Discrete Integrable Systems and Convergence Acceleration Methods

（離散可積分系と加速法）

永　井　　敦

①

THE UNIVERSITY OF TOKYO

1995 Doctoral Thesis

Discrete Integrable Systems and Convergence Acceleration Methods

(離散可積分系と加速法)

Atsushi Nagai

(永井 敦)

Contents

Chapter 1

Introduction

1.1 History of soliton equations

One hundred years ago, Korteweg and de Vries proposed the equation[1],

$$u_t + 6uu_x + u_{xxx} = 0 \tag{1.1}$$

as a model system for shallow water wave. Equation (1.1) is now called the *KdV equation* and is one of the most fundamental and important equations in the soliton theory. It was no less than seventy years later in 1965 when Zabusky and Kruskal found numerical solutions with periodic boundary condition. The solution, which they named a *soliton solution*, represents a stable solitary wave and retains its identity after it interacts with other solitons. In 1967, Gardner, Greene, Kruskal, and Miura discovered an epoch-making method for solving eq. (1.1) under a boundary condition $u \to 0$ as $|x| \to \infty$, which is now called the *inverse scattering method.*

After the discovery of the inverse scattering method, this successful method has been applied to many other nonlinear evolution equations and exact solutions for

[1]Equation (1.1) is written in a scaled form, which we now adopt.

them have been found. Typical examples of such nonlinear equations are the nonlinear Schrödinger equation,

$$iu_t + u_{xx} + 2u|u|^2 = 0, \tag{1.2}$$

the Sine-Gordon equation,

$$u_{xx} - u_{tt} = \sin u, \tag{1.3}$$

and the Toda equation,

$$\frac{\mathrm{d}^2 Q_n}{\mathrm{d}t^2} = -\exp(Q_n - Q_{n+1}) + \exp(Q_{n-1} - Q_n), \ n \in \mathbb{Z}. \tag{1.4}$$

They are first proposed as model equations for optical soliton, dislocation in crystals, and nonlinear lattices, respectively. Though they originate with different physical background, all of them are called *integrable*[2] in the sense that

- They admit an exact solution called the N soliton solution.
- They have an infinite number of conserved quantities and symmetries.
- They possess the Lax pair and their initial value problem can be exactly solved by the inverse scattering method.
- They satisfy the Painlevé property.

Soliton equations, including the above fundamental equations, have been actively studied from various aspects such as mathematics, physics, and engineering. In early 80's, Sato [29] discovered that the solutions for these equations are related with infinite dimensional Grassmann manifold and that there are linearized equations hidden behind the nonlinear integrable equations.

[2]It is still difficult to give a precise definition of the term *integrable*.

1.2 Integrable discretization

In this section, we review integrable discretization of soliton equations. Development of computers has enabled us to solve numerically nonlinear differential equations which do not admit analytical solutions. It should be noted, however, that simple discretization of given continuous equations does not necessarily preserve the original properties, or sometimes brings about numerically induced chaos.

As a simple but famous example, let us consider the logistic equation,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = au(1-u). \tag{1.5}$$

The above equation has an exact solution,

$$u(t) = \frac{Ae^{at}}{1 + Ae^{at}}, \tag{1.6}$$

where A is an arbitrary parameter determined by the initial condition. As one possible discretization of eq. (1.5), we may consider the following difference equation,

$$\frac{u_{n+1} - u_n}{\varepsilon} = au_n(1 - u_n), \tag{1.7}$$

where ε stands for a lattice parameter. If a exceeds a certain critical value, eq. (1.7) shows a chaotic behavior and does not conserve the structure of solution for eq. (1.5). On the other hand, another discretization of eq. (1.5),

$$\frac{u_{n+1} - u_n}{\varepsilon} = au_n(1 - u_{n+1}), \tag{1.8}$$

does not show chaotic behavior and admits an exact solution,

$$u_n = \frac{A(1 + a\varepsilon)^n}{1 + A(1 + a\varepsilon)^n} \tag{1.9}$$

for any $\varepsilon > 0$. The solution (1.9) reduces to eq. (1.6) if we take the continuous limit $n \to \infty$, $\varepsilon \to 0$ with $t = n\varepsilon$ kept finite. We call the discretized eq. (1.8) the *integrable discretization* of eq. (1.5).

Integrable discretization of soliton equations was first proposed by Hirota [18] in 1977, in which he presents an integrable discretization of the KdV equation given by

$$X(n+1,m) - X(n-1,m+1) = \frac{1}{X(n,m+1)} - \frac{1}{X(n,m)}. \qquad (1.10)$$

Since then, discrete analogues of many other soliton equations, including eqs. (1.2)–(1.4), have been constructed.

1.3 Discrete soliton equations and convergence acceleration methods

Recently, it has been revealed that discrete soliton equations appear in rather unexpected area, the numerical analysis. In 1982, Symes [35] pointed out that the finite, nonperiodic Toda equation, which we call the Toda molecule equation in this context, relates with the QR algorithm to calculate eigenvalues of a given matrix. Since then relations between soliton equations and matrix eigenvalue algorithms have been actively studied [8, 11, 17, 25, 24]. In 1993, Papageorgiou, Grammaticos, and Ramani [27] showed that one of the well-known convergence acceleration schemes, the ε−algorithm, is nothing but the discrete potential KdV equation.

Our main interest in this thesis is on the convergence acceleration methods. We first present a simple introduction of convergence acceleration methods. Let $\{S_m\}$ be a sequence of numbers which converges to S_∞. In order to find an approximate value of S_∞ by direct calculation, we often need a large amount of data. Sequences

$$S_m = 1 - \frac{1}{2} + \frac{1}{3} - \cdots + \frac{(-1)^m}{m+1}, \qquad (1.11)$$

$$S_m = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{(m+1)^2} \qquad (1.12)$$

are typical examples. Besides these simple cases, one has often to deal with slowly convergent sequences in the field of applied mathematics. In such cases we transform the original sequence $\{S_m\}$ into another sequence $\{T_m\}$ instead of calculating directly. If $\{T_m\}$ converges to S_∞ faster than $\{S_m\}$, that is

$$\lim_{m\to\infty} \frac{T_m - S_\infty}{S_{\sigma(m)} - S_\infty} = 0, \tag{1.13}$$

we say that the transformation $T : \{S_m\} \to \{T_m\}$ *accelerates the convergence* of the sequence $\{S_m\}$. In eq. (1.13), $\sigma(m)$ is a function in m such that sequence $\{T_i\}_{i=0}^{m}$ is determined by $\{S_i\}_{i=0}^{\sigma(m)}$. In the case of the Aitken acceleration, for instance, $\sigma(m)$ is given by $m+2$.

As proved by Delahaye and Germain-Bonne [12], a universal transformation T accelerating all the converging sequences cannot exist. This negative result, however, means that it will be always interesting to find and to study new sequence transformations since each of them is only able to accelerate the convergence of certain classes of sequences. In fact, we now have many convergence acceleration methods such as $\varepsilon-$algorithm [40], $\eta-$algorithm [3], $\rho-$algorithm [41], BS-algorithm [7], $\theta-$algorithm [4], Levin's $t-$, $u-$, and $v-$transformation [20], and $E-$algorithm [5]. We focus our attention mainly on the $\varepsilon-$, $\eta-$, and $\rho-$algorithms. We show there being a strong tie between these algorithms and discrete soliton equations.

1.4 Outline of the thesis

The thesis is organized as follows. In chapter 2, we discuss the $\eta-$algorithm,

$$\left\{\begin{array}{rcl} \eta_{2n+1}^{(m)} + \eta_{2n}^{(m)} & = & \eta_{2n}^{(m+1)} + \eta_{2n-1}^{(m+1)} \\ \dfrac{1}{\eta_{2n+2}^{(m)}} + \dfrac{1}{\eta_{2n+1}^{(m)}} & = & \dfrac{1}{\eta_{2n+1}^{(m+1)}} + \dfrac{1}{\eta_{2n}^{(m+1)}} \end{array}\right. , \tag{1.14}$$

and ε−algorithm,

$$(\varepsilon_{n+1}^{(m)} - \varepsilon_{n-1}^{(m+1)})(\varepsilon_n^{(m+1)} - \varepsilon_n^{(m)}) = 1. \tag{1.15}$$

which are equivalent to the discrete KdV and the discrete potential KdV equation, respectively. The continuous limit from the discrete KdV eq. (2.7) to the KdV eq. (1.1) is also considered. We mention the performance of these algorithms as convergence accelerators and the reasons for which these algorithms accelerate (or fail to accelerate) the convergence of given sequences. These algorithms, when viewed as difference equations, possess solutions expressed as ratios of Hankel determinants, which appear in the field of the Padé approximation. We also relate the asymptotic behavior of solutions for the Toda molecule equation with convergence acceleration by means of continued fractions and give one reason why the ε−(or η−)algorithm appears in the field of soliton theory.

In chapter 3, we introduce a different type of convergence acceleration algorithm, the ρ−algorithm,

$$(\rho_{n+1}^{(m)} - \rho_{n-1}^{(m+1)})(\rho_n^{(m+1)} - \rho_n^{(m)}) = n. \tag{1.16}$$

In spite of its apparent similarity with the ε−algorithm (1.15), it possesses noticeably different characteristics not only as a convergence accelerator but also as a discrete soliton equation. We show that one generalized version of the ρ−algorithm is considered as an integrable discretization of the cylindrical KdV equation [21],

$$u_t + 6uu_x + u_{xxx} + \frac{u}{2t} = 0. \tag{1.17}$$

The ρ−algorithm, when viewed as a difference equation, admits a double Casorati determinant solution. We explain that this fact is quite natural if we discuss the algorithm in relation with Thiele's rational interpolation formula [36]. By changing elements of the double Casorati determinants, the ρ−algorithm is naturally extended to the algorithm of the following form;

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = \sigma(n+m) - \sigma(m). \qquad (1.18)$$

We show that the above extended version of $\rho-$algorithm accelerates larger class of sequences than the original $\rho-$algorithm (1.16).

Chapter 4 is mainly devoted to one of quite generalized rhombus algorithms, the PGR algorithm. The algorithm, when viewed as a difference equation, satisfies the so-called singularity confinement condition. We discuss its relation with discrete integrable equations and the performance as a convergence accelerator.

Finally in chapter 5, we give concluding remarks.

Chapter 2

The $\eta-$ and $\varepsilon-$algorithms

2.1 The $\eta-$algorithm and the discrete KdV eqution

In this section we show that Bauer's $\eta-$algorithm [3], which is one of the famous convergence acceleration algorithms, is equivalent to the discrete KdV equation. The $\eta-$algorithm involves a two-dimensional array called the $\eta-$table (Figure 2.1). The table is constructed from its first two columns. Let initial values $\eta_0^{(m)}$ and $\eta_1^{(m)}$ be

$$\eta_0^{(m)} = \infty, \ \eta_1^{(m)} = c_m \equiv \Delta S_{m-1}, \ (m = 0, 1, 2, \ldots), \ S_{-1} \equiv 0, \tag{2.1}$$

where Δ is the forward difference operator given by $\Delta a_k = a_{k+1} - a_k$. Then all the other elements are calculated from the following recurrence relations called the $\eta-$algorithm;

$$\left\{ \begin{array}{rcl} \eta_{2n+1}^{(m)} + \eta_{2n}^{(m)} &=& \eta_{2n}^{(m+1)} + \eta_{2n-1}^{(m+1)} \\ \dfrac{1}{\eta_{2n+2}^{(m)}} + \dfrac{1}{\eta_{2n+1}^{(m)}} &=& \dfrac{1}{\eta_{2n+1}^{(m+1)}} + \dfrac{1}{\eta_{2n}^{(m+1)}} \end{array} \right. \quad \text{(rhombus rules)}. \tag{2.2}$$

Equation (2.2) defines a transformation of a given series $c_m = \eta_1^{(m)}, m = 0, 1, 2, \ldots$ to a new series $c'_n = \eta_n^{(0)}, n = 1, 2, \ldots$ such that $\sum_{n=1}^{\infty} c'_n$ converges more rapidly to the same limit S_∞.

$$\begin{array}{ccccccc}
 & \eta_1^{(0)} & & & & & \\
(\infty =)\eta_0^{(1)} & & \eta_2^{(0)} & & & & \\
 & \eta_1^{(1)} & & \eta_3^{(0)} & & & \\
(\infty =)\eta_0^{(2)} & & \eta_2^{(1)} & & \eta_4^{(0)} & & \\
 & \eta_1^{(2)} & & \eta_3^{(1)} & & \eta_5^{(0)} & \\
(\infty =)\eta_0^{(3)} & & \eta_2^{(2)} & & \eta_4^{(1)} & \vdots & \ddots \\
 & \eta_1^{(3)} & & \eta_3^{(2)} & \vdots & & \\
(\infty =)\eta_0^{(4)} & & \eta_2^{(3)} & \vdots & & & \\
\vdots & \eta_1^{(4)} & \vdots & & & & \\
 & \vdots & & & & &
\end{array}$$

Figure 2.1: The $\eta-$table.

As an empirical example we consider a slowly convergent series for $\log 2$,

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \cdots \to \log 2 = 0.693147\cdots \tag{2.3}$$

and construct the $\eta-$table for the series (2.3). We see from Figure 2.2 that the transformed series

$$1 - \frac{1}{3} + \frac{1}{30} - \frac{1}{130} + \frac{1}{975} - \frac{1}{4725} + \cdots$$

converges more rapidly to $\log 2$ than the original series. While the sum of the first seven terms of the original series gives $0.7595\cdots$, that of the corresponding seven terms of the transformed series does $0.693152\cdots$.

