Complexity Theory and Numerical Analysis *

Steve Smale †

January 2, 2000

0 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 roundoff 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 inspirations 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 as Newton's method dominate.

One can indeed be skeptical about a formal development of complexity into the domain of numerical analysis, where problems are solved only to a certain precision and roundoff 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:

$$T(y) \leq K(\text{size } y)^c \tag{1}$$

*Written for *Acta Numerica*

†Department of Mathematics, City University of Hong Kong

Here 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 traditions 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 $\log \frac{1}{\varepsilon}$. Moreover, the time required more typically depends on the condition of the problem. Therefore it is reasonable for "polynomial time" now to be put in the form:

$$T(y, \varepsilon) \leq K \left(\log \tfrac{1}{\varepsilon} + \mu(y) + \text{size}(y)\right)^c \tag{1'}$$

Here $y = (y_1, \cdots, y_n)$, with $y_i \in \mathbb{R}$ is the input of a numerical problem, with $\text{size}(y) = n$. 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 μ by $\log \mu$ or $\log \frac{1}{\varepsilon}$ by $\log \log \frac{1}{\varepsilon}$ for example. Moreover using the notion of approximate zero, described below, the ε might be eliminated.

I see much of the complexity theory ("upper bound" aspect) of numerical analysis conveniently represented by a 2-part scheme. Part 1 is the estimate (1′). Part 2 is an estimate of the probability distribution of μ, and takes the form

$$\text{prob}\left\{y \mid \mu(y) \geq K\right\} \leq \left(\frac{1}{K}\right)^c \tag{2}$$

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

Then Part 1 and 2 combine, eliminating the μ, to give a probability bound on the complexity of the algorithm. The following sections illustrate this theme. One needs to understand the condition number μ 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 notion of 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 lead to exciting new perspectives and problems.

In computer science consideration of polynomial time bounds led to the fundamentally important and notoriously difficult problem "is P=NP?". There is a corresponding problem for real number machines "is 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 [10], referred to hereafter as the [Manifesto] and its references for more background, history, 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 4 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.

Peter Henrici and his coworkers gave a substantial development for analyzing algorithms and a complexity theory for the FTA. See Dejon-Henrici (1969) and Henrici (1977). Also Collins (1977) 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), Smale (1981), which systematically pursued the issue of complexity for the FTA. Coincidently both authors gave main talks at the ICM, Berkeley 1986, Schönhage (1987), Smale (1987a), on this subject.

These articles show well two contrasting approaches.

Schönhage's algorithm is in the tradition of Weyl, with a number of added features which give very good, polynomial, 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 (1987b), Bini and Pan (1994), Neff (1994), Neff and Reif (1996). See Pan (1996) for an account of this approach to the FTA.

In contrast, in Smale (1981), the algorithm is based continuation methods as in Kellog, Li, and Yorke (1976), Smale (1976), Keller (1978), and Hirsch-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. 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 [12], here-

after referred to as BSS (1989). Subsequent developments along this line include Shub and Smale (1985, 1986), Kim (1988), Renegar (1987b), [68-72], hereafter referred to as Bez I–V respectively and [11], hereafter referred to as BCSS (1989).

Here is a brief account of some of the ideas of Smale (1981). A notion "approximate zero" of a polynomial is introduced. A point z is a *approximate zero* if Newton's method starting at z converges well in a certain precise sense. See a more developed notion in Section 4 below. The main theorem of this paper asserts:

Theorem 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 $\frac{1}{\sigma}$ where σ 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 2 parts, 1 and 2.
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 $\left(\frac{1}{k}\right)^c$, some constant c.

