A DESIGN OF THE NUMERICAL CALCULATION METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS

Kajiro WATANABE*, Makoto TERAOKA**

Abstract

This paper proposes a new design procedure of single-step numerical calculation methods for ordinary differential equations. This procedure is composed of the following two processes (a) determination of the appropriate shift operator function which transforms differential equations to difference equations and (b) construction of the algorithm based on the above shift operator. First stage is prepared for the method to satisfy the specified accuracy and stability. Second stage is for the method to be adaptable to the given equation form. Following to this procedure, a method for large scale Laqrange's equation and a method for nonlinear stiff equations are designed.

1. Introduction

Research investigations of the numerical methods for ordinary differential equations have attracted much attention as an important tool of an industrial system design and analysis, during past several years.[1, 2, 3, 4, 5] Most of the investigations have been the comparison of the various numerical methods and the development of the special method applicable to specified problems.

Among these investigations, Henrici[6, 7] has established a theory concerned with the discretization error, numerical stability and error propagation. Gear[1] has investigated the numerical period elongation phenomenon of the oscillatory equations. Watanabe and Shimizu[8] have described a systematic error characteristics of various numerical methods by the transfer function approach. Meanwhile Newmark[9] proposed his β-parameter method for the oscillatory equations. Wilson[10] and Argyris[11] also proposed methods for the similar equations.

This paper intends to summarize the above investigations as a design problem of the numerical method and proposes a general procedure by which a suitable numerical method is designed for given equations. Further, this paper proposes two methods ; a method for Lagrange equation of motion and a method for nonlinear stiff equations, so as to explain this design procedure concretely.

2. Design procedure of the numerical calculation method for ordinary differential equations

2.1 Characteristics of the equations and demands for the solutions

* Associate Prof. Electrical Eng'ng of Technical Dept. of Hosei Univ.

** Graduate Student Electrical Eng'ng of Technical Dept. of Hosei Univ.

3-7-2 Kajino-Cho, Koganei-Shi, Tokyo, Japan (〒184)

Table 1 Classification of the ordinary differential equations when they are solved by the numerical calculation method

linear	large scale	ruled sparse	(1)	The equations with the band structure or the ruled sparse coefficient matrix, these sparsities can be possitively made use of whatever the algorithm are.
		nonruled sparse	(2)	The equations with nonruled sparse matrix. Any algorithm can not be adaptable, except the expliated or the similar scheme.
	middle scale		(3)	The equations which do not need any accounts for sparsity. Numerically tough method is better even if any algorithm can be used.
	distribution of eigen value in sr plane	widely	(4)	The equations including the very high and low frequency solution components and is called stiff equations.
		around the imaginary axis	(5)	The oscillatory equations like as the equation of wave propagation and Lagrange equations of motion.
		around the real axis	(6)	The nonoscillatory equations like as the thermal equation. The optimal method making use of this properties can be designed.
		near origin	(7)	The equations which are easily processable by the standard method.
nonlinear	middle scale		(8)	The equations almost same as following item (9).
	large scale		(9)	The equations of which mathematical characteristics are not known so that a numerical tough method is desiarable to obtain the solution as the first approximation.
	linearizable nonlinear		(10)	The equations of which mathematical properties are not always clarified so that the similar approaches as item (9) should be taken.
	essential nonlinear		(11)	The equations of which mathematical properties are not clarified at all so that the prudent selection of the method and treatments are necessary.

Table 1 shows a classification of the ordinary differential equations, appeared in the technological system design and also in the scientific system analysis. A proper algorithm must be selected corresponding to the classification in Table 1.

The demands for the numerical solutions depend on the purposes of the problems and also on the fields of applications. These demands are classified roughly into three cases as shown in Table 2. The most desirable solution is the one given in the case 1 of Table 2. But, in this case, it is necessary to use a high speed and large scale computer. When such a computer of high capacity is not available or when the solutions do not need to satisfy all of thre requirements of case 1, case 2 and case 3 become the main objects of the design.

2. 2 Foundation and classification of the numerical calculation methods

The usual numerical calculation method for ordinary differential equations is consisted of a process to transform the ordinary differential equation (1) to the difference equation (2) and of a process to solve this difference equation.

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\ \mathbf{u}) \tag{1}$$

$$\mathbf{x}_{n+1}=\mathbf{F}(\mathbf{x}_n,\ \mathbf{u}_{n-k},\ \ldots,\ \mathbf{u}_n,\ \ldots,\ \mathbf{u}_{n+l}) \tag{2}$$

Table 2 Demands for the numerical solutions

Case	Accuracy		Processing Speed		Numerical Stability		
	high	low	fast	slow	good	bad	
1	0		0		0		The case when the high speed processings are necessary. Examples; Orbit calculation of the artifical satellite. Calculation of the atomic Energy plant.
2		0	0		0		The case to obtain the first approximate solution of the large scale and stiff equation. Examples; Simulation of the control system responce calculation of the structures excited by the earthquake.
3	0			0	0		The case which needs the most accurate solutions. Examples; Calculation in astronomy and nuclear physics.

where $\mathbf{x}$ is an n dimensional solution vector, $\mathbf{u}$ is the nonhomogenious term of an m dimensional vector, and the time coordinate of the difference equation are identified with subscripts.