	1						
∞		$-1/3$					
	$-1/2$		$1/30$				
∞		$1/5$		$-1/130$			
	$1/3$		$-1/105$		$1/975$		
∞		$-1/7$		$1/350$		$-1/4725$	
	$-1/4$		$1/252$		$-1/4100$		$1/32508$
∞		$1/9$		$-1/738$		$1/15867$	
	$1/5$		$-1/495$		$1/12505$		
∞		$-1/11$		$1/1342$			
	$-1/6$		$1/858$				
∞		$1/13$					
	$1/7$						

Figure 2.2: The $\eta-$table for $\log 2$

The quantities $\eta_n^{(m)}$ are given by the following ratios of Hankel determinants;

$$\eta_{2n+1}^{(m)} = \frac{\begin{vmatrix} c_m & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} c_{m+1} & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n-1} \end{vmatrix}}{\begin{vmatrix} \Delta c_m & \cdots & \Delta c_{m+n-1} \\ \vdots & & \vdots \\ \Delta c_{m+n-1} & \cdots & \Delta c_{m+2n-2} \end{vmatrix} \cdot \begin{vmatrix} \Delta c_{m+1} & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n-1} \end{vmatrix}}, \qquad (2.4)$$

$$\eta_{2n+2}^{(m)} = \frac{\begin{vmatrix} c_m & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} c_{m+1} & \cdots & c_{m+n+1} \\ \vdots & & \vdots \\ c_{m+n+1} & \cdots & c_{m+2n+1} \end{vmatrix}}{\begin{vmatrix} \Delta c_m & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} \Delta c_{m+1} & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n-1} \end{vmatrix}}. \tag{2.5}$$

We next show the equivalence between the $\eta-$algorithm and the discrete KdV equation. If we introduce dependent variable transformations,

$$X_{2n}^{(m)} = \frac{1}{\eta_{2n}^{(m)}}, \; X_{2n-1}^{(m)} = \eta_{2n-1}^{(m)} \tag{2.6}$$

the $\eta-$algorithm (2.2) is rewritten as

$$X_{n+1}^{(m)} - X_{n-1}^{(m+1)} = \frac{1}{X_n^{(m+1)}} - \frac{1}{X_n^{(m)}}, \tag{2.7}$$

which is the discrete KdV equation.

Let us derive the KdV eq. (1.1) from the discrete KdV eq. (2.7). We replace variables n and m by

$$n = \frac{t}{\epsilon^3}, \; m - \frac{1}{2} = \frac{x}{\epsilon} - \frac{ct}{\epsilon^3}, \tag{2.8}$$

respectively and rewrite $X_n^{(m)}$ as $p + \epsilon^2 u(x - \epsilon/2, t)$, where ϵ is a small parameter and c, p are finite constants related by

$$1 - 2c = \frac{1}{p^2}. \tag{2.9}$$

Then eq. (2.7) becomes

$$\epsilon^2 u(x - \frac{\epsilon}{2} + c\epsilon, t + \epsilon^3) - \epsilon^2 u(x + \frac{\epsilon}{2} - c\epsilon, t - \epsilon^3) = \frac{1}{p + \epsilon^2 u(x + \frac{\epsilon}{2}, t)} - \frac{1}{p + \epsilon^2 u(x - \frac{\epsilon}{2}, t)}. \tag{2.10}$$

If we take the small limit of ϵ, eq. (2.10) yields the KdV equation,

$$u_t - \frac{1}{p^3}uu_x + \frac{1}{48p^2}(1-\frac{1}{p^4})u_{xxx} = 0 \tag{2.11}$$

at the order of ε^5.

The discrete KdV eq. (2.7) is rewritten as

$$\begin{aligned}&\{\tau(n+2,m-1)\tau(n-1,m)+\tau(n+1,m)\tau(n,m-1)\}\tau(n-1,m+1)\\ &=\{\tau(n-2,m+1)\tau(n+1,m)+\tau(n-1,m)\tau(n,m+1)\}\tau(n+1,m-1),\end{aligned} \tag{2.12}$$

through a dependent variable transformation,

$$X_n^{(m)} = \frac{\tau(n+1,m-1)\tau(n-1,m)}{\tau(n,m-1)\tau(n,m)}. \tag{2.13}$$

Subtracting $\tau(n-1,m+1)\tau(n+1,m-1)\tau(n,m)$ from both sides of eq. (2.12), we obtain the bilinear form of the discrete KdV equation,

$$\tau(n+2,m-1)\tau(n-1,m)+\tau(n+1,m)\tau(n,m-1)-\tau(n+1,m-1)\tau(n,m)=0. \tag{2.14}$$

It is noted that the solutions (2.4) and (2.5) are recovered by putting

$$\tau(2n,m) \equiv \begin{vmatrix} c_{m+1} & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n-1} \end{vmatrix}, \tag{2.15}$$

$$\tau(2n+1,m) \equiv \begin{vmatrix} \Delta c_{m+1} & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n-1} \end{vmatrix} \tag{2.16}$$

in eq. (2.13). Substitution of eqs. (2.15) and (2.16) into the bilinear eq. (2.14) gives Jacobi's (or Sylvester's) identity for determinants.

2.2 The $\varepsilon-$algorithm and the discrete potential KdV equation

In this section, following the result by Papageorgiou et al., we briefly review the equivalence between the $\varepsilon-$algorithm and the discrete potential KdV equation. The $\varepsilon-$algorithm originates with Shanks [32] and Wynn [40]. It also involves two-dimensional array called the $\varepsilon-$table (Figure 2.3).

Figure 2.3: The $\varepsilon-$table

$$\begin{array}{ccccccc}
 & \varepsilon_1^{(0)} & & & & & \\
(0=)\varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & & & & \\
 & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & & & \\
(0=)\varepsilon_0^{(2)} & & \varepsilon_2^{(1)} & & \varepsilon_4^{(0)} & & \\
 & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & & \varepsilon_5^{(0)} & \\
(0=)\varepsilon_0^{(3)} & & \varepsilon_2^{(2)} & & \varepsilon_4^{(1)} & \vdots & \ddots \\
 & \varepsilon_1^{(3)} & & \varepsilon_3^{(2)} & \vdots & & \\
(0=)\varepsilon_0^{(4)} & & \varepsilon_2^{(3)} & \vdots & & & \\
\vdots & \varepsilon_1^{(4)} & \vdots & & & & \\
 & \vdots & & & & &
\end{array}$$

Define $\varepsilon_0^{(m)}$ and $\varepsilon_1^{(m)}$ by

$$\varepsilon_0^{(m)} = 0,\ \varepsilon_1^{(m)} = S_m\ (m = 0, 1, 2, \ldots). \tag{2.17}$$

Then all the other quantities obey the following rhombus rule called the $\varepsilon-$algorithm;

$$(\varepsilon_{n+1}^{(m)} - \varepsilon_{n-1}^{(m+1)})(\varepsilon_n^{(m+1)} - \varepsilon_n^{(m)}) = 1. \tag{2.18}$$

According as n becomes large, $\varepsilon_{2n+1}^{(m)}$ converges more rapidly to S_∞ as $m \to \infty$. On the other hand, $\varepsilon_{2n}^{(m)}$ diverges as $m \to \infty$. This fact reminds us of the idea of the singularity confinement [15], since a singularity at $(2n, m)$ is confined and $\varepsilon_{2n+1}^{(m)}$ converges to the same limit as the original sequence $\varepsilon_1^{(m)}$. As previously stated, it has been shown in 1993 by Papageorgiou et. al that the $\varepsilon-$algorithm (2.18) is regarded as the discrete potential KdV equation. It should be noted, however, that Arai, Okamoto, and Kametaka [1] had already indicated the relation between the $\varepsilon-$algorithm (or Aitken accelaration method) and the soliton equation seven years before the result by Papageorgiou et. al.

The quantities $\varepsilon_n^{(m)}$ are also given by the following ratios of Hankel determinants;

$$\varepsilon_{2n+1}^{(m)} = \frac{\begin{vmatrix} S_m & S_{m+1} & \cdots & S_{m+n} \\ S_{m+1} & S_{m+2} & \cdots & S_{m+n+1} \\ \vdots & \vdots & & \vdots \\ S_{m+n} & S_{m+n+1} & \cdots & S_{m+2n} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_m & \Delta^2 S_{m+1} & \cdots & \Delta^2 S_{m+n-1} \\ \Delta^2 S_{m+1} & \Delta^2 S_{m+2} & \cdots & \Delta^2 S_{m+n} \\ \vdots & \vdots & & \vdots \\ \Delta^2 S_{m+n-1} & \Delta^2 S_{m+n} & \cdots & \Delta^2 S_{m+2n-2} \end{vmatrix}}. \tag{2.19}$$

$$\varepsilon_{2n+2}^{(m)} = \frac{\begin{vmatrix} \Delta^3 S_m & \Delta^3 S_{m+1} & \cdots & \Delta^3 S_{m+n-1} \\ \Delta^3 S_{m+1} & \Delta^3 S_{m+2} & \cdots & \Delta^3 S_{m+n} \\ \vdots & \vdots & & \vdots \\ \Delta^3 S_{m+n-1} & \Delta^3 S_{m+n} & \cdots & \Delta^3 S_{m+2n-2} \end{vmatrix}}{\begin{vmatrix} \Delta S_m & \Delta S_{m+1} & \cdots & \Delta S_{m+n} \\ \Delta S_{m+1} & \Delta S_{m+2} & \cdots & \Delta S_{m+n+1} \\ \vdots & \vdots & & \vdots \\ \Delta S_{m+n} & \Delta S_{m+n+1} & \cdots & \Delta S_{m+2n} \end{vmatrix}}. \tag{2.20}$$

Equation (2.19) is called the Shanks transformation [32]. Substitution of $n = 1$ in eq. (2.19) gives the well-known Aitken acceleration algorithm.

We have so far seen that the $\eta-$ and the $\varepsilon-$algorithms are interpreted as the discrete KdV and the discrete potential KdV equations, respectively. Therefore, these two algorithms are the same in their performance as convergence acceleration algorithms. This equivalence can also be understood from the fact [3] that the quantities $\eta_n^{(m)}$ and $\varepsilon_n^{(m)}$ are related by

$$\eta_{2n}^{(m)} = \varepsilon_{2n+1}^{(m-1)} - \varepsilon_{2n-1}^{(m)}, \tag{2.21}$$

$$\eta_{2n+1}^{(m)} = \varepsilon_{2n+1}^{(m)} - \varepsilon_{2n+1}^{(m-1)}. \tag{2.22}$$

Let us apply the $\varepsilon-$algorithm to the following sequences[1];

$$S_m = 2^m \sin\left(\frac{\pi}{2^m}\right) \to \pi = 3.1415926535897\cdots, \tag{2.23}$$

$$S_m = \sum_{k=1}^{m} \frac{(-1)^{k-1}}{\sqrt{k}} \to (1-\sqrt{2})\zeta(0.5) = 0.604898643421\cdots. \tag{2.24}$$

Numerical results are given in Tables 2.1 and 2.2.

[1]It is interesting to remark that Japanese mathematician Takakazu Seki considered the quantities $\varepsilon_3^{(m)}$, namely the Aitken acceleration method, in calculating π and obtained its numerical value correctly to the 16th degit.

Table 2.1: The $\varepsilon-$algorithm for the sequence $S_m = 2^m \sin(\pi/2^m)$

m	$\varepsilon_1^{(m)}$	$\varepsilon_3^{(m)}$	$\varepsilon_5^{(m)}$	$\varepsilon_7^{(m)}$
1	2.000000	3.152682	3.141590268	3.141592653609
2	2.828427	3.142231	3.141592617	3.141592653589
3	3.061467	3.141632	3.141592653	3.141592653589
4	3.121445	3.141594	3.141592658	
5	3.136548	3.14159280	3.141592653589	
6	3.140331	3.14159266		
7	3.141277	3.141592654		
8	3.141514			
9	3.141573			

Table 2.2: The $\varepsilon-$algorithm for the sequence $S_m = \sum_{k=1}^{m}(-1)^{k-1}/k^{1/2}$

m	$\varepsilon_1^{(m)}$	$\varepsilon_3^{(m)}$	$\varepsilon_5^{(m)}$	$\varepsilon_7^{(m)}$	$\varepsilon_9^{(m)}$
1	1.000000	0.610730	0.605044	0.604903	0.604898755
2	0.292893	0.602294	0.604850	0.604897	0.604898614
3	0.870243	0.606311	0.604919	0.6048994	0.604898652
4	0.370243	0.604035	0.604889	0.6048984	0.604898640
5	0.817457	0.605470	0.604904	0.6048987	
6	0.409209	0.604497	0.604896	0.6048986	
7	0.787173	0.605193	0.604900		
8	0.433620	0.604676	0.604897		
9	0.766953	0.605073			
10	0.450725	0.604760			
11	0.752237				
12	0.463562				

As far as we see from these tables, the $\varepsilon-$algorithm seems like a very powerful acceleration tool. The algorithm, however, fails to accelerate rationally decaying sequences such as[2]

$$S_m = \sum_{k=1}^{m} \frac{1}{k^2} \to \frac{\pi^2}{6} = 1.644934066848\cdots. \tag{2.25}$$

[2]See Table 2.3.

Table 2.3: The $\varepsilon-$algorithm for the sequence $S_m = \sum_{k=1}^{m} 1/k^2$

m	$\varepsilon_1^{(m)}$	$\varepsilon_3^{(m)}$	$\varepsilon_5^{(m)}$	$\varepsilon_7^{(m)}$	$\varepsilon_9^{(m)}$
1	1.000000	1.450000	1.551617	1.590305	1.609087
2	1.250000	1.503968	1.571767	1.599984	1.614474
3	1.361111	1.534722	1.584826	1.606777	1.618468
4	1.423611	1.554520	1.593954	1.611799	1.621542
5	1.464611	1.568312	1.600687	1.615658	
6	1.491389	1.578464	1.605854	1.618716	
7	1.511797	1.586246	1.609943		
8	1.527422	1.592399	1.613260		
9	1.539768	1.597387			
10	1.549768	1.601510			
11	1.558032				
12	1.564977				

2.3 The Richardson and the Aitken acceleration methods

We see in the previous section that the $\varepsilon-$algorithm accelerates exponentially and alternatively decaying sequences but cannot do rationally decaying sequences. In order to give its intuitive account, we review in this section the Richardson and the Aitken acceleration methods, on which the $\varepsilon-$algorithm is based. Let $\{S_m\}$ be a sequence whose asymptotic behavior is given by

$$S_m \sim S_\infty + c_1\lambda_1^m + c_2\lambda_2^m + \cdots, \tag{2.26}$$

where c_i's are constants and λ_i's satisfy

$$1 > |\lambda_1| > |\lambda_2| > |\lambda_3| > \cdots. \tag{2.27}$$

We first consider the case that each λ_i is known. Let us define a new sequence $\{T_m^{(1)}\}$ by

$$T_m^{(1)} = \frac{S_{m+1} - \lambda_1 S_m}{1-\lambda_1}. \tag{2.28}$$

Then the sequence $\{T_m^{(1)}\}$ converges to S_∞ in $O(\lambda_2^m)$ and therefore faster than the original sequence $\{S_m\}$. Similarly, if we recursively define sequences $\{T_m^{(n)}\}(n = 2, 3, \ldots)$ by

$$T_m^{(n)} = \frac{T_{m+1}^{(n-1)} - \lambda_n T_m^{(n-1)}}{1-\lambda_n}, \tag{2.29}$$

the new sequences $\{T_m^{(n)}\}$ converge to S_∞ in $O(\lambda_{n+1}^m)$. The algorithm (2.29) is called the Richardson acceleration method.

