*Acta Numerica* (1997), *pp.* 523–551

# Complexity theory and numerical analysis

Steve Smale
*Department of Mathematics*
*City University of Hong Kong*
*Hong Kong*
*E-mail: masmale@math.cityu.edu.hk*

## CONTENTS

## Preface

Section 5 is written in collaboration with Ya Yan Lu of the Department of Mathematics, City University of Hong Kong.

## 1. Introduction

Complexity theory of numerical analysis is the study of the number of arithmetic operations required to pass from the input to the output of a numerical problem.

To a large extent this requires the (global) analysis of the basic algorithms of numerical analysis. This analysis is complicated by the existence of ill-posed problems, conditioning and round-off error.

A complementary aspect ('lower bounds') is the examination of efficiency for all algorithms solving a given problem. This study is difficult and needs a formal definition of algorithm.

Highly developed complexity theory of computer science provides some inspiration to the subject at hand. Yet the nature of theoretical computer science, with its foundations in discrete Turing machines, prevents a simple transfer to a subject where real number algorithms such as Newton's method dominate.

One can indeed be sceptical about a formal development of complexity into the domain of numerical analysis, where problems are solved only to a certain precision and round-off error is central.

Recall that, according to computer science, an algorithm defined by a Turing machine is *polynomial time* if the computing time (measured by the number of Turing machine operations) $T(y)$ on input $y$ satisfies:

$$T(y) \leq K(\mathrm{size}(y))^c. \tag{1.1}$$

Here, $\mathrm{size}(y)$ is the number of bits of $y$. A problem is said to be in $P$ (or tractable) if there is a polynomial time algorithm (*i.e.* machine) solving it.

The most natural replacement for a Turing machine operation in a numerical analysis context is an arithmetical operation, since that is the basic measure of cost in numerical analysis. Thus, one can say with little objection that the problem of solving a linear system $Ax = b$ is tractable because the number of required Gaussian pivots is bounded by $cn$ and the input size of the matrix $A$ and vector $b$ is about $n^2$. (There remain some crucial questions of conditioning to be discussed later.) In this way complexity theory is part of the tradition of numerical analysis.

But this situation is no doubt exceptional in numerical analysis in that one obtains an exact answer, and most algorithms in numerical analysis solve problems only approximately with, say, accuracy $\varepsilon > 0$, or precision $\log \varepsilon^{-1}$. Moreover, the time required depends more typically on the condition of the problem. Therefore it is reasonable for 'polynomial time' to be recast in the form:

$$T(y, \varepsilon) \leq K\Big(\mu(y) + \mathrm{size}(y) - \log \varepsilon\Big)^c. \tag{1.2}$$

Here, $y = (y_1, \cdots, y_n)$, with $y_i \in \mathbb{R}$ is the input of a numerical problem, with $\mathrm{size}(y) = n$. The accuracy required is $\varepsilon > 0$ and $\mu(y)$ is a number representing the condition of the particular problem represented by $y$ ($\mu(y)$ could be a condition number). There are situations where one might replace $\mu$ by $\log \mu$ or $\log \varepsilon^{-1}$ by $\log \log \varepsilon^{-1}$, for example. Moreover, using the notion of approximate zero, described below, the $\varepsilon$ might be eliminated.

I see much of the complexity theory ('upper bound' aspect) of numerical analysis conveniently represented by a two-part scheme. Part 1 is the estimate (1.2). Part 2 is an estimate of the probability distribution of $\mu$, and takes the form

$$\text{prob}\left\{y : \mu(y) \geq K\right\} \leq \left(\frac{1}{K}\right)^c, \tag{1.3}$$

where a probability measure has been put on the space of inputs.

Then Parts 1 and 2 combine, eliminating the $\mu$, to give a probability bound of the complexity of the algorithm. The following sections illustrate this theme. One needs to understand the condition number $\mu$ with great clarity for the procedure to succeed.

I hope this gives some immediate motivation for a complexity theory of numerical analysis and even to indicate that, all along, numerical analysts have often been thinking in complexity terms.

Now, complexity theory of computer science has also studied extensively the problem of finding lower bounds for certain basic problems. For this one needs a formal definition of algorithm, and the Turing machine begins to play a serious role. That makes little sense when the real numbers of numerical analysis dominate the mathematics. However without too much fuss we can extend the concept of a machine to deal with real numbers, and one can also start dealing with lower bounds of real number algorithms. This last is not so traditional for numerical analysis, yet the real number machine leads to exciting new perspectives and problems.

In computer science, consideration of polynomial time bounds led to the fundamentally important and notoriously difficult problem 'P = NP?'. There is a corresponding problem for real number machines, namely 'P = NP over $\mathbb{R}$?'.

The above is a highly simplified, idealized snapshot of a complexity theory of numerical analysis. Some details follow in the sections below. Also see Blum, Cucker, Shub and Smale (1996), referred to hereafter as the Manifesto, and its references for more background, history and examples.

## 2. Fundamental theorem of algebra

The fundamental theorem of algebra (FTA) deserves special attention. Its study in the past has been a decisive factor in the discovery of algebraic numbers, complex numbers, group theory and more recently in the development of the foundations of algorithms.

Gauss gave four proofs of this result. The first was in his thesis which, in spite of a gap (see Ostrowski in *Gauss*), anticipates some modern algorithms (see Smale 1981). Constructive proofs of the FTA were given in 1924 by Brouwer and Weyl.

Further, Peter Henrici and his co-workers have given a substantial development for analysing algorithms and a complexity theory for the FTA. See Dejon and Henrici (1969) and Henrici (1977). Also, Collins (1975) gave a contribution to the complexity of FTA. See especially Pan (1996) and McNamee (1993) for historical background and references.

In 1981–82, two articles appeared with approximately the same title, Schönhage (1982) and Smale (1981), which systematically pursued the issue of complexity for the FTA. Coincidentally, both authors gave main talks at the International Congress of Mathematicians, Berkeley 1986, on this subject; see Schönhage (1987) and Smale (1987*a*).

These articles fully illustrate two contrasting approaches.

Schönhage's algorithm is in the tradition of Weyl, with a number of added features which give very good polynomial time complexity bounds. The Schönhage analysis includes the worst case and the implicit model is the Turing machine. On the other hand, the methods have never extended to more than one variable, and the algorithm is complicated. Some subsequent developments in a similar spirit include Renegar (1987*b*), Bini and Pan (1987), Neff (1994), and Neff and Reif (1996). See Pan (1997) for an account of this approach to the FTA.

In contrast, in Smale (1981), the algorithm is based on continuation methods such as Kellog, Li, and Yorke (1976), Smale (1976), Keller (1978), and Hirsch and Smale (1979). See Allgower and Georg (1990, 1993) for a survey. The complexity analysis of the 1981 paper was a probabilistic polynomial time bound on the number of arithmetic operations, but much cruder than Schönhage's. The algorithm, based on Newton's method, was simple, robust, easy to program, and extended eventually to many variables. The implicit machine model was that of Blum, Shub and Smale (1989), hereafter referred to as BSS (1989). Subsequent developments along these lines include Shub and Smale (1985, 1986), Kim (1988), Renegar (1987*b*), Shub and Smale (1993*a*, 1993*b*, 1993*c*, 1996 and 1994), hereafter referred to as Bez I–V, respectively, and Blum, Cucker, Shub and Smale (1997), hereafter referred to as BCSS (1997).

Here is a brief account of some of the ideas of Smale (1981). A point $z$ is called an *approximate zero* if Newton's method starting at $z$ converges well in a certain precise sense; see Section 4 below. The main theorem of this paper asserts the following.

**Theorem 2.1** A sufficient number of steps of a modified Newton's method to obtain an approximate zero of a polynomial $f$ (starting at 0) is polynomially bounded by the degree of the polynomial and $1/\sigma$, where $\sigma$ is the probability of failure.

For the proof, an invariant $\mu = \mu(f)$ of $f$ is defined akin to a condition number of $f$. Then the proof is broken into two parts.

**Part 1:** A sufficient number of modified Newton steps to obtain an approximate zero of $f$ is polynomially bounded by $\mu(f)$.

The proof of Part 1 relies on a Loewner estimate related to the Bieberbach conjecture.

**Part 2:** The probability that $\mu(f)$ is larger than $k$ is less than $k^{-c}$, some constant $c$.

The proof of Part 2 uses elimination theory of algebraic geometry and geometric probability theory, Crofton's formula, as in Santaló (1976).

The crude bounds given in Smale (1981), and the mathematics too, were substantially developed in Shub and Smale (1985, 1986).

Here is a more detailed, more developed, complexity theoretic version of the FTA in the spirit of numerical analysis. See BCSS (1997) for the details.

Assume given (or input):

a complex polynomial $f(z) = \sum a_i z^i$ in one complex variable,
a complex number $z_0$, and an $\varepsilon > 0$.