A linear ordinary differential equation of the following kind

$$\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}\sum_{i=0}^{l}\frac{t^i}{i!}\nabla^i\mathbf{u}_n \qquad (3)$$

is considered as a special case of (1), where $\mathbf{A}$ is $n\times n$ and $\mathbf{B}$ is $n\times m$ matrix. When the nonhomogenious term is an lth order polynomial, interpolated by the given discrete data u_n, (3) can be transformed to the difference equation with stepwidth τ, as follows;

$$\mathbf{x}_{n+1}=\mathbf{F}(\mathbf{A}\tau)\mathbf{x}_n+\sum_{i=0}^{l}\{(\mathbf{F}(\mathbf{A}\tau)-\mathbf{I})\mathbf{A}^{-(i+1)}-\sum_{j=0}^{i-1}\frac{\tau^{i-j}}{(i-j)!}\mathbf{A}^{-(j+1)}\}\mathbf{B}\nabla^i\mathbf{u}_n/\tau^i \qquad (4)$$

If (3) is transformed exactly to (4), $\mathbf{F}(\mathbf{A}\tau)$ in (4) must be

$$\mathbf{F}(\mathbf{A}\tau)=e^{\mathbf{A}\tau} \qquad (5)$$

The direct numerical processing of (5) needs a large number of numerical operations, so that the function $\mathbf{F}(\mathbf{A}\tau)$ is approximated by some proper function which is easily accessible. This is the same with using the approximate shift operator $\mathbf{F}(s\tau)$ instead of $e^{s\tau}(=z)$ of z transformation. The single-step numerical calculation methods can be classified by the function form of $\mathbf{F}(s\tau)$. The function form is classified into three cases as listed in Table 3. These cases are corresponded to an explicit, an implicit and a predictor-corrector algorithm.

The above approaches are mainly for the linear differential equations, but the approch by the approximate shift operator $\mathbf{F}(s\tau)$ is adaptable even to the nonlinear equations. Because in the formation of the algorithm, the nonlinear equation can be reqarded as locally linearized by the mean value theorem.

2.3 Characteristic criteria for designing the numerical calculation method

For designing the numerical calculation method, it is necessary to evaluate the following items; (a) Adaptivity to the sparsity of a coefficient matrix of equation, (b)

Table 3 Classification of the numerical calculation method by the approximate functions of $e^{s\tau}$ function

Approximate function of $e^{s\tau}$	Scheme	Characteristics	Typical algorithm
$\sum_{i=0}^{n} a_i(s\tau)^i$	Explicit	This scheme provides the simple algorithm which is adaptable to nonlinear equations and linear equations with nonruled sparse matrix. The accuracy of the solution can be improved easily by ajusting the order of the function, but the numerical stability is not assured.	Euler's Runge-Kutta's optimized Runge-Kutta's Milne's
$\sum_{i=0}^{n} a_i(s\tau)^i / \sum_{j=0}^{m} b_j(s\tau)^j$	Implicit	This function form can assure the numerical stability and is adaptable to the ruled sparsity of the coefficient matrix. But the algorithm is not always simple and it is difficult to apply it to nonlinear equation.	Backward Euler's Crank Nicholson's Newmark-β's Stiffly stable's Houbolt's
$\sum_{i=0}^{n} a_i(s\tau)^i / \sum_{j=0}^{m} b_j(s\tau)^j$	Predictor -corrector	Characteristics are similar to the explicit scheme. In this case, the denominator is processed by Taylor expanded form.	Milne-Hamming's Midpoint -trapezoidal Adams Bashforth Moulton's Argyrise's

Numerical stability, (c) Accuracy, (d) Processing speed, (e) Discretization error, (f) Simplicity of an algorithm. Among these criteria, the items (b) and (c) are evaluated by the characteristics of $F(s\tau)$ and the items (a) and (f) are by the function form of $F(s\tau)$ itself. In these items, item (b) and (c) are so important that these items must be treated quantitatively. The items (d), (e) and (f) are evaluated derivatively from the above discussions of the item (b) and (c).

2.3.1 Accuracy

Let $s\tau$ be (a characteritic root of the differential equation) × (stepwidth) and let z be a characteristic root of the difference equation.

Consider the mappings of $s\tau$ to z by $e^{s\tau}$ and by its approximate function $F(s\tau)$. The mapping by $e^{s\tau}$ is shown in Fig 1. (b) and the mapping by $F(s\tau)$ is in Fig. 1 (c). The inverse mapping of Fig. 1 (b) by $\ln(z)$ gives exact $s\tau$ value as in Fig. 1 (d), whereas the mapping of Fig. 1 (c) by $\ln(z)$ leads different value from original one as in Fig. 1 (e). Such a plane as shown in Fig. 1 (e) is generated by

$$\overline{s\tau}=\ln\{F(s\tau)\} \qquad (6)$$

and this $\overline{s\tau}$ plane is distorted by the characteristics of $F(s\tau)$ function. When such $F(s\tau)$ is adopted as a shift operator of a numerical calculation method, period and damping ratio of numerical solution deviate systematically from those of correct solution. If $\overline{s\tau}$ shown by a symbol ○ in Fig. 1 (e) is near to the original value $s\tau$ on the broad region of the plane, the method solves the equations with the high accuracy. Therefore the magnitude of the distortion of the $\overline{s\tau}$ plane as in Fig. 1 (e) can serve as the assessment of the accuracy of the given methods.

As examples, $\overline{s\tau}$ of Euler method and 4th order Runge Kutta method are shown in Fig. 2 and Fig. 3, where the dotted lines are $\overline{s\tau}$ and solid lines are $s\tau$, the corresponding original value. The figures show only the left upper quater of the complex plane.

