので、残りの $n-h$ 個の要素を仮定して与える。すなわち、

$$\begin{cases} X_{q0^{v_i}}(t_0) \equiv a^{v_i} & i=1, 2, \dots, k \\ X_{q0^{v_i}}(t_0) = \text{assumed} & i=h+1, \dots, n \end{cases} \quad (6.13)$$

$$(6.14)$$

で、 a^{v_i} は初期境界条件の規定値であり、 v_i は整数で

$$1 \leq v_1 < v_2 < \dots < v_k \leq n$$

$$1 \leq v_{k+1} < \dots < v_n \leq n$$

の関係があり、 $v_i (i=1, 2, \dots, n)$ を大きさの順に並べ換えると $(1, 2, \dots, n)$ となるものである。

ここで、(6.12) 式で与えられる $X_{qR}(t_0)$ について説明を補足する。右辺第 1 項の $X_{qR-1}(t_0)$ は、第 R-1 次近似の初期値として与えられるもので、最初の第 0 次近似は (6.13), (6.14) で与えられる。この値を初期値として、(6.11) 式を $t=t_f$ まで積分すれば、第 R-1 次近似の準基準解が得られるものである。右辺第 2 項の $\Delta X_{R-1}(t_0)$ は、 $X_{qR-1}(t_0)$ を初期値として (6.11) 式を最終点まで積分して、端末境界条件と比較した後初めて計算できる量である。 $\Delta X_{R-1}(t_0)$ の各要素のうち、初期境界条件として規定されている $X_{qR-1}(t_0)$ の k 個の要素に対応している要素は常に 0 である。すなわち、

$$\Delta X_{R-1}^{v_i}(t_0) \equiv 0 \quad i=1, 2, \dots, k \quad (6.15)$$

なるとなれば、初期境界条件が規定されて要素に関しては、準基準解からのずれがあってはならないからである。

$\Delta X_{R-1}(t_0)$ が計算されれば、(6.12) 式によって、次の第 R 回目の逐次近似計算の初期値 $X_{qR}(t_0)$ が設定

される。なお、逐次近似計算を始める前には、 $\Delta X_0(t_0) = 0$ としておく。

次に、現代制御理論で用いられる遷移行列の概念を導入しよう。問題に与えられた原方程式 (3.1) 式を、準基準解のまわりで線型化すると、

$$\dot{X}_R = f(X_{qR}, t) + J(X_{qR})(X_R - X_{qR}) \quad (6.16)$$

$$J(X_{qR}) \triangleq \left[\frac{\partial f}{\partial X^T} \right]_{X=X_{qR}(t)} \quad (6.17)$$

となる。ここで、(6.10), (6.11) 式を用いて書き直すと、

$$\Delta \dot{X}_R(t) = J(X_{qR}) \Delta X_R(t) \quad (6.18)$$

と表わされる。

ここで、遷移行列を導入すると、(6.18) 式の解は次のように書ける。

$$\Delta X_R(t) = \Phi_R(t, t_0) \Delta X_R(t_0) \quad (6.19)$$

ここで、 $\Phi_R(t, t_0)$ は第 R 次近似の Quasi-reference に関して与えられる遷移行列で、次式から計算される。

$$\dot{\Phi}_R(t, t_0) = J(X_{qR}) \Phi_R(t, t_0), \quad (6.20)$$

$$\Phi_R(t_0, t_0) = I: n \times n \text{ 単位行列}$$

したがって、遷移行列を用いれば、(6.10) 式は、

$$X_R(t) = X_{qR}(t) + \Phi_R(t, t_0) \Delta X_R(t_0) \quad (6.21)$$

のように書き換えられる。

ここで、今までの議論に出てきた方程式のうち、実際の計算に必要なものを整理しておく、次のようになる。

1) 準基準解 (Quasi-reference Solution):

$$\dot{X}_{qR} = f(X_{qR}, t) \quad t_0 \leq t \leq t_f \quad (6.22)$$

$$X_{qR}(t_0) \triangleq X_{qR-1}(t_0) + \Delta X_{R-1}(t_0) \quad (6.23)$$

$$\begin{cases} X_{q0^{v_i}}(t_0) \equiv a^{v_i} & (i=1, 2, \dots, k) \\ & (R=0, 1, \dots, R) \end{cases} \quad (6.24)$$

$$\begin{cases} X_{q0^{v_i}}(t_0) = \text{assumed}, & (i=k+1, \dots, n) \end{cases} \quad (6.25)$$

$$\Delta X_{R-1}^{v_i}(t_0) \triangleq 0 \quad (6.26)$$

2) 遷移行列 (Transition Matrix):

$$\dot{\Phi}_R(t, t_0) = J(X_{qR}) \Phi_R(t, t_0), \quad \Phi_R(t_0, t_0) = I \quad (6.27)$$

$$J(X_{qR}) \triangleq \left[\frac{\partial f}{\partial X^T} \right]_{X=X_{qR}(t)} \quad (6.28)$$

3) 一般解 (General Solution):

$$X_R(t) = X_{qR}(t) + \Phi_R(t, t_0) \Delta X_R(t_0) \quad (6.29)$$

QLMD の逐次近似計算は、 $\Delta X_R(t_0)$ の計算が中心となるが、その詳細は次節に述べる。

6.3 QLMD の逐次近似修正法

ここで述べる QLMD に限らず、GNR 法でも同様であるが、代数方程式の根を求める普通の Newton-

Raphson 法の場合と異なり、一般に 2 点境界値問題においては、逐次近似修正法 (Iteration logic) は、一般的な公式としては与えられない。境界条件の与え方によって異ってくるからである。ただ、原理は後述するように全く同じである。

いま、状態変数は n 次元ベクトルと考え、境界条件として、初期点で k 個の要素を、終端点で残りの $n-k$ 個の要素を規定するとしよう。規定される状態ベクトルの要素は、初期点と終端点で同一要素であっても構わない。したがって、初期境界条件の与え方は組み合わせで ${}_nC_k$ 通りあり、終端境界条件の与え方は ${}_nC_{n-k}$ 通りあるので、二点境界条件の与え方は全体で

$${}_nC_k \times {}_nC_{n-k} = {}_nC_n^2$$

通り考えられる。例えば、 $n=6, k=3$ の場合には、 ${}_6C_3^2=400$ 通りとなる。二点境界条件の与え方は、組み合わせとしては多くの場合があるが、我々の当面する問題は ${}_nC_k^2$ 個の組み合わせのうちの一つであるから心配は不要である。ただ、逐次近似修正法を構成するための logic が単純な一般公式として書き表わせないというにすぎない。ただし、原理は全ての場合を通じて同じであるから、当面する問題の境界条件の与えられ方によって、後述する原理にしたがって、explicit な形の logic を作ればよい。