Next we consider another case; i.e. each λ_i is unknown. We replace λ_1 in eq. (2.28) by its estimated value $\hat\lambda_1$ defined by

$$\hat\lambda_1 = \frac{S_{m+2} - S_{m+1}}{S_{m+1} - S_m}. \tag{2.30}$$

Then eq. (2.28) gives the following transformation;

$$S_m^{(1)} = \frac{\begin{vmatrix} S_m & S_{m+1} \\ S_{m+1} & S_{m+2} \end{vmatrix}}{\Delta^2 S_m}. \tag{2.31}$$

The above transformation is called the Aitken acceleration method. The Shanks transformation (2.19) is a generalized version of the Aitken acceleration. Since one needs a large amount of calculation in order to find the numerical value of determinants (2.19), Wynn [40] proposed the $\varepsilon-$algorithm, from which one finds it without calculating directly.

Since the $\varepsilon-$algorithm originates with the Richardson and the Aitken acceleration methods, it may well accelerate the sequence (2.23), which decays exponentially. For rationally decaying sequences such as eq. (2.25), λ_1 is very close to 1. This makes the denominator of eq. (2.28) near 0. Therefore, the $\varepsilon-$algorithm, intuitively speaking, fails to accelerate rationally decaying sequences.

Then how is the acceleration for alternatively decaying sequences? Since λ_1 is very close to -1, approximation of $\lambda_1 = -1$ in eq. (2.28) gives

$$T_m^{(1)} = \frac{S_{m+1} + S_m}{2}, \tag{2.32}$$

which is the mean value of S_m and S_{m+1}. It is well-known that the mean value of two neighboring elements of alternatively decaying sequence $\{S_m\}$ gives a better approximation of S_∞. We remark that the Richardson acceleration method for alternatively decaying sequences is equivalent to the Euler transformation [23].

2.4 The Toda molecule equation and convergence acceleration methods

In this section, we first consider the Toda molecule equation,

$$\begin{aligned}
\frac{\mathrm{d}^2 Q_1}{\mathrm{d}t^2} &= -e^{-(Q_2-Q_1)}, \\
\frac{\mathrm{d}^2 Q_n}{\mathrm{d}t^2} &= -e^{-(Q_{n+1}-Q_n)} + e^{-(Q_n-Q_{n-1})} \qquad (n = 1, 2, \ldots, N-1), \\
\frac{\mathrm{d}^2 Q_N}{\mathrm{d}t^2} &= e^{-(Q_N-Q_{N-1})}.
\end{aligned} \tag{2.33}$$

Equation (2.33) is obtained by imposing the formal boundary condition,

$$Q_0 = -\infty, \ Q_{N+1} = \infty \tag{2.34}$$

in the Toda lattice eq. (1.4). Owing to its boundary condition (2.34), each particle described by the Toda molecule eq. (2.33) moves freely and distance between two neighboring particles becomes infinite. We here relate such behaviors of solutions for the Toda molecule eq. (2.33) with convergence acceleration. It should be noted that the relation between the Toda molecule eq.(2.33) and the matrix eigenvalue algorithms has been clarified (See Appendix A or ref. [25].).

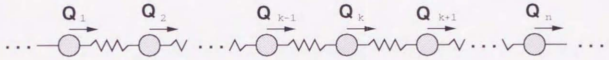


Figure 2.4: Toda molecule equation

Since the $\varepsilon-$algorithm (2.18) is regarded as a full discrete equation, we here consider the time discrete version of the Toda molecule equation [17],

$$\left\{ \begin{array}{rcl} I_{n+1}^{(m)} V_n^{(m)} & = & I_n^{(m+1)} V_n^{(m+1)} \\ I_n^{(m+1)} - I_n^{(m)} & = & V_n^{(m)} - V_{n-1}^{(m+1)} \end{array} \right. . \tag{2.35}$$

The quantities $I_n^{(m)}$ and $V_n^{(m)}$ correspond to $\dot{Q}_n(t)$ and $\exp(-(Q_{n+1}(t) - Q_n(t)))$, respectively. Therefore $V_n^{(m)}$ tends exponentially to zero as $m \to \infty$. We relate the time evolution of eq. (2.35) with the $\varepsilon-$algorithm by means of the Padé approximation and continued fraction.

2.4.1 Padé approximation

Let

$$f(x) = \sum_{i=0}^{\infty} c_i x^i \tag{2.36}$$

be a given formal power series. Substitution of $x = 1$ in eq. (2.36) gives an infinite series. Padé approximation of $f(x)$ consists in finding a rational function $R_{m,n}(x)$ with the numerator of the $m-$th degree and the denominator of the $n-$th degree,

$$R_{m,n}(x) = \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m}{b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n} \equiv \frac{P_{m,n}(x)}{Q_{m,n}(x)}, \tag{2.37}$$

such that

$$f(x) - R_{m,n}(x) = O(x^{m+n+1}), \tag{2.38}$$

or equivalently,

$$Q_{m,n}(x) f(x) - P_{m,n}(x) = O(x^{m+n+1}) \tag{2.39}$$

holds. Putting $b_0 = 1$, we see that eq. (2.39) is equivalent to the following Padé equations for $(a_0, a_1, \ldots, a_m)$ and $(b_1, b_2, \ldots, b_n)$;

$$\begin{pmatrix} c_{m-n+1} & c_{m-n+2} & c_{m-n+3} & \cdots & c_m \\ c_{m-n+2} & c_{m-n+3} & c_{m-n+4} & \cdots & c_{m+1} \\ c_{m-n+3} & c_{m-n+4} & c_{m-n+5} & \cdots & c_{m+2} \\ \vdots & \vdots & & & \vdots \\ c_m & c_{m+1} & c_{m+2} & \cdots & c_{m+n-1} \end{pmatrix} \begin{pmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \vdots \\ b_1 \end{pmatrix} = - \begin{pmatrix} c_{m+1} \\ c_{m+2} \\ c_{m+3} \\ \vdots \\ c_{m+n} \end{pmatrix}, \tag{2.40}$$

$$\begin{aligned} a_0 &= c_0, \\ a_1 &= c_1 + b_1 c_0, \\ a_2 &= c_2 + b_1 c_1 + b_2 c_0, \\ &\vdots \\ a_m &= c_m + \sum_{i=1}^{\min(m,n)} b_i c_{m-i}. \end{aligned} \tag{2.41}$$

Solving linear eqs. (2.40) and (2.41), we have a Padé approximant $R_{m,n}(x)$ for $f(x)$. By changing degrees m and n, we see that the functions $R_{m,n}(x)$ are arranged in a two-dimensional array called the Padé table (Figure 2.5).

$$\begin{array}{cccccc}
R_{0,0} & R_{0,1} & R_{0,2} & R_{0,3} & R_{0,4} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
R_{m-1,0} & R_{m-1,1} & R_{m-1,2} & R_{m-1,3} & R_{m-1,4} & \cdots \\
R_{m,0} & R_{m,1} & R_{m,2} & R_{m,3} & R_{m,4} & \cdots \\
R_{m+1,0} & R_{m+1,1} & R_{m+1,2} & R_{m+1,3} & R_{m+1,4} & \cdots \\
R_{m+2,0} & R_{m+2,1} & R_{m+2,2} & R_{m+2,3} & R_{m+2,4} & \cdots \\
R_{m+3,0} & R_{m+3,1} & R_{m+3,2} & R_{m+3,3} & R_{m+3,4} & \cdots \\
R_{m+4,0} & R_{m+4,1} & R_{m+4,2} & R_{m+4,3} & \cdots & \\
R_{m+5,0} & R_{m+5,1} & R_{m+5,2} & R_{m+5,3} & \cdots & \\
R_{m+6,0} & R_{m+6,1} & R_{m+6,2} & \cdots & & \\
\vdots & & & & &
\end{array}$$

Figure 2.5: The Padé table

Example 2.1 *Let function $f(x)$ be*

$$f(x) = \sqrt{\frac{1+\frac{x}{2}}{1+2x}} = 1 - \frac{3}{4}x + \frac{39}{32}x^2 - \cdots.$$

The Padé approximants for $f(x)$ are

$$R_{0,0} = \frac{1}{1},\ R_{1,0} = \frac{1-\frac{3}{4}x}{1}$$

$$R_{0,1} = \frac{1}{1+\frac{3}{4}x},\ R_{1,1} = \frac{1+\frac{7}{8}x}{1+\frac{13}{8}x}.$$

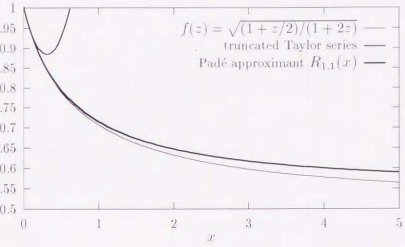


Figure 2.6: Values of $f(x)$, its Padé approximant $R_{1,1}$, and its truncated Taylor series $1 - 3z/4 + 39z^2/32$.

2.4.2 Padé approximation and the $\varepsilon-$algorithm

We here reveal the relation between the Padé approximation and the $\varepsilon-$algorithm (2.18). We see from Cramer's formula that the numerator and the denominator of the Padé approximant $R_{m,n}(x)$ are expressed as the following determinant form;

$$P_{m,n}(x) = \begin{vmatrix} c_{m-n+1} & c_{m-n+2} & \cdots & c_{m+1} \\ c_{m-n+2} & c_{m-n+3} & \cdots & c_{m+2} \\ \vdots & \vdots & & \vdots \\ c_m & c_{m+1} & \cdots & c_{m+n} \\ \sum_{i=0}^{m-n} c_i x^{n+i} & \sum_{i=0}^{m-n+1} c_i x^{n+i-1} & \cdots & \sum_{i=0}^{m} c_i x^i \end{vmatrix}, \quad (2.42)$$

$$Q_{m,n}(x) = \begin{vmatrix} c_{m-n+1} & c_{m-n+2} & \cdots & c_{m+1} \\ c_{m-n+2} & c_{m-n+3} & \cdots & c_{m+2} \\ \vdots & \vdots & & \vdots \\ c_m & c_{m+1} & \cdots & c_{m+n} \\ x^n & x^{n-1} & \cdots & 1 \end{vmatrix}. \tag{2.43}$$

Substitution of $x = 1$ in eqs. (2.42) and (2.43) gives

$$R_{m,n}(1) = \frac{P_{m,n}(1)}{Q_{m,n}(1)} = \frac{\begin{vmatrix} S_{m-n} & S_{m-n+1} & \cdots & S_m \\ S_{m-n+1} & S_{m-n+2} & \cdots & S_{m+1} \\ \vdots & \vdots & & \vdots \\ S_m & S_{m+1} & \cdots & S_{m+n} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_{m-n} & \Delta^2 S_{m-n+1} & \cdots & \Delta^2 S_{m-1} \\ \Delta^2 S_{m-n+1} & \Delta^2 S_{m-n+2} & \cdots & \Delta^2 S_m \\ \vdots & \vdots & & \vdots \\ \Delta^2 S_{m-1} & \Delta^2 S_m & \cdots & \Delta^2 S_{m+n-2} \end{vmatrix}}, \tag{2.44}$$

where S_m is a summation of c_i's,

$$S_m = \sum_{i=0}^{m} c_i. \tag{2.45}$$

Equation (2.44) immediately reminds us of the Shanks transformation (2.19). In fact, the Padé approximation and the $\varepsilon-$algorithm are related by

$$\varepsilon_{2n+1}^{(m)} = R_{m+n,n}(1). \tag{2.46}$$

2.4.3 Padé approximation and the discrete Toda molecule equation

We here clarify the relation between the Padé approximation and the QD algorithm by means of continued fractions.[3] Let us begin with the following theorem.

Theorem 2.1 (Gragg [14], Baker and Graves-Morris [2])

Let

$$T_m = \{R_{m,0}, R_{m+1,0}, R_{m+1,1}, R_{m+2,1}, R_{m+2,2}, \ldots\} \ (m \geq 0) \tag{2.47}$$

denote a sequence of Padé approximants for $f(x)$, any three consecutive elements of which are different. Then there exists a continued fraction,

$$d_0 + d_1 x + \cdots + d_m x^m + \frac{d_{m+1}x^{m+1}|}{|1} + \frac{d_{m+2}x|}{|1} + \frac{d_{m+3}x|}{|1} + \cdots \tag{2.48}$$

such that the $n-$th approximant of eq. (2.48) is equal to the $(n+1)-$st term of T_m for any $n \geq 1$.