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

The crude bounds given in Smale (1981), and the mathematics as well 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 as given (or as input):

$$\text{a complex polynomial } f \text{ in one variable,} \quad f(z) = \sum a_i z^i,$$

$$\text{a complex number } z_0, \text{ and an } \varepsilon > 0.$$

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

$$|f(z^*)| < \varepsilon. \tag{1}$$

Let $t_0 = 0$, $t_i = t_{i-1} + \Delta t$, $\Delta t = \frac{1}{k}$, k a positive integer so that $t_k = 1$, be a partition of $[0, 1]$. Define (Newton's method) for any polynomial g,

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

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

$$z_i = N_{f_{t_i}}(z_{i-1}), \qquad i = 1, \ldots, k, \quad z^* = z_k, \tag{2}$$

It is easily shown that for almost all (f, z_0), z_i will be defined, $i = 1, \ldots, k$, provided Δt is small enough. We may say that $k = \frac{1}{\Delta t}$ is the "complexity". It is the main measure of complexity in any case; the problem at hand is: "how big may we choose Δt and still have z^* satisfying (1) and (2)?" (i.e. so that the complexity is the lowest).

Next a "condition number" $\mu(f, z_0)$ is defined which measures how close ζ_t is to being ill-defined. (More precisely $\mu(f, z_0) = \frac{1}{\sin\theta}$ where θ is the supremum of the angles of sectors about $f(z_0)$ where the inverse f^{-1} sending $f(z_0)$ into z_0, is defined.)

Theorem 2. *A sufficient number k of Newton steps defined in (2) to achieve (1) is given by*

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

Remarks. (a) We are assuming $0 < \varepsilon < \frac{1}{2}$.
(b) Note that the degree d of f plays no role, and the result holds for any (f, z_0, ε).
(c) The proof is based on "point estimates" (α-theory) (see Section 4 below) and an estimate of Loewner from Schlicht function theory. Thus it doesn't quite extend to n variables. But see a closely related theorem, Theorem 1 of Section 6 below. It remains a good problem to find the unity between the theorem above and Theorem 1 of Section 6.

For the next result suppose that f has the form

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

Theorem 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*

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

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

It is straight forward to combine theorem 2 and 3 to eliminate the μ and obtain both probabilistic and deterministic complexity bounds for approximating a zero of a polynomial. The probabilistic estimate improves by a factor of d the deterministic one. Theorem 3 and these results are in Shub and Smale (1985, 1986), but see also BCSS (1997), and Smale (1985).
Remarks. 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 (1987b) 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 that now, especially some recent developments. For linear systems, $Ax = b$, the condition number is defined in every basic NA text.

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 Σ_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 illposed problems following Bez I, II, IV, Dedieu (1996a), (1996b), (1996c) and BCSS (1997).

Consider first the context of the implicit function theorem,

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

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

Think of $F_a : \mathbb{R}^m \to \mathbb{R}^m$, $F_a(y) = F(a, y)$, 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.

Then 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), Wozniakowski (1977), Demmel (1987), Bez IV, Dedieu (1996a)) is defined by

$$\mu(a_0, y_0) = \|DG(a_0)\|, \qquad \text{operator norm}$$

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 a.

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 and also its norm, the condition number $\mu(a_0, y_0)$.

An example, already in 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 ζ, $F(f,\zeta) = f(\zeta)$.

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 $\hat{f} : \mathbb{C}^{n+1} \to \mathbb{C}$ so that $\hat{f}(1, z_1, \ldots, z_n) = f(z_1, \ldots, z_n)$ 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 $\hat{f} = (\hat{f}_1, \ldots, \hat{f}_n)$, 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

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

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

$$\langle f, g\rangle = \sum_\alpha \binom{d}{\alpha_1 \cdots \alpha_{n+1}}^{-1} \bar{f}_\alpha g_\alpha$$

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 multi index and

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

The weighting by the multinomial coefficient is important and yields unitary invariance of the inner product as will be noted.

Proposition (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.

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

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

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

Dedieu has suggest the weighting by $\frac{1}{d_i}$ 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 e.g. 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, Stein and Weiss (1971), Dedieu (1996a).

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. *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 $\{g \in \mathcal{H}_d \mid g(\zeta) = 0\}$ to the subset of where ζ 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 (1996a) has generalized this result quite substantially, and later (1996b) to sparse polynomial systems. 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 ζ of f (the "associated zero") and

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

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 (Frechet) 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 (1).