QLMD アルゴリズムの iteration logic 構成の原理を簡条書にすれば、以下の通りである。

- (1) 第 0 次近似として、準基準解の初期値 $X_{q0}(t_0)$ を、初期境界条件が規定されている k 個の要素は与えられた値を用い、残り $n-k$ 個の未規定要素には適当な推定値を用いて設定する。
- (2) 逐次近似の繰り返し計算を開始するとに当たって、準基準からのずれ $\Delta X_R(t_0)=0 (R=0, 1, 2, \dots)$ と仮定する。
- (3) 第 R 次近似として、最終点 $t=t_f$ まで積分計算が終了したところで、準基準解 $X_{qR}(t_f)$ の要素のうち、終端境界条件として値が規定されている $n-k$ 個の要素が、規定値として一致しているかどうかを検討する。一般には一致しない。計算開始点で、 $\Delta X_R(t_0)=0$ と仮定したが、 $X_{qR}(t_0)$ の要素のうち初期境界条件によって規定されていない残りの $n-k$ 個の要素に誤差があったために、終端点でずれが生じたと考えられる。そこで、同時に計算しておいた遷移行列を用いて、終端点におけるずれから初期点における誤差を逆算する。
- (4) 求められた初期誤差を用いて、次の第 $R+1$ 次近似の準備として、(6.23) 式によって準基準解

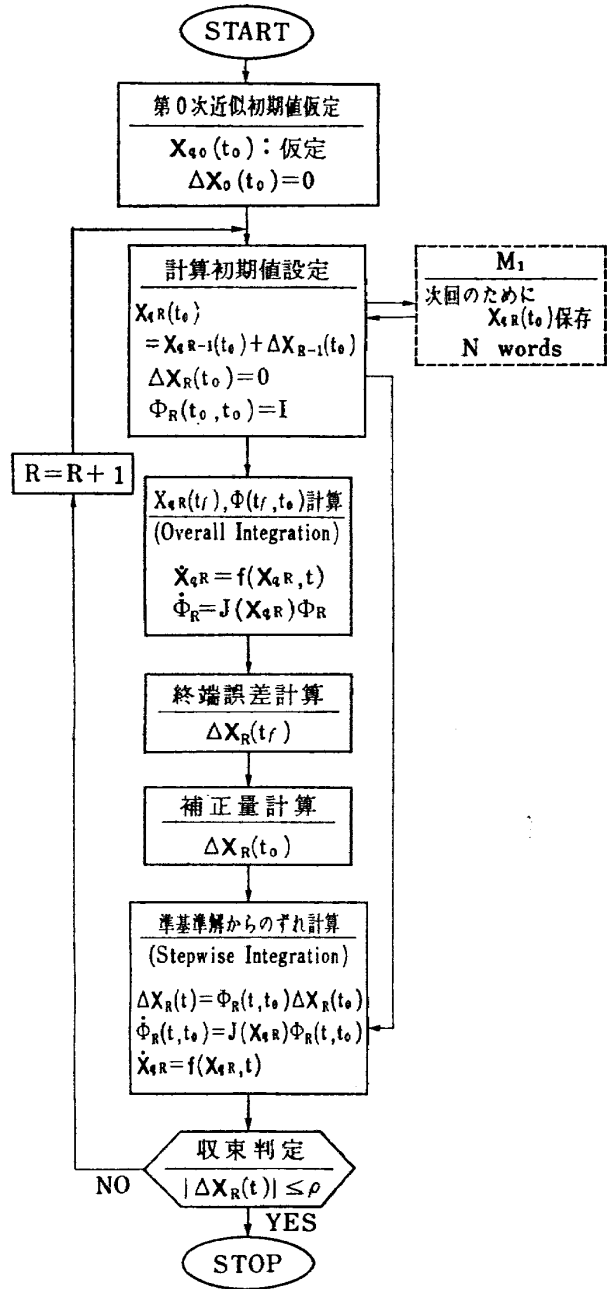


図 5 QLMD アルゴリズムの計算手順

- の初期値を修正する。
 - (5) 第 $R+1$ 次近似として、再び準基準解からのずれはないものとして、 $\Delta X_{R+1}(t_0)=0$ と仮定し、 $X_{qR+1}(t_0)$ を初期値として逐次近似計算を再開する。
 - (6) 解が収束するまで、以上の手順を繰り返す。
 - (7) 解が収束して逐次近似計算が完了した時点で、準基準解は求める解と一致しているはずである。
- 以上に述べた逐次近似修正法 (Iteration logic) を数式的に説明すると以下ようになる。
- 第 R 回目の逐次近似計算が終了した時点 $t=t_f$ では、(6.29) 式から次の関係が成立つ。

$$X_R(t_f) = X_{qR}(t_f) + \Phi_R(t_f, t_0) \Delta X_R(t_0) \quad (6.30)$$

この式で, $X_{qR}(t_f)$, $\Phi_R(t_f, t_0)$ は計算によって与えられ全要素が既知である。 $\Delta X_R(t_0)$ は定義から, 初期境界条件の与えられている k 個の要素に対応するものは,

$$\delta X_R^{vi}(t_0) \equiv 0 \quad (i=1, 2, \dots, k) \quad (6.31)$$

であり, 残りの $n-k$ 個の要素は未知である。また, $X_R(t_f)$ の要素に関しては, 終端境界条件 (3.3) 式から,

$$X_R^{\mu j}(t_f) \equiv b^{\mu j} \quad (j=1, 2, \dots, n-k) \quad (6.32)$$

の恒等式が成立つ。

(6.30) 式を行列およびベクトルの要素を用いて書き改めると, 次のようになる。

$$\begin{pmatrix} X_R^1(t_f) - X_{qR}^1(t_f) \\ \vdots \\ X_R^{\mu j}(t_f) - X_{qR}^{\mu j}(t_f) \\ \vdots \\ X_R^n(t_f) - X_{qR}^n(t_f) \end{pmatrix} = \begin{pmatrix} \varphi_{11} & \dots & \varphi_{1\nu_i} & \dots & \varphi_{1n} \\ \vdots & & \vdots & & \vdots \\ \varphi_{\mu j 1} & \dots & \varphi_{\mu j \nu_i} & \dots & \varphi_{\mu j n} \\ \vdots & & \vdots & & \vdots \\ \varphi_{n1} & \dots & \varphi_{n\nu_i} & \dots & \varphi_{nn} \end{pmatrix} \times \begin{pmatrix} \delta X_R^1(t_0) \\ \vdots \\ \delta X_R^{vi}(t_0) \\ \vdots \\ \delta X_R^n(t_0) \end{pmatrix} \quad (6.33)$$