Let $g_m(x)$ be a continued fraction whose $n-$th approximant is equal to the $(n+1)-$st term of T_m for any n. We write $g_m(x)$ as follows;

$$\begin{aligned} g_m(x) &= c_0 + c_1 x + \cdots + c_m x^m + \frac{c_{m+1}x^{m+1}|}{|1} - \frac{I_1^{(m+1)}x|}{|1} \\ &\quad - \frac{V_1^{(m+1)}x|}{|1} - \frac{I_2^{(m+1)}x|}{|1} - \frac{V_2^{(m+1)}x|}{|1} - \frac{I_3^{(m+1)}x|}{|1} - \cdots. \end{aligned} \tag{2.49}$$

Taking the even part of eq. (2.49), we have another expression for $g_m(x)$,

$$g_m(x) = c_0 + c_1 x + \cdots c_m x^m + \frac{c_{m+1}x^{m+1}|}{|1 - I_1^{(m+1)}x}$$

[3]Elementary properties of continued fractions are given in Appendix B or in refs. [39, 19].

$$+\frac{-I_1^{(m+1)}V_1^{(m+1)}x^2}{\left|1-(I_2^{(m+1)}+V_1^{(m+1)})x\right.}+\frac{-I_2^{(m+1)}V_2^{(m+1)}x^2}{\left|1-(I_3^{(m+1)}+V_2^{(m+1)})x\right.}+\cdots. \tag{2.50}$$

In the same way, we consider a continued fraction,

$$\begin{aligned} g_{m-1}(x) &= c_0+c_1x+\cdots+c_{m-1}x^{m-1}+\frac{c_mx^m}{|1}-\frac{I_1^{(m)}x}{|1} \\ &\quad -\frac{V_1^{(m)}x}{|1}-\frac{I_2^{(m)}x}{|1}-\frac{V_2^{(m)}x}{|1}-\frac{I_3^{(m)}x}{|1}-\cdots, \end{aligned} \tag{2.51}$$

the $n-$th approximant of which equals the $(n+1)-$st term of T_{m-1}. Taking the odd part of eq. (2.51), we have

$$\begin{aligned} g_{m-1}(x) &= c_0+c_1x+\cdots c_mx^m+\frac{c_mI_1^{(m)}x^{m+1}}{\left|1-(I_1^{(m)}+V_1^{(m)})x\right.} \\ &\quad +\frac{-I_2^{(m)}V_1^{(m)}x^2}{\left|1-(I_2^{(m)}+V_2^{(m)})x\right.}+\frac{-I_3^{(m)}V_2^{(m)}x^2}{\left|1-(I_3^{(m)}+V_3^{(m)})x\right.}+\cdots. \end{aligned} \tag{2.52}$$

The $n-$th approximants of eqs. (2.50) and (2.52) are equal to $(2n+1)-$st term of T_m and $(2n+2)-$nd term of T_{m-1}, respectively, both of which is nothing but $R_{m+n,n}(x)$. Therefore, two continued fractions (2.50) and (2.52) are equivalent. This leads to the following relation between the quantities $\{I_n^{(m)}\}$ and $\{V_n^{(m)}\}$,

$$I_{n+1}^{(m)}V_n^{(m)} = I_n^{(m+1)}V_n^{(m+1)}, \tag{2.53}$$

$$I_n^{(m)}+V_n^{(m)} = I_n^{(m+1)}+V_{n-1}^{(m+1)}, \tag{2.54}$$

$$V_0^{(m)}=0, \qquad I_1^{(m)}=\frac{c_{m+1}}{c_m}, \tag{2.55}$$

which is nothing but the time-discrete Toda molecule eq. (2.35), or equivalently the QD algorithm in terms of numerical analysis [34]. The QD algorithm is used to solve an algebraic equation and is equivalent to the LR algorithm to calculate eigenvalues of a given tridiagonal matrix [38].

We see from the above discussion that the formal power series,

$$f(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + \cdots \tag{2.56}$$

has a continued fraction expression,

$$\begin{aligned} c_0 + c_1x + \cdots + c_mx^m + \frac{c_{m+1}x^{m+1}|}{|\ 1} - \frac{I_1^{(m+1)}x|}{|\ 1} - \frac{V_1^{(m+1)}x|}{|\ 1} \\ - \frac{I_2^{(m+1)}x|}{|\ 1} - \frac{V_2^{(m+1)}x|}{|\ 1} - \frac{I_3^{(m+1)}x|}{|\ 1} - \frac{V_3^{(m+1)}x|}{|\ 1} - \cdots \end{aligned} \tag{2.57}$$

and that each coefficient is governed by the time-discrete Toda molecule equation. Putting $x = 1$ in eq. (2.57), we have a continued fraction expression for an infinite series,

$$\begin{aligned} \sum_{i=0}^{\infty} c_i &= c_0 + c_1 + \cdots + c_m + \frac{c_{m+1}|}{|\ 1} - \frac{I_1^{(m+1)}|}{|\ 1} - \frac{V_1^{(m+1)}|}{|\ 1} \\ &\quad - \frac{I_2^{(m+1)}|}{|\ 1} - \frac{V_2^{(m+1)}|}{|\ 1} - \frac{I_3^{(m+1)}|}{|\ 1} - \frac{V_3^{(m+1)}|}{|\ 1} - \cdots \end{aligned} \tag{2.58}$$

We again consider the time-discrete Toda molecule eq. (2.35). Since variables $V_n^{(m)}$ and $I_n^{(m)}$ correspond to $\exp(-(Q_{n+1}(t) - Q_n(t)))$ and $\dot{P}_n(t)$, respectively, $V_n^{(m)}$ decays exponentially to 0 as m tends to infinity. If we approximate $V_n^{(m+1)} = 0$ in eq. (2.57) for fixed $n \geq 1$, we have one approximation for the infinite series $\sum_{i=0}^{\infty} c_i$. The approximation of $V_n^{(m+1)} = 0$ is equivalent to taking the $2n-$th convergent of the continued fraction (2.58). From the theorem 2.1, the $2n-$th convergent of eq. (2.58) equals $(2n + 1)-$st term of the sequence $T_m(1)$, namely $R_{m+n,n}(1)$.

Combining the above result with eq. (2.46), we arrive at the following relation[4] between the $\varepsilon-$algorithm and the time-discrete Toda molecule eq. (2.35):

$$\begin{aligned}\varepsilon_{2n+1}^{(m)} &= S_m + \frac{c_{m+1}|}{|1} - \frac{I_1^{(m+1)}|}{|1} - \frac{V_1^{(m+1)}|}{|1} - \frac{I_2^{(m+1)}|}{|1} \\ &\quad - \frac{V_2^{(m+1)}|}{|1} - \frac{I_3^{(m+1)}|}{|1} - \frac{V_3^{(m+1)}|}{|1} - \cdots - \frac{I_n^{(m+1)}|}{|1}. \end{aligned} \tag{2.59}$$

We remark that approximation of $V_1^{(m+1)} = 0$ in eq. (2.58) gives the Aitken acceleration. We see from properties of continued fraction that the error between the quantities $\varepsilon_{2n+1}^{(m)}$ and the limit $S_\infty = \lim_{m\to\infty} S_m$ is roughly estimated as

$$|\varepsilon_{2n+1}^{(m)} - S_\infty| \sim V_1^{(m+1)} V_2^{(m+1)} \cdots V_n^{(m+1)}, \tag{2.60}$$

or equivalently

$$\frac{\varepsilon_{2n+1}^{(m)} - S_\infty}{\varepsilon_{2n-1}^{(m)} - S_\infty} \sim V_n^{(m+1)}, \tag{2.61}$$

provided that $I_n^{(m)} \sim O(1)$. Since the quantities $V_n^{(m+1)}$ tends exponentially to zero as $m \to \infty$, the right hand side of eq. (2.61) also tends in the same order to zero, which matches exactly the definition of convergence acceleration.

The Toda molecule equation, though it may not be interesting in a physical sense, stands for a dynamical system which converges to a diagonal matrix and appear in many other fields of engineering. We should in future consider such molecule-type soliton equations which are expected to find applications in mathematical engineering.

[4] See also Cuyt [10]

Chapter 3

The ρ–algorithm

3.1 The ρ–algorithm and the Cylindrical KdV equation

The ρ–algorithm is traced back to Thiele's rational interpolation [36]. It was first used as a convergence accelerator by Wynn [41]. The initial values of the algorithm are given by

$$\rho_0^{(m)} = 0,\ \rho_1^{(m)} = S_m\ (m = 0, 1, 2, \ldots), \tag{3.1}$$

and all the other elements fulfill the following rhombus rule;

$$(\rho_{n+1}^{(m)} - \rho_{n-1}^{(m+1)})(\rho_n^{(m+1)} - \rho_n^{(m)}) = n. \tag{3.2}$$

The ρ–algorithm is almost the same as the ε–algorithm except that "1" in the right hand side of eq. (2.18) is replaced by "n" in eq. (3.2). This slight change, however, yields considerable differences in various aspects between these two algorithms.

The first difference is in their performance [33]. The ε–algorithm accelerates exponentially or alternatively decaying sequences, while the ρ–algorithm does rationally decaying sequences as eq. (2.25) (See Table 3.1).

Table 3.1: The $\rho-$algorithm for the sequence $S_m = \sum_{k=1}^{m} 1/k^2$

m	$\rho_1^{(m)}$	$\rho_3^{(m)}$	$\rho_5^{(m)}$	$\rho_7^{(m)}$	$\rho_9^{(m)}$
1	1.000000	1.650000	1.644895	1.64493437	1.644934064
2	1.250000	1.646825	1.644923	1.64493414	1.6449340662
3	1.361111	1.645833	1.644930	1.64493409	1.6449340666
4	1.423611	1.645429	1.644932	1.64493407	1.6449340667
5	1.463611	1.645235	1.644933	1.64493407	
6	1.491389	1.645130	1.64493457	1.64493406	
7	1.511797	1.645069	1.64493478		
8	1.527422	1.645031	1.64493489		
9	1.539768	1.645006			
10	1.549768	1.644989			
11	1.558032				
12	1.564977				

The second difference is in their determinant expressions. The quantities $\varepsilon_n^{(m)}$ are given by ratios of Hankel determinants, while the quantities $\rho_n^{(m)}$ are given by [36]

$$\rho_n^{(m)} = (-1)^{\left[\frac{n-1}{2}\right]} \frac{\bar{\tau}_n^{(m)}}{\tau_n^{(m)}}, \tag{3.3}$$

where $[x]$ stands for the greatest integer less than or equal to x. Moreover, the functions $\tau_n^{(m)}$ and $\bar{\tau}_n^{(m)}$ are expressed as the following double Casorati determinants;

$$\tau_n^{(m)} = \begin{cases} u^{(m)}(k;k) & n = 2k, \\ u^{(m)}(k+1;k) & n = 2k+1, \end{cases} \tag{3.4}$$

$$\tilde{\tau}_n^{(m)} = \begin{cases} u^{(m)}(k+1;k-1) & n=2k, \\ u^{(m)}(k;k+1) & n=2k+1, \end{cases} \tag{3.5}$$

where

$$u^{(m)}(p;q) = \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ m & m+1 & \cdots & m+p+q-1 \\ m^2 & (m+1)^2 & \cdots & (m+p+q-1)^2 \\ \vdots & \vdots & & \vdots \\ m^{p-1} & (m+1)^{p-1} & \cdots & (m+p+q-1)^{p-1} \\ S_m & S_{m+1} & \cdots & S_{m+p+q-1} \\ mS_m & (m+1)S_{m+1} & \cdots & (m+p+q-1)S_{m+p+q-1} \\ m^2S_m & (m+1)^2S_{m+1} & \cdots & (m+p+q-1)^2S_{m+p+q-1} \\ \vdots & \vdots & & \vdots \\ m^{q-1}S_m & (m+1)^{q-1}S_{m+1} & \cdots & (m+p+q-1)^{q-1}S_{m+p+q-1} \end{bmatrix}. \tag{3.6}$$

The third difference is in the relation with discrete soliton equations. We have seen in the previous section that the ε-algorithm is regarded as the discrete potential KdV equation. Instead of the ρ-algorithm (3.2) itself, we consider the algorithm of the following form;

$$(\rho_{n+1}^{(m)} - \rho_{n-1}^{(m+1)})(\rho_n^{(m+1)} - \rho_n^{(m)}) = an + b(m+1), \tag{3.7}$$

where a and b are constant. Employing a dependent variable transformation,

$$Y_n^{(m)} = \rho_n^{(m)} - \rho_n^{(m-1)}, \tag{3.8}$$

we obtain

$$Y_{n+1}^{(m)} - Y_{n-1}^{(m+1)} = \frac{an+bm+b}{Y_n^{(m+1)}} - \frac{an+bm}{Y_n^{(m)}} \tag{3.9}$$

from eq. (3.7). Equation (3.9) possesses a form similar to the discrete KdV eq. (2.7). However, the nonautonomous property of eq. (3.9) yields an essential difference in its continuous limit. Let us introduce new variables t, x defined by

$$\frac{t}{\epsilon^3} = an + bm, \quad \frac{x}{\epsilon} = cn + m - \frac{1}{2}, \tag{3.10}$$

and rewrite $Z_n^{(m)}$ as $\epsilon^{-3/2}\sqrt{t}\,\{p + \epsilon^2 u(x - \epsilon/2, t)\}$, where ϵ is a small parameter and p, c are finite constants satisfying

$$p^2 = \frac{b}{2a-b}, \quad c = \frac{1}{2} - \frac{1}{2p^2} = \frac{b-a}{b}. \tag{3.11}$$

Then eq. (3.9) becomes

$$\begin{aligned}
&\epsilon^2 u(x - \frac{\epsilon}{2} + c\epsilon, t + a\epsilon^3) - \epsilon^2 u(x + \frac{\epsilon}{2} - c\epsilon, t + (b-a)\epsilon^3) \\
- &\frac{1}{p + \epsilon^2 u(x + \frac{\epsilon}{2}, t + b\epsilon^3)} + \frac{1}{p + \epsilon^2 u(x - \frac{\epsilon}{2}, t)} \\
+ &\frac{\epsilon^3}{2t}\left[a\left\{p + \epsilon^2 u(x - \frac{\epsilon}{2} + c\epsilon, t + a\epsilon^3)\right\}\right. \\
&\quad + (a-b)\left\{p + \epsilon^2 u(x + \frac{\epsilon}{2} - c\epsilon, t + (b-a)\epsilon^3)\right\} \\
&\quad - \left.\frac{b}{p + \epsilon^2 u(x + \frac{\epsilon}{2}, t + b\epsilon^3)}\right] = 0.
\end{aligned} \tag{3.12}$$

Taking the small limit of ϵ, we have

$$(2a-b)u_t - \frac{1}{p^3}uu_x + \frac{1}{48p^2}(1 - \frac{1}{p^4})u_{xxx} + \frac{(2a-b)}{2t}u = 0 \tag{3.13}$$

at the order of ϵ^5 from eq. (3.12). Since the coefficient of u_t is always twice as large as that of u/t, eq. (3.7) is considered as one integrable discretization of the cylindrical KdV equation. It is interesting to note that the $\rho-$algorithm (3.2) is not exactly discretization of the cylindrical KdV equation. This is because we have $p = 0$ in eq. (3.11) and coefficients of uu_x and u_{xxx} become infinite in the case of the $\rho-$algorithm.

