28
On the Work of Steve Smale on the Theory of Computation

MICHAEL SHUB*

The theory of computation is the newest and longest segment of Steve Smale's mathematical career. It is still evolving and, thus, it is difficult to evaluate and isolate the more important of Smale's contributions. I think they will be as important as his contributions to differential topology and dynamical systems. He has firmly grounded himself in the mathematics of practical algorithms, Newton's method, and the simplex method of linear programming, inventing the tools and methodology for their analysis. With the experience gained, he is laying foundations for the theory of computation which have a unifying effect on the diverse subjects of numerical analysis, theoretical computer science, abstract mathematics, and mathematical logic. I will try to capture some of the points in this long-term project. Of course, the best thing to do is to read Smale's original papers; I have not done justice to any of them.

Smale's work on the theory of computation begins with economics [Smale, 1976]. Prices $p = (p_1, \ldots, p_\ell) \in R^\ell_+$ for ℓ commodities give rise to demand and supply functions $D(p)$ and $S(p)$. The excess demand function $f(p) = D(p) - S(p) \in R^\ell$ has as the ith coordinate the excess demand for the ith good at prices p. A price equilibrium is a system of prices p for which $f(p) = 0$, that is, supply equals demand.

$$f\colon R^\ell_+ \to R^\ell$$

and the problem is to find a zero of f[1]. Given a C^2 function $f\colon M \to R^n$ defined on a domain $M \subset R^n$, Smale suggested using the "global Newton" differential equation

$$Df(x)\frac{dx}{dt} = -\lambda f(x)$$

* Partially supported by an NSF grant.

[1] Actually, f is assumed to be scale invariant $f(\lambda P) = f(P)$ for $\lambda > 0$ so f may be restricted to the unit sphere intersect R^ℓ_+, $S^{\ell-1}_+$. Walras' law is $p \cdot f(P) = 0$, so f is tangent to $S^{\ell-1}_+$ and the problem is to find a zero of this vector field.

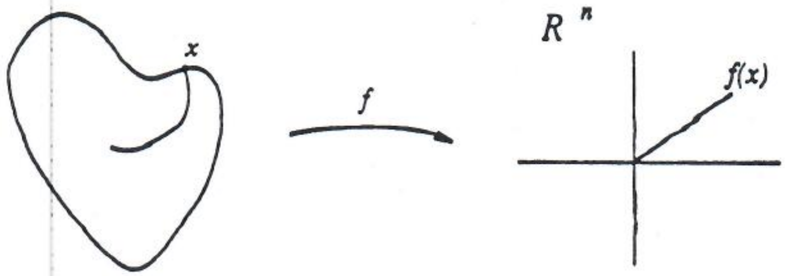


FIGURE 1

where λ is a real number depending on the sign of $\mathrm{Det}(Df(x))$. The solution curves of this equation through points x with $f(x) \neq 0$ are inverse images of rays pointing to zero.

or what is the same, inverse images of points $x \in S^{n-1}$ for the function

$$g: M - E \to S^{n-1}$$

defined by $g(x) = f(x)/\|f(x)\|$ and where $E = \{x \in M \mid f(x) = 0\}$. With the right boundary conditions, the "global Newton" vector field is transverse to the boundary and g is nonsingular on the boundary.

By Sard's theorem, almost every value in S^{n-1} is a regular value, and for almost every $m \in \partial M$, $g^{-1}(g(m))$ is a smooth curve. If M is compact, this curve must lead to the set of zeros E.

Smale proved the existence of price equilibria this way. Differential equation solvers can then be used to locate these points. Smale pursued these ideas with Hirsch in [Hirsch-Smale, 1979] where they suggested various explicit algorithms for solving $f(x) = 0$. In particular, their work included polynomial mappings

$$f: \mathbb{C}^m \to \mathbb{C}^m \quad (\text{or } R^m \to R^m)$$

which are proper and have nonvanishing Jacobian outside of a compact set.

An interesting feature of these algorithms is that their natural starting points are quite far from the zero set; for example, in the discussion of global Newton above, they are in ∂M.

The global Newton differential equation had been considered previously and independently by Branin without convergence results. The polynomial system $g: \mathbb{R}^2 \to \mathbb{R}^2$ proposed by Brent,

$$g_1(x_1, x_2) = 4(x_1 + x_2),$$

$$g_2(x_1, x_2) = 4(x_1 + x_2) + (x_1 + x_2)((x_1 - 2)^2 + x_2^2 - 1),$$

is illustrated in [Branin, 1972] and reproduced below. The curve $|J| = 0$ is $\mathrm{Det}(g) = 0$. Note the region of closed orbits and nonconvergence which are close to zero.