ここで, $X_R(t_f)$ の要素のうち, 終端境界条件として値が規定されている部分だけに着目して取り出すと, 次式が得られる。

$$\begin{pmatrix} X_R^{\mu 1}(t_f) - X_{qR}^{\mu 1}(t_f) \\ \vdots \\ X_R^{\mu j}(t_f) - X_{qR}^{\mu j}(t_f) \\ \vdots \\ X_R^{\mu n-k}(t_f) - X_{qR}^{\mu n-k}(t_f) \end{pmatrix} = \begin{pmatrix} \varphi_{\mu 1 1} & \dots & \varphi_{\mu 1 \nu_i} & \dots & \varphi_{\mu 1 n} \\ \vdots & & \vdots & & \vdots \\ \varphi_{\mu j 1} & \dots & \varphi_{\mu j \nu_i} & \dots & \varphi_{\mu j n} \\ \vdots & & \vdots & & \vdots \\ \varphi_{\mu n-k 1} & \dots & \varphi_{\mu n-k \nu_i} & \dots & \varphi_{\mu n-k n} \end{pmatrix} \begin{pmatrix} \delta X_{R1}(t_0) \\ \vdots \\ \delta X_{R^{vi}}(t_0) \\ \vdots \\ \delta X_{R^n}(t_0) \end{pmatrix} \quad (6.34)$$

上の (6.34) 式の右边を整理して, 初期境界条件が規定されている要素に対応するものと, 未定の要素に対応する二つの部分に分解すると, 次のように書ける。

$$\begin{pmatrix} X_R^{\mu 1}(t_f) - X_{qR}^{\mu 1}(t_f) \\ \vdots \\ X_R^{\mu n-k}(t_f) - X_{qR}^{\mu n-k}(t_f) \end{pmatrix} = \begin{pmatrix} \varphi_{\mu 1 \nu_1} & \dots & \varphi_{\mu 1 \nu_k} \\ \vdots & & \vdots \\ \varphi_{\mu n-k \nu_1} & \dots & \varphi_{\mu n-k \nu_k} \end{pmatrix} \begin{pmatrix} \delta X_{R^{\nu_1}}(t_0) \\ \vdots \\ \delta X_{R^{\nu_k}}(t_0) \end{pmatrix} + \begin{pmatrix} \varphi_{\mu 1 \nu_{k+1}} & \dots & \varphi_{\mu 1 \nu_n} \\ \vdots & & \vdots \\ \varphi_{\mu n-k \nu_{k+1}} & \dots & \varphi_{\mu n-k \nu_n} \end{pmatrix} \begin{pmatrix} \delta X_{R^{\nu_{k+1}}}(t_0) \\ \vdots \\ \delta X_{R^{\nu_n}}(t_0) \end{pmatrix} \quad (6.35)$$

ここで, (6.31), (6.32) の条件を (6.35) 式に代入

すると, (6.35) 式の右边第1項は消去されて, 次のようになる。

$$\begin{pmatrix} b^{\mu 1} - X_{qR}^{\mu 1}(t_f) \\ \vdots \\ b^{\mu n-k} - X_{qR}^{\mu n-k}(t_f) \end{pmatrix} = \begin{pmatrix} \varphi_{\mu 1 \nu_{k+1}} & \dots & \varphi_{\mu 1 \nu_n} \\ \vdots & & \vdots \\ \varphi_{\mu n-k \nu_{k+1}} & \dots & \varphi_{\mu n-k \nu_n} \end{pmatrix} \begin{pmatrix} \delta X_{R^{\nu_{k+1}}}(t_0) \\ \vdots \\ \delta X_{R^{\nu_n}}(t_0) \end{pmatrix} \quad (6.36)$$

この式は, $(n-k)$ 元連立一次方程式である。簡単のために, 行列・ベクトル表示を用いると,

$$\Phi_{\mu\nu} \delta X = B \quad (6.35)$$

と表わされる。ここで,

$$\Phi_{\mu\nu} \triangleq \begin{pmatrix} \varphi_{\mu 1 \nu_{k+1}} & \dots & \varphi_{\mu 1 \nu_n} \\ \vdots & & \vdots \\ \varphi_{\mu n-k \nu_{k+1}} & \dots & \varphi_{\mu n-k \nu_n} \end{pmatrix} \quad \delta X \triangleq \begin{pmatrix} \delta X_{R^{\nu_{k+1}}}(t_0) \\ \vdots \\ \delta X_{R^{\nu_n}}(t_0) \end{pmatrix}, \quad B \triangleq \begin{pmatrix} b^{\mu 1} - X_{qR}^{\mu 1}(t_f) \\ \vdots \\ b^{\mu n-k} - X_{qR}^{\mu n-k}(t_f) \end{pmatrix}$$

の定義式を用いる。なお, $\Phi_{\mu\nu}$ は遷移行列 $\Phi_R(t_f, t_0)$ の $(n-k)$ 位の Minor となっている。

$\Phi_{\mu\nu}$ が逆行列をもち ill-condition でなければ,

$$\delta X = \Phi_{\mu\nu}^{-1} B \quad (6.36)$$

として (6.35) 式の解が得られる。これは $\Delta X_R(t_0)$ の未知の要素を与えるものである。したがって, 第 R 回目の逐次近似計算の結果として, $\Delta X_R(t_0)$ が決定する。

次の第 $R+1$ 回目の逐次近似計算は,

$$\begin{cases} X_{qR+1}(t_0) = X_{qR}(t_0) + \Delta X_R(t_0) \\ \Delta X_{R+1}(t_0) = 0 \end{cases}$$