The third difference can be understood clearly from a viewpoint of the Hirota's formalism. Employing the same dependent variable transformation as eq. (3.8), we obtain

$$Y'^{(m)}_{n+1} - Y'^{(m+1)}_{n-1} = \frac{n}{Y'^{(m+1)}_{n}} - \frac{n}{Y'^{(m)}_{n}} \tag{3.14}$$

from eq. (3.2). Moreover, through the same dependent variable transformation as eq. (2.13),

$$Y_n^{(m)} = \frac{F(n+1,m-1)F(n-1,m)}{F(n,m-1)F(n,m)}, \tag{3.15}$$

eq. (3.9) is rewritten as the following trilinear form;

$$\begin{vmatrix} -F(n+2,m-1) & F(n+1,m-1) & nF(n,m-1) \\ F(n+1,m) & 0 & F(n-1,m) \\ -nF(n,m+1) & F(n-1,m+1) & F(n-2,m+1) \end{vmatrix} = 0. \tag{3.16}$$

The functions $F(n,m)$ and $\tau_n^{(m)}$ in eq. (3.4) are related by

$$F(n,m) = (-1)^{a(n)}\tau_n^{(m)}, \tag{3.17}$$

where $a(n)$ satisfies

$$a(n) \equiv a(n-2) + \left[\frac{n-2}{2}\right] \pmod 2,\ a(0) = a(1) = 0. \tag{3.18}$$

Because of nonautonomous property of eq. (3.9), there is no way to derive a bilinear form with a single variable $F(n,m)$ from the trilinear eq. (3.16). This fact reminds us of the similarity constraint of the discrete KdV equation [30]. It should be noted, however, that a pair of functions $\tau_n^{(m)}$ and $\tilde{\tau}_n^{(m)}$ given by eqs. (3.4) and (3.5) satisfy bilinear equations,

$$\tau_{n+1}^{(m)}\tau_{n-1}^{(m+1)} - \tau_n^{(m)}\tilde{\tau}_n^{(m+1)} + \tau_n^{(m+1)}\tilde{\tau}_n^{(m)} = 0, \tag{3.19}$$

$$\tau_{n-1}^{(m+1)}\tilde{\tau}_{n+1}^{(m)} + \tau_{n+1}^{(m)}\tilde{\tau}_{n-1}^{(m+1)} - n\tau_n^{(m)}\tau_n^{(m+1)} = 0, \tag{3.20}$$

which are considered to be the Jacobi and the Plücker identities for determinants, respectively.

3.2 Reciprocal differences

In the previous section, we show that the ρ–algorithm accelerates rationally decaying sequences and that the quantities $\rho_n^{(m)}$ are given by the ratios of double Casorati determinants. In this section, we show that these facts are quite natural if we discuss the ρ–algorithm in relation with Thiele's interpolation formula [36].

Let the values of an unknown function $f(x)$ be given for the values $x_0, x_1, \cdots, x_n$, no two of which are equal. Then reciprocal differences $\rho_i(x_k x_{k+1} \cdots x_{k+i})$ are defined by

$$\begin{aligned}
\rho_1(x_0) &= f(x_0), && (3.21)\\
\rho_2(x_0x_1) &= \frac{x_0 - x_1}{\rho_1(x_0) - \rho_1(x_1)}, && (3.22)\\
\rho_3(x_0x_1x_2) &= \frac{x_0 - x_2}{\rho_2(x_0x_1) - \rho_2(x_1x_2)} + \rho_1(x_1), && (3.23)\\
\rho_4(x_0x_1x_2x_3) &= \frac{x_0 - x_3}{\rho_3(x_0x_1x_2) - \rho_3(x_1x_2x_3)} + \rho_2(x_1x_2), && (3.24)\\
&\vdots\\
\rho_{n+1}(x_0x_1\cdots x_n) &= \frac{x_0 - x_n}{\rho_n(x_0x_1\cdots x_{n-1}) - \rho_n(x_1x_2\cdots x_n)}\\
&\quad + \rho_{n-1}(x_1x_2\cdots x_{n-1}). && (3.25)
\end{aligned}$$

We remark that substitution of $x_k = k$ in the above equations gives the ρ–algorithm (3.2).

Let us replace x_0 by x in eqs. (3.22)–(3.25). Then eqs. (3.21)–(3.25) are equivalent to the following identities;

$$\begin{aligned}
f(x) &= \rho_1(x), && (3.26)\\
\rho_1(x) &= \rho_1(x_1) + \frac{x - x_1}{\rho_2(xx_1)}, && (3.27)\\
\rho_2(xx_1) &= \rho_2(x_1x_2) + \frac{x - x_2}{\rho_3(xx_1x_2) - \rho_1(x_1)}, && (3.28)\\
\rho_3(xx_1x_2) &= \rho_3(x_1x_2x_3) + \frac{x - x_3}{\rho_4(xx_1x_2x_3) - \rho_2(x_1x_2)}, && (3.29)
\end{aligned}$$

$$\vdots$$

$$\begin{aligned}\rho_{n+1}(xx_1x_2\cdots x_n) &= \rho_{n+1}(x_1x_2\cdots x_{n+1}) \\ &\quad + \frac{x-x_n}{\rho_{n+2}(xx_1x_2\cdots x_{n+1})-\rho_n(x_1x_2\cdots x_n)}. \qquad (3.30)\end{aligned}$$

We obtain the following continued fraction expression for $f(x)$ from eqs. (3.26)–(3.30);

$$f(x)=\rho_1(x_1)+\cfrac{x-x_1}{\rho_2(x_1x_2)+\cfrac{x-x_2}{\rho_3(x_1x_2x_3)-\rho_1(x_1)+\cfrac{x-x_3}{\rho_4(x_1x_2x_3x_4)-\rho_2(x_1x_2)+\ddots}}}. \qquad (3.31)$$

When we take n–th convergent of eq. (3.31), we obtain a rational function, which agrees in value with $f(x)$ at the points

$$x_1, x_2, \cdots, x_n,$$

which is called Thiele's interpolation formula.

Let us consider continued fraction of the form,

$$a_1+\cfrac{b_2}{a_2+\cfrac{b_3}{a_3+\cfrac{b_4}{a_4+\ddots}}}. \qquad (3.32)$$

If $\dfrac{p_n}{q_n}$ denotes the n–th convergent of eq. (3.32), we have the well-known recurrence relations,

$$\left\{\begin{aligned} p_n &= a_np_{n-1}+b_np_{n-2} \\ q_n &= a_nq_{n-1}+b_nq_{n-2}\end{aligned}\right. . \qquad (3.33)$$

If we write $p_0 = 1, q_0 = 0$, we have

$$\begin{bmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{bmatrix} = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_2 & 1 \\ b_2 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_{n-1} & 1 \\ b_{n-1} & 0 \end{bmatrix} \begin{bmatrix} a_n & 1 \\ b_n & 0 \end{bmatrix} \tag{3.34}$$

In particular, the components p_n, q_n are given by

$$\begin{bmatrix} p_n \\ q_n \end{bmatrix} = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_2 & 1 \\ b_2 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_{n-1} & 1 \\ b_{n-1} & 0 \end{bmatrix} \begin{bmatrix} a_n \\ b_n \end{bmatrix} \tag{3.35}$$

Let us return to eq. (3.31). We rewrite

$$y = f(x),\ y_s = f(x_s),\ \rho_s = \rho_s(x_1 x_2 \cdots x_s) \tag{3.36}$$

for brevity. If we put n−th convergent of eq. (3.31) as $\dfrac{p_n(x)}{q_n(x)}$, we see from the above fact that the components $p_n(x)$ and $q_n(x)$ are given by

$$\begin{aligned} \begin{bmatrix} p_n(x) \\ q_n(x) \end{bmatrix} &= \begin{bmatrix} y_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \rho_2 & 1 \\ x - x_1 & 0 \end{bmatrix} \begin{bmatrix} \rho_3 - y_1 & 1 \\ x - x_2 & 0 \end{bmatrix} \begin{bmatrix} \rho_4 - \rho_2 & 1 \\ x - x_3 & 0 \end{bmatrix} \cdots \\ &\quad \times \begin{bmatrix} \rho_n - \rho_{n-2} & 1 \\ x - x_{n-1} & 0 \end{bmatrix} \begin{bmatrix} \rho_{n+1} - \rho_{n-1} \\ x - x_n \end{bmatrix}. \end{aligned} \tag{3.37}$$

We easily see that $p_{2n+1}(x)$, $q_{2n+1}(x)$, and $p_{2n}(x)$ are polynomials in x of degree n while function $q_{2n}(x)$ is a polynomial of degree $n-1$, and that these polynomials are written as

$$p_{2n}(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1}x^{n-1} + x^n, \tag{3.38}$$

$$q_{2n}(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{n-2}x^{n-2} + \rho_{2n}x^{n-1}, \tag{3.39}$$

$$p_{2n+1}(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1}x^{n-1} + \rho_{2n+1}x^n, \tag{3.40}$$

$$q_{2n+1}(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_{n-1}x^{n-1} + x^n. \tag{3.41}$$

Regarding

$$\frac{p_{2n}(x_s)}{q_{2n}(x_s)} = y_s \quad (s = 1, 2, \cdots, 2n), \tag{3.42}$$

and

$$\frac{p_{2n+1}(x_s)}{q_{2n+1}(x_s)} = y_s \quad (s = 1, 2, \cdots, 2n+1) \tag{3.43}$$

as simultaneous equations for $(a_0, a_1, \cdots, a_{n-1}, b_0, b_1, \cdots, b_{n-2}, \rho_{2n})$ and $(c_0, c_1, \cdots, c_{n-1}, d_0, d_1, \cdots, d_{n-1}, \rho_{2n+1})$, respectively, we see from Cramer's formula that the quantities ρ_{2n} and ρ_{2n+1} are given by

$$\rho_{2n} = \rho_{2n}(x_1 x_2 \cdots x_{2n}) = \frac{\det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_1 & y_2 & \cdots & y_{2n} \\ x_1 & x_2 & \cdots & x_{2n} \\ x_1 y_1 & x_2 y_2 & \cdots & x_{2n} y_{2n} \\ \vdots & \vdots & & \vdots \\ x_1^{n-2} & x_2^{n-2} & \cdots & x_{2n}^{n-2} \\ x_1^{n-2} y_1 & x_2^{n-2} y_2 & \cdots & x_{2n}^{n-2} y_{2n} \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_{2n}^{n-1} \\ x_1^{n} & x_2^{n} & \cdots & x_{2n}^{n} \end{bmatrix}}{\det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_1 & y_2 & \cdots & y_{2n} \\ x_1 & x_2 & \cdots & x_{2n} \\ x_1 y_1 & x_2 y_2 & \cdots & x_{2n} y_{2n} \\ \vdots & \vdots & & \vdots \\ x_1^{n-2} & x_2^{n-2} & \cdots & x_{2n}^{n-2} \\ x_1^{n-2} y_1 & x_2^{n-2} y_2 & \cdots & x_{2n}^{n-2} y_{2n} \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_{2n}^{n-1} \\ x_1^{n-1} y_1 & x_2^{n-1} y_2 & \cdots & x_{2n}^{n-1} y_{2n} \end{bmatrix}}, \tag{3.44}$$

and

$$\rho_{2n+1} = \rho_{2n+1}(x_1 x_2 \cdots x_{2n+1}) = \frac{\det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_1 & y_2 & \cdots & y_{2n+1} \\ x_1 & x_2 & \cdots & x_{2n+1} \\ x_1 y_1 & x_2 y_2 & \cdots & x_{2n+1} y_{2n+1} \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_{2n+1}^{n-1} \\ x_1^{n-1} y_1 & x_2^{n-1} y_2 & \cdots & x_{2n+1}^{n-1} y_{2n+1} \\ x_1^n y_1 & x_2^n y_2 & \cdots & x_{2n+1}^n y_{2n+1} \end{bmatrix}}{\det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_1 & y_2 & \cdots & y_{2n+1} \\ x_1 & x_2 & \cdots & x_{2n+1} \\ x_1 y_1 & x_2 y_2 & \cdots & x_{2n+1} y_{2n+1} \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_{2n+1}^{n-1} \\ x_1^{n-1} y_1 & x_2^{n-1} y_2 & \cdots & x_{2n+1}^{n-1} y_{2n+1} \\ x_1^n & x_2^n & \cdots & x_{2n+1}^n \end{bmatrix}}. \qquad (3.45)$$

Determinants in eqs. (3.44) and (3.45) become double Casorati determinants by changing their rows. It should be noted that eq. (3.6) is recovered by putting

$$x_k = k, \ y_k = S_k \qquad (3.46)$$

in eqs. (3.44) and (3.45). From the above discussion, we see that the ρ-algorithm originates with rational interpolation formula. This fact makes it clear that the algorithm accelerates sequences whose asymptotic behavior is given by

$$S_m \sim S_\infty + \frac{C_1}{m} + \frac{C_2}{m^2} + \frac{C_3}{m^3} + \cdots, \qquad (3.47)$$

exact proof of which is given in ref. [26].

3.3 Thiele's ρ-algorithm

In this section, we extend the ρ-algorithm according to the notion of Thiele's interpolation. When we redefine $u^{(m)}(p;q)$ in eq. (3.6) by

$$u^{(m)}(p;q) = \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \sigma(m) & \sigma(m+1) & \cdots & \sigma(m+p+q-1) \\ \sigma(m)^2 & \sigma(m+1)^2 & \cdots & \sigma(m+p+q-1)^2 \\ \vdots & \vdots & & \vdots \\ \sigma(m)^{p-1} & \sigma(m+1)^{p-1} & \cdots & \sigma(m+p+q-1)^{p-1} \\ S_m & S_{m+1} & \cdots & S_{m+p+q-1} \\ \sigma(m)S_m & \sigma(m+1)S_{m+1} & \cdots & \sigma(m+p+q-1)S_{m+p+q-1} \\ \sigma(m)^2 S_m & \sigma(m+1)^2 S_{m+1} & \cdots & \sigma(m+p+q-1)^2 S_{m+p+q-1} \\ \vdots & \vdots & & \vdots \\ \sigma(m)^{q-1} S_m & \sigma(m+1)^{q-1} S_{m+1} & \cdots & \sigma(m+p+q-1)^{q-1} S_{m+p+q-1} \end{bmatrix}, \tag{3.48}$$

we can construct an extended version of the ρ-algorithm,

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = \sigma(n+m) - \sigma(m). \tag{3.49}$$

Since we obtain the ρ-algorithm (3.2) by putting $\sigma(x) = x$, we call eq. (3.49) *Thiele's ρ-algorithm*. This algorithm accelerates sequences of the following form;

$$S_m \sim S_\infty + \frac{C_1}{\sigma(m)} + \frac{C_2}{(\sigma(m))^2} + \frac{C_3}{(\sigma(m))^3} + \cdots. \tag{3.50}$$

This is because application of the algorithm (3.49) is equivalent to interpolating original sequence $\{S_m\}$ by

$$\frac{c_0 + c_1\sigma(m) + c_2\sigma(m)^2 + \cdots + c_{n-1}\sigma(m)^{n-1} + \rho_{2n+1}\sigma(m)^n}{d_0 + d_1\sigma(m) + d_2\sigma(m)^2 + \cdots + d_{n-1}\sigma(m)^{n-1} + \sigma(m)^n} \quad (n = 1, 2, 3, \ldots). \tag{3.51}$$

Let us apply Thiele's $\rho-$algorithm for two examples and compare its performance with the $\rho-$algorithm (3.2). First we consider a sequence,

$$S_m = \sum_{k=1}^{m} \frac{1}{k^{3/2}} \to 2.61237534868\cdots, \tag{3.52}$$

whose asymptotic behavior is given by

$$S_m \sim S_\infty + \frac{C_1}{m^{1/2}} + \frac{C_2}{m^{3/2}} + \cdots. \tag{3.53}$$

In eq. (3.53), $C_i(i = 1, 2, \ldots)$ are constants. We put $\sigma(x) = x^{1/2}$ in eq. (3.49) and compare the result with the $\rho-$algorithm.