Toward 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 kth 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 for any reason take "$\gamma = \infty$". (see Smale (1986), Smale (1987a), Bez I for details of this development).

The invariant γ 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 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 ζ.

Now let

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

Theorem 2 (Smale (1986)). *There exists a universal constant $\alpha_0 > 0$ so 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 ζ of f).*

Remark 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 ζ.

Remark 2. For this definition of approximate zero, the best value of α_0 is probably no smaller than $\frac{1}{10}$. See developments, details, discussions in Smale (1987a), Wang (1993) with Han Dan Fu, Bez I, BCSS (1997).

Now how might one estimate γ? In Smale (1986, 1987a), 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 γ projectively invariant. Recall that $D = \max(d_i)$, $d = (d_1, \ldots, d_n)$, $d_i = \deg f_i$.

Theorem 3 (Bez I).

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

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

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

As a corollary of Theorem 3 and a projective version of Theorem 1, one obtains

Theorem 4 (Separation of zeros, Dedieu (1996b, 1996d), BCSS (1997)). *Let $f \in \mathcal{H}_d$, and ζ, ζ' be two distinct zeros of f. Then*

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

Remark 3. One has also the stronger result

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

Remark 4. The strength of Theorem 4 lies in its global aspect. It is not asymptotic even though μ is defined just by a derivative.

We end this section by stating a global perturbation theorem (Dedieu (1996b)).

Theorem 5. *Let f, $g:\mathbb{C}^n \to \mathbb{C}^n$, $\zeta \in \mathbb{C}^n$ with $f(\zeta)=0$. Then if*

$$\begin{aligned} &\alpha(g,\zeta) \le \frac{|3-3\sqrt{17}|}{4} \qquad \text{and} \\ &\|I - Df(\zeta)^{-1}Dg(\zeta)\| \le \frac{9-\sqrt{17}}{16} \end{aligned}$$

there is a zero ζ' of g such that

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

Here everything is affine including $\mu(f,\zeta)$. This uses Theorem 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 requirement 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 easily 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 κ 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 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, $\kappa = ||A||\ ||A^{-1}||$ be its condition number in the 2-norm, then*

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

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 the 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 by Trefethen reveals that large growth factors only appear exponentially rarely.

Theorem 2 (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*

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

where $\alpha > 2$ for real random matrices, $\alpha > 3/2$ for complex random matrices.

For iterative methods, we mention that a complexity result is available for the conjugate gradient method [35]. Let A be a real symmetric positive definite matrix, x_0 be an initial guess for the exact solution x_* of $Ax = b$, x_j be the j-th iteration by the conjugate gradient method, the following result is well known (see [3], Appendix B)

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

where the A-norm of a vector v is defined as $||v||_A = (v^T A v)^{1/2}$. 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 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 $x_j = A^j x_0/||A^j x_0||$ 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 ϵ (ϵ-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 in some distribution.

Theorem 3 (Kostlan (1988)). *For a real symmetric $n \times n$ matrix A with eigenvalues $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n|$, the number of iterations $\tau_\epsilon(A)$ required for the power method to produce an ϵ-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 $\psi(x) = d(\log \Gamma(x))/dx$.

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 η are excluded.

Theorem 4 (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 ϵ-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 (Ostrowski (1958), Parlett, Kahan (1969), Batterson, 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 θ_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 the number of iterations required to produce a small angle, say, $\theta_j < \epsilon$ for a small constant ϵ. 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 6 (Batterson & 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 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 reflectors 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 2×2 block of T that is closer to the (n, n) entry of T, $QR = T - sI$ is the QR factorization of $T - sI$. Wilkinson proved that the $(n, n-1)$ entries of the obtained sequence of T always converges 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 7. *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 "small enough" is also desirable, since the usual decoupling criterion is based on a comparison with the two nearby 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, $\{c_j\}$ 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. It

would be interesting to study the complexity of this method. Because of its recursive nature, a satisfactory complexity result given in terms of quantities related to the original matrix T could be more difficult to obtain.