Here is the algorithm to produce a solution (output) $z^*$ satisfying

$$|f(z^*)| < \varepsilon. \tag{2.1}$$

Let $t_0 = 0$, $t_i = t_{i-1} + \Delta t$, where $\Delta t = 1/k$, for some positive integer $k$; thus $t_k = 1$, and we have a partition of $[0, 1]$. For any polynomial $g$, we define Newton's method by

$$N_g(z) = z - \frac{g(z)}{g'(z)}, \qquad \text{for all } z \in \mathbb{C}, \text{ such that } g'(z) \neq 0.$$

Let $f_t(z) = f(z) - (1-t)f(z_0)$. Then, generally, there is a unique path $\zeta_t$ such that $f_t(\zeta_t) = 0$ all $t \in [0, 1]$ and $\zeta_0 = z_0$. Define inductively

$$z_i = N_{f_{t_i}}(z_{i-1}), \qquad i = 1, \ldots, k, \quad z^* = z_k. \tag{2.2}$$

It is easily shown that for almost all $(f, z_0)$, $z_i$ will be defined, $i = 1, \ldots, k$, provided $\Delta t$ is small enough. We may say that $k = 1/\Delta t$ is the 'complexity'. It is the main measure of complexity in any case: the problem at hand is, 'how big may we choose $\Delta t$ and still have $z^*$ satisfying (2.1) and (2.2)?' (*i.e.* so that the complexity is the lowest).

Next a 'condition number' $\mu(f, z_0)$ is defined which measures how close $\zeta_t$ is to being ill-defined. (More precisely $\mu(f, z_0) = \operatorname{cosec}\theta$ where $\theta$ is the supremum of the angles of sectors about $f(z_0)$ for which the inverse $f^{-1}$ mapping $f(z_0)$ to $z_0$ is defined.)

**Theorem 2.2** A sufficient number $k$ of Newton steps defined in (2.2) to achieve (2.1) is given by

$$k < 26\mu(f, z_0)\left(\log\frac{|f(z_0)|}{\varepsilon} + 1\right).$$

**Remark 2.1**

(a) We are assuming $0 < \varepsilon < 1/2$.
(b) Note that the degree $d$ of $f$ plays no role, and the result holds for any $(f, z_0, \varepsilon)$.
(c) The proof is based on 'point estimates' ($\alpha$-theory) (see Section 4 below) and an estimate of Loewner from Schlicht function theory. Thus it doesn't quite extend to $n$ variables. It remains a good problem to find the connection between Theorem 2.2 and Theorem 6.1.

For the next result suppose that $f$ has the form

$$f(z) = \sum_{i=0}^{d} a_i z^i, \qquad a_d = 1, \quad |a_i| \leq 1.$$

**Theorem 2.3** The set of points $z_0 \in \mathbb{C}$, $|z_0| = R > 2$, such that $\mu(f, z_0) > b$, is contained in the union of $2(d-1)$ arcs of total angle

$$\frac{2}{d}\left(\frac{1}{b} + \sin^{-1}\frac{1}{R-1}\right).$$

This result is an estimate on how infrequently poorly conditioned pairs $(f, z_0)$ occur.

It is straightforward to combine Theorems 2.2 and 2.3 to eliminate the $\mu$ and obtain both probabilistic and deterministic complexity bounds for approximating a zero of a polynomial. The probabilistic estimate improves the deterministic one by a factor of $d$. Theorem 2.3 and these results are in Shub and Smale (1985, 1986), but see also BCSS (1997), and Smale (1985).

**Remark 2.2** The above-mentioned development might be improved in sharpness in two ways.

(A) Replace the hypothesis on the polynomial $f$ by assuming as in Renegar (1987*b*) and Pan (1996) that all the roots of $f$ are in the unit disk.
(B) Suppose that the input polynomial $f$ is described not by its coefficients, but by a 'program' for $f$.

## 3. Condition numbers

The condition number as studied by Wilkinson (1963), important in its own right in numerical analysis, also plays a key role in complexity theory. We review it now, especially some recent developments. For linear systems, $Ax = b$, the condition number is defined in most basic numerical analysis texts.

The Eckart and Young (1936) theorem is central, and may be stated as

$$\|A^{-1}\|^{-1} = d_f(A, \Sigma_n),$$

where $A$ is a non-singular $n \times n$ matrix, with the operator norm on the left and the Frobenius distance on the right. Moreover, $\Sigma_n$ is the subspace of singular matrices.

The case of 1-variable polynomials was studied by Wilkinson (1963) and Demmel (1987), among others. Demmel gave estimates on the condition number and the reciprocal of the distance to the set of polynomials with multiple roots.

We now give a more general context for condition numbers and give exact formulae for the condition number as the reciprocal of a distance to the set of ill-posed problems following Bez I, II, IV, Dedieu (1997*a*, 1997*b*, 1997*c*) and BCSS (1997).

Consider first the context of the implicit function theorem:

$$F : \mathbb{R}^k \times \mathbb{R}^m \to \mathbb{R}^m, \quad C^1, \quad F(a_0, y_0) = 0,$$
$$\frac{\partial F}{\partial y}(a_0, y_0) : \mathbb{R}^m \to \mathbb{R}^m \quad \text{non-singular.}$$

Then there exists an open neighbourhood $\mathcal{U}$ of $a_0$ in $\mathbb{R}^k$ and a $C^1$ map $G : \mathcal{U} \to \mathbb{R}^m$ such that $G(a_0) = y_0$ and $F(a, G(a)) = 0$, for $a \in \mathcal{U}$.

Regard $F_a : \mathbb{R}^m \to \mathbb{R}^m$, $F_a(y) = F(a, y)$, as a system of equations parameterized by $a \in \mathbb{R}^k$. Then $a$ might be the input of a problem $F_a(y) = 0$ with output $y$; $G$ is the 'implicit function'.

Let us call the derivative $DG(a_0) : \mathbb{R}^k \to \mathbb{R}^m$ the *condition matrix* at $(a_0, y_0)$. Then the condition number $\mu(a_0, y_0) = \mu$, as in Wilkinson (1963), Rice (1966), Wozniakowski (1977), Demmel (1987), Bez IV, and Dedieu (1997*a*), is defined by

$$\mu(a_0, y_0) = \|DG(a_0)\|,$$

the operator norm. Thus $\mu(a_0, y_0)$ is the bound on the infinitesimal output error of the system $F_a(y) = 0$ in terms of the infinitesimal input error.

It is important to note that, while the map $G$ is given only implicitly, the condition matrix

$$DG(a_0) = \frac{\partial F}{\partial y}(a_0, y_0)^{-1} \frac{\partial F}{\partial a}(a_0, y_0)$$

is given explicitly, as is its norm, the condition number $\mu(a_0, y_0)$.

An example, given by Wilkinson, is the case where $\mathbb{R}^k$ is the space of real polynomials $f$ in one variable of degree $\leq k - 1$, and $\mathbb{R}^m = \mathbb{R}$ the space of $\zeta$, $F(f, \zeta) = f(\zeta)$. One may compute that in this case

$$\mu(f, \zeta) = \frac{\left(\sum_0^d |\zeta^i|^2\right)^{1/2}}{|f'(\zeta)|}.$$

For the discussion of several variable polynomial systems, it is convenient to use complex numbers and homogeneous polynomials.

If $f : \mathbb{C}^n \to \mathbb{C}$ is a polynomial of degree $d$, we may introduce a new variable, say $z_0$, and define $\hat{f} : \mathbb{C}^{n+1} \to \mathbb{C}$ by $\hat{f}(1, z_1, \ldots, z_n) = f(z_1, \ldots, z_n)$ and $\hat{f}(\lambda z_0, \lambda z_1, \ldots, \lambda z_n) = \lambda^d \hat{f}(z_0, \ldots, z_n)$. Thus $\hat{f}$ is a homogeneous polynomial.

If $f : \mathbb{C}^n \to \mathbb{C}^n$, $f = (f_1, \ldots, f_n)$, $\deg f_i = d_i$, $i = 1, \ldots, n$, is a polynomial system, then by letting $\hat{f}$ equal $(\hat{f}_1, \ldots, \hat{f}_n)$, we obtain a homogeneous system $\hat{f} : \mathbb{C}^{n+1} \to \mathbb{C}^n$. Any zero of $f$ will also be a zero of $\hat{f}$ and justification can be made for the study of such systems in their own right. Thus now we will consider such systems, say $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$ and denote the space of all such $f$ by $\mathcal{H}_d$, $d = (d_1, \ldots, d_n)$, degree $f_i = d_i$.

Recall that an Hermitian inner product on $\mathbb{C}^{n+1}$ is defined by

$$\langle z, w \rangle = \sum_{i=0}^{n} \bar{z}_i w_i, \qquad z, w \in \mathbb{C}^{n+1}.$$

Now, define for degree $d$ homogeneous polynomials $f$, $g : \mathbb{C}^{n+1} \to \mathbb{C}$,

$$\langle f, g \rangle = \sum_{\alpha} \binom{d}{\alpha}^{-1} \bar{f}_\alpha g_\alpha,$$

where

$$f(z) = \sum_{|\alpha|=d} f_\alpha z^\alpha, \qquad g(z) = \sum_{|\alpha|=d} g_\alpha z^\alpha.$$

Here $\alpha = (\alpha_1, \ldots, \alpha_{n+1})$ is a multi-index and

$$\binom{d}{\alpha} = \frac{d!}{\alpha_1! \cdots \alpha_{n+1}!}, \qquad |\alpha| = \sum_{i=1}^{n+1} \alpha_i.$$

The weighting by the multinomial coefficient is important, and yields unitary invariance of the inner product, as below.

**Proposition 3.1 (Reznick 1992)** Let $f, N_x : \mathbb{C}^{n+1} \to \mathbb{C}$ be degree $d$ homogeneous polynomials, where $N_x(z) = \langle x, z \rangle^d$. Then $f(x) = \langle f, N_x \rangle$.

