<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>OISHI, Shin'ichi</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (1992), 787: 37-71</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1992-06</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/82605">http://hdl.handle.net/2433/82605</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>
The Self-Validating Numerical Method
—A New Tool for Computer Assisted Proofs of Nonlinear Problems—

Shin’ichi OISHI
School of Science and Engineering, Waseda University

Summary
The purpose of the present paper is to review a state of the art of nonlinear analysis with the self-validating numerical method. The self-validating numerics based method provides a tool for performing computer assisted proofs of nonlinear problems by taking the effect of rounding errors in numerical computations rigorously into account. First, Kantorovich’s approach of a posteriori error estimation method is surveyed, which is based on his convergence theorem of Newton’s method. Then, Urabe’s approach for computer assisted existence proofs is likewise discussed. Based on his convergence theorem of the simplified Newton method, he treated practical nonlinear differential equations such as the Van der Pol equation and the Duffing equation, and proved the existence of their periodic and quasi-periodic solutions by the self-validating numerics. An approach of the author for generalization and abstraction of Urabe’s method are also described to more general functional equations. Furthermore, methods for rigorous estimation of rounding errors are surveyed. Interval analytic methods are discussed. Then an approach of the author which uses rational arithmetic is reviewed. Finally, approaches for computer assisted proofs of nonlinear problems are surveyed, which are based on the self-validating numerics.

1 Introduction
Recent development of soliton theory (see, for example, Ref.[1]) reveals that exact analysis of nonlinear problems allows us to achieve a through understanding of nonlinear phenomena. In fact, soliton theory provides us a deep insight into a miraculous world of completely integrable nonlinear systems. Namely, we can write down exact solutions for many soliton equations. Such an exact solution delineates various interesting properties of solitons. One view of soliton theory involves nonlinear Fourier analysis[2]. In general, a soliton equation has a well-organized underlying algebraic structure related, for example, to infinite dimensional Lie algebras[1].

Although the number of exactly solvable interesting soliton equations exceeds one hundred and soliton equations are scattered in various fields, there remain many more nonlinear equations, which are interest but cannot be solved by the soliton theory. Thus, tools are desired for the exact analysis of such nonlinear equations. Since such equations have, in general, poor algebraic structures, we must use topological methods such as functional analysis combined with algebraic analysis. This is the philosophy of Poincaré. Moreover, in order to obtain a concrete
result, we must also use a computer as an assistant. Thus, a kind of algorithmic functional analysis is needed in this area.

Fortunately, it has recently become clear that computer-assisted proofs of various kinds of nonlinear problems can be performed by validating the accuracy of the numerical calculations[3]-[8]. Here, the concept "validating the accuracy of the numerical calculations" means that it is necessary to consider the effect of rounding errors of numerical computations. A key role in computer-assisted proofs using self-validating numerics is played by the Newton method. The truly pioneering work of Kantorovich[9, 10] shows that by the numerical proof the sufficient conditions for the convergence of the Newton method, proofs of the existence and the local uniqueness of solutions for a wide class of nonlinear functional equations can be done by computer. However, since exact analysis of rounding errors of numerical calculations was considered to be extremely difficult, such an approach was thought to be too restrictive. Recent advances in the study of machine interval analysis break through these difficulties and show that with reasonable effort, one can completely remove the effects of rounding errors. In fact, recently, several programming languages which support machine interval analysis have been developed such as FORTRAN-SC, ACRITH-XSC, PASCAL-(X)SC, and ACRIMOTH, and many computer-assisted proofs of nonlinear problems have been conducted with self-validating numerics [3]-[8], [11]-[16].

The purpose of the present paper is to review assess the current state of research in computer-assisted proofs for nonlinear problems using the self-validating numerics.

2 The Newton Method and Kantorovich's Convergence Theorem

Since, in computer assisted proofs using self-validating numerics, the Newton method plays a fundamentally important role, this paper begins with a review of the Newton method. For theoretical background, see for example Ref.[10] and for historical remarks, see Refs.[18]-[20].

Let $f$ be a continuously differentiable map from an open set $B$ of a Banach space $X$ into another B-space $Y$. We are concerned with the problem of finding a zero of $f$:

$$f(x) = 0.$$  \hspace{1cm} (1)

For present purposes, we take any element $x_0 \in B$. If $f(x_0) \neq 0$, $x_0$ should be updated by

$$x_1 = x_0 + \Delta x.$$  \hspace{1cm} (2)

Substituting this into the r.h.s of Eq.(1), we have

$$f(x) = f(x_0) + f'(x_0)\Delta x + o(\Delta x),$$ \hspace{1cm} (3)

where $o(\Delta x)$ is a higher-order infinitesimal of $\Delta x$. Thus by approximating $f(x)$ by $f(x_0) + f'(x_0)\Delta x$, we have

$$f(x_0) + f'(x_0)\Delta x = 0,$$ \hspace{1cm} (4)
which yields

\[ x_1 = x_0 - [f'(x_0)]^{-1}f(x_0), \tag{5} \]

provided that \([f'(x_0)]^{-1}\) exists and \(x_1\) remains inside of \(B\). By repeating this process, we have the recursion formula:

\[ x_{n+1} = x_n - [f'(x_n)]^{-1}f(x_n). \tag{6} \]

This process of forming the sequence \(\{x_n\}\) is called the Newton method.

If the operator \([f'(x_n)]^{-1}\) is approximated by a linear operator \(F^{-1}\), then the modified Newton method is obtained, in which a sequence is calculated by the formula:

\[ x_{n+1} = x_n - F^{-1}f(x_n). \tag{7} \]

The (modified) Newton method is known to be very powerful in solving nonlinear equations\[10\]. In order to demonstrate this efficacy with nonlinear equations, it is useful to consider examples.

For this purpose, we consider the following simple problem of obtaining the square root of a positive number:

**Example 2.1 (The Newton Method for Obtaining a Square Root)** Let us consider the problem of obtaining the square root of a positive rational number \(c\). Since if \(c = 4^ma\) then \(\sqrt{c} = 2^m\sqrt{a}\), we may assume that \(\frac{1}{4} < a < 1\) and seek a value for \(\sqrt{a}\). To obtain \(\sqrt{a}\), we consider to solve

\[ f(x) = x^2 - a = 0. \tag{8} \]

In this case, Eq.(6) becomes as

\[ x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right). \tag{9} \]

We begin this iteration from \(x_0 = 1\). It is easily seen that \(x_n\) makes a monotonically decreasing sequence. Moreover it is easily found that if \(y_n\) is a sequence generated by

\[ y_{n+1} = \frac{1}{2} \left( y_n + \frac{b}{y_n} \right), \quad y_0 = 1 \tag{10} \]

for \(\frac{1}{4} < b < a < 1\), then \(|\sqrt{a} - x_n| < |\sqrt{b} - y_n|\) holds true. In this case, the sequence \(y_n\) is said to majorize the sequence \(x_n\). Thus for any \(a \in \left(\frac{1}{4}, 1\right)\), \(|\sqrt{a} - x_n| < |\sqrt{\frac{1}{4}} - t_n|\) holds true. Here, \(t_n\) is a sequence obtained by Eq.(10) with \(b = \frac{1}{4}\). The first few \(t_n\)'s are given by

\[
\begin{align*}
t_0 &= 1, \\
t_1 &= \frac{5}{8}, \\
t_2 &= \frac{41}{80}, \\
t_3 &= 3281, \\
t_4 &= 6560', \\
t_5 &= 21523361, \\
t_6 &= 43046720', \\
t_7 &= 926510094425921 \\
t_8 &= 185302018851840', \\
t_9 &= 1716814910146256242328924544641 \\
t_{10} &= 3433683820292512484657849089280' \\
\cdots
\end{align*}
\]
From this we can conjecture that if we let \( t_n = \frac{s_n}{r_n} \), \( s_n \) and \( r_n \) being mutually prime integers, \( r_n = 2(s_n - 1) \). In fact, a computer experiment shows that this relation holds for at least \( n \leq 12 \) and by mathematical induction, we have

\[
t_{n+1} = \frac{2s_n(s_n - 1) + 1}{4s_n(s_n - 1)}, \quad s_1 = 5. 
\] (12)

Thus we have proven that \( t_n \) has the form \( t_n = \frac{s_n}{2(s_n - 1)} \) and \( s_n \) is generated by

\[
s_{n+1} = 2s_n(s_n - 1) + 1, \quad s_1 = 5. 
\] (13)

This implies that

\[
\left| \sqrt{\frac{1}{4}} - t_n \right| = \frac{1}{2(s_n - 1)} = \delta_n. 
\] (14)

Finally we have \( |\sqrt{a} - x_n| < \left| \sqrt{\frac{1}{4}} - t_n \right| = \delta_n. \) This gives an exact error estimate.