2.3.2 Numerical stability

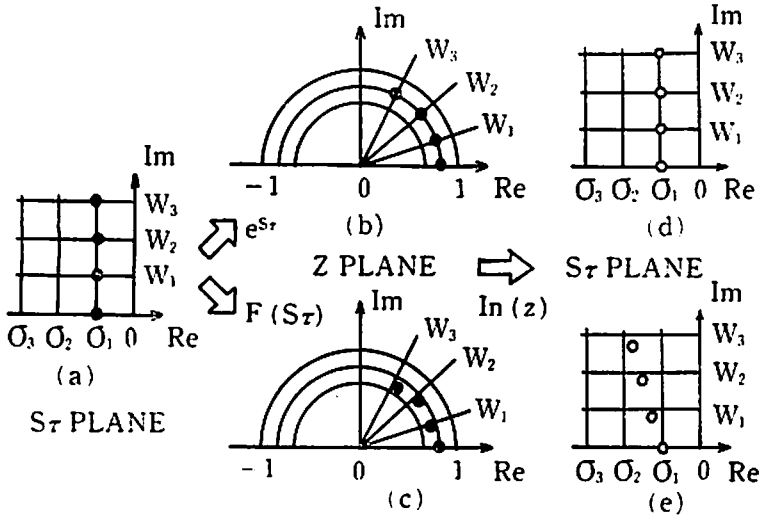

Fig. 1. Mapping relation between the sτ plane and the z plane

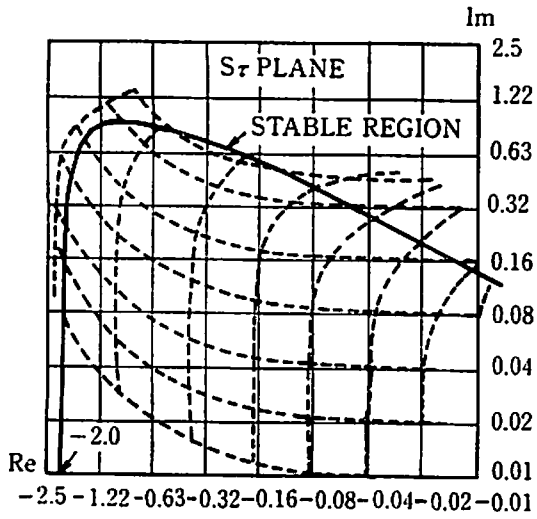

Fig. 2. Distortion characteristics of the sτ plane and stable region of the Euler method

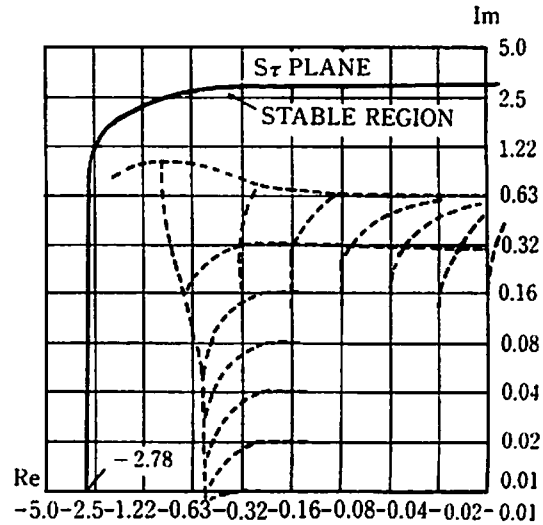

Fig. 3. Distortion characteristics of the sτ plane and stable region of the Runge Kutta method

The real part of $\overline{s\tau}$ may happen to be positive, even if the real part of the value $s\tau$ is negative. This means that the stable equation is taken as the unstable equation in the approximate method. Such a phenomenon is called the numerical instability. The numerical instability border on the $s\tau$ plane is given by the root locus of the following equation.

$$F(s\tau)=e^{i\omega\tau} \quad (\omega\tau=0\sim\pi) \tag{7}$$

Where $e^{i\omega\tau}$ is the stable border on the z plane. The stable borders of the approximate shift operators

$$F(s\tau)=\sum_{i=0}^{n}(s\tau)^i/i! \quad (n=1\sim9) \tag{8}$$

are shown in Fig. 4. The curve for n=1 is the border of Euler method and the curve for n=4 is one of 4th order Runge Kutta method.

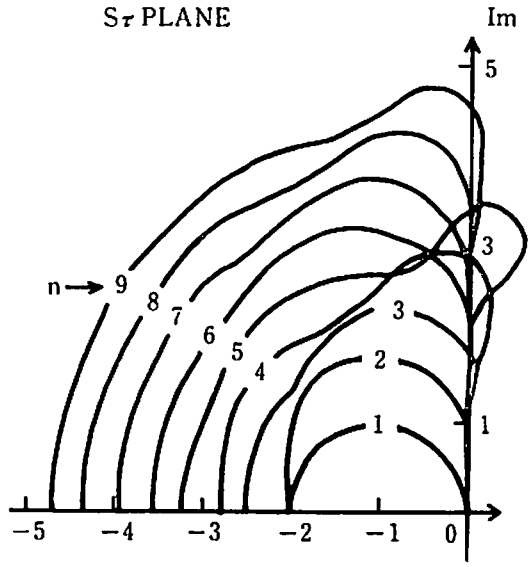

Fig. 4. Stable region of the approximate function $\sum_{i=0}^{n} (s\tau)^i/i!$ of the $e^{s\tau}$ function