**Corollary 3.1**

$$|f(x)| \leq \|f\| \, \|N_x\| \leq \|f\| \, \|x\|^d.$$

For $f$, $g \in \mathcal{H}_d$, define

$$\langle f, g \rangle = \sum \frac{\langle f_i, g_i \rangle}{d_i}, \qquad \|f\| = \langle f, f \rangle^{1/2}.$$

Dedieu has suggested weighting by $1/d_i$ to make the Condition Number Theorem below more natural.

The unitary group $U(n+1)$ is the group of all linear automorphisms of $\mathbb{C}^{n+1}$ which preserve the Hermitian inner product.

There is an induced action of $U(n+1)$ on $\mathcal{H}_d$ defined by

$$(\sigma f)(z) = f(\sigma^{-1}z), \qquad \sigma \in U(n+1), \qquad z \in \mathbb{C}^{n+1}, \quad f \in \mathcal{H}_d.$$

Then it can be proved (see, for instance, BCSS 1997) that

$$\langle \sigma f, \sigma g \rangle = \langle f, g \rangle, \qquad f, g \in \mathcal{H}_d, \qquad \sigma \in U(n+1).$$

This is unitary invariance.

There is a history of this inner product going back at least to Weyl (1932), with contributions or uses in Kostlan (1993), Brockett (1973), Reznick (1992), Bez I–V, Dégot and Beauzamy (1997), Stein and Weiss (1971), Dedieu (1997*a*).

Now we may define the condition number $\mu(f,\zeta)$ for $f \in \mathcal{H}_d$, $\zeta \in \mathbb{C}^{n+1}$, $f(\zeta) = 0$ using the previously defined implicit function context. To be technically correct, one must extend this context to Riemannian manifolds to deal with the implicitly defined projective spaces. See Bez IV for details.

The following is proved in Bez I (but see also Bez III, Bez IV).

**Condition Number Theorem 1** Let $f \in \mathcal{H}_d$, $\zeta \in \mathbb{C}^{n+1}$, $f(\zeta) = 0$. Then

$$\mu(f,\zeta) = \frac{1}{d((f,\zeta),\Sigma_\zeta)}.$$

Here the distance $d$ is the projective distance in the space $\{g \in \mathcal{H}_d : g(\zeta) = 0\}$ to the subset where $\zeta$ is a multiple root of $g$.

The proof uses unitary invariance of all the objects. Thus one can reduce to the point $\zeta = (1, 0, \cdots, 0)$, and then to the linear terms, and then to the Eckart–Young theorem.

Dedieu (1997*a*) has generalized this result quite substantially, and has considered sparse polynomial systems (Dedieu 1997*b*). Thus a formula for the eigenvalue problem becomes a special case.

## 4. Newton's method and point estimates

Say that $z \in \mathbb{C}^n$ is an *approximate zero* of $f : \mathbb{C}^n \to \mathbb{C}^n$ (or $\mathbb{R}^n \to \mathbb{R}^n$, or even for Banach spaces) if there is an actual zero $\zeta$ of $f$ (the 'associated zero') and

$$\|z_i - \zeta\| \leq \left(\frac{1}{2}\right)^{2^i-1} \|z - \zeta\|, \tag{4.1}$$

where $z_i$ is given by Newton's method

$$z_i = N_f(z_{i-1}), \quad z_0 = z, \quad N_f(z) = z - Df(z)^{-1}f(z).$$

Here $Df(z) : \mathbb{C}^n \to \mathbb{C}^n$ is the (Fréchet) derivative of $f$ at $z$.

An approximate zero $z$ gives an effective termination for an algorithm provided one can determine whether $z$ has the property (4.1).

Towards that end, the following invariant is decisive.

$$\gamma = \gamma(f, z) = \sup_{k\geq 2} \left\| \frac{Df(z)^{-1}D^{(k)}f(z)}{k!} \right\|^{\frac{1}{k-1}}.$$

Here $D^{(k)}f(z)$ is the $k$th derivative of $f$ considered as a $k$-linear map and we have taken the operator norm of its composition with $Df(z)^{-1}$; if the expression is not defined, then use $\gamma = \infty$. See Smale (1986), Smale (1987*a*) and Bez I for details of this development.

The invariant $\gamma$ turns out to be a key element in the complexity theory of non-linear systems. Although it is defined in terms of all the higher derivatives, in many contexts it can be estimated in terms of the first derivative, or even the condition number.

**Theorem 4.1 (Smale 1986; see also Traub and Wozniakowski 1979)** Let $f : \mathbb{C}^n \to \mathbb{C}^n$, $\zeta \in \mathbb{C}^n$ with $f(\zeta) = 0$. If

$$\gamma(f, \zeta)\|z - \zeta\| \leq \frac{3 - \sqrt{7}}{2},$$

then $z$ is an approximate zero of $f$ with associated zero $\zeta$.

Now let

$$\alpha = \alpha(f, z) = \beta(f, z)\gamma(f, z), \quad \beta(f, z) = \|Df(z)^{-1}f(z)\|.$$

**Theorem 4.2 (Smale 1986)** There exists a universal constant $\alpha_0 > 0$ such that: if $\alpha(f, z) < \alpha_0$ for $f : \mathbb{C}^n \to \mathbb{C}^n$, $z \in \mathbb{C}^n$, then $z$ is an approximate zero of $f$ (for some associated actual zero $\zeta$ of $f$).

**Remark 4.1** This is the result that motivates 'point estimates'. One uses it to conclude that $z$ is an approximate zero $f$ by checking an estimate at the point $z$ only. Nothing is assumed about $f$ in a region or $f$ at $\zeta$.

**Remark 4.2** For this definition of approximate zero, the best value of $\alpha_0$ is probably no smaller than 1/10. See developments, details and discussions in Smale (1987*a*), Wang (1993), Bez I, and BCSS (1997).

Now how might one estimate $\gamma$? In Smale (1986, 1987*a*), there is an estimate in terms of the first derivative of $f$, but an estimate in Bez I seems much more useful. In the context of Section 3, let $f \in \mathcal{H}_d$, $\zeta \in \mathbb{C}^{n+1}$, $f(\zeta) = 0$, and $\gamma_0(f, \zeta) = \|\zeta\|\gamma(f, \zeta)$. The last is to make $\gamma$ projectively invariant. Recall that $D = \max(d_i)$, $d = (d_1, \ldots, d_n)$, $d_i = \deg f_i$.

**Theorem 4.3 (Bez I)**

$$\gamma_0(f, \zeta) \leq \frac{D^2}{2}\mu(f, \zeta).$$

Recall that $\mu(f, \zeta)$ is the condition number.

**Remark 4.3** One has a similar estimate without assuming $f(\zeta)=0$.

As a corollary of Theorem 4.3 and a projective version of Theorem 4.1, one obtains the following.

**Theorem 4.4 (Separation of zeros, Malajovich-Munoz 1993, BCSS 1997, Dedieu 1997*b*, 1997*d*)** Let $f \in \mathcal{H}_d$, and $\zeta$, $\zeta'$ be two distinct zeros of $f$. Then

$$\begin{aligned} d(\zeta,\zeta') &\geq \frac{3-\sqrt{7}}{D^2\mu(f)}, \\ D &= \max(\deg f_i), \qquad f=(f_1,\ldots,f_n), \\ \mu(f) &= \max_{\zeta,\ f(\zeta)=0} \mu(f,\zeta) \text{ is the condition number of } f, \\ & \text{and } d \text{ is the distance in projective space.} \end{aligned}$$

**Remark 4.4** One has also the stronger result

$$d(\zeta,\zeta') \geq \frac{3-\sqrt{7}}{D^2\mu(f,\zeta)}.$$

**Remark 4.5** The strength of Theorem 4.4 lies in its global aspect. It is not asymptotic even though $\mu$ is defined just by a derivative.

We end this section by stating a global perturbation theorem (Dedieu (1997*b*)).

**Theorem 4.5** Let $f$, $g:\mathbb{C}^n \to \mathbb{C}^n$, $\zeta \in \mathbb{C}^n$ with $f(\zeta)=0$. Then, if

$$\alpha(g,\zeta) \leq \frac{|3-3\sqrt{17}|}{4} \qquad \text{and}$$

$$\|I - Df(\zeta)^{-1}Dg(\zeta)\| \leq \frac{9-\sqrt{17}}{16},$$

there is a zero $\zeta'$ of $g$ such that

$$\|\zeta-\zeta'\| \leq 2\mu(f,\zeta)\|f-g\|.$$

Here everything is affine including $\mu(f,\zeta)$. This uses Theorem 4.2.

## 5. Linear algebra

Complexity theory is quite implicit in the numerical linear algebra literature. Indeed, numerical analysts have studied the execution time and memory requirements for many linear algebra algorithms. This is particularly true for direct algorithms that solve a problem (such as a linear system of equations) in a finite number of steps. On the other hand, for more difficult linear algebra problems (such as the matrix eigenvalue problem) where iterative methods are needed, the complexity theory is not fully developed. It is our belief that a more detailed complexity analysis is desirable and such a study could help lead to better algorithms in the future.