By similar observations, we can find that if \( y_n \) is generated by

\[
y_{n+1} = \frac{1}{2} \left( y_n + \frac{b}{y_n} \right), \quad y_0 = 1 
\] (15)

with \( b = \frac{(m-1)^2}{m^2} \), we have

\[
y_{n+1} = \frac{2u_n(u_n - 1) + 1}{m^2(2u_n(2m - 1) + 1)}, \quad u_1 = 2m(m-1) + 1. 
\] (16)

Thus if \( a \geq \frac{(m-1)^2}{m^2} \), we have a more precise error estimate as

\[
|\sqrt{a} - x_n| \leq \frac{(m-1)}{m(u_n - 1)}, 
\] (17)

where \( u_n \) is generated by

\[
u_{n+1} = 2u_n(u_n - 1), \quad u_1 = 2m(m-1) + 1. 
\] (18)

\[
\square
\]

In this example, exact error estimates are given between the exact solution \( \sqrt{a} \) and its approximations \( x_n \)'s obtained by the Newton method using particular properties of the problem. In order to prove the convergence of the Newton method in more general situations, Kantorovich[10] and Kantorovich and Akilov[17] considered a general iteration process

\[
x_{n+1} = S(x_n), 
\] (19)

where \( S \) is a \( C^1 \)-map defined in the sphere \( \|x - x_0\| < R \) of some B-space \( X(x_0 \in X) \). Along with Eq.(19), he considered a real equation

\[
t_{n+1} = g(t_n), 
\] (20)

where \( g \) is a \( C^1 \)-map defined in the interval \([t_0, t']\) \( (t' = t_0 + r < t_0 + R) \). The function \( g \) is said to majorize the operator \( S \) if
(1) \[ \|S(x_0) - x_0\| \leq g(t_0) - t_0, \] (21)

(2) \[ \|S'(x)\| \leq g'(t) \text{ whenever } \|x - x_0\| \leq t - t_0. \] (22)

**Theorem 2.1 (Kantorovich[10] and Kantorovich and Akilov[17])** In the above-mentioned situation, if \( g \) majorizes \( S \) and if \( t = g(t) \) has a root in \([t_0, t']\), then the equation \( x = S(x) \) also has a solution \( x^* \), to which the sequence \( \{x_n\} \) starting from \( x_0 \) is convergent. Also, \[ \|x^* - x_n\| \leq t^* - t_n, \] (25)

where \( t^* \) denotes the least root of the equation \( t = g(t) \).

Using this theorem, Kantorovich[10] proved the following famous convergence theorem.

**Theorem 2.2 (Kantorovich[10] and Kantorovich and Akilov[17])** Let \( B = \{x||x - x_0\| \leq r\} \) and \( f \) is in \( C^2 \) on \( B \). Moreover, let

(1) the linear operator \( L = [f'(x_0)]^{-1} \) exist;

(2) \[ \|Lf(x_0)\| \leq c; \] (26)

(3) \[ \|Lf''(x)\| \leq K \quad (x \in B). \] (27)

Now, if \( h = cK < \frac{1}{2} \) and

\[ r \geq r_0 = \frac{1 - \sqrt{1 - 2h}}{h} c \] (29)

hold, Eq.(1) has a solution \( x^* \) to which both the original Newton method

\[ x_{n+1} = x_n - [f'(x_n)]^{-1}f(x_n) \] (30)

and the simplified Newton method

\[ x_{n+1} = x_n - [f'(x_0)]^{-1}f(x_n) \] (31)

are convergent and \( \|x^* - x_0\| \leq r_0 \) holds. Furthermore, if for \( h < \frac{1}{2} \)

\[ r < r_1 = \frac{1 + \sqrt{1 - 2h}}{h} c, \] (32)
the solution $x^*$ is unique in $B$.

The speed of convergence of (30) is characterized by

$$\|x^* - x_n\| \leq \frac{1}{2^n} (2h)^{2^n} \frac{c}{h} \quad (n = 0, 1, \ldots)$$

and that of (31), for $h < \frac{1}{2}$, by

$$\|x^* - x_n\| \leq \frac{c}{h} (1 - \sqrt{1 - 2h})^{n+1} \quad (n = 0, 1, \ldots).$$

Remark 2.1 The conditions $f \in C^2$ and (3) can be replaced by $f \in C^1$ and

(3')

$$\|f'(x) - f'(y)\| \leq \alpha \|x - y\| \quad \text{for any } x, y \in B.$$  

In this case $K = \|L\|\alpha$. This was done by Feny[21]. Moreover, various extensions of this theorem have been presented. See for example, Ortega and Rheinboldt[22]. Sharp error bounds are obtained by several authors. See for example Refs.[23]-[27].

Moreover Kantorovich and Akilov[17] considered a special equation written by

$$f(x) = p(x) + q(x) = 0.$$ 

Let $x_0$ be an approximate solution of

$$p(x) = 0.$$ 

He showed that if the following conditions are satisfied

(1)

$$\|\left[p'(x_0)\right]^{-1}f(x_0)\| \leq c,$$ 

(2)

$$\|\left[p'(x_0)\right]^{-1}f'(x_0)\| \leq d < 1,$$ 

(3)

$$\|\left[p'(x_0)\right]^{-1}f''(x)\| \leq K \quad (x \in B),$$ 

and if $h = \frac{cK}{(1-d)\gamma} < 2^{-1}$ and $r \geq r_0 = \frac{(1-\sqrt{1-2h})c}{h(1-d)}$, then Eq.(36) has a solution in $B$.

Using the theorem 2.2 and its extensions, in Ref.[17], Kantorovich and Akilov presented the following examples of inclusions for exact solutions to functional equations:

(1) A single real and complex equation;

(2) A system of algebraic equations; in particular, they give the an example

$$3x_1^2x_2 + x_2^2 = 1,$$

$$x_1^4 + x_1x_2^3 = 1.$$ 

They showed an inclusion of an exact solution as

$$0.991173 \leq x_1^* \leq 0.991205; \quad 0.327366 \leq x_2^* \leq 0.327398;$$
(3) A nonlinear integral equation of the form
\[ x(s) = \int_{0}^{1} K(s, t, x(t))dt. \]  
Specifically, they considered the case of \( K(s, t, u) = \frac{u^{2}\sin st}{2} \) and showed an inclusion
\[ |x^*(s) - (1 + 0.38617s - 0.0345s^{3})| < 0.0119 \quad (s \in [0, 1]); \]  
(4) An initial value problem of a differential equation
\[ x'(t) - g(x(t), t) = 0, \quad x(0) = 0 \]  
provided that \( g(u, t) \) is continuous and is \( C^2 \) with respect to \( u; \)
(5) Periodic solution of the differential equation
\[ x''(t) + x(t) + \mu g(x(t), x'(t), t) = 0, \]  
where \( g(u, v, t) \) is continuous and is \( C^2 \) with respect to \( u \) and \( v \), and is periodic in \( t \) with period \( k > 0; \)
(6) An eigenvalue problem of the operator \( U_t = U + tV \), where \( U \) and \( V \) are linear operators from a Banach space \( X \) into itself, provided that an eigenvalue and eigenfunction of \( U \) are known;
(7) A certain boundary value problem of a second order quasilinear differential equation with two independent variables.

Examples of existence proofs based on Kantorovich's theorem up to 1967 can be found in Ref.[10, p.723, 749] and Ref.[28, p.138]. Refs.[29]-[39] also give examples.