と設定し直すことによって開始される。以後, 収束判定の停止条件が満足されるまで, 同様の手順が繰り返され, 逐次近似計算が続行して行く。なお, 停止条件については次節で述べる。

以上述べたように, 逐次近似修正法を構成するための原理を一般論として記述することはできるが, 計算機プログラムを作る立場からみると, 与えられた問題の境界条件の規定の仕方によって, それぞれプログラムに variation をつけなければならない。この点, 代数方程式の根を求めるための Newton-Raphson 法では, 与えられた問題の方程式の特殊性にかかわらず, iteration logic が与えられるのとは異なり, 微分方程式の二点境界値問題の解法では, むしろ与えられた境界条件の特殊性に応じて iteration logic を構成しなければならないことに注意して欲しい。

なお, QLMD アルゴリズムの逐次近似計算の手順

を表わすフロー・チャートを図5に示す。図3のGNRMアルゴリズムの計算手順と比べると、かなりすっきりとしたものになっている。

6.4 QLMDにおける逐次近似の概念

QLMDアルゴリズムでは、前述のIteration logicを用いて、Quasi-reference (準基準) を逐次修正して行くという立場を採っている。換言すれば、逐次近似によって準基準解 $X_{qR}(t)$ を真の解 $X(t)$ に近づけて行く、すなわち、

$$X_{q1}(t), X_{q2}(t), \dots, X_{qR}(t) \Rightarrow X(t)$$

という考え方である。逆に言えば、準基準からのずれ $\Delta X_R(t)$ が、

$$\Delta X_1(t), \Delta X_2(t), \dots, \Delta X_R(t) \Rightarrow 0$$

と収束して行くと考えている訳である。したがって、収束判定は $\Delta X_R(t)$ についてチェックして行けばよい。

収束判定のための停止条件としては、積分区間全域、すなわち、 $t_0 \leq t \leq t_f$ にわたって収束を要求される場合には、次のように与えればよい。

$$|\Delta X_R(t)| \leq \rho \quad (6.37)$$

ここで、 ρ はあらかじめ与える微小な値で、要求する収束精度を表わす。|| は普通の l_2 ノルムである。

図6は、QLMDアルゴリズムの逐次近似の概念を示す説明図である。図中の影をつけた面が第 R 次近似 $X_{qR}(t)$ を表わす超平面であり、終端点において終端境界条件とのずれから誤差 $\Delta X_R(t_f)$ が求まり、それを遷移行列を用いて初期点まで逆行して初期誤差 $\Delta X_R(t_0)$ を計算する。この初期誤差で第 R 次近似の準基準解の初期値を修正して、第 $R+1$ 次近似の計算を開始する訳である。第 $R+1$ 次近似 $X_{qR+1}(t)$ を表わす超平面は、図中でハッチングを施した面である。

逐次近似計算を繰り返すにしたがって、このような面が次第に真の解を表わす超平面に近づいて行く訳である。従来のGNR法では、図4に示すような、斉次解 $H_R(t)$ 、特殊解 $p_R(t)$ 、定数ベクトル c_R の組み合わせによって形成される超空間における三角柱が、次第に真の解に近づいて行くという概念を持っている点が異なっている。

6.5 QLMDアルゴリズムの改良結果

QLMDアルゴリズムでは、解くべき連立微分方程式は(6.22)、(6.27)式となり、状態変数を N 次元ベクトルとすれば、 N_2+N 個の連立微分方程式を解けばよいことになる。これはGNRMより N 個少ない。

連立微分方程式を解くために必要な記憶容量 M_1 としては、次回の逐次近似計算の準備として、現在の準基準解の初期値 $X_{qR}(t_0)$ を保存しておかなければならないので、

$$M_1 = N \text{ words} \quad (6.38)$$

の記憶容量が必要となる。また、1回の逐次近似計算において、連立微分方程式を解くに要する積分時間 T_{integ} は、

$$T_{\text{integ}} = T_H + T_N \quad (6.39)$$

であり、 T_H は遷移行列; T_N は準基準解を求めるのに要する積分時間である。GNRMと比べると、 M_1 の方は変わらないが、積分時間の方は特殊解を求める分だけ減少している。

収束判定に要する記憶容量 M_2 としては、GNRMと同様に、積分計算を Overall integration と Step-wise integration の2段階を繰り返すことによって、記憶容量を占有する代わりに、計算時間で置き換えるので、

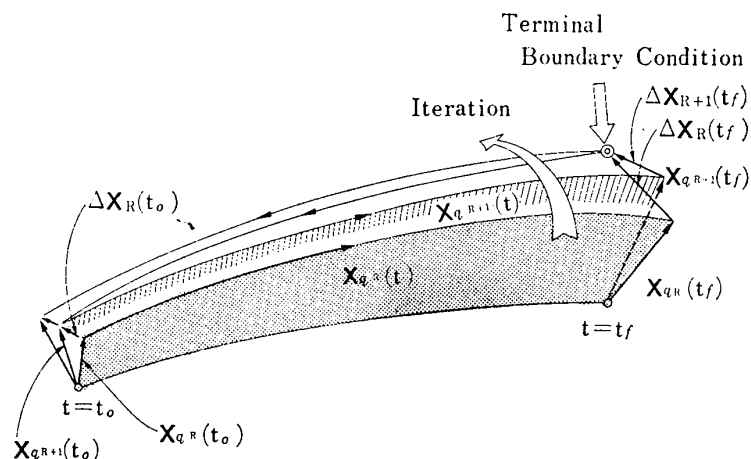


図6 QLMDにおける逐次近似の概念

$$M_2=0 \quad (\text{words}) \quad (6.40)$$

とすることができる。その反面,

$$T_{M_2}=T_H+T_N \quad (6.41)$$

だけ積分時間を余計に必要とする。この T_{M_2} は、 K 分割した各積分刻みごとに遷移行列 $\Phi_R(t, t_0)$ と準基準解 $X_{QR}(t)$ を記憶させる場合に必要なる $(N^2+N)K$ words の記憶容量を代替した結果増えた積分時間である。

6.6 計算例

以上に述べた QLMD アルゴリズムの有効性を確かめるために、簡単な計算例の一つ挙げておく。本報告の目的は、例えば、宇宙飛行体の軌道最適化問題のように、状態変数ベクトルの次元の高い複雑な問題を解く際に直面する所要記憶容量の障害を除去する改良アルゴリズムを提出することにあるので、記憶容量の問題を解決した GNRM と QLMD の二つのアルゴリズムについて比較を行なっている。