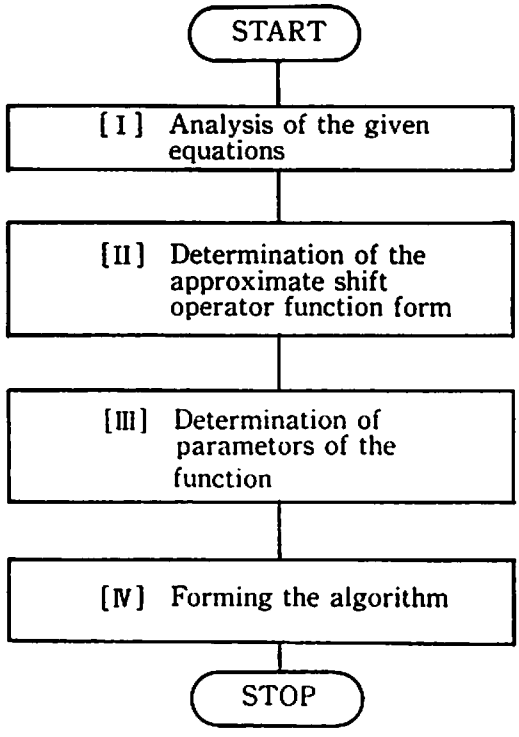

Fig. 5. Design procedure of the numerical calculation method

2. 4 Design procedure

A design procedure of the numerical calculation method is pictorially shown in Fig. 5 and the details of each block [I]~[IV] are as follows ;

[I] Analysis of the given equations

This is a stage to determine the design specifications of the method, referring to the characteristics of equations in Table 1 and to the demands for numerical solutions in Table 2.

[II] Determination of the approximate shift operator function form

This is a stage to determine the function form, following to the result of [I]. Table 3 is referred in order to investigate the characteristics of the function and Table 4 is referred to decide the function form.

[III] Determination of parameters of the function

This is a stage for adjusting the parameters of the function. This adjustment makes it possible (a) to expand the numerical stable border, (b) to improve the accuracy of the specified region of the sτ plane and (c) to make the numerical method have the specified filter characteristics. As the method of the parameter adjustment, we consider the least square method, the least square method with constraints and the min-max method.

Table 4 A table to decide the approximate z-operator for the numerical calculation

case	m. n	conditions
1	m=0 n≦3	For linear equations when their coefficient matrices are ruled sparse and have the eigen values around the origin of $s\tau$. For nonlinear equations, when it is necessary to obtain the first approximate solutions.
2	m=0 n≧4	Under the conditions of case 1, when it is necessary to obtain the solutions with the higher accuracy or with the toughness.
3	m=1 n=1	For the linear, middle scale or large scale with the ruled sparsity equations when it is necessary to obtain the not so accurate but with faster processing speed and numerical stability.
4	m≧2 n≧2	For the linear, middle scale or large scale with the ruled sparsity equations when it is necessary to obtain the more accurate solution stably.
Remarks		In this table, m and n denote the order of denominator and numerater of the approximate shift operator $\sum_{i=0}^{n} a_i(s\tau)^i / \sum_{j=0}^{m} b_j(s\tau)^j$ respectively. When m=0 this operator provide the explicit scheme and m≠0, implicit scheme.

[IV] Formation of the algorithm

This is a stage to construct the algorithm of numerical method for the selected shift operator function. Here, following items are taken into account. (a) adaptivity to the sparsity of the coefficient matrix, (b) adaptivity to the nonlinear equations, (c) processing efficiency.

To illustrate this procedure, we consider two examples, the large scale Lagrange equation of motion and the stiff nonlinear equations, which appear frequently in technological problems.

3. Design of a numerical calculation method for the large scale Lagrange equation of motion

3.1 Design specifications

Let's design the numerical calculation method for the following large scale Lagrange equation of motion

$$\mathbf{M}\ddot{\mathbf{x}}+\mathbf{C}\dot{\mathbf{x}}+\mathbf{K}\mathbf{x}=\mathbf{f} \tag{9}$$

where **M**, **C** and **K** are symmetrical, and large scale matrices of band structure and the equation is stiff.

Given below are the specifications proposed in this paper. (a) The accuracy is higher than 4th order Runge Kutta method. (b) The numerical stability is assured. (c) The algorithm must be adaptable to the sparsity of the matrices.

3.2 Design of the algorithm

In order to assure the numerical stability and the accuracy of higher than 4th order Runge Kutta method, we adopt 2nd order Padé's expansion of $e^{s\tau}$

$$F(s\tau)=\frac{1+s\tau/2+(s\tau)^2/12}{1-s\tau/2+(s\tau)^2/12}=\frac{(1+s\tau/c_1)(1+s\tau/c_2)}{(1-s\tau/c_1)(1-s\tau/c_2)} \tag{10}$$

as the approximate shift operator, where c_1 and c_2 are

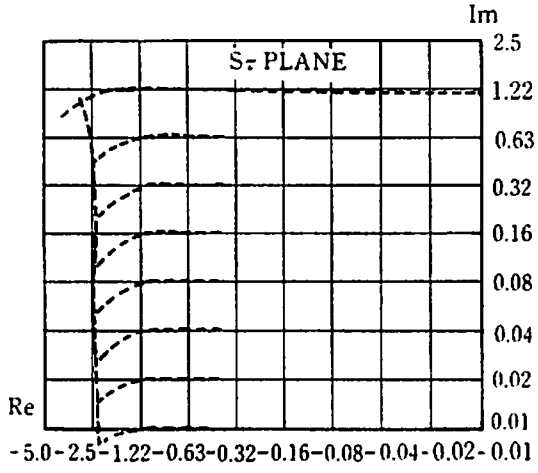

Fig. 6. Distortion characteristics of the sτ plane of the Padé-22 function

$$c_1=3+i\sqrt{3},\quad c_2=3-i\sqrt{3} \tag{11}$$

This function satisfies the following relation

$$\mathrm{Re}\{\ln(F(s\tau))\}\leq 0.0 \text{ for } \mathrm{Re}(s\tau)\leq 0.0 \tag{12}$$

which assures the numerical stability. Fig. 6 shows the distorted sτ plane by this function and it is clear that this distortion is less than that of 4th order Runge Kutta method in Fig. 3.