### *5.1. Linear systems*

Consider the classical problem of a system of linear equations $Ax = b$, where $A$ is a $n \times n$ non-singular matrix, $b$ is a column vector of length $n$. The standard method for solving this problem is Gaussian elimination (say, with partial pivoting). The number of arithmetic operations required for this method can be found in most numerical analysis textbooks: it is $2n^3/3 + O(n^2)$. Most of these operations come from the $LU$ factorization of the matrix $A$, with suitable row exchanges. Namely, $PA = LU$, where $L$ is a unit lower triangular matrix (whose entries satisfy $|l_{ij}| \leq 1$), $U$ is an upper triangular matrix, and $P$ is the permutation matrix representing the row exchanges. When this factorization is completed, the solution of $Ax = b$ can be found in $O(n^2)$ operations. Similar operation counts are also available for other direct methods for linear systems, for example, the Cholesky decomposition for symmetric positive definite matrices. Another method for solving $Ax = b$, and, more importantly, for least squares problems, is to use the $QR$ factorization of $A$. The number of required operations is $4n^3/3 + O(n^2)$. All these direct methods for linear systems involve only a finite number of steps to find the solution. The complexity of these methods can be found by counting the total number of arithmetic operations involved.

A related problem is to investigate the average loss of precision for solving linear systems. It is well known that the condition number $\kappa$ of the matrix $A$ bounds the relative errors introduced in the solution by small perturbations in $b$ and $A$. Therefore, $\log \kappa$ is a measure of the loss of numerical precision. To find its average, a statistical analysis is needed. The following result for the expected value of $\log \kappa$ is obtained by Edelman.

**Theorem 5.1 (Edelman 1988)** Let $A$ be a random $n \times n$ matrix whose entries (real and imaginary parts of the entries, for the complex case) are independent random variables with the standard normal distribution, and let $\kappa = ||A|| \, ||A^{-1}||$ be its condition number in the 2-norm; then

$$E(\log \kappa) = \log n + c + o(1), \quad \text{for} \quad n \to \infty,$$

where $c \approx 1.537$ for real random matrices and $c \approx 0.982$ for complex random matrices.

The above result on the average loss of precision is a general result valid for any method, as a lower bound. If one uses the singular value decomposition to solve $Ax = b$, the average loss of precision should be close to $E(\log \kappa)$ above. For a more practical method like Gaussian elimination with partial pivoting, the same average could be larger. In fact, Wilkinson's backward error analysis reveals that the numerical solution $\hat{x}$ obtained from a finite precision calculation is the exact solution of a perturbed system $(A+E)\hat{x} = b$. The magnitude of $E$ could be larger than the round-off of $A$ by an extra growth factor $\rho(A)$. This gives rise to the extra loss of precision caused by the particular method used, namely, Gaussian elimination with partial pivoting. Well-known examples indicate that the growth factor can be as large as $2^{n-1}$. But the following result suggests that large growth factors only rarely appear exponentially.

**Conjecture 5.1 (Trefethen)** For any fixed constant $p > 0$, let $A$ be a random $n \times n$ matrix, whose entries (real and complex parts of the entries for the complex case, scaled by $\sqrt{2}$) are independent samples of the standard normal distribution. Then, for all sufficiently large $n$,

$$\text{Prob}\,(\rho(A) > n^{\alpha}) < n^{-p},$$

where $\alpha > 1/2$.

For iterative methods, we mention that a complexity result is available for the conjugate gradient method (Hestenes and Stiefel 1952). Let $A$ be a real symmetric positive definite matrix, $x_0$ be an initial guess for the exact solution $x_*$ of $Ax = b$, and $x_j$ be the $j$th iterate of the conjugate gradient method. Then the following result is well known (Axelsson (1994), Appendix B):

$$\|x_j - x_*\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^j \|x_0 - x_*\|_A,$$

where the $A$-norm of a vector $v$ is defined as $\|v\|_A = (v^TAv)^{1/2}$. From this, one easily concludes that if

$$j \geq \frac{\log \frac{2}{\epsilon}}{\log\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)} = O\left(\frac{\sqrt{\kappa}}{2}\log\frac{2}{\epsilon}\right),$$

then $\|x_j - x_*\|_A \leq \epsilon\|x_0 - x_*\|_A$.

### *5.2. Eigenvalue problems*

In this subsection, we consider a number of basic algorithms for eigenvalue problems. Complexity results for these methods are more difficult to obtain.

For a matrix $A$, the power method approximates the eigenvector corresponding to the dominant eigenvalue (largest in absolute value). If there is one dominant eigenvalue, for almost all initial guesses $x_0$, the sequence generated by the power method $x_j = A^jx_0/\|A^jx_0\|$ converges to the dominant eigenvector. A statistical complexity analysis for the power method tries to determine the average number of iterations required to produce an approximation to the exact eigenvector, such that the angle between the approximate and exact eigenvectors is less than a given small number $\epsilon$ ($\epsilon$-dominant eigenvector). These questions have been studied by Kostlan. The average is first taken for all initial guesses $x_0$ and a fixed matrix $A$, then extended to all matrices for some distribution.

**Theorem 5.2 (Kostlan 1988)** For any real symmetric $n \times n$ matrix $A$ with eigenvalues $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$, the number of iterations $\tau_\epsilon(A)$ required for the power method to produce an $\epsilon$-dominant eigenvector, averaged over all initial vectors, satisfies

$$\frac{\log \cot \epsilon}{\log|\lambda_1| - \log|\lambda_2|} < \tau_\epsilon(A) < \frac{\frac{1}{2}[\psi(n/2) - \psi(1/2)] + \log \cot \epsilon}{\log|\lambda_1| - \log|\lambda_2|} + 1,$$

where $\psi(x) = \Gamma'(x)/\Gamma(x)$.

When an average is taken for the set of $n \times n$ random real symmetric matrices (the entries are independent random variables with Gaussian distributions of zero mean, the variance of any diagonal entry is twice the variance of any off-diagonal entry), the required number of iterations is infinite. However, a finite bound can be obtained if a set of 'bad' initial guesses and 'bad' matrices of normalized measure $\eta$ are excluded.

**Theorem 5.3 (Kostlan 1988)** For the above $n \times n$ random real symmetric matrix, with the probability $1 - \eta$, the average required number of iterations to produce an $\epsilon$-dominant eigenvector satisfies

$$\tau_{\epsilon,\eta} < \frac{3n(n+1)}{4\sqrt{2}\eta} \left(\psi(n/2) - \psi(1/2) + 2\log\cot\epsilon\right).$$

Similar results hold for complex Hermitian matrices. Furthermore, a finite bound on random symmetric positive definite matrices is also available. Statistical complexity analysis for a different method of dominant eigenvector calculation can be found in Kostlan (1991).

In practice, the Rayleigh quotient iteration method is much more efficient. Starting from an initial guess $x_0$, a sequence of vectors $\{x_j\}$ is generated from

$$\mu = \frac{x_{j-1}^T A x_{j-1}}{x_{j-1}^T x_{j-1}}, \quad (A - \mu I)y = x_{j-1}, \quad x_j = \frac{y}{\|y\|}.$$

For symmetric matrices, the following global convergence result has been established.

**Theorem 5.4 (Ostrowski 1958, Parlett and Kahan 1969, Batterson and Smillie 1989)** Let $A$ be a symmetric $n \times n$ matrix. For almost any choice of $x_0$, the Rayleigh quotient iteration sequence $\{x_j\}$ converges to an eigenvector and $\lim_{j\to\infty}\theta_{j+1}/\theta_j^3 \leq 1$, where $\theta_j$ is the angle between $x_j$ and the closest eigenvector.

A statistical complexity analysis for this method is still not available. In fact, even for a fixed symmetric matrix $A$, there is no upper bound on the number of iterations required to produce a small angle, say, $\theta_j < \epsilon$ for a small constant $\epsilon$. In general, for a given initial vector $x_0$, one can not predict which eigenvector it converges to (if the sequence does converge). On the other hand, for nonsymmetric matrices, we have the following result on non-convergence.

**Theorem 5.5 (Batterson and Smillie 1990)** For each $n \geq 3$, there is a nonempty open set of matrices, each of which possesses an open set of initial vectors for which the Rayleigh quotient iteration sequence does not converge to an invariant subspace.

Practical numerical methods for matrix eigenvalue problems are often based on reductions to the condensed forms by orthogonal similarity transformations. For an $n\times n$ symmetric matrix $A$, one typically uses Householder reflections to obtain a symmetric tridiagonal matrix $T$. The reduction step is a finite calculation that requires $O(n^3)$ arithmetic operations. While many numerical methods are available for calculating the eigenvalues and eigenvectors of symmetric tridiagonal matrices, we see the lack of a complexity analysis for these methods.

The $QR$ method with Wilkinson's shift always converges; see Wilkinson (1968). In this method, the tridiagonal matrix $T$ is replaced by $sI + RQ$ (still symmetric tridiagonal), where $s$ is the eigenvalue of the last $2\times 2$ block of $T$ that is closer to the $(n,n)$ entry of $T$, and $QR = T - sI$ is the $QR$ factorization of $T - sI$. Wilkinson proved that the $(n, n-1)$ entries of this sequence of $T$ always converge to zero. Hoffman and Parlett (1978) gave a simpler proof for the global linear convergence. The following is an easy corollary of their result.