Next we consider the problem of evaluating

$$S_\infty = \int_0^1 g(x)\mathrm{d}x \tag{3.54}$$

by the trapezoidal rule. Define S_m by

$$S_m = \frac{1}{m}\left\{\frac{1}{2}g(0) + \sum_{k=1}^{m-1} g\left(\frac{k}{m}\right) + \frac{1}{2}g(1)\right\}. \tag{3.55}$$

If $g(x)$ is sufficiently smooth, an asymptotic behavior of S_m is given by

$$\begin{aligned} S_m &\sim S_\infty + \frac{D_1}{m^2} + \frac{D_2}{m^4} + \frac{D_3}{m^6} + \cdots, \qquad (3.56)\\ &D_1 = \frac{1}{12}\{g'(1) - g'(0)\},\ D_2 = -\frac{1}{720}\{g'''(1) - g'''(0)\},\ldots. \end{aligned}$$

We put $g(x) = (0.05 + x)^{-1/2}$ in eq. (3.54) and apply Thiele's $\rho-$algorithm with $\sigma(x) = x^2$.

As one can see from Figures 3.1 and 3.2, Thiele's $\rho-$algorithm accelerates convergence better than the original $\rho-$algorithm (3.2). We should select of $\sigma(x)$ in eq. (3.49) appropriately according to an asymptotic behavior of a given sequence S_m.

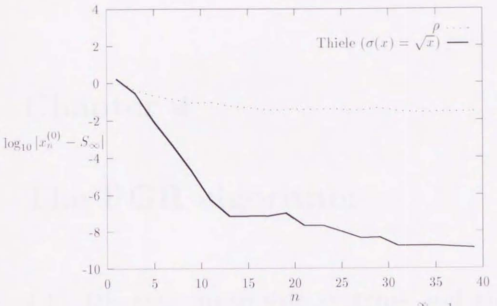


Figure 3.1: Acceleration methods for $\sum_{k=1}^{\infty} \frac{1}{k^{3/2}}$

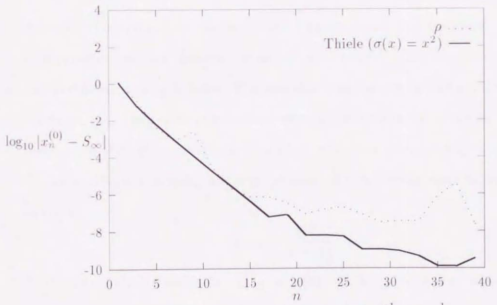


Figure 3.2: Acceleration methods for $\int_0^1 \frac{1}{(0.05+x)^{1/2}}\mathrm{d}x$

Chapter 4

The PGR algorithm

4.1 Discrete integrable systems and singularity confinement method

Recently, Grammaticos, Ramani, and Papageorgiou [15] proposed the singularity confinement test as a discrete version of the Painlevé test. The idea of the singularity confinement is as follows; The movable singularities of integrable mappings are confined, i.e., they are canceled out after a finite number of steps. Moreover, the memory of the initial condition is not lost whenever a singularity is crossed.

As a simple example, we first consider the following discrete integrable equation [31],

$$X_{n+1} = \frac{8X_n}{1-X_n^2} - X_{n-1}. \tag{4.1}$$

Under the initial conditions $X_{n-2} = \alpha(\text{finite}), X_{n-1} = 1 + \epsilon$, where ϵ is a small parameter, we have from straightforward calculation

$$X_n = -\frac{4}{\epsilon} - (2+\alpha) + O(\epsilon), \tag{4.2}$$

$$X_{n+1} = -1 + O(\epsilon), \tag{4.3}$$
$$X_{n+2} = -\alpha + O(\epsilon). \tag{4.4}$$

We see from the above equations that singularity of X_n is confined and that the initial value α reappears in X_{n+2}.

Following the result by Grammaticos et al. [15], we next derive the discrete Painlevé equation I from the viewpoint of the singularity confinement. We start from a nonautonomous mappings,

$$x_{n+1} = -x_{n-1} - x_n + B(n) + \frac{C(n)}{x_n}. \tag{4.5}$$

Under the initial conditions $x_{n-1} = a$(finite) and $x_n = \epsilon$, we have

$$\begin{aligned}
x_{n+1} &= \frac{C(n)}{\epsilon} + (B(n) - a) - \epsilon & (4.6)\\
x_{n+2} &= \frac{-C(n)}{\epsilon} + (B(n+1) - B(n) + a) + \frac{C(n+1)}{C(n)}\epsilon + O(\epsilon^2) & (4.7)\\
x_{n+3} &= B(n+2) - B(n+1) + \frac{C(n) - C(n+1) - C(n+2)}{C(n)}\epsilon + O(\epsilon^2) & (4.8)\\
x_{n+4} &= \frac{C(n)}{\epsilon} + \frac{C(n+3)}{B(n+2) - B(n+1) + \frac{C(n) - C(n+1) - C(n+2)}{C(n)}\epsilon}\\
&\quad -B(n+2) + B(n) - a + O(\epsilon). & (4.9)
\end{aligned}$$

The singularity confinement condition, i.e., x_{n+4} finite is given by

$$B(n+2) - B(n+1) = 0, \tag{4.10}$$
$$C(n+3) - C(n+2) - C(n+1) + C(n) = 0. \tag{4.11}$$

The solutions for these equations are

$$B(n) = b(= \text{const}),\ C(n) = \alpha n + \beta. \tag{4.12}$$

Then we obtain

$$x_{n+1} + x_n + x_{n-1} = b + \frac{\alpha n + \beta}{x_n} \tag{4.13}$$

from eq. (4.5), i.e., the discrete Painlevé equation of type I[1].

Although the singularity confinement test is not mathematically well grounded, discrete soliton equations are known to pass this test. Therefore it has been used to check integrability of given discrete nonlinear maps[2].

4.2 The PGR algorithm

In this section, we discuss "the PGR algorithm". First of all, we consider the algorithm given by

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = z_n^{(m)}, \tag{4.14}$$

where $z_n^{(m)}$ is a given function in m and n. Papageorgiou et al. [27] applied the singularity confinement test to eq. (4.14) (See Fig. 4.1). If we have $x_n^{(m+1)} = x_n^{(m)} + \delta$, then $x_{n+1}^{(m)}$ diverges as $z_n^{(m)}/\delta$. At the next iteration, we find

$$\begin{aligned} x_{n+2}^{(m-1)} &= x_n^{(m)} + \frac{z_{n+1}^{(m-1)}}{z_n^{(m)}}\delta + O(\delta^2), & (4.15)\\ x_{n+2}^{(m)} &= x_n^{(m+1)} + \left(1 - \frac{z_{n+1}^{(m)}}{z_n^{(m)}}\right)\delta + O(\delta^2). & (4.16)\end{aligned}$$

The singularity confinement condition, i.e. $x_{n+3}^{(m-1)}$ finite, gives the following condition for $z_n^{(m)}$

$$z_n^{(m)} - z_{n+1}^{(m-1)} - z_{n+1}^{(m)} + z_{n+2}^{(m-1)} = 0. \tag{4.17}$$

We call this generalized class of algorithms (4.14), where $z_n^{(m)}$ satisfies eq. (4.17), "the PGR algorithm". The ε ($z_n^{(m)} = 1$), ρ ($z_n^{(m)} = n$), and Thiele's ρ ($z_n^{(m)} =$

[1]The derivation of the other discrete Painlevé equations is given, for example, in refs. [15, 28].

[2]See refs. [28, 31, 34], for example.

$$\begin{tikzcd}
 & x_{n+1}^{(m-1)} & & \\
x_n^{(m)} & & x_{n+2}^{(m-1)}(finite) & \\
 & x_{n+1}^{(m)}(\infty) & & x_{n+3}^{(m-1)}(?) \\
x_n^{(m+1)}(=x_n^{(m)}+\delta) & & x_{n+2}^{(m)}(finite) & \\
 & x_{n+1}^{(m+1)} & &
\end{tikzcd}$$

Figure 4.1: The PGR algorithm

$\sigma(m+n)-\sigma(m))$ algorithms belong to the class of the PGR algorithm. It is interesting to remark that Cordellier [9] has already proposed the algorithm (4.14) satisfying the singularity confinement condition (4.17), which he termed "the homographic invariance".

Two question arises; (1) what kind of integrable equations is the PGR algorithm associated with if it is considered as a difference equation? (2) how is the performance of the PGR algorithm as a convergence accelerator?

As one solution to the first question, we show that it is related to the discrete Painlevé equation of type I (4.13). Let us consider a special case of the PGR algorithm,

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = \alpha n - \alpha m + C \ (C = \text{const}), \tag{4.18}$$

which passes the singularity confinement condition (4.17). Through variable transformations,

$$k = n - m, \ l = m, \tag{4.19}$$

we have

$$(X(k+1,l) - X(k-2,l+1))(X(k-1,l) - X(k,l)) = \alpha k + C. \tag{4.20}$$

Elimination of dependence of the variable l in eq. (4.20) gives the following equation;

$$X(k+1) - X(k-2) = \frac{-\alpha k - C}{X(k) - X(k-1)}. \tag{4.21}$$

Through dependent variable transformation,

$$Y(k) = X(k) - X(k-1), \tag{4.22}$$

we have

$$Y(k+1) + Y(k) + Y(k-1) = \frac{-\alpha k - C}{Y(k)} \tag{4.23}$$

from eq. (4.21). The equation (4.23) is the discrete Painlevé equation I (4.13) with $b = 0$.

Let us go to the second question, i.e. acceleration performance of the PGR algorithm. Intuitively speaking, most of the PGR algorithm do not well accelerate convergence as far as we have tested. However, when we take

$$z_n^{(m)} = \frac{1}{m+n+1}, \tag{4.24}$$

the algorithm accelerates both alternatively and monotonically decaying sequences.

We employ $\varepsilon-$algorithm, $\rho-$algorithm, and PGR algorithm with $z_n^{(m)} = \dfrac{1}{m+n+1}$ to accelerate the series,

$$S_m = \sum_{k=1}^{m} \frac{(-1)^{k-1}}{k} \quad \to \quad \log 2 (= 0.69314718\cdots), \tag{4.25}$$

$$S_m = \sum_{k=1}^{m} \frac{1}{k^2} \quad \to \quad \frac{\pi^2}{6} (= 1.6449336\cdots), \tag{4.26}$$

$$S_m = \sum_{k=1}^{m} \frac{(2k-1)!!}{(2k)!!(4k+1)} \quad \to \quad \int_0^1 \frac{\mathrm{d}x}{\sqrt{1-x^4}} - 1 (= 0.31102877\cdots), \tag{4.27}$$

numerical results of which are given in Figures 4.2 – 4.4.

The relation between discrete integrability and convergence acceleration has not been clarified yet. The PGR algorithm, which is considered as a nonlinear integrable mapping, does not necessarily work well as a convergence accelerator. It is interesting to remark, on the other hand, that the θ–algorithm [4],

$$\theta_{2n}^{(m)} = \theta_{2n-2}^{(m+1)} + \frac{1}{\theta_{2n-1}^{(m+1)} - \theta_{2n-1}^{(m)}} \tag{4.28}$$

$$\theta_{2n+1}^{(m)} = \theta_{2n-1}^{(m+1)} + \frac{\Delta\theta_{2n-1}^{(m+1)}\Delta\theta_{2n}^{(m+1)}}{\Delta^2\theta_{2n}^{(m)}} \tag{4.29}$$

$$\theta_0^{(m)} = 0, \qquad \theta_1^{(m)} = S_m = c_0 + c_1 + \cdots + c_m.$$

is not integrable in the sense of the singularity confinement. The algorithm, however, accelerates both alternatively and rationally decaying sequences (See Tables 4.1 and 4.2.).