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 et al (1996), the quasi-Laguerre method is used. An asymptotic convergence result has been established in [26], 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 double shift is widely used to 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 [4] 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 ζ_t could be given by the implicit function theorem from f_t and the initial ζ_0 with $f_0(\zeta_0) = 0$.

Next suppose $[0,1]$ is given a partition into k parts by $t_0 = 0$, $t_i = t_{i-1} + \Delta t$, $\Delta t = \frac{1}{k}$. Thus $t_k = 1$.

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

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

For Δt small enough the z_i are well defined and are good approximations of ζ_i. But $k = \frac{1}{\Delta t}$ represents the complexity so the problem is to avoid taking Δt much smaller than necessary. What is a sufficient number of Newton steps?

Theorem 1 (The main theorem of Bez I). *The biggest integer k in*

$$cLD^2\mu^2$$

is a sufficient to yield z_i by (1) to be an approximate zero of f_{t_i} with associated actual zero ζ_{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, $0 \le t \le 1$, D is the max of the d_i, $i = 1, \ldots, n$ and $\mu = \max_{0 \le t \le 1} \mu(f_t, \zeta_t)$ 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 ordinary Newton's method applied to the restriction of f to

$$z + \Big\{ y \in \mathbb{C}^{n+1} \mid \langle y, z \rangle = 0 \Big\}.$$

As a consequence of the Condition Number Theorem and Theorem 1, the complexity depends mainly on how close the path (f_t, ζ_t) comes to the set of ill-conditioned problems. An improved proof of Theorem 1 may be found in BCSS (1997).

Earlier work on complexity theory for Newton's method, several variables, is Renegar (1987a).

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 this:

Theorem 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).*

The result on one hand is surprising for its polynomial time bound on a problem on the boundary of the 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 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 the $f : \mathbb{C}^n \to \mathbb{C}^n$ is unlikely to be accomplished in polynomial time even using exact arithmetic (see Section 8), an astute analysis of the above theorem can give insight into the basic problem: "what are the limit of computation?" For example, is it "on the average" that gives the possibility of polynomial time?

A real (rather than complex) analogue of Theorem 2 also remains to be found.

Let us give some mathematical detail about the statement of Theorem 2.

"Approximate zero" has been defined in Section 4 are of course exact zeros cannot be found (Abel, Galois, et al). Averaging is done relative to a measure induced by our unitarily invariant inner product on the space of homogenized polynomials of degree $d = (d_1, \ldots, d_n)$, $d_i = \deg f_i$, $f = (f_1, \ldots, f_n)$ (see Section 3).

If $N = N(d)$ is the number of coefficients of such a system f, then unless $n \le 4$ or some $d_i = 1$, the number of arithmetic operations is bounded by cN^4. If $n \le 4$ 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 $N \le n^3$. 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 1-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 Δt 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 $g_i(z) = z_0^{d_i-1} z_i$, $i = 1, \ldots, n$ and $\zeta = (1, 0, \ldots, 0)$. A proof of this conjecture would yield a uniform algorithm.

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

The proof of Theorem 2 use Theorem 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 Section 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 $f \in \mathcal{H}_d$, $d = (d_1, \ldots, d_n)$, 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 it is proved in Bez II:

Theorem 1.

$$\textit{prob measure } \left\{ f \in \mathcal{H}_d \,\middle|\, \mu(f) > \frac{1}{\varepsilon} \right\} \leq C_d \varepsilon^4$$

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

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 as the subset of pairs (system, solution):

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

Denote by $\pi_1 : V \to P(\mathcal{H}_d)$, $\pi_2 : V \to P(\mathbb{C}^{n+1})$, the restriction of the corresponding projections.

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

Theorem 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) and the references).

There is an aspect of these kinds of 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 be an integer bigger than 1 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 $f \in S_d$, $\mu(f) \leq d$, for each for $d = 2, 3, 4, \ldots$.

Problem (Bez III). *Construct a family of $f_d \in \mathcal{H}_d$, $d = 2, 3, 4, \ldots$ so that*

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

c any constant.