**Theorem 5.6** Let $T$ be a real symmetric $n \times n$ tridiagonal matrix. For any $\epsilon > 0$, let $m$ be a positive integer satisfying

$$m > 6\log_2\frac{1}{\epsilon} + \log_2(T_{n,n-1}^4 T_{n-1,n-2}^2) + 1.$$

Then, after $m$ $QR$ iterations with Wilkinson's shift, the last subdiagonal entry of $T$ satisfies

$$|T_{n,n-1}| < \epsilon.$$

It would be interesting to develop better complexity results based on the higher asymptotic convergence rate. Alternative definitions for the last subdiagonal entry to be sufficiently small are desirable, because the usual decoupling criterion is based on a comparison with the two adjacent diagonal entries.

The divide and conquer method suggested by Cuppen (1981) calculates the eigensystem of an unreduced symmetric tridiagonal matrix based on the eigensystems of two tridiagonal matrices of half size and a rank-one updating scheme. The computation of the eigenvalues is reduced to solving the following nonlinear equation

$$1+\rho\sum_{j=1}^{n}\frac{c_j^2}{d_j-\lambda}=0,$$

where $\{d_j\}$ are the eigenvalues of the two smaller matrices and $\{c_j\}$ are related to their eigenvectors. This method is complicated by the possibilities that the elements in $\{d_j\}$ may be not distinct and the set $\{c_j\}$ may contain zeros. Dongarra and Sorensen (1987) developed an iterative method for solving the nonlinear equation based on simple rational function approximations. See Bini and Pan (1994) for a complexity analysis of a related algorithm.

A related method for computing just the eigenvalues uses the set $\{d_j\}$ to separate the eigenvalues and a nonlinear equation solver for the characteristic polynomial. In Du, Jin, Li and Zeng (1997*b*), the quasi-Laguerre method is used. An asymptotic convergence result has been established in Du, Jin, Li and Zeng (1997*a*), but a complexity analysis is still not available. The method is complicated by the switch to other methods (the bisection or Newton's method) to obtain good starting points for the quasi-Laguerre iterations.

For a general real nonsymmetric matrix $A$, the $QR$ iteration with Francis's double shift is widely used to triangularize the Hessenberg matrix $H$ obtained from the reduction by orthogonal similarity transformations from $A$. In this case, there are simple examples for which the $QR$ iteration does not lead to a decoupling. In Batterson and Day (1992), matrices where the asymptotic rate of decoupling is only linear are identified. For normal Hessenberg matrices, Batterson discovered the precise conditions for decoupling under the $QR$ iteration. See Batterson (1994) for details. To develop a statistical complexity analysis for this method is a great challenge.

## 6. Complexity in many variables

Consider the problem of following a path, implicitly defined, by a computationally effective algorithm. Let $\mathcal{H}_d$ be as in Section 3.

Let $F : [0,1] \to \mathcal{H}_d \times \mathbb{C}^{n+1}$, $F(t) = (f_t, \zeta_t)$, satisfy $f_t(\zeta_t) = 0$, $0 \le t \le 1$, with the derivative $Df_t(\zeta_t)$ having maximum rank. For example, $\zeta_t$ could be given by the implicit function theorem from $f_t$ and the initial $\zeta_0$ with $f_0(\zeta_0) = 0$.

Next, suppose $[0,1]$ is partitioned into $k$ parts by $t_0 = 0$, $t_i = t_{i-1} + \Delta t$, $\Delta t = 1/k$; thus $t_k = 1$.

Define via Newton's method $\hat{N}_{f_{t_i}}$

$$z_i = \hat{N}_{f_{t_i}}(z_{i-1}), \quad i = 1, \ldots, k, \quad z_0 = \zeta_0. \tag{6.1}$$

For sufficiently small $\Delta t$, the $z_i$ are well defined and are good approximations of $\zeta_i$. But $k = 1/\Delta t$ represents the complexity, so the problem is to avoid taking $\Delta t$ much smaller than necessary. What is a sufficient number of Newton steps?

**Theorem 6.1 (The main theorem of Bez I)** The biggest integer $k$ in

$$cLD^2\mu^2$$

is sufficient to yield $z_i$ by (6.1) which is an approximate zero of $f_{t_i}$ with associated actual zero $\zeta_{t_i}$, each $i = 1, \ldots, k$.

In this estimate $c$ is a rather small universal constant, $L$ is the length of the curve $f_t$ in the projective space, $P(\mathcal{H}_d)$, $0 \leq t \leq 1$, $D$ is the max of the $d_i$, $i = 1, \ldots, n$ and $\mu = \max_{0 \leq t \leq 1} \mu(f_t, \zeta_t)$, where $\mu(f_t, \zeta_t)$ is the condition number as defined in Section 3.

Newton's method and approximate zero have been adapted to projective space. Thus $\hat{N}_f$ for $f \in \mathcal{H}_d$ at $z \in \mathbb{C}^{n+1}$ is the ordinary Newton method applied to the restriction of $f$ to

$$z + \left\{y \in \mathbb{C}^{n+1} : \langle y, z \rangle = 0\right\}.$$

As a consequence of the Condition Number Theorem and Theorem 6.1, the complexity depends mainly on how close the path $(f_t, \zeta_t)$ comes to the set of ill-conditioned problems. An improved proof of Theorem 6.1 may be found in BCSS (1997).

For earlier work on complexity theory for Newton's method in several variables, see Renegar (1987*a*). Malajovich (1994) has implemented the algorithm and developed some of the ideas of Bez I.

The main theorem of the final paper of the series Bez I–Bez V is as follows.

**Theorem 6.2** The average number of arithmetic operations sufficient to find an approximate zero of a system $f : \mathbb{C}^n \to \mathbb{C}^n$ of polynomials is polynomially bounded in the input size (the number of coefficients of $f$).

On one hand, this result is surprising, because it gives a polynomial time bound for a problem that is almost intractable. On the other hand, the 'algorithm' is not uniform: it depends on the degrees of the $(f_i)$ and even the desired probability of success. Moreover, the algorithm isn't known! It is only proved to exist. Thus Theorem 6.2 cries out for understanding and development. In fact, Mike Shub and I were unable to find a sufficiently good exposition to include in BCSS (1997).

Since deciding if there is a solution to $f : \mathbb{C}^n \to \mathbb{C}^n$ is unlikely to be accomplished in polynomial time, even using exact arithmetic (see Section 8), an astute analysis of Theorem 6.2 can give insight into the basic problem 'What are the limits of computation?' For example, is it 'on the average' that gives the possibility of polynomial time?

A real (rather than complex) analogue of Theorem 6.2 also remains to be found.

Let us give some mathematical detail about the statement of Theorem 6.2. An 'approximate zero' has been defined in Section 4, as, of course, exact zeros cannot be found (Abel, Galois, et al.). Averaging is performed relative to a measure induced by the unitarily invariant inner product on homogenized polynomials of degree $d = (d_1, \dots, d_n)$, where $d_i = \deg f_i$, $f = (f_1, \dots, f_n)$ (see Section 3). If $N = N(d)$ is the number of coefficients of such a system $f$, then unless $n \leq 4$ or some $d_i = 1$, the number of arithmetic operations is bounded by $cN^4$. If $n \leq 4$ or some $d_i = 1$, then we get $cN^5$.

An important special case is that of quadratic systems, when $d_i = 2$ and so $N \leq n^3$. Then the average arithmetic complexity is bounded by a polynomial function of $n$.

'On the average' in the main result is needed because certain polynomial systems, even affine ones of the type $f : \mathbb{C}^2 \to \mathbb{C}^2$, have one-dimensional sets of zeros, extremely sensitive to any (practical) real number algorithm; one would say such $f$ are ill posed.

The algorithm (non-uniform) of the theorem is similar to those of Section 2. It is a continuation method where each step is given by Newton's method (the step size $\Delta t$ is no longer a constant). The continuation starts from a given 'known' pair $g : \mathbb{C}^{n+1} \to \mathbb{C}^n$ and $\zeta \in \mathbb{C}^{n+1}$, $g(\zeta) = 0$. It is conjectured in Bez V that one could take for $g$, the system defined by $g_i(z) = z_0^{d_i - 1} z_i$, $i = 1, \dots, n$ and $\zeta = (1, 0, \dots, 0)$. A proof of this conjecture would yield a uniform algorithm.

Finally, we remark that in Bez V, Theorem 6.2 is generalized to the problem of finding $\ell$ zeros, when $\ell$ is any number between one and the Bézout number $\prod_{i=1}^n d_i$ and the number of arithmetic operations is augmented by the factor $\ell^2$.

The proof of Theorem 6.2 uses Theorem 6.1 and the geometric probability methods of the next section.

## 7. Probabilistic estimates

As described in the Introduction, our complexity perspective has two parts, and the second deals with probability estimates of the condition number. We have already seen some aspects of this in Sections 2 and 5. Here are some further results.

Section 3 describes a condition number for studying zeros of polynomial systems of equations. We have dealt especially with the homogeneous setting and defined projective condition number $\mu(f,\zeta)$ for $f \in \mathcal{H}_d$, $d = (d_1, \dots, d_n)$, degree $f_i = d_i$, and $\zeta \in \mathbb{C}^{n+1}$ with $f(\zeta) = 0$. Then

$$\mu(f) = \max_{\zeta,\ f(\zeta)=0} \mu(f,\zeta).$$

The unitarily invariant inner product (Section 3) on $\mathcal{H}_d$ induces a probability measure on $\mathcal{H}_d$ (or equivalently on the projective space $P(\mathcal{H}_d)$). With this measure the following is proved in Bez II.