In 1969, Rall[40] published a beautiful introductory text of the Newton method and its applications. In this book, techniques of the interval analysis initiated by Moore[41] and automatic differentiations are supplemented to the points mentioned above. Since the interval analysis is the topic of another section of the paper, we note here only that this method is based on the doctoral thesis of Moore[41]. Detailed bibliographies can be found in Ref.[42]:

In his book[40], Rall presented a method of automatically implementing the Newton method by making use of automatic differentiations. As an example, he treated the following examples:

(1) A system of algebraic equations; specifically, he gave the an example
\[ 16x_1^4 + 16x_2^4 + x_3^4 - 16 = 0, \]
\[ x_1^2 + x_2^2 + x_3^2 - 3 = 0, \]
\[ x_1^3 - x_2 = 0. \]  
He showed an inclusion of an exact solution as
\[ x(1) = \left( \frac{223}{224}, \frac{63}{80}, \frac{79}{60} \right) \quad ||x^* - x(1)|| \leq 1.98526343 \times 10^{-4}, \]  
where \( x(1) \) is obtained from an initial approximation \( x(0) = (1, 1, 1) \) by applying the Newton iteration once.
(2) An initial value problem of an ordinary differential equation.

(3) Two-point boundary value problem of the ordinary differential equation. In particular, he considered

$$x''(t) - g(t, x) = 0, \quad x(0) = x(1) = 0,$$

provided that $g(t, x)$ is continuous and is $C^2$ with respect to $x$ on $[0, 1]$. Since this problem demonstrates a typical application of the Newton method to functional equations, we follow Rall’s example: Let

$$f(x) = \frac{d^2x}{dt^2} - g(t, x).$$

In this case, the Newton iteration becomes

$$x_{m+1} = x_m + u_m,$$

where $u_m$ is a solution of the linear boundary value problem

$$u''_m - g_x(t, x_m)u_m = -f(x_m), \quad u_m(0) = u_m(1) = 0.$$

Using the Green function $G(t, s)$ of the linear differential operator $\frac{d^2}{dx^2}$ with the boundary condition $x(0) = x(1) = 0$,

$$G(t, s) = \begin{cases} s(t-1)t(s-1) & 0 \leq s \leq t \\ t(s-1) & t \leq s \leq 1, \end{cases}$$

we can transform Eq.(52) into the following Fredholm type integral equation

$$u_m(t) - \int_0^1 G(t, s)g_x(s, x_m(s))u_m(s)ds = -\int_0^1 G(t, s)f(x_m(s))ds.$$  (54)

Now let us consider the specific example of $g(t, x) = tx^2 - 1$. In this case, if we take $x_0 = \frac{x(1-x)}{2}$, then Eq.(54) becomes

$$u_0(t) - \int_0^1 G(t, s)s^2(1-s)u_0(s)ds = \frac{s^7}{42} - \frac{s^6}{15} + \frac{s^5}{20} - \frac{s}{140}.  \quad (55)$$

If we consider the linear integral operator with the kernel $G(t, s)s^2(1-s)$ as a map from $C[0, 1]$ to $C^2[0, 1]$, then we have a bound $\|K\| \leq \frac{4}{27}$, provided that the norm of $C^2[0, 1]$ is given by

$$\|x\| = \max\{\|x\|_{\infty}, \|x\|_{\infty}, \|x\|_{\infty}\}.  \quad (56)$$

Then a solution of Eq.(55) can be obtained by the Neumann series expansion and we have an estimate

$$\|u_0(t)\| \leq \frac{1}{1 - \|K\|} \|s^7 - \frac{s^6}{15} + \frac{s^5}{20} - \frac{s}{140}\| \leq \frac{27}{23} \frac{31}{42}  \quad (57)$$

Moreover we have

$$\|f'(x_0)\| \leq \frac{1}{1 - \|K\|} = \frac{27}{23},  \quad (58)$$

and $\|f''(x_0)\| = 2xI_2 \leq 2$. Thus we have $h \leq 0.20532 < 0.5$ where $h$ is a constant in Theorem 2.2 so that it becomes evident that an exact solution exists near $x_0 = \frac{x(1-x)}{2}$.  

Similar discussions are also given by Moore [43]. In this paper, illustrative examples given in an unpublished paper by Talbot are presented:

(a) \[ x''(t) = (x + a)(x + a - 2)[1 + 2t^2(x + a - 1)], \quad x(-1) = x(1) = 0; \quad (59) \]

(b) \[ x''(t) = \exp(x(t)), \quad x(-1) = x(1) = 0; \quad (60) \]

(c) \[ x''(t) = \exp(-x(t)), \quad x(-1) = x(1) = 0 \text{ (this problem has two solutions)}. \quad (61) \]

In Ref. [44], the two-point boundary value problem

\[ x''(t) + g(t, x', x'') = 0, \quad x(a) = x(b) = 0 \quad (62) \]

is considered and a method is given for calculating Kantorovich's constants. A more general two-point boundary value problem

\[ x''(t) = g(t, x(t)), \quad B_1y(a) + B_2y(b) = w, \quad (63) \]

where \( g : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n, g \in C^2, B_1, B_2 \) are matrices and \( w \in \mathbb{R}^n \), is considered by Kedem [45]. As examples, he considered

(a) \[ \varepsilon x'' = (x^2 - (t - 1)^2)x', \quad x(0) = A, x(1) = B; \quad (64) \]

(b) \[ h'''' + hh'' + gg' = 0, g'' + hg' - h'g = 0, h(0) = h(1) = h'(0) = h'(1) = 0, g(0) = \Omega_0, g(1) = \Omega_1. \quad (65) \]

See also Ref. [46]. Applications to control and oscillation theory is presented in Ref. [47]. Recently, two different approaches have been given by Nakao [48] and Plum [49]. We will discuss their approaches later.

3 Simplified Newton Method and Urabe’s Convergence Theorem

3.1 Historical Development

In Ref. [50], Cesari discussed the existence analysis concerning solutions of linear and nonlinear equations \( Kx = y \) in function spaces. His method is related to Galerkin’s method and reduces the problem to the study of a finite system of transcendental determining equations in a finite-dimensional Euclidean space. He discussed a process which may provide an answer to two questions: (1) If a certain \( m \)-th approximation \( x(m) \) is known, is it possible to argue whether an exact solution \( X \) also exists? (2) If the answer to (1) is affirmative, is it possible to
give an upper estimate for the difference \(X - x(m)\) (error bound)? He reduced these problems to Banach's contraction mapping principle or Schauder's fixed point theorem. He treated as an example the nonlinear ordinary differential equation

\[
x'' + x + ax^3 = \beta t \quad (0 \leq t \leq 1)
\]  
(66)

with homogeneous boundary conditions \(x(0) = 0, x'(1) + hx(1) = 0\). He showed, for example, that in case of \(h = 1\) and \(\alpha = \beta = \frac{1}{2}\), an exact solution \(x(t)\) of Eq. (66) exists in the neighborhood of the first Galerkin approximation \(x(1) = 0.11873 \sin 2.0288t\) as \(\|x(t) + x(1)\| \leq d = 0.0038\).

In Ref.[51], Cesari treated as an example the following nonlinear ordinary differential equation

\[
x'' + x^3 = \sin t,
\]  
(67)

and showed that it has a periodic solution of period \(2\pi\), \(X(t)\), in the neighborhood of an approximate solution \(x(t) = 1.434 \sin t - 0.124 \sin 3t\) as \(\|X(t) - x(t)\| \leq d = 0.124\).

Knobloch presented a remark on Cesari's work[52] and gave another example of computer-assisted existence proof of periodic solutions of a nonlinear ordinary differential equation of the second order[53].