The meaning of construct is to construct a polynomial time algorithm (e.g. in the sense of the machine of Section 8) which with 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:

Theorem 3 (Bez II). *The average number of real zeros of a real homogeneous polynomial system is exactly the square root of the Bezout number $\mathcal{D} = \prod_{i=1}^n d_i$ ($\mathcal{D}$ is the number of complex solutions).*

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

For the complexity results of Bez IV, V, Theorem 1 is inadequate. There one has similar type 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 need lower bound estimates for problem solving.

For this endeavor, one must consider all possible algorithms which 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.

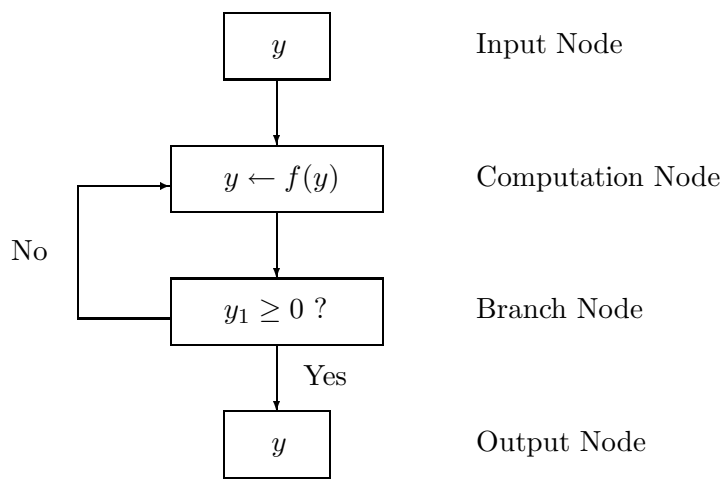
A "real number machine" is the most natural vehicle to deal with problem solving schemes based on Newton's method for example.

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

Inputs are a string y of real numbers of the form

$$\cdots 000 \cdots y_1 y_2 y_3 \cdots y_n 000 \cdots .$$

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.



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 explain themselves from the diagram.

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 a 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, eventually land on a output node. An *input-output map* ϕ is defined on the halting set by "following the flow" of the flow chart. For precise definitions and developments see BSS (1989), BCSS (1997).

Polynomial Time Complexity of a machine (sometimes with a restricted class of inputs) is the property.

$$T(y) \leq S(y)^c, \qquad \text{input } y \tag{1}$$

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 the there are no loops, i.e., the machine is a tree of nodes, 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), Ben-Or (1983), based on a real algebraic geometry estimate of Oleinik and Petrovsky (1949), Oleinik (1951), Milnor (1964), Thom (1965). Another is that of Smale (1987b), Vassiliev (1992) and based on the cohomology of the braid group.

Lower bounds tend to be modest and difficult, but are necessary towards understanding the fundamental problem. "What are the limits of computation?"

Note that the definition of real machine is valid with strings of numbers lying in any field if one replaces the branch node with the question, is "$y_1 = 0$?" If this field is the field of 2 elements, one has a Turing machine. The size becomes the number of bits. If one uses the complex numbers, one has a "complex machine".

Side Remarks: The study of zeros of polynomial systems plays a central role in both mathematics and computation theory. Deciding if 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 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 3 great unsolved problems of mathematics (together with the Riemann Hypothesis and Poincare's conjecture in 3 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:

(i) The condition number $\mu(y)$, or its log, or similar invariants of y.

(ii) The desired precision $\log \frac{1}{\varepsilon}$ where ε is the desired accuracy (or perhaps depending on the problem, ε, or even $\log\log \frac{1}{\varepsilon}$).

(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 justify to some extend taking the condition number of the particular problem as input. It is analagous, for example, to running a path following program inputting an initial step size as a guess.

Let me give an example of an open problem which fits into this framework. Let $d = (d_1, \ldots, d_m)$ and $\mathcal{P}_{n,d}$ be the space of m-triples of real polynomials $f = (f_1, \ldots, f_m)$ 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 $f_i(x) \geq 0$, all $i = 1, \ldots, m$ 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 ∞) of $\dim \mathcal{P}_{n,d}$ and $\mu(f)$.