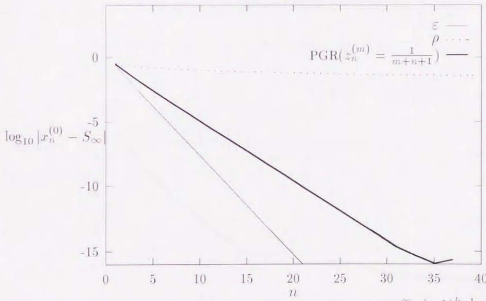


Figure 4.2: Acceleration methods for $\sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k}$

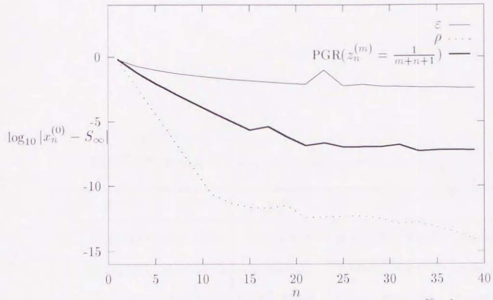


Figure 4.3: Acceleration methods for $\sum_{k=1}^{\infty} \frac{1}{k^2}$

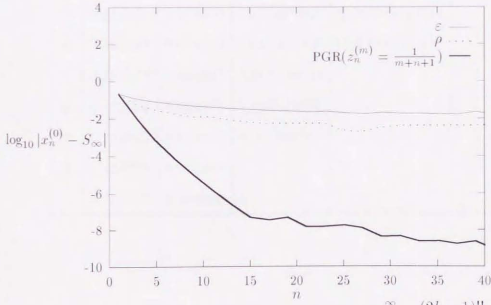


Figure 4.4: Acceleration methods for $\sum_{k=1}^{\infty} \frac{(2k-1)!!}{(2k)!!(4k+1)}$

Table 4.1: $\theta-$algorithm for the sequence $S_m = \sum_{k=1}^{m}(-1)^{k-1}/k^{1/2}$

m	$\theta_1^{(m)}$	$\theta_3^{(m)}$	$\theta_5^{(m)}$	$\theta_7^{(m)}$	$\theta_9^{(m)}$
1	1.000000	0.606153	0.604902	0.604898644927	0.604898643422
2	0.292893	0.604430	0.604899	0.604898642991	0.604898643421
3	0.870243	0.605116	0.6048989	0.604898643562	0.604898643421
4	0.370243	0.604783	0.60489854	0.604898643370	
5	0.817457	0.604966	0.60489869	0.604898643441	
6	0.409209	0.604856	0.60489862	0.604898643412	
7	0.787173	0.604927	0.604898655		
8	0.433620	0.604879	0.604898636		
9	0.766953	0.604913	0.604898647		
10	0.450725	0.604888			
11	0.752237	0.604906			

Table 4.2: $\theta-$algorithm for the sequence $S_m = \sum_{k=1}^{m} 1/k^2$

m	$\theta_1^{(m)}$	$\theta_3^{(m)}$	$\theta_5^{(m)}$	$\theta_7^{(m)}$	$\theta_9^{(m)}$
1	1.000000	1.638889	1.6449349	1.6449340822	1.6449340672
2	1.250000	1.642361	1.6449345	1.6449340703	1.6449340676
3	1.361111	1.643611	1.6449343	1.6449340678	1.6449340677
4	1.423611	1.644167	1.6449342	1.6449340673	
5	1.463611	1.644450	1.6449341	1.6449340666	
6	1.491389	1.644610	1.6449341	1.6449340689	
7	1.511797	1.644706	1.64493409		
8	1.527422	1.644768	1.64493408		
9	1.539768	1.644809	1.64493408		
10	1.549768	1.644838			
11	1.558032	1.644858			

Chapter 5

Concluding remarks

We have seen that the $\eta-$algorithm and the $\varepsilon-$algorithm are equivalent to the discrete KdV and the discrete potential KdV equations, respectively and that their performance as convergence acceleration algorithms is completely the same. The application of the $\varepsilon-$algorithm is equivalent to the Diophantine approximation for the limit $\lim_{m\to\infty} S_m$ and coefficients appearing in the continued fraction obey the time-discrete Toda molecule equation. The time evolution of the Toda molecule equation relates strongly with convergence acceleration by the $\varepsilon-$algorithm.

We have also shown that the $\rho-$algorithm is considered to be one integrable discretization of the cylindrical KdV equation. The $\varepsilon-$ and the $\rho-$ algorithms, despite their apparent similarity, possess different properties both as convergence accelerators and as discrete soliton equations. The difference in performance of these two algorithms depends on their different determinant expressions. By changing the determinant in the $\rho-$algorithm, we can naturally derive Thiele's $\rho-$algorithm, which accelerates larger class of sequences.

The PGR algorithm, quite general integrable rhombus algorithm, relates with the discrete Painlevé equation of type I besides the discrete KdV-type equations. It is an interesting problem to consider the relation between convergence acceleration and the Painlevé property.

When we apply $\varepsilon-$ and $\rho-$algorithms to a convergent sequence, odd terms converge to the same limit as the original sequence though even terms diverge. This fact agrees with the idea of the singularity confinement. It is a future problem to clarify how two different notions, acceleration and integrability, are associated with each other. In other words, we should consider whether we can construct new convergence acceleration algorithms from the other discrete soliton equations[1] and what kind of equations the other algorithms correspond to.

Why are these two areas, soliton theory and numerical analysis, related with each other even though they have different targets? The solution of these problems will shed a new light on the study of integrable systems and numerical analysis.

[1]Papageorgiou et al. have proposed a new algorithm based on the discrete modified KdV equation.

Acknowledgment

I want to thank all the people who helped me with this thesis.

First of all, I would like to thank Professor Ryogo Hirota of Waseda University and Professor Yoshimasa Nakamura of Doushisha University. It is no exaggeration to say that their very fascinating lectures are base of the research. It has been my great pleasure to have many opportunities of discussion with them. I also thank Mr. Satoshi Tsujimoto of Waseda University and Mr. Ken'ichi Maruno of Kyushu University for fruitful discussions on soliton theory, convergence acceleration, and many other things.

Thanks are also due to Professor Naoki Osada of Nagasaki Institute of Applied Science. Since I have been studying acceleration methods for only one year, my knowledge about them is very little. His lecture on acceleration methods was very instructive.

I am also grateful to the members of the "Rikigaku Kyoshitsu". Although I entered the Department of Mathematical Sciences after finishing the master course, the Rikigaku Kyoshitsu is very comfortable for me. It is due to the active and friendly atmosphere of the laboratory that I managed to write this thesis. Especially I would like to thank Professor Tetsuji Tokihiro, Dr. Tetsu Yajima, Dr. Yasuhiro Ohta, Dr. Junta Matsukidaira, Dr. Narimasa Sasa, Dr. Kenji Kajiwara, Dr. Saburo

Kakei, Dr. Katsuhiro Nishinari, Dr. Daisuke Furihata, Dr. Makoto Torii, Mr. Kai Matsui, Mr. Satoshi Moriwaki for many useful discussions and instructions in the seminar.

Finally, and most of all, I wish to give special thanks to Professor Junkichi Satsuma for many valuable instructions for these five years in spite of various duties. His motto is the best teaching is not teaching at all but creating a climate for students, which gives me all the confidence to choose my path of study.

Bibliography

[1] M. Arai, K. Okamoto, and Y. Kametaka, Proc. Japan Acad., Ser. A **62** (1986) 5. Y. Kametaka, Suugaku-seminar, Jan.−May (1986) [in Japanese].

[2] G. A. Baker Jr. and P. Graves-Morris, Padé Approximants Part I : Basic Theory, Encyclopedia of Mathematics and its Applications 13 (Addison-Wesley, Massachusetts, 1981).

[3] F. L. Bauer, in: On Numerical Approximation, ed. R. E. Langer (University of Wisconsin Press, Madison, 1959) p.361. in: Approximation of Functions, ed. H. L. Garabedian (Elsevier, Amsterdam, 1965) p. 134.

[4] C. Brezinski, Lecture notes in mathematics, Vol. 584, Accélération de la convergence and analyse numérique (Springer, Berlin, 1977).

[5] C. Brezinski, Numer. Math. **35** (1980) 175.

[6] C. Brezinski and M. Redivo Zaglia, Extrapolation Methods. Theory and Practice (North-Holland, Amsterdam, 1991).

[7] R. Bulirsh and J. Stoer, Numer. Math. **8** (1966) 1.

[8] M. T. Chu, SIAM J. Alg. Disc. Methods **5** (1984) 187.

[9] F. Cordellier, in: Lecture notes in mathematics Vol. 1071, Padé Approximation and its Applications, eds. H. Werner and H. J. Bünger (Springer, Berlin, 1984) p. 124.

[10] W. Cuyt, in: Lecture notes in mathematics Vol. 1071, Padé Approximation and its Applications, eds. H. Werner and H. J. Bünger (Springer, Berlin, 1984) p. 137.

[11] P. A. Deift, SIAM J. Numer. Anal. **20** (1983) 1.

[12] J. P. Delahaye and B. Germain-Bonne, Numer. Math. **35** (1980) 443; SIAM J. Numer. Anal. **19** (1982) 840.

[13] H. Flaschka, Phys. Rev. B **9** (1974) 1924.

[14] W. B. Gragg, SIAM Review **14** (1972) 1.

[15] B. Grammaticos, A. Ramani, and V. Papageorgiou, Phys. Rev. Lett. **67** (1991) 1825.

[16] R. Hirota, Direct Method in Soliton Theory (Iwanami, Tokyo, 1992) [in Japanese].

[17] R. Hirota, S. Tsujimoto, and T. Imai, RIMS Kokyuroku **822** (1992) 144.

[18] R. Hirota, J. Phys. Soc. Jpn. **43** (1977) 1424, **50** (1981) 3785.

[19] W. B. Jones and W. J. Thron, Continued Fractions, Analytic Theory and Applications, Encyclopedia of Mathematics and its Applications 11 (Addison-Wesley, Massachusetts, 1980).

[20] D. Levin, Int. J. Comput. Math. **B3** (1973) 371.

[21] S. Maxon and J. Viecelli, Phys. Fluids **17** (1974) 1614.

[22] L. M. Milne-Thomson, The Calculus of Finite Differences (Chelsea, New York, 1933).

[23] S. Moriguti, K. Udagawa, and S. Hitotsumatsu, Mathematical Formulae II (Iwanami, Tokyo, 1960) [in japanese].

[24] A. Nagai and J. Satsuma, J. Phys. Soc. Jpn. **64** (1995) 3669.

[25] Y. Nakamura, Japan J. Indust. Appl. Math. **9** (1992) 133, **11** (1994) 1, RIMS Kokyuroku **889** (1994) 1.

[26] N. Osada, SIAM J. Numer. Anal. **27** (1990) 178; to appear in Numer. Math.

[27] V. Papageorgiou, B. Grammaticos, and A. Ramani, Phys. Lett. A **179** (1993) 111.

[28] A. Ramani, B. Grammaticos, and J. Hietarinta, Phys. Rev. Lett. **67** (1991) 1829.

[29] M. Sato, RIMS Kokyuroku **439** (1981) 59.

[30] J. Satsuma, A. Ramani, and B. Grammaticos, Phys. Lett. A **174** (1993) 387.

[31] J. Satsuma, RIMS Kokyuroku **822** (1993) 153.

[32] D. Shanks, J. Math. Phys. **34** (1955) 1.

[33] D. A. Smith and W. F. Ford, SIAM J. Numer. Anal. **16** (1979) 223.

[34] K. Sogo, J. Phys. Soc. Jpn. **62** (1993) 1081.

[35] W. W. Symes, Physica **4**D (1982) 275.

[36] T. N. Thiele, Interpolationsrechnung (Taubner, Leibzig, 1909).

[37] M. Toda, Prog. Thoer. Phys. Suppl. **45** (1970) 174.

[38] H. Togawa, Matrix Computations (Ohm, Tokyo, 1971) [in Japanese].

[39] H. S. Wall, Analytic Theory of Continued Fractions (Chealsea, Bronx, N. Y. 1948).

[40] P. Wynn, Math. Tables Aids Comput. **10** (1956) 91.

[41] P. Wynn, Proc. Cambridge Philos. Soc. **52** (1956) 663.

Appendix A

The Toda molecule equation and matrix eigenvalue algorithms

A.1 The Toda molecule equation and the QR algorithm

We here give a simple introduction of the QR algorithm and review its relation with the Toda molecule equation [35]. In addition, we discuss how solutions for the Toda molecule equation reflect on the evolution of a given matrix in iterations of the QR algorithm.

The QR algorithm is the most popular method to find eigenvalues of a given matrix $X \in M(n, \mathbb{C})$. This is based on the QR decomposition[1],

$$X = QR, \tag{A.1}$$

where Q is a unitary matrix (or an orthogonal matrix in the case of real) and R is

[1]The QR decomposition is equivalent to Gram-Schmidt's orthogonalization.

an lower-triangular matrix with all of its diagonal entries positive. If we decompose $X = QR$ and multiply the factors in reverse order we obtain a matrix $X' = RQ$ whose eigenvalues are equal to those of the initial matrix X. If we rename the initial matrix X_0, then the algorithm is defined by the equations,

$$X_{n-1} = Q_{n-1}R_{n-1}, \quad R_{n-1}Q_{n-1} = X_n \ (n = 1, 2, \ldots). \tag{A.2}$$

Since the relation

$$X_n = Q_{n-1}^T X_{n-1} Q_{n-1}$$

holds, the matrix X_n is similar to X_{n-1} and hence by induction to X_0. It was shown that under certain restrictions the matrix X_n and its diagonal entries converge to upper-triangular matrix and eigenvalues of the initial matrix X_0, respectively as n tends to ∞.

We next explain about the QR flow. Given a fixed matrix $X_0 \in M(n, \mathbb{C})$, let $G(z)$ be an analytic function defined in Ω containing all eigenvalues of X_0. The QR flow is written in the following matrix differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}X(t) = [X(t), \Pi_1(G(X(t)))], \ X(0) = X_0 \in M(n, \mathbb{C}). \tag{A.3}$$

In eq. $(A.3)$, the operator Π_1 maps a certain matrix Y into its skew-hermite part when the matrix Y is decomposed into the direct sum of skew-hermite part and upper-triangular part. The QR flow is equivalent to the following set of equations for matrix $Q(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}Q(t) = Q(t) \cdot (\Pi_1(G(X(t)))), \ Q(0) = I, \ X(t) = Q^*(t)X_0Q(t). \tag{A.4}$$

The matrix $Q(t)$ is unitary for all $t \in \mathbb{R}$. In addition, if the matrix $\mathrm{e}^{tG(X_0)}$ has a QR decomposition as

$$e^{tG(X_0)} = Q(t)R(t), \tag{A.5}$$

then the matrix $Q(t)$ solves eq. $(A.4)$. Equation $(A.5)$ implies that initial value problem of the QR flow can be solved by using the QR decomposition.

If we put

$$X(t) = \begin{bmatrix} b_1(t) & a_1(t) & & & 0 \\ a_1(t) & b_2(t) & a_2(t) & & \\ & a_2(t) & \ddots & \ddots & \\ & & \ddots & \ddots & a_{n-1}(t) \\ 0 & & & a_{n-1}(t) & b_n(t) \end{bmatrix},$$

and

$$G(z) = -z$$

in the QR flow $(A.3)$, we obtain the Toda molecule equation in the Flaschka's form [13],

$$\begin{cases} \dot{a_n} &=& a_n(b_{n+1} - b_n) \quad (n = 1, 2, \ldots, N-1), \\ \dot{b_n} &=& 2(a_n^2 - a_{n-1}^2) \quad (n = 1, 2, \ldots, N), \\ & & a_0 = a_n = 0, \end{cases} \tag{A.6}$$

where $a_n(t), b_n(t)$ and $Q_n(t)$ in eq. (2.33) are related by

$$a_n(t) = \frac{1}{2}\exp\left[-\frac{Q_{n+1}(t) - Q_n(t)}{2}\right], \tag{A.7}$$

$$b_n(t) = \frac{1}{2}\dot{Q}(t). \tag{A.8}$$

Furthermore, the following theorem connecting the QR algorithm and the Toda molecule eq. $(A.6)$ holds;

Theorem A.1 (Symes [35])
If the matrix $\exp[-X(k)]$ *has the QR decomposition,*

$$\exp[-X(k)] = Q(k)R(k), \tag{A.9}$$

then the relation,

$$R(k)Q(k) = \exp[-X(k+1)] \tag{A.10}$$

holds.