**Theorem 7.1**

$$\text{Probability}\left\{f \in \mathcal{H}_d \,:\, \mu(f) > \frac{1}{\varepsilon}\right\} \leq C_d \varepsilon^4$$

$$C_d = n^3(n+1)(N-1)(N-2)\mathcal{D}, \qquad N = \dim \mathcal{H}_d, \qquad \mathcal{D} = \prod_{i=1}^{n} d_i$$

In the background of this and a number of related results is a geometric picture (from geometric probability theory), briefly described as follows. It is convenient to use the projective spaces $P(\mathcal{H}_d)$, $P(\mathbb{C}^{n+1})$ and their product for the environment of this analysis. Define $V$ to be the subset of ordered pairs (system, solution):

$$V = \Big\{(f,\zeta) \in P(\mathcal{H}_d) \times P(\mathbb{C}^{n+1}) : f(\zeta) = 0\Big\}.$$

Let $\pi_1 : V \to P(\mathcal{H}_d)$, $\pi_2 : V \to P(\mathbb{C}^{n+1})$ be the restrictions of the corresponding projections, as shown below.

$$\begin{tikzcd}
& V \subset P(\mathcal{H}_d) \times P(\mathbb{C}^{n+1}) \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
P(\mathcal{H}_d) & & P(\mathbb{C}^{n+1})
\end{tikzcd}$$

**Theorem 7.2 (Bez II)** Let $U$ be an open set in $V$, then

$$\int_{x \in P(\mathcal{H}_d)} \#\Big(\pi_1^{-1}(x) \cap U\Big) = \int_{z \in P(\mathbb{C}^{n+1})} \int_{(a,z) \in \pi_2^{-1}(z) \cap U} \det\Big(DG(a)DG(a)^*\Big)^{-1/2}$$

Here $DG(a)$ is the condition matrix, $DG(a)^*$ its adjoint and # means cardinality.

This result and the underlying theory is valid in great generality (see Bez II, IV, V, BCSS (1997)).

There is one aspect of these results and arguments that is quite unsettling and pervades Bez II–V: the implicit existence theory is not very constructive.

Consider the simplest case (Bez III). For the moment, let $d > 1$ be an integer and $\mathcal{H}_d$ the space of homogeneous polynomials in two variables of degree $d$. It follows from the above geometric probability arguments that there is a subset $S_d$ of $P(\mathcal{H}_d)$ of probability measure larger than one-half such that, for $f \in S_d$, $\mu(f) \leq d$.

**Problem 7.1 (Bez III)** Construct a family $\{f_d \in \mathcal{H}_d : d = 2, 3, \ldots\}$ so that

$$\mu(f_d) \leq d, \qquad \text{or even} \quad \mu(f_d) \leq d^c,$$

for $c$ any constant.

By 'construct', we mean to provide a polynomial time algorithm (*e.g.* in the sense of the machine of Section 8) which, given input $d$, outputs $f_d$ satisfying the above condition. (This amounts to constructing elliptic Fekete polynomials.) See also Rakhmanov, Saff and Zhou (1994, 1995).

Another example of an application of the above setting of geometric probability is the following result. For $d = (d_1, \ldots, d_n)$, let $\mathcal{H}_d^{\mathbb{R}}$ denote the space of real homogeneous systems $(f_1, \ldots, f_n)$ in $n+1$ variables with degree $f_i = d_i$. One can average just as before and obtain the following.

**Theorem 7.3 (Bez II)** The average number of real zeros of a real homogeneous polynomial system is exactly the square root of the Bézout number $\mathcal{D} = \prod_{i=1}^{n} d_i$ ($\mathcal{D}$ being the number of complex solutions).

See Kostlan (1993) for earlier special cases. See also Edelman and Kostlan (1995).

For the complexity results of Bez IV, V, Theorem 7.1 is inadequate. There one has similar theorems where the maximum of the condition number along an interval is estimated.

## 8. Real machines

Up to now, our discussion might be called the complexity analysis of algorithms, or upper bounds for the time required to solve problems. To complement this theory one needs lower bound estimates for problem solving.

For this endeavour, one must consider all possible algorithms that solve a given problem. In turn this needs a formal definition and the development of algorithms and machines. The traditional Turing machine is ill-suited for this purpose, as is argued in the Manifesto. A 'real number machine' is the most natural vehicle to deal with problem-solving schemes based on Newton's method, for example.

There is a recent development of such a machine in BSS (1989) and BCSS (1997), which we will review very briefly.

Each input is a string $y$ of real numbers of the form

$$\cdots 000 y_1 \cdots y_n 000 \cdots;$$

the size $S(y)$ of $y$ is $n$. These inputs may be restricted to code an instance of a problem. An 'input node' transforms an input into a state string.

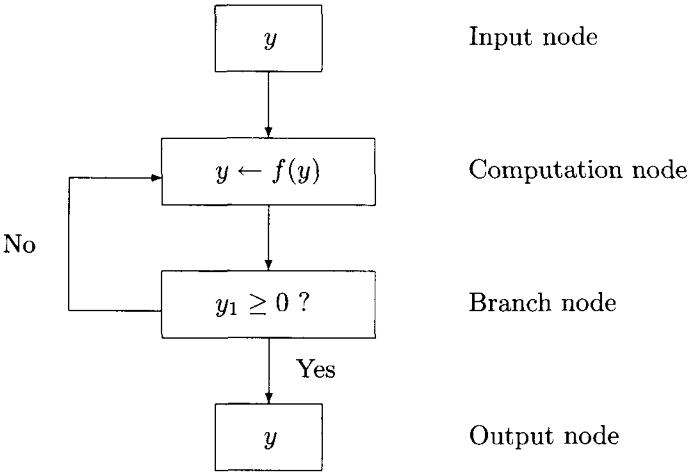

Fig. 1. Example of a real number machine

The computation node replaces the state string by a shifted one, right or left shifted, or does an arithmetic operation on the first elements of the string. The branch nodes and output nodes are self-explanatory.

The definition of a real machine (or a 'machine over $\mathbb{R}$') is suggested by the example and consists of an input node and a finite number of computation, branch, and output nodes organized into a directed graph. It is the flow chart of a computer program seen as a mathematical object. One might say that this real number machine is a 'real Turing machine' or an idealized Fortran program.

The *halting set* of a real machine is the set of all inputs such that, acting on the nodal instructions, we eventually land on an output node. An *input–output map* $\phi$ is defined on the halting set by 'following the flow' of the flow chart. For precise definitions and developments see BSS (1989) and BCSS (1997).

A machine has *polynomial time complexity* (sometimes with a restricted class of inputs) if it enjoys the property

$$T(y) \leq S(y)^c, \qquad \text{for all inputs } y, \tag{8.1}$$

where $c$ is independent of $y$. In this estimate, $T(y)$ is the time to the output for the input $y$ measured by the number of nodes encountered in the computation of $\phi(y)$. Recall that the size $S(y)$ of $y$ is the length of the input string $y$.

If the size of the inputs is bounded, and there are no loops, *i.e.*, the machine is a tree of nodes, then one has a *tame machine*, or an algebraic computation tree. These objects have been used to obtain lower bounds for real number problems. One such development is that of Steele and Yau (1982) and Ben-Or (1983), based on a real algebraic geometry estimate of Oleinik and Petrovski (1949), Oleinik (1951), Milnor (1964) and Thom (1965). Another is that of Smale (1987*b*) and Vassiliev (1992), and based on the cohomology of the braid group.

Lower bounds tend to be modest and difficult to obtain, but are necessary for the understanding of the fundamental problem: 'What are the limits of computation?'

Note that the definition of a real machine is valid with strings of numbers lying in any field if one replaces the branch node with the question, '$y_1 = 0$?' If this field is the field of two elements, one has a Turing machine, and the size becomes the number of bits. If one uses complex numbers, then one has a 'complex machine'.

**Side remarks 8.1** The study of zeros of polynomial systems plays a central role in both mathematics and computation theory. Deciding whether a set of polynomial equations has a zero over $\mathbb{R}$ is even universal in a formal sense in the theory of real computation. This problem is called 'NP-complete over $\mathbb{R}$' and hence its solution in polynomial time is equivalent to 'P = NP over $\mathbb{R}$.' For machines over $\mathbb{C}$, this problem is that of the Hilbert Nullstellensatz, and Brownawell's (1987) work was critical in getting the fastest-known algorithm (but not polynomial time!) The relation to NP-complete over $\mathbb{C}$ and 'P = NP over $\mathbb{C}$' is as in the real case. The same applies to the field $\mathbb{Z}_2$ of two elements and 'P = NP over $\mathbb{Z}_2$?' is the same as the classical Cook–Karp problem 'P = NP?' of computer science. See BCSS (1997).

My own belief is that this problem is one of the three great unsolved problems of mathematics (together with the Riemann hypothesis and Poincaré's conjecture in three dimensions).

The rest of Section 8 is more tentative, as we present suggestions in the direction of a 'second generation' real machine.

For an input $y$ of a problem, an extended notion of size still denoted by

$S(y)$ could be convenient. The extended notion would be the maximum of the length of the string (*i.e.* the previously defined size) and other ingredients, as follows:

(i) the condition number $\mu(y)$, or its log, or similar invariants of $y$
(ii) the precision $\log \varepsilon^{-1}$, where $\varepsilon$ is the required accuracy (or perhaps, depending on the problem, $\varepsilon$, or even $\log\log \varepsilon^{-1}$) of the output
(iii) for integer machines, the number of bits.