(2) **Problem:** 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.

I would guess that the answer is yes, and that the answer would be interesting even assuming $\mu(f)$ is an explicitly known part of the input. My belief that the above is a more natural and interesting problem in the context of numerical analysis than the problem of deciding if a set of real polynomials has a zero.

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

A *roundoff machine over* $\mathbb{R}$ is a real machine together with a function $\delta > 0$ of inputs that at each input, computation and output node, adds a state vector of magnitude less than δ of which

one has no a priori knowledge (an adversary?). This idealization has the virtue of simplicity which hopefully compensates for its ignorance of important detail.

A problem will be called *robustly solvable* if it can be solved by a roundoff machine (with certainty).

The problem of deciding real polynomial systems is not robustly solvable. More important is the concept of *robustly solvable* in *polynomial time*. In addition to the estimate (1) with extended size, $S(y)$, one adds the requirement

$$\log \frac{1}{\delta(y)} \leq S(y)^c. \tag{3}$$

One can now sharpen the problem (2) to ask for a decision which is robustly solvable in polynomial time.

The above gives some sense to the notion of a robust or numerically stable algorithm, perhaps improving on the attempts in Isaacson-Keller (1966), Wozniakowski (1979), Smale (1990), 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 the Proceedings of "The Mathematics of Numerical Analysis" by Renegar, Shub and Smale (1996) 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 (1996).

Also well-developed is the area of information based complexity. In spite of its relevance and important 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 in the science of mathematical programming. I believe that numerical analysts interested in complexity considerations can learn much from what has happened and is happening in that field. I like especially the perspective and work of Renegar (1996).

References

[1] Allgower, E. and K. Georg (1990). *Numerical Continuous Methods*. Springer-Verlag.

[2] Allgower, E. and K. Georg (1993). Continuation and path following. *Acta Numerica*. 1–64.

[3] Axelsson, O. (1994). *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK.

[4] Batterson, S. (1994). Convergence of the Francis shifted QR algorithm on normal matrices. *Linear Algebra And Its Applications, 207*, 181-195.

[5] Batterson, S. and Day, D. (1992). Linear convergence in the shifted QR algorithm. *Mathematics of Computation, 59*, 141-151.

[6] Batterson, S. and Smillie, J. (1989). The dynamics of Rayleigh quotient iteration. *SIAM J. Numer. Anal. 26*, 624-636.

[7] Batterson, S. and Smillie, J. (1990). Rayleigh quotient iteration for nonsymmetric matrices. *Mathematics of Computation. 55*, 169-178.

[8] Ben-Or, M. (1983). Lower bounds for algebraic computation trees. In *15th annual ACM Symp. on the Theory of Computing*, pp.80–86.

[9] Bini, D. and V. Pan (1994). *Polynomial and Matrix Computations*. Birkhäuser.

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

[11] Blum, L., F. Cucker, M. Shub, and S. Smale (1997). *Complexity and Real Computation*. Springer. (or BCSS) To appear.

[12] Blum, L., M. Shub, and S. Smale (1989). On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the Amer. Math. Soc. 21*, 1–46. (or BSS)

[13] Brockett, R. (1973). In D. Mayne and R. Brockett (Eds.), *Geometric Methods in Systems Theory, Proceedings of the NATO Advanced Study Institute*. D. Reidel Publishing Co.

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

[15] Collins, G. (1975). *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, Volume 33 of *Lect. Notes in Comp. Sci.*, pp.134–183. Springer-Verlag.

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

[17] Dedieu, J.-P. (1996a). Approximate solutions of numerical problems, condition number analysis and condition number theorems. To appear in *Proceeding of the Summer Seminar on "Mathematics of Numerical Analysis: Real Number Algorithm", AMS Lectures in Applied Mathematics*.

[18] Dedieu, J.-P. (1996b). Condition number analysis for sparse polynomial systems. Preprint.

[19] Dedieu, J.-P. (1996c). Condition operators, condition numbers and condition number theorem for the generalized eigenvalue problem. To appear in *J. of Linear Algebra and Applic.*

[20] Dedieu, J.-P. (1996d). Estimations for separation number of a polynomial system. To appear in *J. of Symbolic Computation*.