例題は文献 30 において、その成果を利用しているが、静止衛星の軌道保持の問題として、ずれた軌道からもとの静止軌道へ戻すための変換軌道を求める問題である。図 7 に示すように、軌道 1 から軌道 2 へ最小エネルギーで移行する軌道を決定する問題で、両軌道に接する長円軌道で与えられる、いわゆる Hohmann 軌道移行が、その解であるが、実際にその軌道を求めるには、図に示す軌道上の r_1, r_2 の二つの位置ベクトルと時間間隔 T を与えられた場合の軌道決定の問題を解かなければならない。この問題は Lambert の問題と呼ばれることがあるが、二点境界値問題となっている。

問題を簡単化するために、二体問題とし、両軌道とも同一平面上にあるとすれば、求める移行軌道の方程

式は、次のように与えられる。

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (6.42)$$

$$\mu = G(m_1 + m_2) \quad (6.43)$$

G : 万有式力定数

m_1 : 地球質量

m_2 : 衛星質量

境界条件としては、

$$\mathbf{r}_1 = \begin{pmatrix} 42,133.766666 \\ 0 \end{pmatrix} \text{ km}$$

$$\mathbf{r}_2 = \begin{pmatrix} -42,146.494716 \\ 0 \end{pmatrix} \text{ km}$$

$$T = \pi/7.292115 \times 10^{-5} \text{ sec}$$

の数値を与えて、速度 $\mathbf{v}(t_0)$ および $\mathbf{v}(t_f)$ を求めれば軌道移行の問題は解け、移行に必要な速度修正量も決定される。

以上の問題を QLMD および GNRM で解かせるために、本報告で用いている形に式を書き換えよう。

(6.42) 式を書き改めると、次のようになる。

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\frac{\mu}{r^3} \mathbf{r} \end{bmatrix} \quad (6.44)$$

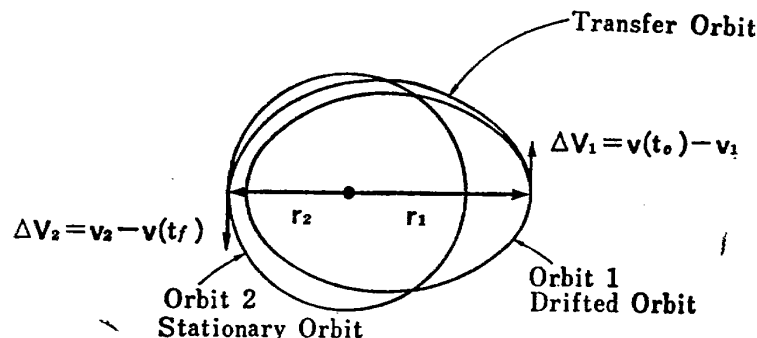
$$\mathbf{v}(t) = \dot{\mathbf{r}}(t)$$

$$t_0=0, \quad t_f=T$$

上の式は

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ v_x \\ v_y \end{bmatrix} \quad (6.45)$$

と定義すれば、非線型な原方程式 (3.1) の具体的な表現にほかならない。境界条件は



ΔV_i : Impulsive Velocity Correction

v_i : Velocity of Orbit i ($i=1, 2$)

$v(t_0)$: Initial Velocity of Transfer Orbit

$v(t_f)$: Final Velocity of Transfer Orbit

図 7 軌道移行による軌道保持

第 1 表 逐次近似計算の結果

GNRM (Computing Time=149 sec)

Iteration	$v_x(t_0)$	$v_y(t_0)$	$v_x(t_f)$	$v_y(t_f)$	Max RMS Error
0	0.0 $\times 10^{-40}$	0.30730817×10^4	—	—	—
1	$-0.44624903 \times 10^{-1}$	0.30733136×10^4	$-0.45382480 \times 10^{-1}$	-0.30723866×10^4	0.30330535×10^5
2	$-0.44922734 \times 10^{-1}$	0.30733137×10^4	$-0.44922734 \times 10^{-1}$	-0.30723856×10^4	0.75391728×10^1
3	$-0.44922734 \times 10^{-1}$	0.30733137×10^4	$-0.44922734 \times 10^{-1}$	-0.30723856×10^4	$0.18678099 \times 10^{-5}$
4	$-0.44922734 \times 10^{-1}$	0.30733137×10^4	$-0.44922734 \times 10^{-1}$	-0.30723856×10^4	$0.55471220 \times 10^{-6}$

QLMD (Computing Time=94 sec)

Iteration	$v_x(t_0)$	$v_y(t_0)$	$v_x(t_f)$	$v_y(t_f)$	Max RMS Error
0	0.0 $\times 10^{-40}$	0.30730817×10^4	—	—	—
1	$-0.44624903 \times 10^{-1}$	0.30733137×10^4	$-0.45382480 \times 10^{-1}$	-0.30723866×10^4	0.30330535×10^5
2	$-0.44922734 \times 10^{-1}$	0.30733137×10^4	$-0.44922734 \times 10^{-1}$	-0.30723856×10^4	0.73391726×10^1
3	$-0.44922734 \times 10^{-1}$	0.30733137×10^4	$-0.44922734 \times 10^{-1}$	-0.30723856×10^4	$0.13762875 \times 10^{-6}$

$$\begin{cases} X^1(t_0) = 42, 133.766666 \\ X^2(t_0) = 0 \end{cases} \quad (6.46)$$

$$\begin{cases} X^1(t_f) = -42, 146.494716 \\ X^2(t_f) = 0 \end{cases} \quad (6.47)$$

となり, (3.2), (3.3) 式において, それぞれ, $v_1 = \mu_1 = 1$, $v_2 = \mu_2 = 2$ および $n=4, k=2$ と置いたことに相当している。