Then this function satisfies the design specification (a) and (b) without any adjustments of the parameters.

So that the design procedures [I], [II] and [III] are omitted. The remaining problem is how to develop the algorithm, based on this function, which satisfies the specification (c).

(9) is rewritten in the alternate form.

$$\dot{\mathbf{y}}=\mathbf{A}\mathbf{y}+\mathbf{B}\mathbf{f} \tag{13}$$

where **A**, **B** and **y** are defined as follows;

$$\mathbf{A}=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C}\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}\end{bmatrix},\quad \mathbf{y}=\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} \tag{14}$$

substitution of the factorized function (10) to (4) with $l=1$ and substitution (14) to this resultant equation lead

$$\begin{bmatrix}\mathbf{I} & -\tau/c_1\mathbf{I}\\ \tau/c_1\mathbf{M}^{-1}\mathbf{K} & \mathbf{I}+\tau/c_1\mathbf{M}^{-1}\mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{v}\\ \dot{\mathbf{v}}\end{bmatrix}_{n+1}$$
$$\begin{bmatrix}\mathbf{I} & \tau/c_1\mathbf{I}\\ -\tau/c_1\mathbf{M}^{-1}\mathbf{K} & \mathbf{I}-\tau/c_1\mathbf{M}^{-1}\mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \dot{\mathbf{u}}\end{bmatrix}_n+\frac{\tau}{c_1}\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}\end{bmatrix}(\mathbf{f}_{n+1}+\mathbf{f}_n) \tag{15}$$

where $\mathbf{u}_n$, $\mathbf{v}_n$ are respectively

$$\begin{aligned}\mathbf{u}_n&=\mathbf{x}_n+\tau/c_2\dot{\mathbf{x}}_n\\ \mathbf{v}_n&=\mathbf{x}_n-\tau/c_2\dot{\mathbf{x}}_n\end{aligned} \tag{16}$$

By the forward multiplication of matrix $\begin{bmatrix}c_1/\tau\mathbf{M}+\mathbf{C} & \mathbf{M}\\ -\mathbf{K} & c_1/\tau\mathbf{M}\end{bmatrix}$ to

(15), (15) is written in the alternate form

$$\begin{bmatrix} \mathbf{R} & \mathbf{O} \\ \mathbf{O} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \dot{\mathbf{v}} \end{bmatrix}_{n+1} = \begin{bmatrix} \mathbf{S} & 2\mathbf{M} \\ -2\mathbf{K} & \mathbf{T} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_n + \begin{bmatrix} \tau/c_1\mathbf{I} \\ \mathbf{I} \end{bmatrix} (\mathbf{f}_{n+1}+\mathbf{f}_n) \tag{17}$$

where **R**, **S** and **T** are respectively

$$\begin{aligned} \mathbf{R} &= c_1/\tau\mathbf{M}+\mathbf{C}+\tau/c_1\mathbf{K} \\ \mathbf{S} &= c_1/\tau\mathbf{M}+\mathbf{C}-\tau/c_1\mathbf{K} \\ \mathbf{T} &= c_1/\tau\mathbf{M}-\mathbf{C}-\tau/c_1\mathbf{K} \end{aligned} \tag{18}$$

Elimination of $\ddot{\mathbf{x}}$ from (9) and (17) leads the following equations

$$\mathbf{R}\mathbf{w}_{n+1} = -\tau\mathbf{K}\mathbf{x}_n + c_1\mathbf{M}\dot{\mathbf{x}}_n + \tau/2(\mathbf{f}_{n+1}+\mathbf{f}_n) - c_1\tau/12(\mathbf{f}_{n+1}-\mathbf{f}_n) \tag{19}$$

$$\mathbf{w}_{n+1} = (\mathbf{x}_{n+1}-\mathbf{x}_n) - c_1\tau/12(\dot{\mathbf{x}}_{n+1}-\dot{\mathbf{x}}_n) \tag{20}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathrm{Re}(\mathbf{w}_{n+1}) - \sqrt{3}\,\mathrm{Im}(\mathbf{w}_{n+1}) \tag{21}$$

$$\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_n - 4\sqrt{3}/\tau\,\mathrm{Im}(\mathbf{w}_{n+1}) \tag{22}$$

$$\ddot{\mathbf{x}}_{n+1} = \ddot{\mathbf{x}}_n + 6/\tau(\dot{\mathbf{x}}_{n+1}+\dot{\mathbf{x}}_n) - 12/\tau^2(\mathbf{x}_{n+1}-\mathbf{x}_n) \tag{23}$$

The equations (19), (21), (22) and (23) form the following algorithm (a) compute $\mathbf{w}_{n+1}$ using the $\mathbf{x}_n$, $\dot{\mathbf{x}}_n$ ' and $\mathbf{f}_n$, $\mathbf{f}_{n+1}$ by (19), (b) compute the solutions $\dot{\mathbf{x}}_{n+1}$, $\mathbf{x}_{n+1}$ by (21) and (22), (c) compute the $\ddot{\mathbf{x}}_{n+1}$ by the use of the result of (b), and then replace n+1 by n and return to (a).

To obtain $\mathbf{w}_{n+1}$ by (19), it is necessary to invert matrix **R**. In this case, $\mathbf{LL}^T$ decomposition of matrix **R** can make use of the band structure of matrices **M**, **C** and **K**, which assures the calculation speed to be faster without wasting many memories of a computer. When **M**, **C** and **K** do not vary, the necessary $\mathbf{LL}^T$ decomposition is only once at the start.

Consequency the above algorithm for the large scale Lagrange equation of motion satisfies all of the specifications.