[21] Dégot, J. and B. Beauzamy (TA). Differential identities. To appear in *Transactions of the Amer. Math. Soc.*

[22] Dejon, B. and P. Henrici (1969). *Constructive Aspects of the Fundamental Theorem of Algebra*. John Wiley & Sons.

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

[24] Dongarra, J. J. and Sorensen, D. C. (1987). A fully parallel algorithm for the symmetric eigenvalue problem. *SIAM J. Sci. Stat. Comput. 8*, s139-s154.

[25] Du, Q., Jin, M., Li, T. Y. and Zeng, Z. (1996a). Quasi-Laguerre iteration in solving symmetric tridiagonal eigenvalue problems. To appear in *SIAM J. Scientific Computing*.

[26] Du, Q., Jin, M., Li, T. Y. and Zeng, Z. (1996b). The quasi-Laguerre iteration, To appear in *Math. Comp.*

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

[28] Edelman, A. (1988). Eigenvalues and condition numbers of random matrices. *SIAM J. of Matrix Anal. and Applic. 9*, 543–556.

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

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

[31] Grigoriev, D. (1987). *Computational complexity in polynomial algebra*, Proc. Internat. Congr. Math. (Berkeley, 1986), vol. 1, 2, Amer. Math. Soc., Providence, RI, 1987, 1452–1460.

[32] Giusti, M., J. Heintz, J.E. Morais, J. Morgenstern, and L.M. Pardo (1996). Straight-line program in geometric elimination theory, to appear in *Journal of Pure and Applied Algebra*.

[33] Golub,G. and C. Van Loan (1989). *Matrix Computations*, John Hopkins Univ. Press.

[34] Henrici, P. (1977). *Applied and Computational Complex Analysis*, John Wiley & Sons.

[35] Hestenes, M. R. and Stiefel, E. (1952). Method of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand. 49*, 409-436.

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

[37] Hoffman, W. and Parlett, B. N. (1978). A new proof of global convergence for the tridiagonal QL algorithm, *SIAM J. Num. Anal. 15*, 929-937.

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

[39] Keller, H. (1978). Global homotopic and Newton methods. In *Recent advances in Numerical Analysis*, pp.73–94, Academic Press.

[40] Kellog, R., T. Li, and J. Yorke (1976). A constructive proof of Brouwer fixed-point theorem and computational results, *SIAM J. of Numer. Anal. 13*, 473–483.

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

[42] Kostlan, E. (1988). Complexity theory of numerical linear algebra, *J. of Computational and Applied Mathematics 22*, 219–230.

[43] Kostlan, E. (1991). Statistical complexity of dominant eigenvector calculation. *Journal of Complexity 7*, 371-379.

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

[45] Malajovich, G. (1994). On generalized Newton algorithms: quadratic convergence, path-following and error analysis, *Theoretical Computer Science 133*, 65–84.

[46] McNamee, J. M. (1993). A bibliography on roots of polynomials, *J. of Computational and Applied Math. 47, 3*, 391–394.

[47] Milnor, J. (1964). On the Betti numbers of real varieties, *Proceeding of the Amer. Math. Soc. 15*, 275–280.

[48] Neff, C. (1994). Specified precision root isolation is in NC, *J. of Computer and System Sci. 48*, 429–463.

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

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

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

[52] Ostrowski, A. (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.

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

[54] Pan, V. (1996). Solving a polynomial equation: some history and recent progress, *SIAM Review.*

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

[56] Rakhmanov, E.A., E.B. Saff and Y.M. Zhou (1995). Electrons on the sphere, In: *Computational Methods and Function Theory* (R.M. Ali, St. Ruscheweyh and E.B. Saff, Eds.), World Scientific, pp.111–127.

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

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

[59] Renegar, J. (1996). Condition numbers, the Barrier method, and the conjugate gradient method, *SIAM J. on Optimization*, to appear.