In 1965, Urabe[54] considered periodic nonlinear differential systems

\[
x' = X(x, t),
\]  
(68)

where \(x\) and \(X\) are vectors of the same dimension and \(X(x, t)\) is smooth. He has proved that if an isolated periodic solution \(x(t)\) of Eq. (68) exists in a suitable bounded region, then \(x(t)\) can always be approximated by means of the Galerkin process. Then, he presented a convergence theorem of the simplified Newton method. Using this, he further showed that if the conditions of his convergence theorem are met at a known Galerkin approximation \(x_m(t)\), an exact isolated periodic solution \(x(t)\) can be proven to exist in the neighborhood of \(x_m(t)\), and an error bound for \(x(t) - x_m(t)\) can be determined. More precisely, to determine a periodic solution of Eq. (68) he considered the trigonometric polynomial

\[
x_m(t) = a_0 + \sum_{n=1}^{m} (a_{2n-1} \sin nt + a_{2n} \cos nt).
\]  
(69)

Substituting Eq. (69) into Eq. (68), he obtained transcendental nonlinear equations for undetermined coefficients \(a_0, a_1, a_2, \ldots, a_{2m-1}, a_{2m}\). This procedure is nothing but the well-known Galerkin procedure. He proved

**Theorem 3.1 (Urabe[54])** Let \(X(x, t)\) and its derivatives with respect to the \(x\)-coordinates be continuously differentiable with respect to the same \(x\)-coordinates and \(t\) in the region \(D \times L\), where \(D\) is a closed bounded region of the \(x\)-space and \(L\) is the real line. If there is an isolated periodic solution \(x = x(t)\) of Eq. (68) lying inside \(D\), then there exists a Galerkin approximation \(x = x_m(t)\) for any order \(m \geq m_0\) lying in \(D\) provided \(m_0\) is sufficiently large. Such Galerkin approximations \(x = x_m(t)\) converge uniformly as \(m \to \infty\) to the initial exact solution \(x = \hat{x}(t)\) together with their first-order derivatives.
As an example, in Ref.[54] he treated
\[ x'' + 1.52x + (x - 1.5 \sin t)^3 = 2 \sin t. \]  
(70)
He showed that in the neighborhood of the 3rd-order Galerkin approximation,
\[ x = x^\#(t) = 1.5994l \sin t - 0.00004 \sin 3t, \]  
(71)
there exists an exact periodic solution \( \hat{x}(t) \) of Eq.(70) which satisfies
\[ ||\hat{x}(t) - x^\#(t)|| < 0.000141. \]  
(72)
In the succeeding paper[55], Urabe and Reiter treated the Van der Pol equation with a harmonic forcing term
\[ x'' - \varepsilon(1 - x^2)x' + x = \varepsilon E \sin \omega t. \]  
(73)
They calculated the 15th Galerkin approximation and showed that in its neighborhood there is an exact periodic solution. This equation may be the first practical equation treated by self-validating numerics. Then, his method, which is now called Urabe's method, has been applied for the purpose of numerical analysis of periodic solutions of many nonlinear periodic systems[56, 57, 58, 59]. In Ref.[58], in order to solve a determining nonlinear equation, Shinohara developed a geometrical method, which is a kind of continuation method. Related problems are also treated by Refs.[56]-[73].

Moreover, Urabe's method is extended to nonlinear autonomous systems[74]-[77] and to numerical analysis of quasi-periodic solutions of quasi-periodic differential systems [78]-[80]. In the same philosophy, Urabe developed a theory for the method of computing solutions of the multi-point boundary value problem of ordinary differential equations[81].

Componentwise error estimates for approximate solutions of nonlinear equations are discussed in Refs.[82, 83]. An application is discussed to control problems in Refs.[84]-[87].

Similar approaches of Urabe are presented to include closed orbits of chaotic nonlinear differential equations in Refs.[88, 89, 90].

3.2 Generalization and Abstraction—Infinite Dimensional Homotopy Method—

In this section we would like to point out that Urabe's method can be extended to more general functional equations. In general, it is well known that nonlinear problems require Banach space formalism, and we consider the problem of finding a solution of
\[ f(x) = y, \]  
(74)
where \( f \) is a continuous map from a suitable B-space \( X \) into another B-space \( Y \). We would like to point out that operator equations solvable by Galerkin's method can be abstracted and generalized as A–proper operator equations, whose notion was developed by Petryshyn[91]. First we show that Theorem 3.1 can be generalized in the context of A–proper operator theory. This is achieved by using the infinite dimensional homotopy method developed by the author and his coworkers[92]-[97].

The A–proper operator is defined through a projection scheme[91]:
**Definition 3.1** [91] Let $X$ be a B-space and $\{x_n\}$ be a sequence of finite dimensional subspaces of $X$. Moreover, let $P_n : X \rightarrow X_n$ be a linear continuous projection operator. If for any $x \in X P_n x \rightarrow x$ holds true as $n \rightarrow \infty$, then $X$ is called a B-space with a projection scheme $\Pi = \{x_n, P_n\}$.

It is known[91] that there are various kinds of projection schemes corresponding to, for example, a difference scheme, Galerkin's scheme and so on.

**Definition 3.2** [91] Let $X$ and $Y$ be B-spaces with projection schemes $\{X_n, P_n\}$ and $\{Y_n, Q_n\}$, respectively. If for any $n > 0 \dim X_n = \dim Y_n$ holds true, $\Gamma = \{X_n, P_n; Y_n, Q_n\}$ is called an operator projection scheme.

**Definition 3.3** [91] Let $X$ and $Y$ be B-spaces having an operator projection scheme $\Gamma = \{X_n, P_n; Y_n, Q_n\}$. Let $D$ be an open set in $X$. An operator $f : cl(D) \rightarrow Y$ is $A$-proper iff the following holds true: $Q_m f$ is continuous and for any bounded infinite sequence $\{x_m\} \subset D$ $(x_m \in D_m = D \cap X_m)$ satisfying $Q_m f(x_m) \rightarrow y (m \rightarrow \infty)$ there exist a subsequence $\{x_{m_j}\}$ of $\{x_m\}$ and $x$ such that $x_{m_j} \rightarrow x$ as $j \rightarrow \infty$ and $f(x) = y$ hold true.

**Example 3.1** [91]

(1) Let $\Pi = \{X_n, P_n\}$ be a projection scheme satisfying $\|P_n\| = 1$. Then, if $f : X \rightarrow X$ is a ball condensing operator, $I - f$ becomes an $A$-proper operator with respect to $\Pi$.

(2) There are many operators in a class of monotone operators which become $A$-proper operators.

The concept of $A$-proper homotopy plays an important role. This concept is introduced by Makino and the present author[92].

**Definition 3.4** Let $X$ and $Y$ be B-spaces having an operator projection scheme $\Gamma = \{X_n, P_n; Y_n, Q_n\}$. Let $D$ be an open set in $X$. A homotopy $h : cl(D) \times [0, 1] \rightarrow Y$ is called an $A$-proper homotopy with respect to $\Gamma$ iff the following holds true: $Q_m h$ is continuous. For any $t_m \rightarrow t$ $(t_m \in [0, 1])$ and for any bounded infinite sequence $\{x_m\} \subset D$ $(x_m \in D_m = D \cap X_m)$ satisfying $Q_m h(x_m, t_m) \rightarrow y (m \rightarrow \infty)$, there exists a subsequence $\{x_{m_j}\}$ of $\{x_m\}$ and $x$ such that $x_{m_j} \rightarrow x$ as $j \rightarrow \infty$ and $h(x, t) = y$ hold true.

**Example 3.2** (1) ($A$-properness of the fixed point homotopy)[91] Let $X$ be a B-space with a projection scheme $\Pi = \{X_n, P_n\}$ and $D$ be a bounded open set in $X$. If $f(cl(D))$ is bounded and the fixed point homotopy $h(x, t) = (1-t)(x-x)+tf(x), (x, t) \in cl(D) \times [0, 1]$, is $A$-proper for each fixed $t \in [0, 1]$ with respect to $\Pi$, then $h$ is $A$-proper with respect to $\Pi$.

(2) ($A$-properness of the odd homotopy)[92] Let $X$ be a B-space with a projection scheme $\{X_n, P_n\}$ and $D \subset X$ be a bounded open set symmetric with respect to the origin. If $f(cl(D))$ is bounded and the odd homotopy

\[ h(x, t) = \frac{(1-t)(f(x) - f(-x))}{2} + tf(x) + (1-t)y, \quad (x, t) \in cl(D) \times [0, 1], \tag{75} \]
is A-proper for each fixed \( t \in [0,1] \) with respect to \( \Pi \), then \( h \) is A-proper with respect to \( \Pi \).

We are now in a position to state the main theorem of this subsection:

**Theorem 3.2** Let \( X \) and \( Y \) be B-spaces having an operator projection scheme \( \Gamma = \{ X_n, P_n; Y_n, Q_n \} \), and \( D \) be an open set in \( X \). Moreover, let \( f : cl(D) \to Y \) be an A-proper operator and \( h(x, t) : cl(D) \times [0,1] \to Y \) be an A-proper homotopy. If \( h \) satisfies the following conditions, then at least a solution of \( f(x) = y \) can be numerically obtained: (1) \( h \) is continuous with respect to \( t \) and satisfies \( h(x, 0) = g(x) \) and \( h(x, 1) = f(x) \). (2) For each \( n \), \( Q_n y \) is a regular value of \( Q_n g(x) \) and \( Q_n g(x) = Q_n y \) has an odd number of solutions in \( D_n = D \cap X \). Moreover, we assume that we can obtain a solution \( x_0n \) of \( Q_n g(x) = Q_n y \), which is not connected with other solutions of \( Q_n g(x) = Q_n y \) by the solution curve of \( Q_n h(x, t) = Q_n y \). (3) \( h(x,t) \neq y \) on \( \partial D \times [0,1] \).