4. Design of the numerical calculation method for nonlinear stiff ordinary differential equations

4.1 Design specifications

We consider another design problem with following specifications.

(a) The method is adaptable to the nonlinear and large scale equations easily.

(b) The method must be tough numerically (the numerical stable region must be wide).

(c) The accuracy is higher than Euler method and well adaptable to nonoscillatory differential equations.

(d) The method can process with high speed.

Such methods are suitable for the following applications.

(I) The application to obtain the first approximate solutions.

(II) The application to solve the thermal equation and dynamic chemical process.

(III) As a method for the simulator of the continuous control system.

4.2 Determination of the function form

As the approximate shift operator, the function of case 2 in Table 4

$$F(s\tau)=\sum_{i=0}^{m} a_i(s\tau)^i \tag{24}$$

is desirable for the above specifications. To satisfy the specification (d), m in (24) must not be so greater and let m=4 as 4th order Runge Kutta method, and further to satisfy the specification (c), let

$$a_0=1, \quad a_1=1 \tag{25}$$

which leads that $F(s\tau)$ and its first derivative are equal to the corresponding ones of $e^{S\tau}$ on the origin of $s\tau$ plane.

4.3 Determination of the parameters

Let's determine the residual parameters $a_2 \sim a_4$ in (24) by the least square method with the following square function.

$$\begin{aligned} J=&\int_0^W\int_{-R}^0[\{X-e^X\cos(Y)\}^2+\{Y-e^X\sin(Y)\}^2]dxdy \\ &+\int_0^Q\int_{-P}^0[X^2+Y^2]dxdy-\int_0^W\int_{-R}^0[X^2+Y^2]dxdy \end{aligned} \tag{26}$$

where x, y are real and imaginary part of $s\tau$ and X, Y are real and imaginary part of function $F(s\tau)$ as follows;

$$\begin{aligned} X&=(1-a_2y^2+a_4y^4)+(1-3a_3y^2)x+(a_2-6a_4y^2)x^2+a_3x^3+a_4x^4 \\ Y&=(y-a_3y^3)+(2a_2y-4a_4y^3)x+3a_3yx^2+4a_4yx^3 \end{aligned} \tag{27}$$

The rectangular zone (O, −R, W) in Fig. 7 is the area for fitting $F(s\tau)$ to $e^{S\tau}$ and the hook shaped zone (−P, Q, W, −R) in Fig. 7 is for making $F(s\tau)$ vanishingly small by which the numerical stable region is expanded.

Letting the partial derivative of the square J by **a** be equal to zero, that is

$$\partial J/\partial \mathbf{a}=0 \tag{28}$$

the following equation is derived.

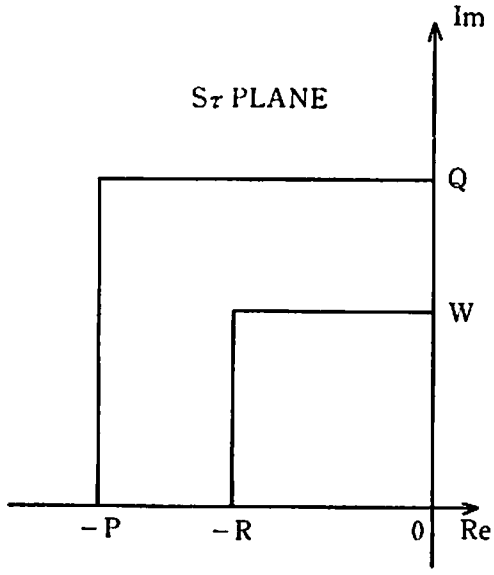

Fig. 7. Integral intervals of the criterion function of the least square method

$$\mathbf{Ha}=\mathbf{t} \tag{29}$$

where **a** and **t** are given by

$$\mathbf{a}=[a_2,\ a_3,\ a_4]^T,$$

$$\mathbf{t}=-[A_2+A_{12},\ A_3+A_{13},\ A_4+A_{14}]^T \tag{30}$$

and **H** is the Jacobian of $\partial J/\partial \mathbf{a}$ for **a** given as follows;

$$\mathbf{H}=\begin{bmatrix} 2A_{22} & A_{23} & A_{24} \\ A_{23} & 2A_{33} & A_{34} \\ A_{24} & A_{34} & 2A_{44} \end{bmatrix} \tag{31}$$

the elements A_{IJ} and A_K of **H** and **t** are the functions of the variables R, W, P and Q, as listed in Table 5.

4.4 Formation of the algorithm

The parameters of (24) are determined by

Table 5 Coefficients of the element of matrix H and vector t

$A_{11}=(Q^2+P^2)PQ/3$
$A_{12}=-(PQ^2/3+P^3/2)PQ$
$A_{13}=-2(-Q^4+P^4)PQ/5$
$A_{14}=(P^3Q^2/3+3PQ^4/5-P^5/3)PQ$
$A_{22}=(2P^2Q^2/9+P^4/5+Q^4/5)PQ$
$A_{23}=-(P^3Q^2/3+P^5/3+PQ^4/5)PQ$
$A_{24}=2(P^4Q^2/15-P^2Q^4/15+P^6/7-Q^6/7)PQ$
$A_{33}=-(P^4Q^2/5+P^2Q^4/5+Q^6/7+P^6/7)PQ$
$A_{34}=-(P^5Q^2/3+3P^3Q^4/10+PQ^6/7+P^7/4)PQ$
$A_{44}=(4P^6Q^2/21+4P^2Q^6/21+P^8/9+Q^8/9+6P^4Q^4/25)PQ$