It is convenient to consider the traditional size of the input as part of the input (BSS 1989, BCSS 1997). Should the same hold for the extended size? We won't try to give a definitive answer here. Part of this answer is a question of convenience, part interpretation. Should the algorithm assume that the condition number is known explicitly? Probably not, at least very generally. On the other hand, if one has a good theoretical result on the distribution, one can make some guess about the condition number. This can to some extent justify taking the condition number of the particular problem as input. It is analogous, for example, to running a path-following program inputing an initial step size as a guess.

Let me give an example of an open problem that fits into this framework. Let $d = (d_1, \dots, d_m)$ and $\mathcal{P}_{n,d}$ be the space of $m$-tuples of real polynomials $f = (f_1, \dots, f_m)$ in $n$ variables with $\deg f_i \leq d_i$. Put some distance $D$ on $\mathcal{P}_{n,d}$. Say that $f$ is *feasible* if the system of inequalities $f_i(x) \geq 0$, all $i = 1, \dots, m$ has a solution $x \in \mathbb{R}^n$. Let the 'condition number' of $f$ be defined by:

$$\mu(f) = \left(\inf_{g \text{ not feasible}} D(f,g)\right)^{-1} \quad \text{if } f \text{ is feasible,}$$

$$\mu(f) = \left(\inf_{g \text{ feasible}} D(f,g)\right)^{-1} \quad \text{if } f \text{ is not feasible.}$$

Let the extended size $S(f)$ of $f \in \mathcal{P}_{n,d}$ be the maximum (perhaps $\infty$) of $\dim \mathcal{P}_{n,d}$ and $\mu(f)$.

**Problem 8.1** Is there a polynomial time algorithm deciding the above feasibility problem using the extended size?

The problem is formalized in terms of the real machines described above, using exact arithmetic in particular.

We now propose an extension of the earlier notion of real machine to allow round-off error in the computation.

A *round-off machine over* $\mathbb{R}$ is a real machine, together with a function of inputs that, at each input, computation and output node, adds a state vector of magnitude less than some positive constant $\delta$. One has no *a priori* knowledge of the added state vector (it's an adversary). This idealization has the virtue of simplicity; we hope this compensates for its ignorance of important detail.

A problem will be called *robustly solvable* if it can be solved for inputs of finite extended size by a round-off machine, no matter what the round-off error.

More important is the concept of *robustly solvable* in *polynomial time*. In addition to the estimate (8.1) with extended size, $S(y)$, one adds a requirement such as

$$\frac{1}{\delta(y)} \leq S(y)^c. \tag{8.2}$$

One can now sharpen Problem 8.1 to ask for a decision which is robustly solvable in polynomial time.

The above gives some sense of the notion of a robust or numerically stable algorithm, perhaps improving on the attempts in Isaacson and Keller (1966), Wozniakowski (1977), Smale (1990) and Shub (1993).

## 9. Some other directions

Many aspects of complexity theory in numerical analysis have not been dealt with in this brief report. We now refer to some of these omissions.

A general reference is Renegar, Shub and Smale (1997), which expands on the previous topics and those below.

There is the important, well-developed field of algebraic complexity theory, which relates very much to some of our account. I have the greatest admiration for this work, but will only mention here Bini and Pan (1994), Grigoriev (1987), and Giusti et al. (1997).

Also well-developed is the area of information-based complexity. In spite of its relevance and importance to our review, I will only mention Traub, Wasilkowski and Wozniakowski (1988), where one will find a good introduction and survey.

Another area in which the mathematical foundation and development are strong is the science of mathematical programming, or optimization. I believe that numerical analysts interested in complexity considerations can learn much from what has happened and is happening in that field. I especially like the perspective and work of Renegar (1996).

## REFERENCES

E. Allgower and K. Georg (1990), *Numerical Continuous Methods*, Springer.

E. Allgower and K. Georg (1993), Continuation and path following, in *Acta Numerica*, Vol. 2, Cambridge University Press, pp. 1–64.

O. Axelsson (1994), *Iterative Solution Methods*, Cambridge University Press.

S. Batterson (1994), 'Convergence of the Francis shifted $QR$ algorithm on normal matrices', *Linear Algebra Appl.* **207**, 181–195.

S. Batterson and D. Day (1992), 'Linear convergence in the shifted $QR$ algorithm', *Math. Comp.* **59**, 141–151.

S. Batterson and J. Smillie (1989), 'The dynamics of Rayleigh quotient iteration', *SIAM J. Numer. Anal.* **26**, 624–636.

S. Batterson and J. Smillie (1990), 'Rayleigh quotient iteration for nonsymmetric matrices', *Math. Comp.* **55**, 169–178.

M. Ben-Or (1983), Lower bounds for algebraic computation trees, in *15th Annual ACM Symposium on the Theory of Computing*, pp. 80–86.

D. Bini and V. Pan (1987), 'Sequential and parallel complexity of approximating polynomial zeros', *Computers and Mathematics (with applications)* **14**, 591–622.

D. Bini and V. Pan (1994), *Polynomial and Matrix Computations*, Birkhäuser, Basel.

L. Blum, F. Cucker, M. Shub and S. Smale (1996), 'Complexity and real computation: a manifesto', *Int. J. Bifurcation and Chaos* **6**, 3–26. **Referred to as the Manifesto.**

L. Blum, F. Cucker, M. Shub and S. Smale (1997), *Complexity and Real Computation*, Springer. To appear. **Referred to as BCSS (1997).**

L. Blum, M. Shub and S. Smale (1989), 'On a theory of computation and complexity over the real numbers: $NP$-completeness, recursive functions and universal machines', *Bull. Amer. Math. Soc.* **21**, 1–46. **Referred to as BSS (1989).**

R. Brockett (1973), in *Geometric Methods in Systems Theory, Proceedings of the NATO Advanced Study Institute* (D. Mayne and R. Brockett, eds), D. Reidel, Dordrecht.

W. Brownawell (1987), 'Bounds for the degrees in the Nullstellensatz', *Annals of Math.* **126**, 577–591.

G. Collins (1975), *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition*, Vol. 33 of *Lect. Notes in Comp. Sci.*, Springer, pp. 134–183.

J. J. M. Cuppen (1981), 'A divide and conquer method for the symmetric tridiagonal eigenproblem', *Numer. Math.* **36**, 177–195.

J.-P. Dedieu (1997*a*), Approximate solutions of numerical problems, condition number analysis and condition number theorems, in *Proceedings of the Summer Seminar on 'Mathematics of Numerical Analysis: Real Number Algorithms', AMS Lectures in Applied Mathematics* (J. Renegar, M. Shub and S. Smale, eds), AMS, Providence, RI. To appear.

J.-P. Dedieu (1997*b*), Condition number analysis for sparse polynomial systems. Preprint.

J.-P. Dedieu (1997*c*), 'Condition operators, condition numbers and condition number theorem for the generalized eigenvalue problem', *Linear Algebra Appl.* To appear.

J.-P. Dedieu (1997*d*), 'Estimations for separation number of a polynomial system', *J. Symbolic Computation*. To appear.

J. Dégot and B. Beauzamy (1997), 'Differential identities', *Trans. Amer. Math. Soc.* To appear.

B. Dejon and P. Henrici (1969), *Constructive Aspects of the Fundamental Theorem of Algebra*, Wiley.

J. Demmel (1987), 'On condition numbers and the distance to the nearest ill-posed problem', *Numer. Math.* **51**, 251–289.

J. J. Dongarra and D. C. Sorensen (1987), 'A fully parallel algorithm for the symmetric eigenvalue problem', *SIAM J. Sci. Statist. Comput.* **8**, 139–154.

Q. Du, M. Jin, T. Y. Li and Z. Zeng (1997*a*), 'The quasi-Laguerre iteration', *Math. Comp.* To appear.

Q. Du, M. Jin, T. Y. Li and Z. Zeng (1997*b*), 'Quasi-Laguerre iteration in solving symmetric tridiagonal eigenvalue problems', *SIAM J. Sci. Comput.* To appear.

C. Eckart and G. Young (1936), 'The approximation of one matrix by another of lower rank', *Psychometrika* **1**, 211–218.

A. Edelman (1988), 'Eigenvalues and condition numbers of random matrices', *SIAM J. Matrix Anal. Appl.* **9**, 543–556.

A. Edelman and E. Kostlan (1995), 'How many zeros of a random polynomial are real?', *Bull. Amer. Math. Soc.* **32**, 1–38.

C. F. Gauss (1973), *Werke*, Band X, Georg Olms, New York.

M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern and L. M. Pardo (1997), 'Straight-line program in geometric elimination theory', *Journal of Pure and Applied Algebra*. To appear.

G. Golub and C. van Loan (1989), *Matrix Computations*, Johns Hopkins University Press.

D. Grigoriev (1987), in *Computational complexity in polynomial algebra, Proceedings of the International Congress Math. (Berkeley, 1986)*, Vol. 1, 2, AMS, Providence, RI, pp. 1452–1460.