(Proof) Without loss of generality we can assume \( Q_n y \) is a regular value of \( Q_n h(x,t) \). Thus, the solution set of \( Q_n h(x,t) = Q_n y \) consists of disjoint one-dimensional manifolds. From condition (2) there exists at least one solution curve starting from a solution of \( Q_n g(x) = Q_n y \) and reaching the \( t = 1 \) plane or \( \partial D \times (0,1) \) so that there exists \( x_n(t_n) \) satisfying

\[
Q_n h(x_n(t_n), t_n) = Q_n y \quad \text{on } D_n \times \{1\} \text{ or } \partial D_n \times (0,1).
\]  

(76)

In fact, starting from \( (x_0n, t = 0) \) by tracing the solution curve of \( Q_n h(x, t) = Q_n y \) numerically, we can obtain \( (x_n, t_n) \). From the boundedness of \( \{t_n\} \subset (0,1) \), it follows that there exists a subsequence \( \{t_{n_j}\} \) such that \( t_{n_j} \to t^* \in [0,1] \) as \( j \to \infty \). Since \( h \) is an A-proper homotopy, there exists a subsequence \( \{x_{n_j}\} \) of \( \{x_{n_j}\} \) and \( x^* \) such that \( x_{n_j} \to x^* \), \( h(x^*, t^*) = y \) as \( m \to \infty \). If we assume \( t^* < 1 \), it follows that \( (x^*, t^*) \) lies on \( \partial D \times [0,1] \), which contradicts condition (3). Thus, it becomes evident that \( t^* = 1 \) so that \( x^* \) is a solution of \( f(x) = y \).

**Corollary 3.1** Let \( X \) be a B-space with a projection scheme \( \Pi = \{ X_n, P_n \} \), \( D \) is an open bounded set including the origin and \( f : X \to X \). If the fixed point homotopy \( h(x,t) = (1-t)x + tf(x) \) satisfies the following conditions, then the solution to \( f(x) = 0 \) can be obtained numerically: (1) For each \( t \in [0,1] \) \( h \) is A-proper with respect to \( \Pi \), and \( f(cl(D)) \) is bounded. (2) \( h(x,t) \neq y \) on \( \partial D \times [0,1] \).

(Proof) From Example 3.2, \( h \) becomes an A-proper homotopy with respect to \( \Pi \). Since \( P_n h(x,t) = P_n 0 \) has a unique solution, it is easy to see that the conditions of Th.3.2 hold true.

**Corollary 3.2** Let \( X \) be a B-space with a projection scheme \( \Pi = \{ X_n, P_n \} \), \( D \) be an open bounded set symmetric with respect to the origin, and \( f : X \to X \). If the odd homotopy \( h \) satisfies the following conditions, then the solution to \( f(x) = y \) can be obtained numerically: (1) \( h(x,t) = (1-t)0.5(f(x) - f(-x)) + tf(x) + (1-t)y \) is A-proper with respect to \( \Pi \) for each fixed \( t \in [0,1] \) and \( f(cl(D)) \) is bounded. (2) \( h(x,t) \neq y \) on \( \partial D \times [0,1] \). (3) For each \( n \), \( P_n y \in D_n \) is a regular value of \( P_n h(x,0) \).

(Proof) From Example 3.2, it follows that \( h \) is an A-proper homotopy. Moreover \( P_n 0 \) is a trivial solution of \( P_n h(x,0) = 0 \). If the solution curve starting from \( (P_n 0, t = 0) \) does not return to the \( t = 0 \) plane, then all the conditions of Th.3.2 are satisfied. Even if this solution curve returns
to the $t = 0$ plane at $(x_{0n}, t = 0)$, from the oddness of the homotopy, $-(x_{0n}, t = 0)$ is also a solution. Thus, starting from this point we can restart the curve tracing. If the solution curve returns to the $t = 0$ plane again, the process is repeated. Since $P_{n}h(x, 0) = 0$ has odd number of solutions, we can find a solution curve which does not return to the $t = 0$ plane. Thus, in this case, the conditions of Th.3.2 also hold true.

**Corollary 3.3** (Schauder and Darbo's fixed point theorem) Let $X$ be a B-space with a projection scheme $\Pi = \{X_{n}, P_{n}\}$, $D$ be a open bounded convex set in $X$, and $p : cl(D) \rightarrow cl(D)$. If $p$ is a continuous ball condensing operator, $p$ has at least a fixed point in $cl(D)$.

*(Proof)* Let $x_{0} \in D$, $f(x) = p(x) - x$, and $h(x,t) = (1-t)(x_{0} - x) + tf(x)$. Then, $h$ becomes an A-proper homotopy with respect to $\Pi$. Moreover, it is easily seen that the conditions of Th.3.2 are satisfied.

Now, we would like to present a problem. Although by A-proper homotopy theory a method is given for calculating an approximate solution sequence $\{x_{n}\}$ whose subsequence converges to a true solution $x^{*}$. In practice, we cannot choose a convergence subsequence from this approximation sequence! This difficulty can be overcome with the aid of an Urabe-type *a posteriori* error estimation method.

In the following, we assume

**Assumption 3.1** Let $X$ be a B-space with a projection scheme $\{X_{n}, P_{n}\}$ such that $P_{n}P_{m} = P_{\min\{n,m\}}$ and $\|P_{n}\| \leq 1$.

In order to overcome the above-mentioned difficulty, we propose the following projective simplified Newton method in $X$:

$$x_{k+1} = x_{k} - P_{k+1}f(x_{k}), \quad k \geq 0, x_{0} \in X_{0}.$$  \hspace{1cm} (77)

We note that from the definition, $x_{k}$ belongs to $X_{k}$. The following is an Urabe-type convergence theorem for the projective simplified Newton method:

**Theorem 3.3** Let $X$ be a B-space satisfying Assumption 3.1, $Y$ be a B-space, $D \subset X$ is a nonempty open set, and $f : D \rightarrow Y$ is a $C^{1}$-operator. Assume that $x \in D$, being an approximate solution of $f(x) = 0$, and a bounded linear operator $F : X \rightarrow Y$, being an approximation of $f'(x)$, are obtained. Moreover, we assume that there exists a $\delta$ satisfying the following conditions:

- (c1) $B(x_{0}, \delta) \subset D$,
- (c2) $\|f'(x) - F\| \leq K_{0}$ if $x \in B(x_{0}, \delta)$,
- (c3) $F^{-1} : Y \rightarrow X$ exists and satisfies
  \[\|F^{-1}\| \delta^{-1}\|f(x)\| + K_{0} \leq 1,\]
  \hspace{1cm} (78)
- (c4) $\|F^{-1}\|K_{0} < 1$. 


Then the following statements hold true:

(a) There exists a unique solution, $x^*$, of $f(x) = 0$ in $B(x_0, \delta)$,

(b) $x_k \in B(x_0, \delta)$ for any $k \geq 0$,

(c) $x_k \rightharpoonup x^*$ as $k \to \infty$,

(d) $\|x_k - x^*\| \leq (1 - \|F^{-1}\|K_0)^{-1}\|F^{-1}f(x_k)\|$.

Here, $x_k$ is assumed to be generated by Eq.(77).

We now present a method for numerically identifying an approximate solution of Eq.(74) satisfying the conditions of Th.3.3. Let $X$ be a B-space satisfying Assumption 3.1, $Y$ a B-space with a projection scheme $\{Y_n, Q_n\}$, $D \subset X$ a nonempty open set, and $f : D \to Y$, a $C^1$-operator such that $f'$ is $\alpha$-Lipschitz continuous. We assume that there exists an algorithm solving an approximate equation $Q_n f(x) = 0$, $x \in X_n$ for sufficiently large $n$.

**Algorithm 3.1 (Step 1)** Let $n = 1$.

(Step 2) Calculate an approximate solution of $Q_n f(x) = 0$, $x \in X_n$. If the solution cannot be obtained, go to Step 4.