$A_1=-P^2Q-2Re^{-R}\sin(W)-2(e^{-R}-1)W\cos(W)$
$A_2=2(P^2-Q^2)PQ/3+e^{-R}\{(2R^2+2W^2-4)\sin(W)+4WR\cos(W)\}$
$\quad+(W^3-6W)\cos(W)$
$A_3=-P^4Q/2+P^2Q^3-2[e^{-R}\{(3R^2W+6-W^3)\cos(W)+(R^3+3RW^2$
$\quad+6R)\sin(W)\}+(W^3-6W)\cos(W)]$
$A_4=2P^5Q/5-4P^3Q^3/3+2PQ^5/5-2[e^{-R}\{(-4R^3W-24RW+4RW^3)$
$\quad\cos(W)+(-R^4-12R^2+6R^2W^2-24-W^4+12W^2)\sin(W)\}$
$\quad+(-12W^2+W^4+24)\sin(W)]$

(29), and the next procedure is to construct the algorithm basing on (24). Let the algorithm form be as follows ;

$$\Delta_1=\tau f(x_n) \tag{32}$$

$$\Delta_i=\tau f(x_n+d_{i-1}\Delta_{i-1}) \quad (i=2\sim m)$$
$$x_{n+1}=x_n+c_1\Delta_1+c_2\Delta_2+\cdots+c_m\Delta_m \tag{33}$$

where f(x) is the function of the differential equation. This is the form of the Runge Kutta algorithm and can satisfy specification (a). The coefficients $c_i(i=1\sim m)$ and $d_k(k=1\sim m-1)$ of the algorithm are related to the coefficients $a_i(i=0\sim m)$ of (24).

Namely substitution of the equation

$$\dot{x}=f(x),\ f(x)=sx \tag{34}$$

to (32) leads

$$\begin{aligned}\Delta_1&=s\tau x_n\\ \Delta_2&=(s\tau+d_1s^2\tau^2)x_n\\ &\vdots\\ \Delta_m&=(s\tau+d_{m-1}s^2\tau^2+d_{m-2}d_{m-1}s^3\tau^3+\cdots+\prod_{i=1}^{m-1}d_is^m\tau^m)x_n\end{aligned} \tag{35}$$

And substitutions of the above $\Delta_i(i=1\sim m)$ to (33) lead the equation.

$$x_{n+1}=Gx_n \tag{33}'$$

where

$$G=1+(c_1+c_2+\cdots+c_m)s\tau+(d_1c_2+d_2c_3+\cdots+d_{m-1}c_m)s^2\tau^2 +\cdots+\prod_{i=1}^{m-1}d_ic_ms^m\tau^m \tag{36}$$

(33)′ is the difference equation with the shift operator (36). Letting this G be equal to $F(s\tau)$ of (24)

$$F(s\tau)=G \tag{37}$$

and letting each coefficient of $(s\tau)^i$ of (24) be equal to one of (36), the following relations are derived.

$$\begin{aligned} a_1 &= c_1 + c_2 + \cdots + c_m \\ a_2 &= d_1c_2 + d_2c_3 + \cdots + d_{m-1}c_m \\ &\vdots \\ a_m &= \prod_{i=1}^{m-1} d_ic_m \end{aligned} \tag{38}$$

(38) has m equations with (2m−1) unknown variables, and does not have unique solution, so that it is possible to introduce the additional criterion (e. g. the minimization of the discretization error) and to formulate this problem as an optimization problem with the constraints (38). But, in this paper, parameters d_k (k=1∼m) are simply to be equal to the parameters of 4th order Runge Kutta method.

4. 5 Design example

The order m of (24) was equal to 4, and the parameters must be determined so as to lessen the distortion of $s\tau$ plane and expand the stable region around it real axis. Therefore the integral intervals of (26) are set as follows.

$$P=11.0,\ Q=2.0,\ R=5.0,\ W=1.0 \tag{39}$$

The resultant coefficients a_i (i=2∼4) are listed in a_i column of Table 6. The parameters d_k (k=1∼3) are

$$d_1=0.5,\ d_2=0.5,\ d_3=1.0 \tag{40}$$

as discribed above. Form these a_i, d_k, the parameters c_i (i=1∼4) are determined uniquely by (38) as listed in c_i column of Table 6.

Fig. 8 shows the distortion characteristics of $s\tau$ plane and the stable border of the designed method.

Comparison between Fig. 2 (Euler method) and Fig. 8 indicates that the distortions at −0. 02 of the real axis and 0. 03 of the imaginary axis on Fig. 2 appear −0. 1 of the real axis and 0. 06 of the imaginary axis of Fig. 8 respectively. Then the designed method has higher accuracy than Euler method. The stable region of the designed method on real axis is 4. 4 times wider than one of 4th order Runge Kutta method.

Consequently the designed method satisfies all of the specifications.

4. 6 Example of the numerical calculation by the designed method

The dynamic behavior of a chemical plant in Fig. 9 is calculated by 4th order Runge Kutta method and the designed method. Fig. 10 shows these solutions. The curve shown by the solid line is the solution by 4th order Runge Kutta method under

Table 6 Coefficient of the designed numerical method

a_i	c_j
$a_1=1.000000$	$c_1=0.402794$
$a_2=0.301403$	$c_2=0.462322$
$a_3=0.351212$	$c_3=0.129284$
$a_4=0.140000$	$c_4=0.005600$