[60] Renegar, J., M. Shub, and S. Smale (Eds.) (1996). Lectures in Applied Mathematics, American Mathematical Society.

[61] Reznick, B. (1992). *Sums of even powers of real linear forms*, Number 463 in Memoirs of the American Mathematical Society, AMS, Providence, RI.

[62] Santalo, L. (1976). *Integral geometry and geometric probability*, Addison-Wesley, Reading, Mass.

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

[64] Schönhage, A. (1987). Equation solving in terms of computational complexity, In *Proceeding of the International Congress of Mathematicans*, Amer. Math. Soc.

[65] Shub, M. (1993). On the Work of Steve Smale on the Theory of Computation, In M. Hirsch, J. Marsden, and M. Shub (Eds.), *From Topology to Computation: Proceeding of the Smalefest*, pp.443–455, Springer-Verlag.

[66] Shub, M. 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.

[67] Shub, M. and S. Smale (1986). Computational complexity: on the geometry of polynomials and a theory of cost II, *SIAM Journal on Computing 15*, 145–161.

[68] Shub, M. and S. Smale (1993a). Complexity of Bezout's theorem I: geometric aspect, *Journal of the Amer. Math. Soc. 6*, 459–501.

[69] Shub, M. and S. Smale (1993b). Complexity of Bezout's theorem II: volumes and probabilities, In F. Eyssette and A. Galligo (Eds.), *Computational Algebraic Geometry*, Volume 109 of *Progress in Mathematics*, pp267–285,

[70] Shub, M. and S. Smale (1993c). Complexity of Bezout's theorem III: condition number and packing, *Journal of Complexity 9*, 4–14.

[71] Shub, M. and S. Smale (1994). Complexity of Bezout's theorem V: polynomial time, *Theoretical Computer Science 133*, 141–164.

[72] Shub, M. and S. Smale (1996). Complexity of Bezout's theorem IV: probability of success; extensions, *SIAM J. of Numer. Anal. 33*, 128–148.
The preceding are refered to as Bez I–V respectively.

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

[74] Smale, S. (1981). The fundamental theorem of algebra and complexity theory, *Bulletin of the Amer. Math. Soc. 4*, 1–36.

[75] Smale, S. (1985). On the efficiency of algorithms of analysis, *Bulletin of the Amer. Math. Soc. 13*, 87–121.

[76] Smale, S. (1986). Newton's method estimates from data at one point, In R. Ewing, K. Gross, and C. Martin (Eds.), *The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics*, Springer-Verlag.

[77] Smale, S. (1987a). Algorithms for solving equations, In Proceedings of the International Congress of Mathenaticans, pp.172–195, American Mathematical Society.

[78] Smale, S. (1987b). On the topology of algorithms I, *Journal of Complexity 3*, 81–89.

[79] Smale, S. (1990). Some remarks on the foundations of numerical analysis, *SIAM review 32*, 211–220.

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

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

[82] Thom, R. (1965). Sur l'homologie des variétés algébriques réelles, In S. Cairns (Ed.), *Differential and combinatorial topology*, Princeton University Press.

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

[84] Traub, J. and H. Wozniakowski (1979). Convergence and complexity of Newton iteration for operator equations, *Journal of the ACM 29*, 250–258.

[85] Trefethen, L. N. Why Gaussian elimination is stable for almost all matrices, preprint.

[86] Vassiliev, V. A. (1992). Complements of discriminants of smooth maps: topology and applications, Transl. of Math. Monographs, 98, 1992, revised 1994, Amer. Math. Soc., Providence, RI.

[87] Wang, X. (1993). Some results relevent to Smale's reports, In M. Hirsch, J. Marsden, and M. Shub (Eds.), *From Topology to Computation: Proceeding of the Smalefest*, pp.456–465, Springer-Verlag.

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

[89] Wilkinson, J. (1963). *Rounding Errors in Algebraic Processes*, Prentice Hall.

[90] Wilkinson, J. (1968). Global convergence of tridiagonal QR algorithm with origin shifts, *Lin. Alg. and Its Applic. I*, 409–420.

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