(Step 3) Examine whether there exists a $\delta > 0$ such that $B(x_n, \delta) \subset D$ and $\|f'(x_n)^{-1}\|(\delta^{-1}\|f(x_n)\| + \alpha \delta) < 1$. If there exists such a $\delta$, go to Step 5.

(Step 4) Let $n = n + 1$ and go to Step 2.

(Step 5) Then, it is seen that in $B(x_n, \delta)$ there exists a unique solution $x^*$ of $f(x) = 0$. If $\delta$ is greater than the desired precision, iterate the following starting from $x_n$:

$$x_{k+1} = x_k - P_{k+1}f'(x_n)^{-1}f(x_k).$$

An error estimation is given by

$$\|x_k - x^*\| \leq (1 - \|f'(x_n)^{-1}\|\alpha \delta)^{-1}\|f'(x_n)^{-1}f(x_k)\|.$$  \hspace{1cm} (80)

**Theorem 3.4** Together with the conditions of Algorithm 3.1, we assume that $f$ is A-proper, Fredholm with index zero, $f(x) \neq 0$ on $\partial D$ and 0 is a regular value of $f$. Then, Algorithm 3.1 is completed in finite cycles.

The proof can be found in Ref.[96].
4 Arithmetic for Self-Validating Numerics and Computational Complexity

In this section we shall discuss computer arithmetic for self-validating numerics. Since we are concerned with mathematical proofs of nonlinear problems, automatic rigorous estimation of rounding errors is necessary for such self-validating numerics. Although this kind of rigorous estimation of rounding errors had been believed to be difficult, development of the studies of self-validating numerics in the last decade shows that such estimation is not very difficult and that there are methods for practical implementation.

4.1 Machine Interval Analysis

A fundamental tool of such automatic estimation is the interval analysis introduced in Moore's doctoral thesis entitled "Interval arithmetic and automatic errors analysis in digital computing"[41]. In this thesis, machine interval arithmetic is introduced to automatically estimate rounding errors caused by, for example, floating point calculations. Here, a machine interval is an interval with end points being represented by floating point numbers. In this arithmetic system, for example, the number $\pi$ is represented as $\pi \in [3.14, 3.15]$. The process generating a sequence of machine intervals such as $[3.141, 3.142], [3.1415, 3.1416], [3.14159, 3.14160], \ldots$, is a computation of $\pi$ in the interval analysis. Thus in the machine interval analysis, a machine interval is a fundamental data type. Arithmetics on machine intervals can be defined. Let $A$ and $B$ be machine intervals and $\ast \in \{+, -, \times, /\}$, then $A * B$ is defined by

$$A * B = \{a \ast b | a \in A, b \in B\}.$$  \hspace{1cm} (81)

In this case, the following properties hold:

$$[a, b] + [c, d] = [a + c, b + d],$$
$$[a, b] - [c, d] = [a - d, b - c],$$
$$[a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)],$$
$$[a, b] / [c, d] = [a, b] \cdot [1/d, 1/c].$$  \hspace{1cm} (82)

Thus, it turns out that arithmetics between machine intervals can be executed by the arithmetics among end points of intervals. For practical implementations of interval arithmetics, see, for example, Neumaier[42] and Kulish and Miranker[3].

If we define a width and an absolute value of an interval respectively by $w([a, b]) = b - a$ and $||a, b|| = \max(|a|, |b|)$, then for intervals $A$ and $B$ the following holds:

$$w(A \pm B) = w(A) + w(B),$$
$$w(A \cdot B) \leq w(A)|B| + |A|w(B).$$  \hspace{1cm} (83)

The first equation of (83) indicates that

$$w(A - A) = 2w(A),$$  \hspace{1cm} (84)
which implies that the widths of intervals tend to increase as calculations proceed. This drawback has been overcome by Kulish and his coworkers[3, 13]. They showed that widths of intervals can be narrowed by utilizing an interval version of the residue correction method[98]. Architecturally, they proposed the use of a long accumulator to exactly calculate inner products of vectors whose components are floating point numbers[99]. They showed, through many examples, that not only linear problems, but also nonlinear problems, can be solved by this method with guaranteed accuracy[100]. Moreover, they showed that functional equations can also be solved by means of their method[101].

As software for self-validating numerics, FORTRAN–SC, ACRITH–XSC, PASCAL–SC, PASCAL-XSC, and so on have been developed. Since there already exist good reviews[6]-[8], [12, 13, 102] on them, we leave detailed discussion to others. Results related to a rounding error analysis up to 1965 are gathered in Ref.[103].

4.2 Rational Arithmetic

In this section, we describe our approach using rational arithmetic for self-validating numerics. Why do we use rational arithmetic? The following is a partial answer to this question:

(1) Most numerical algorithms are designed by analytical theory which is based on the concept of real numbers forming a field. The fact that the set of rational numbers also forms a field and is dense in the set of real numbers is very conductive to the design of a numerical algorithm. On the other hand, a set of floating numbers with fixed length does not form a field so that even an associative law does not hold.

(2) Also, a numerical algorithm using rational arithmetic should involve rounding, because the number of bits needed to represent rational numbers become extremely large even after a few iterations of rational arithmetic. However, we can round a rational number with desired accuracy, by for example, using its continued fraction expansion. Thus rounding errors can be easily estimated.

(3) Computational complexity theory fits very well with the rational arithmetic model of computation. Namely, to obtain a solution to a numerical problem, the required precision of arithmetic depends on the problem. Although floating point numbers have fixed precision, rational numbers can express arbitrary precisioned numbers. Thus, at least theoretically, rational arithmetic has an advantage. For example, the design of a polynomial time algorithm of linear programming is based on rational arithmetic.

We now describe how to use rational arithmetic for self-validating numerics. For present purposes, we start with a discussion of how to represent natural numbers. Let \( P \) be a fixed natural number greater than one. Then, using \( P \) as a base, an arbitrary natural number \( a \) can be represented as

\[
a = a_nP^n + a_{n-1}P^{n-1} + \cdots + a_1P + a_0,
\]  

(85)

where \( a_i \) satisfying \( 0 \leq a_i < P \) is called a digit and \( a_n \neq 0 \). We shall denote the correspondence (85) as

\[
a = (a_n, a_{n-1}, \cdots, a_1, a_0)_P.
\]  

(86)
Then a rational number \( q \) is represented as
\[
q = s \frac{q_1}{q_2},
\] (87)
where \( s \) is the sign of \( q \) and \( q_1 \) and \( q_2 \) are natural numbers except in the case of \( q = 0 \). If \( q = 0 \), then we represent it as \( s = + \) or \( s = - \), \( q_1 = 0 \) and \( q_2 = 1 \). We assume that in normalized form \( q_1 \) and \( q_2 \) are mutually prime. An efficient implementation of rational arithmetic such as addition, subtraction, multiplication and division is described, by for example, Knuth[104] so that we omit a description here.

We consider here how to round rational numbers. For this purpose, the continued fraction expansion is useful. Let \( \omega \) be a positive real number. Its continued fraction expansion can be obtained as follows: Let \([\omega]\) be an integer part of \( \omega \). Let
\[
a_0 = [\omega].
\] (88)
If \( \omega - a_0 \neq 0 \), then we can write \( \omega \) as
\[
\omega = a_0 + \frac{1}{\omega_1},
\] (89)
where \( \omega_1 = \frac{1}{\omega-a_0} \). Since \( \omega_1 > 1 \) we let
\[
a_1 = [\omega_1].
\] (90)
Continuing this process, we have a continued fraction expansion of \( \omega \):
\[
\omega = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ldots}}}
\] (91)
We shall denote this relationship as \( \omega = [a_0, a_1, a_2, \ldots] \). It is known that if \( \omega \) is a rational number, the numbers \( a_0, a_1, a_2, \ldots \), can be directly obtained by the Euclidean algorithm. A rounding of the real \( \omega \) is obtained by truncating its continued fraction expansion as
\[
\omega \simeq [a_0, a_1, a_2, \ldots, a_n].
\] (92)
If we let
\[
\frac{p_n}{q_n} = [a_0, a_1, a_2, \ldots, a_n],
\] (93)
then,
\[
p_{n+1} = a_{n+1}p_n + p_{n-1}, \quad p_1 = a_0a_1 + 1, \quad p_0 = a_0(n \geq 1),
\]
\[
q_{n+1} = a_{n+1}q_n + q_{n-1}, \quad q_1 = a_1, \quad q_0 = 1(n \geq 1)
\] (94)
hold. From this, it is easy to see that
\[
P_nq_{n-1} - p_{n-1}q_n = (-1)^{n+1}
\] (95)
holds true. Rounding error of approximating \( \omega \) by \( p_n/q_n \) can be easily estimated as follows. The real \( \omega \) can be represented as
\[
\omega = [a_0, a_1, a_2, \ldots, a_n, \omega_{n+1}].
\] (96)
Thus, from Eqs.(94) we have
\[ \omega = \frac{\omega_{n+1}p_{n}p_{n-1}}{\omega_{n+1}q_{n} + q_{n-1}}. \] (97)
From this we have
\[ |\omega| = \frac{p_{n}}{q_{n}} \leq \frac{1}{q_{n}^{2}}. \] (98)
Moreover, if \( n \) is even \( \omega \geq \frac{p_{n}}{q_{n}} \) holds, and if \( n \) is odd \( \omega \leq \frac{p_{n}}{q_{n}} \).