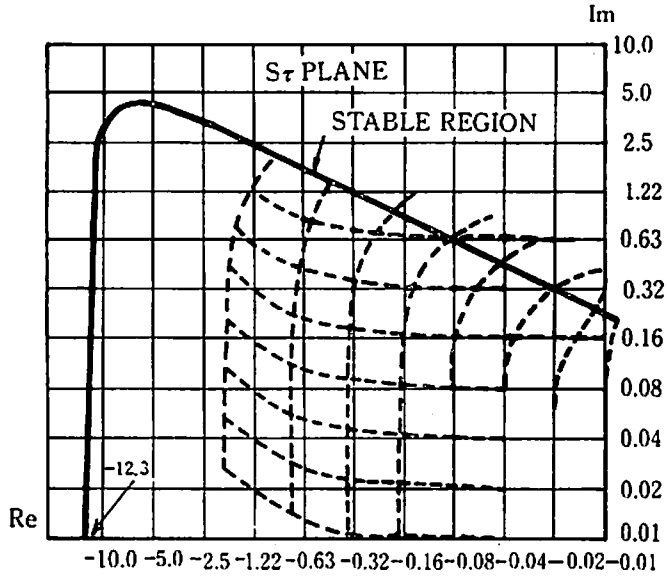

Fig. 8. Distortion characteristics of the sτ plane and stable region of the designed method

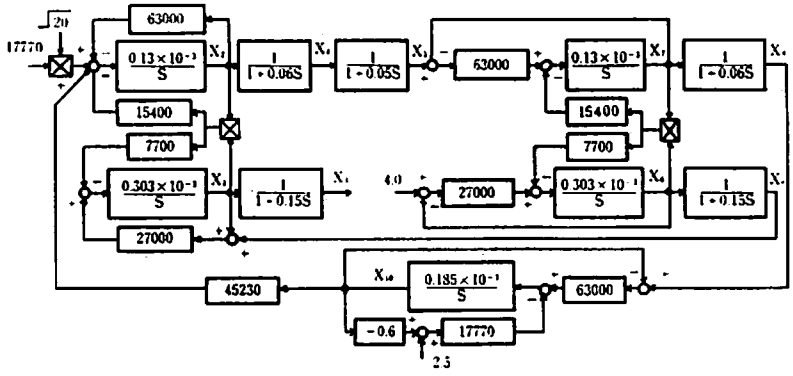

Fig. 9. Block diagram of the nonlinear process

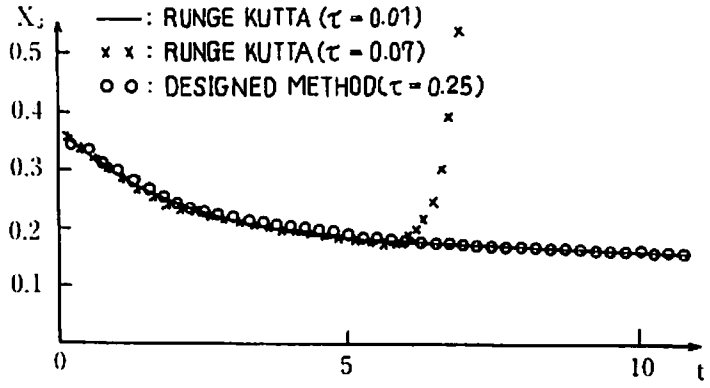

Fig. 10. Numerical solutions solved by the Runge Kutta method and the designed method

the stepwidth $\tau=0.01$ and this curve can be regarded as theoretical value.

The numerical unstable phenomenon happens, when solved by 4th order Runge Kutta method under $\tau=0.07$ as shown by symbol $\times$. The designed method can solve it stably when $\tau\leq 0.265$, and the solution under $\tau=0.25$ is shown by symbol $\bigcirc$. This is very near to the curve given by the solid line and indicates that the processing efficiency is almost 4 times higher than 4th order Runge Kutta method when applyed to the equation of this type.

5. Conclusion

This paper presented a design procedure of the single-step numerical calculation method for ordinary differential equations and two design examples.

The numerical methods designed by this approach, can adapt the characteristics of the given equation and can improve the numerical efficiency. These numerical mathods are designed by simple procedure and are well adaptable to the equstions with special features and to the digital simulator of a industorial process, which uses the numerical calculation frequently.

In this design procedure, the designed methods are restricted to the single-step method and the discretization errors are not always take into consideration. The future theme in this problem is to establish the design procedure, which includes multi-step methods and thinks over the discretization errors.

6. Acknowledgements

The authors would like to express thier thanks to Prof. K. Hasegawa of Tokyo Institute of Technology and Dr. N. Shimizu of Chiyoda Chemical Construction Co., Ltd. for their helpful comments and suggestions.

References

1. C. W. Gear. (1971). Numerical Initial Value Problems in Ordinary Differential Equations, New Jersey : Prentice-Hall.
2. G. Dahlguist. (1956). Convergence and Stability in the Numerical Integration of Ordinary Differential Equations, pp. 33~53, Math. Scand. 4.
3. R. E. Nickell. (1972). A Survey of Direct Integration Methods in Structual Dynamics, Brown Univ.
4. W. Liniger. (1969). Optimization of a Numerical Integration Method for stiff Systems of Ordinarv Differential Equations, IBM Research, rc 2198.
5. Trujillo, D. M. (1975). Int. J. Numr. Methods Engng, pp. 259.
6. P. Henrici. (1968). Discrete Variable Methods in Ordinary Differential Equations, New York : John Wiley & Sons, Inc.
7. P. Henrici. (1963). Error Propagation for Difference Methods, New York : John Wiley & sons, Inc.
8. Shimizu, N and Watanabe, K. (1977). Transactions of the Japan society of mechanical engineers, Vol. 43, No. 368, pp. 1272~1289. (in Japanese)
9. Newmark, N. M. (1959). Proc. Amer. Soc. Civil Engr., EM-3, pp. 67.
10. Wilson, E. L. (1973). Earthquake Eng'ng and Structual Dynamics, pp. 241.