あとは, 本報告に述べてある計算手順にしたがって, 計算プログラムを作ればよい訳であるが, 線型化式に必要な Jacobian の具体的な形を参考のために記しておく, 次のようである。

$$J(X) = \begin{bmatrix} O_3 & I_3 \\ G(t) & O_3 \end{bmatrix} \quad (6.48)$$

$$G(t) = \frac{\mu}{r^5} (3rr^T - r^2 I_3) \quad (6.49)$$

I_3 : 3×3 単位行列

O_3 : 3×3 零行列

積分刻みを $K=1,000$ とし, 収束精度を $\rho=10^{-6}$ として計算した結果を第 1 表に示す。計算時間としてあるのは計算機の Computing time であって, 本報告で定義している積分時間をその中に含むものである。積分時間は計算時間の大部分を占められるので比較の目安にはなろう。GNRM の場合も, QLMD の場合も逐次近似回数の 2 回目で, 端点の値は最終解が現われているが, その後も逐次近似計算が続行されるのは, 収束判定を軌道全域にわたって検討したためである。右端の欄は平方自乗誤差(Root Mean Square Error) の最大値を示す。上の問題を解かせると,

第 2 表 所要記憶容量の比較 (単位 words)

Algorithm	M_1	M_2	Total
GNR	4,000	20,000	24,000
Bellman	40	20,000	20,040
GNRM	4	0	4
QLMD	4	0	4

GNRM では 4 回目で収束し, QLMD では 3 回の逐次近似で収束した。GNRM では線型化方程式を用いているので, やや余分に計算を繰り返さなければならないようである。逐次近似計算 1 回当たりの計算時間にすると, あまり差はないが, それでも QLMD の方が計算時間は少なくなっている。この問題のように簡単な場合には, あまり差は出てこないが, 状態変数ベクトルの次元が高くなるにつれて, 計算時間の差は大きくなって行くはずである。なぜならば, 計算時間は変数の次元の関数でもあるからである。いずれにせよ, この問題に限って言えば, QLMD は所要計算時間を GNR の 3/4 に短縮することができた。

所要記憶容量を, 速立微分方程式を解くための M_1 と収束判定のための M_2 に関して検討してみると, 第 2 表のようになる。なお, Bellman のアルゴリズムの場合には, 最大逐次近似回数を 10 回として算定してある。

表からわかるように, QLMD は GNR とともに所要記憶容量は 4 words で極めて少ない。GNR 法の原アルゴリズムや Bellman のアルゴリズムでは,

第3表 各アルゴリズムの比較対照

アルゴリズム 比較項目	GNR	Bellman	GNRM	QLMD
連立微分方程式の数	$N^2 + N$	$N^2 + RN \ (R=1, 2, \dots)$	$N^2 + 2N$	$N^2 + N$
齊次解方程式 (Homogeneous eq.)		$\dot{H}_R(t) = J(X_{R-1})H_R(t)$ $H_R(t_0) = I$		遷移行列方程式 (Transition Matrix eq.) $\dot{\phi}_R(t, t_0) = J(X_{qR})\phi_R(t, t_0)$ $\phi_R(t_0, t_0) = I$
特殊解方程式 (Particular eq.)		$\dot{p}_R(t) = f(X_{R-1}) + J(X_{R-1})(p_R - X_{R-1})$ $p_R(t_0) = 0$		準基準解方程式 (Quasi-reference eq.) $\dot{X}_{qR} = f(X_{qR}, t)$ $X_{qR}(t_0) = X_{qR-1}(t_0) + \Delta X_{R-1}(t_0)$
記憶方程式 (Memory eq.)	Memory Storage M_1	$\dot{X}_l = f(X_{l-1}) + J(X_{l-1})(X_l - X_{l-1})$ $X_l(t_0) = c_l \ (l=1, 2, \dots, R-1)$	$\dot{X}_{R-1} = f(X_{R-1}, t)$ $X_{R-1}(t_0) = c_{R-1}$	
第0次近似式	$X_0(t) = g(X_0(t_0), t)$	$X_0(t) = g(X_0(t_0), t)$		
1 iteration 当りの 積分時間 $T_{integ.}$	$T_H + T_P$	$T_H + R \times T_P$	$T_H + T_P + T_N$	$T_H + T_N$
記憶容量 M_1 (方程式を解くため)	$N \times K$ words ($X_{R-1}(t)$ Store)	$N \times R$ words (c_0, c_1, \dots, c_{R-1} Store)	N words (c_{R-1} Store)	N words ($X_{qR-1}(t_0)$ Store)
記憶容量 M_2 (収束判定のため)	$(N^2 + N) \times K$ words	$(N^2 + N) \times K$ words	0 但し $T_{M2} = T_H + T_P + T_N$	0 但し $T_{M2} = T_H + T_N$

このような簡単な問題ですら要求される記憶容量がかなり大きくなり、さらに状態変数の数が増えた問題になれば、やはり許算が困難となる。

所要記憶容量の点では、QLMD も GNRM も変わらないが、計算時間の点では、QLMD の方がすぐれているということが、上の例題から結論できる。

7. 結論およびあとがき

これまでの議論の内容を要約した結果を第3表と図8に示す。

第3表は本報告の成果である QLMD アルゴリズムと他のアルゴリズムとの比較対照の結果を表にまとめたものである。表から明らかなように、QLMD は

GNRM とともに所要記憶容量が非常に減少している。さらに、QLMD は解くべき方程式の数も整理統合されていて、所要積分時間が GNRM よりも短縮されている。したがって、QLMD は状態変数ベクトルの次元の大きい複雑な最適化問題を解くことを目的としたアルゴリズムであるということが出来る。

図8は各アルゴリズムと改良点との相互関係を図解したものであり、GNR 法の持つ問題点が各アルゴリズムによってどのように解決されて行なったかを示している。図中で改良のアイディアを二重枠で囲んであり、アルゴリズムを示す枠から左方へ延びている矢印は問題点がまだ残っていることを示す。GNR 法では収束性は良いが、第0次近似の選び方がまずいと問題