**Theorem 4.1 (Lagrange)**

(i) Let \( \omega \) be a positive real number and \( \frac{p_{n}}{q_{n}} \) be its \( n \)-th continued fraction approximation. Then, for any integer \( p \) and any integer \( q \) satisfying \( 0 < q \leq q_{n} \)
\[ |p - \omega q| > |p_{n} - \omega q_{n}| \] (99)
holds true.

(ii) For any integer \( q \) satisfying \( q_{n} < q < q_{n+1} \) and for any integer \( p \)
\[ |p - \omega q| > |p_{n} - \omega q_{n}| \] (100)
holds true.

We now consider intervals with rational number end points.

**Theorem 4.2** [105] Let \( R \) be a set of sequences of intervals \( \{A_{i}\} \) satisfying the conditions

1. \( A_{0} \supseteq A_{1} \supseteq \ldots \supseteq A_{n} \supseteq A_{n+1} \supseteq \ldots \),
2. \( w(A_{n} = [a_{n}, b_{n}]) = b_{n} - a_{n} \to 0 \) as \( n \to \infty \).

Then, the set \( R \) can be identified with the set of real numbers. If we add one more postulation that

3. \( a_{n} \) and \( b_{n} \) are computable, then \( R \) becomes the set of computable reals.

From this theorem, we may consider a data type having the form
\[ \{[r_{1}, r_{2}] | r_{1} \text{ and } r_{2} \text{ are rational numbers} \} \] (101)
to be real type. Arithmetic between real type-data can be defined through Eq.(82). Moreover, a rounding operator for real type data is defined by
\[ \diamond_{n}[r_{1}, r_{2}] = [\nabla_{n}r_{1}, \triangle_{n}r_{2}]. \] (102)
Here, \( \nabla_{n}r_{1} = \frac{2a}{q_{2n}} \) and \( \triangle_{n}r_{2} = \frac{2a+1}{q_{2n+1}} \). Apparently,
\[ [r_{1}, r_{2}] \subseteq \diamond[r_{1}, r_{2}] \] (103)
holds true.

Truncation error bounds for special functions are given, for example, by Ref.[106].

We now consider to solve exactly a matrix equation
\[ Ax = b, \] (104)
where \( A \) is an \( n \times n \) matrix whose elements are all rational numbers and \( b \) is an \( n \)-dimensional vector whose elements are also all rational numbers. Edmonds[107] has shown that
Theorem 4.3 (Edmonds[107]) Let $A$ be an $n \times n$ matrix whose elements are all rational numbers, and suppose that $A$ requires at most $m$ binary digits to write down. Then, the determinant of $A$ requires no more than $O(nm)$ digits to write.

For the proof, see Refs.[107]-[109].

Based on this theory, it is proven that a solution of Eq.(104) can be obtained in polynomial time of input size, which is defined by the number of digits required to describe $A$ and $b$[109].

Recently, computational complexity of numerical problems related to mainly one-dimensional function has been studied by Ko[110]. In this book he has defined a class of polynomial computable functions and shown that

(1) Let $Max$ be an operator which maps a function $f : [0, 1]^2 \to R$ to the function $g : [0, 1] \to R$, defined by $g(x) = \max\{f(x, y)|0 \leq y \leq 1\}$. Then, $P = NP$ iff for all polynomial-time computable real functions $f$, $Max(f)$ is polynomial-time computable.

(2) Let $Int$ be the operator that maps a function $f : [0, 1] \to R$ to the function $g : [0, 1] \to R$, defined by $g(x) = \int_0^x f(t)dt$. Then, $FP = \#P$ iff for all polynomial-time computable real functions $f$, $Int(f)$ is polynomial-time computable.

(3) Let $f : [0, 1] \to [0, 1]$ be a polynomial-time computable one-to-one function. Then $F^{-1}$ is polynomial-time computable. On the other hand, $LOGSPACE = P$ iff for all log-space computable, one-to-one real functions $f$, $F^{-1}$ is log-space computable.

(4) There exists a polynomial-time computable function $f$ on $[0, 1]$ such that the derivative $f'$ exists but is not computable. On the other hand, if the second derivative $f''$ exists and continuous on $[0, 1]$, then $f'$ must be polynomial-time computable.

(5) There is a natural weak Lipschitz condition on function $f : [0, 1] \to [-1, 1]^2$ such that $P = PSPACE$ iff for all first-order ordinary differential equations $y' = f(x, y)$ defined by polynomial-time computable functions $f$ satisfying this weak Lipschitz condition the solutions $y$ are polynomial-time computable.

Definitions of terminologies such as $LOGSPACE$ and $PSPACE$ etc. and related results can be found in Ref.[111].

From these results, it is seen that there exists strong relationship between fundamental problems of the computational complexity theory and numerical algorithms based on rational arithmetic. Error bounds and complexity are given for Fourier analysis by Brass[115] and error bounds of anti-derivatives are given by Refs.[116, 117, 118].

Finally, we note that, based on Urabe's theorem and rounding by the continued fraction expansion, we have developed a self-validating simplified Newton method. This method is implemented by rational arithmetic and avoids exponential explosion of binary digits needed for expressing intermediate results by rounding. For example, we list a result of solving 5-dimensional nonlinear equation

$$f(x) = (f_1(x), \ldots, f_5(x)) = 0,$$

(105)
where

\[ f_k(x) = \frac{(x_1^3 + x_2^3 + \ldots + x_5^3 + \sqrt{5k})}{10}. \]

From this example, the simplified Newton method can well be implemented with rational arithmetic and suitable rounding through the continued fraction expansion. Details will be reported elsewhere.

4.3 Logical Foundation

Since self-validating numerics based on rational arithmetic is a kind of constructive mathematics, in this section we briefly discuss a constructive mathematics as the foundation for approaches described in the previous sections. Constructive mathematics can be developed on various mathematical foundations. Roughly speaking, they are classified into two types[110]. One is based on the recursive analysis and the other is not. The approaches based on the recursive analysis are further roughly divided into two classes. The first one is developed in the framework of classical mathematics. Thus as well as constructive objects, nonconstructive objects are allowed in this class. In this class, the work of Gregorczyk, Lacombe, Mostowski and Pour-El and Richards[111] are included[110]. In the other approach, only recursive objects are studied by constructive logic. Work by the following author is included in this class[110]: Moschovakis, Goodstein, Sanin, Ceitin and Aberth. On the other hand, an approach which uses intuitive logic and does not restrict itself to the notion of recursiveness is further classified into the following four classes[112]:

1. classical mathematics framework(CLASS)
2. Bishop's constructive mathematics(BISH)
3. Brouwer's intuitionism(INT)
4. Russian constructivism(RUSS).

Roughly speaking, in (2) and (3), the notion of an algorithm, or a finite routine, is taken as primitive. On the other hand, (4) operates within a fixed programming language, and an algorithm is a sequence of symbols in that language.

In this paper, we have taken the following standpoint. Namely, we adopt a classical mathematics as the logical foundation, i.e., we allow other than constructive objects, mathematical objects for which we cannot present an algorithm that constructs the objects. Thus, for instance, the concept of the real number is already given, provided that we know the classical mathematics. Our objective is to find a finite computational procedure for identifying an approximation of a mathematical object whose neighborhood is guaranteed to contain the desired mathematical object. Here, I would like to present a comment. As mentioned in the above-discussion, recently, several programming languages which support self-validating numerics have been developed. The fast automatic differentiation program[113, 114, 118] can also be seen as a kind of language supporting self-validating numerics. Thus, it seems interesting, to define a programming language which not only supports self-validating numerics, but also becomes a logical foundation of the mathematics of self-validating numerics.