The above theorem states that the k−th iteration of the QR algorithm is equivalent to the time evolution of the Toda molecule eq. (2.33) from $t = k$ to $t = k + 1$. This equivalence between the Toda molecule eq. (2.33) and the QR algorithm is also interpreted as follows; Since distance between two neighboring particles becomes infinite as t tends to infinity, the off-diagonal quantities $a_n(t)$ decays exponentially to zero. This is nothing but diagonalization of the initial matrix $X(0)$ in terms of numerical analysis.

A.2 The discrete Toda molecule equation and the LR algorithm

In this section, we discuss the integrable discretization of the Toda molecule equation and its relation with another matrix eigenvalue algorithm, the LR algorithm. Through dependent variable transformation,

$$V_n(t) = \exp(Q_n(t) - Q_{n+1}(t)), \tag{A.11}$$

we have

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2} \log V_n(t) &= V_{n+1}(t) - 2V_n(t) + V_{n-1}(t) \\ V_0 = V_n &= 0 \end{aligned} \tag{A.12}$$

from eq. (2.33). Introducing variable $\tau_n(t)$ defined by

$$V_n(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} \log \tau_n(t), \tag{A.13}$$

we obtain the bilinear equation,

$$\frac{\mathrm{d}^2\tau_n}{\mathrm{d}t^2}\tau_n - \left(\frac{\mathrm{d}\tau_n}{\mathrm{d}t}\right)^2 = \tau_{n+1}\tau_{n-1}. \tag{A.14}$$

The solution for eq. (*A*.14) is given in the following Hankel determinant;

$$\tau_n(t) = \begin{vmatrix} \Psi & \frac{\mathrm{d}\Psi}{\mathrm{d}t} & \cdots & \frac{\mathrm{d}^{n-1}\Psi}{\mathrm{d}t^{n-1}} \\ \frac{\mathrm{d}\Psi}{\mathrm{d}t} & \frac{\mathrm{d}^2\Psi}{\mathrm{d}t^2} & \cdots & \frac{\mathrm{d}^{n}\Psi}{\mathrm{d}t^{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{d}^{n-1}\Psi}{\mathrm{d}t^{n-1}} & \frac{\mathrm{d}^{n}\Psi}{\mathrm{d}t^{n}} & \cdots & \frac{\mathrm{d}^{2n-2}\Psi}{\mathrm{d}t^{2n-2}} \end{vmatrix} \tag{A.15}$$

where $\Psi(t)$ is defined by

$$\Psi(t) = \sum_{j=1}^{N} e^{\eta_j},\ \eta_j = p_j t + \eta_j^{(0)}. \tag{A.16}$$

With the above definition of the function $\Psi(t)$, each quantity $\tau_n(t)$ is rewritten as

$$\tau_n(t) = \sum_{i_1<i_2<\cdots<i_n} V(p_{i_1}, p_{i_2}, \cdots, p_{i_n})^2 \exp\left(\eta_{i_1} + \eta_{i_2} + \cdots + \eta_{i_n}\right), \tag{A.17}$$

where $V(p_1, p_2, \cdots, p_k)$ stands for the Vandermonde determinant,

$$V(p_1, p_2, \cdots, p_k) = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ p_1 & p_2 & \cdots & p_k \\ p_1^2 & p_2^2 & \cdots & p_k^2 \\ \vdots & \vdots & & \vdots \\ p_1^{k-1} & p_2^{k-1} & \cdots & p_k^{k-1} \end{vmatrix}. \tag{A.18}$$

In order to discretize the Toda molecule equation, we redefine the discrete analogue of $\tau_n(t)$ by

$$\tau_n(t) = \begin{vmatrix} \Phi(t) & \Phi(t+\delta) & \cdots & \Phi(t+(n-1)\delta) \\ \Phi(t+\delta) & \Phi(t+2\delta) & \cdots & \Phi(t+n\delta) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(t+(n-1)\delta) & \Phi(t+n\delta) & \cdots & \Phi(t+(2n-2)\delta) \end{vmatrix} \tag{A.19}$$

where $\Phi(t)$ is defined by

$$\Phi(t) = \sum_{j=1}^{N} c_j P_j^{t/\delta}. \tag{A.20}$$

Then we see from Jacobi's identity for determinants that $\tau_n(t)$ in eq. $(A.19)$ satisfies the following discrete bilinear equation,

$$\tau_n(t+2\delta)\tau_n(t) - \tau_n(t+\delta)^2 = \delta^2 \tau_{n+1}(t)\tau_{n-1}(t+2\delta). \tag{A.21}$$

Through dependent variable transformations,

$$\begin{aligned} I_n(t) &= \frac{\tau_{n-1}(t)\tau_n(t+\delta)}{\tau_{n-1}(t+\delta)\tau_n(t)}, & \text{(A.22)} \\ V_n(t) &= \frac{\tau_{n+1}(t)\tau_{n-1}(t+\delta)}{\tau_n(t)\tau_n(t+\delta)}, & \text{(A.23)} \end{aligned}$$

we finally obtain the discrete Toda molecule equation,

$$\begin{aligned} I_{n+1}(t)V_n(t) &= I_n(t+\delta)V_n(t+\delta), & \text{(A.24)} \\ I_n(t+\delta) - I_n(t) &= \delta^2[V_n(t) - V_{n-1}(t+\delta)]. & \text{(A.25)} \end{aligned}$$

The above equation is rewritten as

$$L(t+\delta)R(t+\delta) = R(t)L(t), \tag{A.26}$$

where $L(t)$ and $R(t)$ are matrices defined by

$$L(t) = \begin{bmatrix} 1 & & & & \text{\Large 0} \\ V_1(t) & 1 & & & \\ & V_2(t) & \ddots & & \\ & & \ddots & \ddots & \\ \text{\Large 0} & & & V_{n-1}(t) & 1 \end{bmatrix}, \tag{A.27}$$

$$R(t) = \begin{bmatrix} I_1(t) & \delta & & & \text{\Large 0} \\ & I_2(t) & \delta & & \\ & & \ddots & \ddots & \\ & & & \ddots & \delta \\ \text{\Large 0} & & & & I_n(t) \end{bmatrix}. \tag{A.28}$$

The equation $(A.26)$ is nothing but the LR algorithm, which is one of the famous matrix eigenvalue algorithms [17].

We can again see the relation between the time evolution of the Toda molecule equation and the LR algorithm. Through the iterations $(A.26)$, the matrix

$$X(t) = L(t)R(t) = \begin{bmatrix} I_1(t) & \delta & & & & \text{\Large 0} \\ V_1(t)I_2(t) & I_2(t)+\delta V_1(t) & \delta & & & \\ & V_2(t)I_3(t) & \ddots & \ddots & & \\ & & \ddots & \ddots & \delta & \\ \text{\Large 0} & & & V_{n-1}(t)I_n(t) & I_n(t)+\delta V_{n-1}(t) \end{bmatrix}. \tag{A.29}$$

is known to converge to an upper triangular matrix with diagonal entries approaching the eigenvalues of $X(t)$. This is equivalent to the time evolution of the quantities $V_n(t)$, which correspond to $\exp(-(Q_{n+1}(t) - Q_n(t)))$ and decay exponentially to zero.

Appendix B

Fundamental properties of continued fraction

This chapter deals with fundamental properties of continued fractions which have been used in this thesis (See refs. [39, 19] for details.).

B.1 Basic definitions of continued fractions

We first give a recursive definition for the continued fraction,

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots}}}, \tag{B.1}$$

or

$$b_0 + \frac{a_1 |}{| b_1} + \frac{a_2 |}{| b_2} + \frac{a_3 |}{| b_3} + \cdots, \tag{B.2}$$

or

$$b_0 + \mathbf{K}(a_i/b_i) \tag{B.3}$$

as follows. Let $\{a_n\}$ and $\{b_n\}$ be sequences of numbers with all $a_n \neq 0$ and $\{f_n\}$ be a sequence defined by

$$f_n = S_n(0), \quad n = 0, 1, 2, \cdots, \tag{B.4}$$

where

$$S_0(w) = s_0(w), \quad S_n(w) = S_{n-1}(s_n(w)) = s_0 \circ s_1 \circ \cdots \circ s_n(w) \tag{B.5}$$

$$s_0(w) = b_0 + w, \quad s_n(w) = \frac{a_n}{b_n + w}, \; n = 1, 2, 3, \cdots. \tag{B.6}$$

Then each f_n defines $n-$th convergent of the continued fraction $(B.1)$, $(B.2)$, or $(B.3)$.

Theorem B.1 *Let A_n and B_n be numerator and denominator of the $n-$th convergent f_n. Then they are determined recursively by*

$$A_{-1} = 1, A_0 = b_0, B_{-1} = 0, B_0 = 1, \tag{B.7}$$

$$A_n = b_n A_{n-1} + a_n A_{n-2}, \tag{B.8}$$

$$B_n = b_n B_{n-1} + a_n B_{n-2}. \tag{B.9}$$

B.2 Equivalence transformations and contractions

We say that two continued fractions,

$$b_0 \; + \; \mathbf{K}(a_n/b_n) \tag{B.10}$$

$$b_0^* \; + \; \mathbf{K}(a_n^*/b_n^*) \tag{B.11}$$

are said to be *equivalent* if

$$f_n = f_n^* \tag{B.12}$$

holds for any $n \geq 0$.

Theorem B.2 *Two continued fractions (B.10) and (B.11) are equivalent if and only if there exists a sequence of nonzero constants $\{r_n\}$ such that*

$$\begin{aligned} r_0 &= 1 && \text{(B.13)} \\ a_n^* &= r_n r_{n-1} a_n, \; n = 1, 2, 3, \cdots, && \text{(B.14)} \\ b_n^* &= r_n b_n, \; n = 0, 1, 2, \cdots. && \text{(B.15)} \end{aligned}$$

Next we discuss contractions and extensions of continued fractions. A continued fraction $b_0^* + \mathbf{K}(a_n^*/b_n^*)$ is said to be a *contraction* of a continued fraction $b_0 + \mathbf{K}(a_n/b_n)$ if $\{f_n^*\}$ is a subsequence of $\{f_n\}$. Inversely we say that $b_0 + \mathbf{K}(a_n/b_n)$ is an *extension* of $b_0^* + \mathbf{K}(a_n^*/b_n^*)$.

As a special case, if

$$f_n^* = f_{2n} \tag{B.16}$$

holds for any $n \geq 0$, then $b_0^* + \mathbf{K}(a_n^*/b_n^*)$ is called the *even part* of $b_0 + \mathbf{K}(a_n/b_n)$.

Theorem B.3 *A continued fraction $b_0 + \mathbf{K}(a_n/b_n)$ has an even part if and only if*

$$b_{2k} \neq 0, \; k = 1, 2, 3, \ldots. \tag{B.17}$$

If eq. (B.17) holds, then the elements of the even part $b_0^* + \mathbf{K}(a_n^*/b_n^*)$ are given (up to equivalence transformation) by

$$\begin{aligned} b_0^* &= b_0, \; a_1^* = a_1 b_2, \; b_1^* = a_2 + b_1 b_2, \; a_2^* = -a_2 a_3 b_4 && \text{(B.18)} \\ a_k^* &= -a_{2k-2} a_{2k-1} b_{2k-4} b_{2k} && \text{(B.19)} \\ b_k^* &= a_{2k-1} b_{2k} + b_{2k-1}(a_{2k} + b_{2k-1} b_{2k}) && \text{(B.20)} \end{aligned}$$

Similarly if

$$f_n^* = f_{2n+1} \tag{B.21}$$

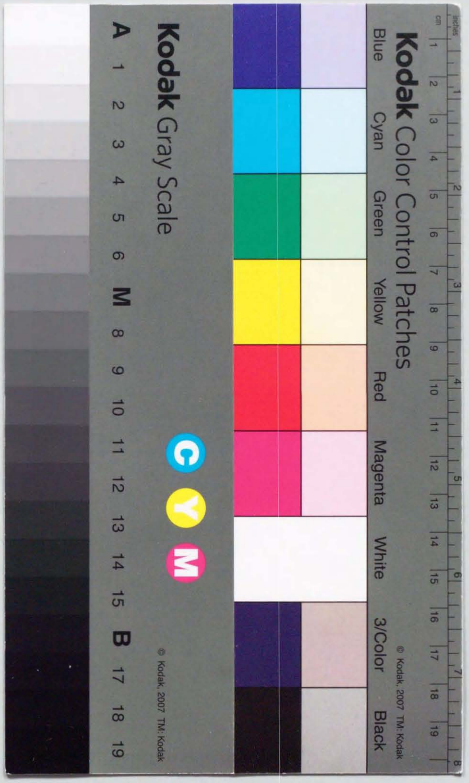
holds for any $n \geq 0$, then $b_0^* + \mathbf{K}(a_n^*/b_n^*)$ is called the *odd part* of $b_0 + \mathbf{K}(a_n/b_n)$.

Theorem B.4 *A continued fraction $b_0 + \mathbf{K}(a_n/b_n)$ has an odd part if and only if*

$$b_{2k-1} \neq 0, \ k = 1, 2, 3, \ldots. \tag{B.22}$$

If eq. (B.22) holds, then the elements of the odd part $b_0^* + \mathbf{K}(a_n^*/b_n^*)$ are given (up to equivalence transformation) by

$$\begin{aligned} b_0^* &= \frac{a_1 + b_0 b_1}{b_1}, & \text{(B.23)} \\ a_1^* &= -\frac{a_1 a_2 b_3}{b_1}, \ b_1^* = a_2 b_3 + b_1(a_3 + b_2 b_3), & \text{(B.24)} \\ a_k^* &= -a_{2k-1} a_{2k} b_{2k+1} b_{2k-3}, & \text{(B.25)} \\ b_k^* &= a_{2k} b_{2k+1} + b_{2k-1}(a_{2k+1} + b_{2k} b_{2k+1}). & \text{(B.26)} \end{aligned}$$


Kodak Color Control Patches
Blue
Cyan
Green
Yellow
Red
Magenta
White
3/Color
Black
© Kodak, 2007 TM: Kodak
Kodak Gray Scale
A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19
C Y M
© Kodak, 2007 TM: Kodak