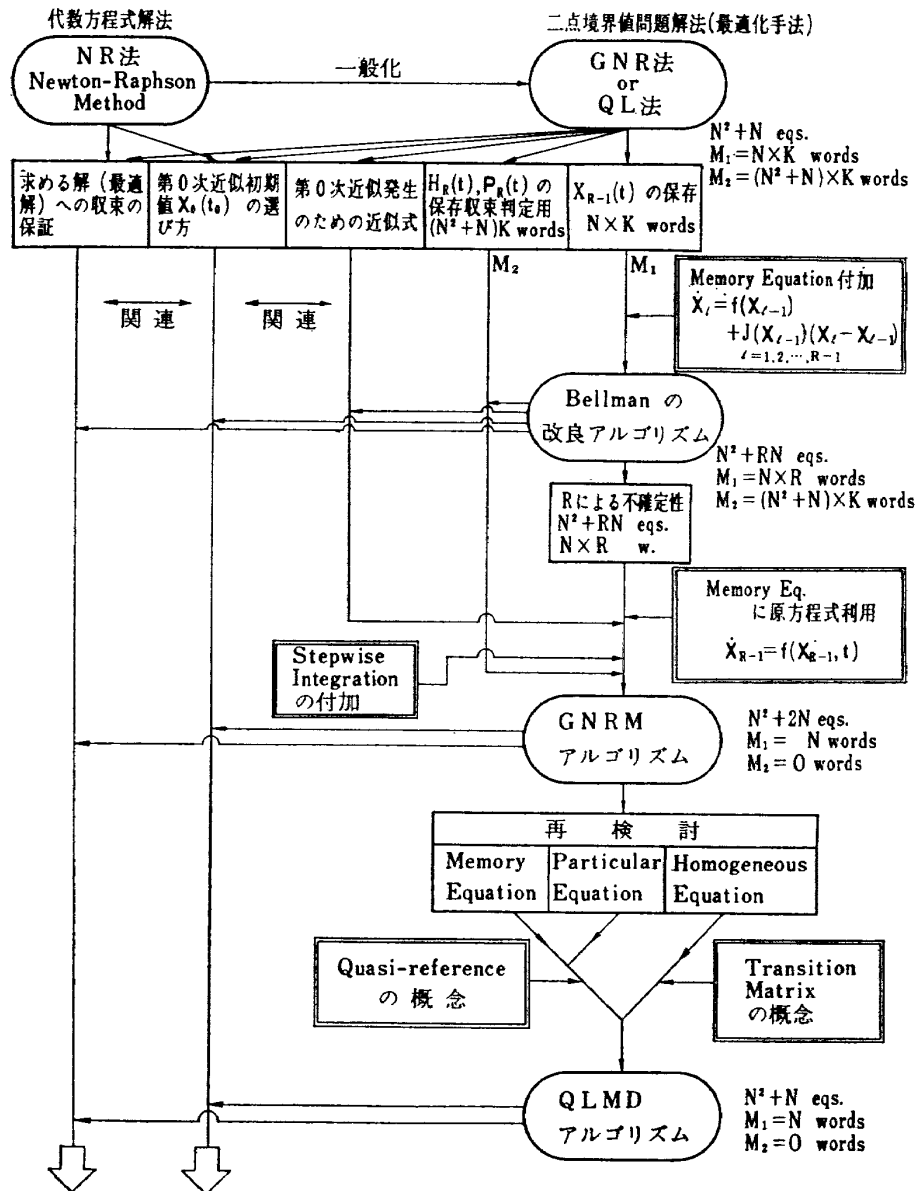


図8 各アルゴリズムの相互関係と改良点

によっては最適解に収束せず局所的最適解に収束してしまう場合があるが、QLMD においても、やはり事情は同じで、これらの問題点は残されている。ただし、与えられた問題に関する物理的解釈が、これらの問題点を除去するための有効な情報を与えてくれるので致命的な欠点とはならないと考えられる。

改良の結果、もとの GNR 法の面影は大分うすれてしまったが、QLMD はやはり GNR 法の系列に属していて収束の状況もよく似ている。なお、QLMD の収束性の証明はまだ完成していないので次の機会にゆずる。本報告はアルゴリズムの構築に重点を置いているので、簡単な計算例一つにとどめたが、本アルゴリズムが目的としているような状態変数の多い複雑な問題への応用例は稿を改めて公表して行く予定である。QLMD のアルゴリズムのみを取り上げて公表するのは、収束性の証明の未完成の故に遅くなったが、すでに発表されている村田の研究³¹⁾において、計算手段として利用されて成果を挙げていることをつけ加えておく。

本報告は研究成果の報告であると同時に、GNR 法に関する邦文の文献がほとんど無いことから GNR 法の紹介をも兼ねて、説明にかなりの頁数を割いた。

なお、本報告では終端時間固定の問題として議論を進めて来たが、終端時間自由の問題、関数型境界条件の問題への拡張は容易である。これらに関しては、稿を改めて公表する。また、状態変数あるいは制御変数に制限のある問題も別の機会に検討してみたいと考えている。この問題には、GNR 法における McGill と Kenneth の文献 (17, 19) が参考になろう。

最後に、アルゴリズムの重要性を強調され本報告の公表を熱心にすすめて下さった樋口一雄計測部長、奥石肇計測研究室長、および討論に参加し貴重な御意見を下さった計測部松島弘一技官、村田正秋技官、宇宙研究グループ新田慶治主任研究官、大坪孔治技官に感謝の意を表す。また、アルゴリズムの検討には当研究所計算センターの HITAC 5020 F を使用したことを付記する。

参 考 文 献

- 1) Athans, M and Falb, P.L.: Optimal Control, McGraw-Hill, New York, 1966
- 2) Eveleigh, V.W.: Adaptive Control and Optimization Techniques, McGraw-Hill, New York, 1967, Chapt. 17
- 3) Bellman, R.E., and Kalaba, R.E.: Quasilinearization and Nonlinear Boundary-Value