Now, we would like to present a comment about the relationship between self-validating numerics and nonlinear functional analysis. As an overview of nonlinear functional analysis, it is
noted that Zeidler, Eberhard has written a huge series of books entitled “Nonlinear Functional Analysis and its Applications” (Springer-Verlag, I (1986), IIA, B(1990), III(1984), IV(1988)). The subtitle of each volume is listed as follows:

I Fixed-Point Theorems,
II/A Linear Monotone Operators,
II/B Nonlinear Monotone Operators,
III Variational Methods and Optimization,
IV Applications to Mathematical Physics.

The areas indicated by the above list, by considering IV as Applications, are main areas of nonlinear functional analysis. From the point of view of the principles which are used in analysis, nonlinear functional analysis can be divided into two areas:

(a) An area which is based on the compactness principle, and

(b) an area in which is the based on the axiom of choice.

Roughly speaking, topics I and II are continued in (a) and topic (III) is in (b). There are, however, quite a few exceptions. As is seen in the previous section (a) can become constructive. For example, constructive Sard’s lemma is discussed in Refs.[119, 120].

On the other hand (b) is not constructive at all, so that, for example, BISH adopted the axiom of countable choice instead of the axiom of choice. A good introduction to computational functional analysis is given by Moore[121] and an interesting theory of discrete functional analysis is presented by Zhou[122].

5 Computer Assisted Proofs for Nonlinear Problems

Self-validating numerics has many applications other than to periodic problems of nonlinear differential equations. In this section, we review such applications.

5.1 Functional Equations

Various functional equations have been solved by the self-validating numerical method. Some of them have already been discussed in the previous sections. Although the methods discussed in the previous sections are based on a posteriori error estimates, many of the self-validating numerics use interval analysis. In this subsection, we have given an overview of the application of self-validating numerics to functional equations. Emphasis is on interval analysis. For introduction of interval analysis, see Refs.[123]-[126].

also treats self-validating numerical methods for monotone-type nonlinear differential equations including partial differential equations. In Ref.[131], this method is further developed.

Recently, translation of Mikhlin's book[132] has been published, in which various arguments related to self-validation can be found. In particular, on p.27, a posteriori error estimation for monotone operator equations is discussed. In this book, an interesting method of opposite functional is also developed for including solutions to boundary value problems of nonlinear equations.

(b) Integral equations: In his book[134], Linz described an approximate solution method for linear operator equations of the second kind, based on, an a posteriori method. This description includes Anselone's collectively compact operator approach[135] to Nestrom's method for integral equations. Linz's book is very readable to engineers and gives a good introductive functional analytic basis for error estimation of various approximation methods. Recently, he has presented a method for determination of precise bounds for inverses of linear integral equations, which is useful to a posteriori error estimation[136]. Since the Newton method uses linearization, this result is also useful for nonlinear integral equations. Noble[138] also treated a problem of inclusion of solutions for integral equations. Using Noble's approach, Spence[137] gave error bounds for eigenvalues of integral equations. Related the problems of Linz are discussed by Sloan[139]. Demmel[140] investigated the relationship between the condition number of a problem and the shortest distance from that problem to an ill-posed one. For the finite element method, see for example Ref.[141]. An application is presented of interval integration to the solution of integral equations by Rall[142].


5.2 Computer-assisted proof for nonlinear problems

In this subsection, we review applications of the self-validating method to nonlinear problems. Recently, numerical study of nonlinear dynamical systems has made a great stride. Beyn's paper[153] is a very good survey on this topic. If a continuous dynamical system is approximated by a discrete system, then a question "what kind of properties of the original dynamical system are reflected in the discrete system?" is a fundamental interest. In general, qualitative properties of a dynamical system are changed by discretization. Thus, the following question becomes important: Is there an invariant curve for the discrete dynamical system, which is an approximation of a continuous dynamical system possessing an invariant curve?
This type of question is studied in Ref.[154]. Related results are cited in Beyn[153]. For Hamiltonian system the KAM theory is related to such a question. In Ref.[155] various papers are gathered for comp.ter assisted proofs of analysis. In particular, computer-assisted KAM theories are presented by several authors. Reference[156] presents a convergence theorem of a Newton-Moser-type method. In Refs.[157, 158], a method is given for branch inclusion in a generic Hopf bifurcation. A computer assisted proof based on interval analysis is given for a problem related to chaos by Ref.[159].

6 Concluding Remarks

In this paper, the current state of research is surveyed for the study of self-validating numerical methods of nonlinear problems. In Sect.2, Kantorovich's approach to this problem is reviewed. His method is based on his convergence theorem of Newton's method and can be seen as an a posteriori error estimation method. Then, in Sect.3, Urabe's approach to this problem is discussed. He treated practical nonlinear differential equations such as the Van der Pol equation and the Duffing equation and proved the existence of their periodic and quasi-periodic solutions using self-validating numerics. Generalizations and abstraction of Urabe's method to more general functional equations are also discussed. Then methods for rigorous estimation of rounding errors are surveyed in Sect.4. First, interval analytic methods are discussed. Then, an approach of the author which uses rational arithmetic is briefly reviewed. Finally, problems related to self-validating numerics are overviewed in Sect.5. Due to the limitation of space, we cannot discuss many of important studies in this area.

Finally, it is noted that many interesting nonlinear problems show potential for treatment by self-validating numerics, such as

(1) problems related to chaos,
(2) problems related to perturbed soliton systems,
(3) problems related to nonlinear large scale circuits simulations, in which numerical solutions are difficult to obtain by the effect of rounding errors,

and so on.

Acknowledgements

The author would like to thank Professor Kazuo Horiuchi of Waseda University for his fruitful discussion and comments. He also would like to express his sincerely thanks to Professor Masao Iri of University of Tokyo for his critical reading of the manuscript. A part of this work is done in collaboration with Dr. Mitsunori Makino and Mr. Masahide Kashiwagi of my laboratory. This paper is partially supported by the Grant-in-Aid for the Ministry of Education, Science Research and Culture;No.03244105 and 02302046, and by the Waseda University Grant for Special Projects.
References


Table 1: Modified Newton Iterations with Guaranteed Accuracy

<table>
<thead>
<tr>
<th>(k)</th>
<th>(x_k)</th>
<th>(\delta_k)</th>
</tr>
</thead>
</table>
| 0    | 829 163 167 159 329  
     | 3121' 455' 389' 325' 607  
 | 4    | 421    |
| 1    | 16945 139 82 91 129  
     | 83793' 388' 191' 186' 238  
 | 1    | 1736   |
| 2    | 15619 441 659 1840 1993  
     | 58601' 1231' 1535' 3761' 3677  
 | 1    | 28593  |
| 3    | 15687 1601 3623 1840 1993  
     | 59057' 4469' 8439' 3761' 3677  
 | 1    | 471209 |
| 4    | 15670 6984 18856 9836 12087  
     | 58993' 19495' 43921' 20105' 22300  
 | 5    | 38827291 |
| 5    | 15670 43505 18856 21671 50341  
     | 58993' 121439' 43921' 44296' 92877  
 | 54   | 6910577605 |
| 6    | 94105 144438 139238 407910 175197  
     | 354278' 403307' 324325' 833777' 323231  
 | 5    | 2108987969 |
| 7    | 203897 534427 538096 352733 412822  
     | 767613' 1491789' 1253379' 720994' 761639  
 | 54   | 34755825669 |
| 8    | 831275 1263826 1275621 21564053 17576149  
     | 3129509' 3527819' 2971285' 44077398' 32427246  
 | 5    | 572771127991 |
| 9    | 24624561 5250276 23698703 21564053 17576149  
     | 92704322' 14655517' 55201036' 44077398' 32427246  
 | 54   | 9439187783494 |
| 10   | 24624561 22994329 23698703 647329500 88293567  
     | 92704322' 64185917' 55201036' 1323155717' 162897869  
 | 5    | 15555489595717 |

There is a solution of Eq.(105) in a ball centered at \(x_k\) with a radius \(\delta_k\).