P. Henrici (1977), *Applied and Computational Complex Analysis*, Wiley.

M. R. Hestenes and E. Stiefel (1952), 'Method of conjugate gradients for solving linear systems', *J. Res. Nat. Bur. Standards* **49**, 409–436.

M. Hirsch and S. Smale (1979), 'On algorithms for solving $f(x) = 0$', *Comm. Pure Appl. Math.* **32**, 281–312.

W. Hoffman and B. N. Parlett (1978), 'A new proof of global convergence for the tridiagonal *QL* algorithm', *SIAM J. Numer. Anal.* **15**, 929–937.

E. Isaacson and H. Keller (1966), *Analysis of Numerical Methods*, Wiley, New York.

H. Keller (1978), Global homotopic and Newton methods, in *Recent Advances in Numerical Analysis*, Academic Press, pp. 73–94.

R. Kellog, T. Li and J. Yorke (1976), 'A constructive proof of Brouwer fixed-point theorem and computational results', *SIAM J. Numer. Anal.* **13**, 473–483.

M. Kim (1988), 'On approximate zeros and rootfinding algorithms for a complex polynomial', *Math. Comp.* **51**, 707–719.

E. Kostlan (1988), 'Complexity theory of numerical linear algebra', *J. Comput. Appl. Math.* **22**, 219–230.

E. Kostlan (1991), 'Statistical complexity of dominant eigenvector calculation', *J. Complexity* **7**, 371–379.

E. Kostlan (1993), On the distribution of the roots of random polynomials, in *From Topology to Computation: Proceedings of the Smalefest* (M. Hirsch, J. Marsden and M. Shub, eds), Springer, pp. 419–431.

G. Malajovich (1994), 'On generalized Newton algorithms: quadratic convergence, path-following and error analysis', *Theoret. Comput. Sci.* **133**, 65–84.

G. Malajovich-Munoz (1993), On the complexity of path-following Newton algorithms for solving polynomial equations with integer coefficients, PhD thesis, University of California at Berkeley.

J. M. McNamee (1993), 'A bibliography on roots of polynomials', *J. Comput. Appl. Math.* **47**(3), 391–394.

J. Milnor (1964), On the Betti numbers of real varieties, in *Proceedings of the Amer. Math. Soc.*, Vol. 15, pp. 275–280.

C. Neff (1994), 'Specified precision root isolation is in NC', *J. Comput. System Sci.* **48**, 429–463.

C. Neff and J. Reif (1996), 'An efficient algorithm for the complex roots problem', *J. Complexity* **12**, 81–115.

O. Oleinik (1951), 'Estimates of the Betti numbers of real algebraic hypersurfaces', *Mat. Sbornik (N.S.)* **28**, 635–640. In Russian.

O. Oleinik and I. Petrovski (1949), 'On the topology of real algebraic surfaces', *Izv. Akad. Nauk SSSR* **13**, 389–402. In Russian; English translation in *Transl. Amer. Math. Soc.* **1**, 399–417 (1962).

A. Ostrowski (1958), 'On the convergence of Rayleigh quotient iteration for the computation of the characteristic roots and vectors, I', *Arch. Rational Mech. Anal.* **1**, 233–241.

V. Pan (1997), 'Solving a polynomial equation: some history and recent progress', *SIAM Review*. To appear.

B. N. Parlett and W. Kahan (1969), 'On the convergence of a practical *QR* algorithm', *Inform. Process. Lett.* **68**, 114–118.

E. A. Rakhmanov, E. B. Saff and Y. M. Zhou (1994), 'Minimal discrete energy on the sphere', *Mathematical Research Letters* **1**, 647–662.

E. A. Rakhmanov, E. B. Saff and Y. M. Zhou (1995), Electrons on the sphere, in *Computational Methods and Function Theory* (R. M. Ali, S. Ruscheweyh and E. B. Saff, eds), World Scientific, pp. 111–127.

J. Renegar (1987*a*), 'On the efficiency of Newton's method in approximating all zeros of systems of complex polynomials', *Math. of Oper. Research* **12**, 121–148.

J. Renegar (1987*b*), 'On the worst case arithmetic complexity of approximating zeros of polynomials', *J. Complexity* **3**, 90–113.

J. Renegar (1996), 'Condition numbers, the Barrier method, and the conjugate gradient method', *SIAM J. Optim.* To appear.

J. Renegar, M. Shub and S. Smale, eds (1997), *Proceedings of the Summer Seminar on 'Mathematics of Numerical Analysis: Real Number Algorithm', AMS Lectures in Applied Mathematics*, AMS, Providence, RI.

B. Reznick (1992), *Sums of Even Powers of Real Linear Forms*, Vol. 463 of *Memoirs of the American Mathematical Society*, AMS, Providence, RI.

J. R. Rice (1966), 'A theory of condition', *SIAM J. Numer. Anal.* **3**, 287–310.

L. Santaló (1976), *Integral Geometry and Geometric Probability*, Addison-Wesley, Reading, MA.

A. Schönhage (1982), The fundamental theorem of algebra in terms of computational complexity, Technical report, Math. Institut der Universität Tübingen.

A. Schönhage (1987), Equation solving in terms of computational complexity, in *Proceedings of the International Congress of Mathematicans*, AMS, Providence, RI.

M. Shub (1993), On the work of Steve Smale on the theory of computation, in *From Topology to Computation: Proceedings of the Smalefest* (M. Hirsch, J. Marsden and M. Shub, eds), Springer, pp. 443–455.

M. Shub and S. Smale (1985), 'Computational complexity: on the geometry of polynomials and a theory of cost I', *Ann. Sci. École Norm. Sup.* **18**, 107–142.

M. Shub and S. Smale (1986), 'Computational complexity: on the geometry of polynomials and a theory of cost II', *SIAM J. Comput.* **15**, 145–161.

M. Shub and S. Smale (1993*a*), 'Complexity of Bézout's theorem I: geometric aspect', *J. Amer. Math. Soc.* **6**, 459–501. **Referred to as Bez I.**

M. Shub and S. Smale (1993*b*), Complexity of Bézout's theorem II: volumes and probabilities, in *Computational Algebraic Geometry* (F. Eyssette and A. Galligo, eds), Vol. 109 of *Progress in Mathematics*, pp. 267–285. **Referred to as Bez II.**

M. Shub and S. Smale (1993*c*), 'Complexity of Bézout's theorem III: condition number and packing', *J. Complexity* **9**, 4–14. **Referred to as Bez III.**

M. Shub and S. Smale (1994), 'Complexity of Bézout's theorem V: polynomial time', *Theoret. Comput. Sci.* **133**, 141–164. **Referred to as Bez V.**

M. Shub and S. Smale (1996), 'Complexity of Bézout's theorem IV: probability of success; extensions', *SIAM J. Numer. Anal.* **33**, 128–148. **Referred to as Bez IV.**

S. Smale (1976), 'A convergent process of price adjustment and global Newton method', *J. Math. Economy* **3**, 107–120.

S. Smale (1981), 'The fundamental theorem of algebra and complexity theory', *Bull. Amer. Math. Soc.* **4**, 1–36.

S. Smale (1985), 'On the efficiency of algorithms of analysis', *Bull. Amer. Math. Soc.* **13**, 87–121.

S. Smale (1986), Newton's method estimates from data at one point, in *The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics* (R. Ewing, K. Gross and C. Martin, eds), Springer, pp. 185–196.

S. Smale (1987*a*), Algorithms for solving equations, in *Proceedings of the International Congress of Mathematicians*, AMS, Providence, RI, pp. 172–195.

S. Smale (1987*b*), 'On the topology of algorithms I', *J. Complexity* **3**, 81–89.

S. Smale (1990), 'Some remarks on the foundations of numerical analysis', *SIAM Review* **32**, 211–220.

J. Steele and A. Yao (1982), 'Lower bounds for algebraic decision trees', *Journal of Algorithms* **3**, 1–8.

E. Stein and G. Weiss (1971), *Introduction to Fourier Analysis on Euclidean Spaces*, Princeton University Press.

R. Thom (1965), Sur l'homologie des variétés algébriques réelles, in *Differential and Combinatorial Topology* (S. Cairns, ed.), Princeton University Press.

J. Traub and H. Wozniakowski (1979), 'Convergence and complexity of Newton iteration for operator equations', *J. Assoc. Comput. Mach.* **29**, 250–258.

J. Traub, G. Wasilkowski and H. Wozniakowski (1988), *Information-Based Complexity*, Academic Press.

L. N. Trefethen (preprint), Why Gaussian elimination is stable for almost all matrices.

V. A. Vassiliev (1992), *Complements of Discriminants of Smooth Maps: Topology and Applications*, Vol. 98 of *Transl. of Math. Monographs*, AMS, Providence, RI. Revised 1994.

X. Wang (1993), Some results relevant to Smale's reports, in *From Topology to Computation: Proceedings of the Smalefest* (M. Hirsch, J. Marsden and M. Shub, eds), Springer, pp. 456–465.

H. Weyl (1932), *The Theory of Groups and Quantum Mechanics*, Dover.

J. Wilkinson (1963), *Rounding Errors in Algebraic Processes*, Prentice-Hall.

J. Wilkinson (1968), 'Global convergence of tridiagonal $QR$ algorithm with origin shifts', *Linear Algebra Appl.* **I**, 409–420.

H. Wozniakowski (1977), 'Numerical stability for solving non-linear equations', *Numer. Math.* **27**, 373–390.