- Problems, American Elsevier Publishing Co., New York, 1965
- 4) Lee, E. Stanley: Quasilinearization and Invariant Imbedding, Academic Press, New York, 1968
- 5) Greensite, A.L.: Analysis and Design of Space Vehicle Flight Control Systems, Vol. IX Optimization Methods, NASA CR-828, 1967
- 6) McIntyre, J.E.: Guidance, Flight Mechanics and Trajectory Optimization, Vol. XIII Numerical Optimization Methods, NASA CR-1012, 1968
- 7) Kantorovich, L.V.: Functional Analysis and Applied Mathematics, Dokl. Akad. Nauk SSSR (N.S.) 59, pp. 1237-1240, 1948. (未入手)
- 8) Hestenes, M. R.: Numerical Methods of Obtaining Solutions of Fixed End Point Problems in the Calculus of Variations, RAND Corp. Memo. No. RM-102, 1949
- 9) Warga, J.: On a Class of Iterative Procedures for Solving Normal Systems of Ordinary Differential Equations, J. Math. Phys. Vol. XXXI, No. 1, pp 223-243, 1953
- 10) Bellman, R.E.: Functional Equations in the Theory of Dynamic Programming, Part V Positivity and Quasilinearity, Proc. of Nat. Acad. Sci., Vol. 41, pp. 743-746, 1955
- 11) Kalaba, R. E.: On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence, J. Math., Vol. 8, No. 4, pp. 519-573, 1959.
- 12) Bellman, R.E., Kagiwada, H. and Kalaba, R.E.: Orbit Determination as a Multi-Point Boundary-Value Problem and Quasilinearization, Proc. of Nat. Acad. Sci., Vol. 48, pp. 1327-1329, 1962
- 13) McGill, R. and Kenneth, P.: A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary-Value Systems, XIVth IAF Cong., Paris, 1963
- 14) McGill, R. and Kenneth, P.: Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator, AIAA J. Vol. 2, No. 10, pp. 1761-1766, 1964.
- 15) Long, R.S.: Newton-Raphson Operator; Problems with Undetermined End Points, AIAA J. Vol. 3, No. 7, pp. 1351-1352, 1965
- 16) Long, R. S.: Quasilinearization and Orbit Determination, AIAA J. Vol. 3, No. 10, pp. 1937-1940, 1965.
- 17) McGill, R.: Optimal Control, Inequality State Constraints, and the Generalized Newton-Raphson Algorithm, J. SIAM Control, Ser. A, Vol. 3, No. 2, pp. 291-298, 1965
- 18) Van Dine, C.P., Fimple, W.R., and Edel-

- baum, T.N.: Application of a Finite-Difference Newton-Raphson Algorithm to Problems of Low-Thrust Trajectory Optimization, Progress in Astronautics, Vol. 17, Academic Press, New York, pp. 377-400, 1966
- 19) Kenneth, P, and McGill, R.: Two-Point Boundary-Value Problem Techniques, Advances in Control Systems, (C.T. Leondes ed.) Vol. 6, Academic Press, pp. 68-108, 1966
- 20) Lewallen, J.M. and Tapley, B.D.: Analysis and Comparison of Several Numerical Optimization Methods, AIAA Paper No. 67-58, 1967
- 21) Tapley, B.D. and Lewallen, J.M.: Comparison of Several Numerical Optimization Methods, J. of Opt. Theor. Appl. Vol. 1, No. 1, pp. 1-32, 1967
- 22) Lewallen, J.M.: A Modified Quasi-Linearization Method for Solving Trajectory Optimization Problems, AIAA J. Vol. 5, No. 5, pp. 962-965, 1967
- 23) Lewallen, J.M.: A Modified Quasilinearization Concept for Solving the Nonlinear Two-Point Boundary Value Problem, NASA TN D-4025, 1967
- 24) Lewallen, J.M., Tapley, B.D., and Williams, S.D.: Iterative Procedures for Indirect Trajectory Optimization Methods, J. Spacecraft, Vol. 5, No. 3, pp. 321-327, 1968.
- 25) Bellman, R.E.: Successive Approximations and Computer Storage Problems in Ordinary Differential Equations, Comm. ACM, Vol. 4, pp. 222-223, 1961.
- 26) Tanabe, T.: Fly-by Process: Guidance and Control, Doctoral Dissertation, Univ. of Tokyo, 1966
- 27) 志甫 徹: 自機搭載用宇宙航行システムのための軌道推定に関する研究, Doctoral Dissertation, Univ. of Tokyo, 1966.
- 28) Shiho, T.: Two Proposed Methods of Trajectory Estimation for Self-Contained Space Navigation System, Proc. of 7th ISTS, Tokyo, pp. 627-642, 1967
- 29) 新田慶治, 松島弘一, 大坪孔治, 志甫 徹: 宇宙飛しょう体の直接式最適誘導法, 航技研報告 TR-161, 1968/8.
- 30) 新田慶治, 松島弘一, 大坪孔治, 志甫 徹: 静止衛星の軌道保持 1969/1 第12回宇宙科学技術講演会予稿集
- 31) 村田正秋: 非線型確率系の準最適制御について, 日本自動制御協会第2回統計学的制御理論シンポジウム予稿集 pp. 105-108, 1970/11
- 32) Baird, C.A., Jr.: Modified Quasilinearization Technique for the Solution of Boundary-Value Problems for Ordinary Differential Equations, J. of Opt. Theo, Appl., Vol. 3, No. 4, pp. 227-242, 1969
- 33) Pitkin, E.T. and Carpenter, G.C.: Modified Newton-Raphson Methods for Preliminary Orbit Determination, Celestial Mechanics, Vol. 1, pp. 72-90, 1969

- | | | | |
|--------|--|----------|----------------------|
| TR-242 | 熱線による3次元平均速度ベクトルとレイノルズ応力の測定
Measurement of Three-Dimensional Mean Velocity Vector and Reynolds Stress by Single Rotatable Hot-Wire | 1971年7月 | 林 良生, 中谷輝臣 |
| TR-243 | 揚力をもつ物体のまわりのポテンシャル流れの計算法 I その理論的基礎
A Method for the Calculation of Lifting Potential Flow Problems
—Part 1 Theoretical Basis— | 1971年7月 | 海老原正夫 |
| TR-244 | 超音速風洞における動安定微係数の測定
Measurements of Dynamic Stability Derivatives in Supersonic Blowdown Wind Tunnel | 1971年7月 | 高島一明, 榊原盛三
関根英夫 |
| TR-245 | ヘリコプタ高度・速度線図および臨界決定点の解析的推定法
An Analytical Method to Predict Height-Velocity Diagram Critical Decision Point of Rotorcraft | 1971年8月 | 古茂田真幸 |
| TR-246 | 梁の大変形問題の解析について
On Analysis of Large Deformation Problems of Beam | 1971年11月 | 泉 日出夫 |
| TR-247 | 任意物体のまわりの自由流線理論
On the Theory of Free Streamlines Past an Arbitrary Shape | 1971年9月 | 高橋 侔 |
| TR-248 | 亜音速流中における二次元翼上の圧力分布の粘性効果を考慮に入れた計算法のプログラム
A Description of the Ideas Underlying a Computer Programme for Predicting the Aerofoil Pressure Distributions in Subcritical Viscous Flows | 1971年11月 | 海老原正夫, 石田洋治
小此木時雄 |

航空宇宙技術研究所報告249号

昭和46年10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 電話武蔵野三鷹(0422)44-9171(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