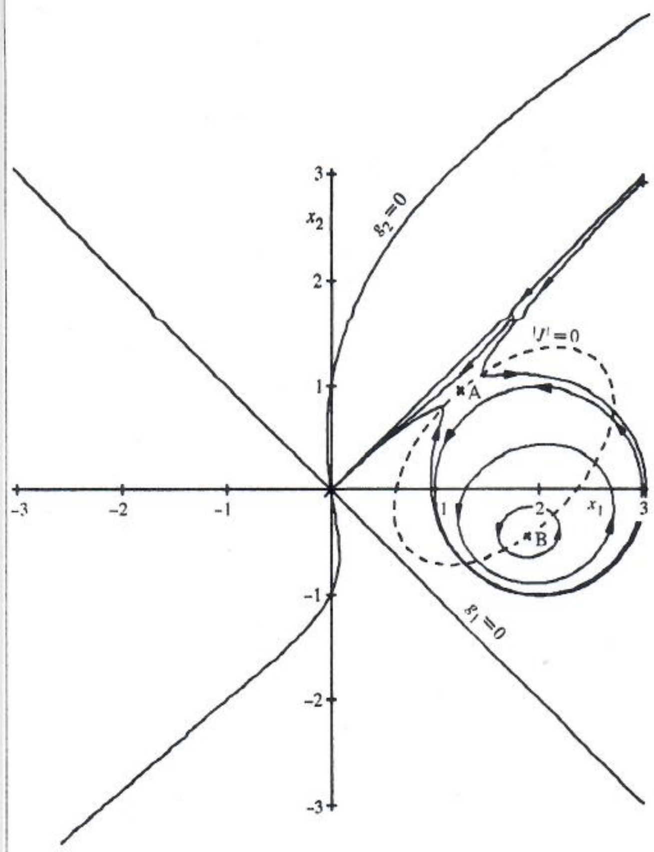


FIGURE 2

Other algorithms for the location of pure equilibria were known. Scarf, in particular, had developed simplicial algorithms; see [Smale, 1976] for a discussion of this. Having competing methodologies to solve the problem led Smale to develop a framework in which to compare their efficiency. Smale's article "The Fundamental Theorem of Algebra and Complexity Theory" [Smale, 1981] is a startling step forward. Consider the first three paragraphs of the paper:

> The main goal of this account is to show that a classical algorithm, Newton's method, with a standard modification, is a tractable method for finding a zero of a complex polynomial. Here, by "tractable" I mean that the cost of finding a zero doesn't grow exponentially with the degree, in a certain statistical sense. This result, our main theorem, gives some theoretical explanation of why certain "fast" methods of equation solving are indeed fast. Also this work has the effect of helping bring the discrete mathematics of complexity theory of computer science closer to classical calculus and geometry.

A second goal is to give the background of the various areas of mathematics, pure and applied, which motivate and give the environment for our problem. These areas are parts of (a) Algebra, the "Fundamental theorem of algebra", (b) Numerical analysis, (c) Economic equilibrium theory and (d) Complexity theory of computer science.

An interesting feature of this tractability theorem is the apparent need for use of the mathematics connected to the Bieberbach conjecture, elimination theory of algebraic geometry, and the use of integral geometry.

The scope of the undertaking is very large and the terrain unclear. It was the discrete theoretical computer scientists who had the most developed notion of algorithm, of cost, and of complexity as a function of input size. For them, the space of problems, complex polynomials of a given degree, so natural for a mathematician is not natural because it is not discrete. From another direction, numerical analysts have practical experience with root-finding, algorithms which are fast and algorithms which are sure, algorithms which are stable and those which are not. In [Smale, 1985], we see Smale grappling again with these questions. First, there is the quote from von Neumann quoted again in [Smale, 1990].

The theory of automata, of the digital, all or none type, as discussed up to now, is certainly a chapter in formal logic. It would, therefore, seem that it will have to share this unattractive property of formal logic. It will have to be from the mathematical point of view, combinatorial rather analytical.

... a detailed, highly mathematical and more specifically analytical, theory of automata and of information is needed.

and Chapter 2, Section 6

6. What is an algorithm?
PROBLEM 11. What is the fastest way of finding a zero of a polynomial? This is a kind of super-problem. I would expect contributions by several mathematicians rather than a single solution. It will take a lot of thought even to find a good mathematical formulation.

In some ways, one could compare this problem with showing the existence of a zero of a polynomial. The concept of complex numbers had to be developed first. For Problem 11, one must develop the concept of algorithm to deal with the kind of mathematics involved. Consistent with the von Neumann statement quoted in the introduction, my belief is that the Turing approach to algorithms is inadequate for these purposes.

Although the definitions of such algorithms are not available at this time, my guess is that some kind of continuous or differentiable machine would be involved. In so much of the use of the digital computer, inputs are treated as real numbers and the output is a continuous function of the input. Of course a continuous machine would be an idealization of an actual machine, as is a Turing machine.

The definition of an algorithm should relate well to an actual program or flow-chart of a numerical analyst. Perhaps one could use a Random Access Machine (RAM, see Aho–Hopcraft–Ullman) and suppose that the registers could hold real numbers.

Then one might with some care expand the list of permissible operations. There are pitfalls along the way and much thought is needed to do this right.

To be able to discuss the fastest algorithm, one has to have a definition of algorithm. I have used the word algorithm throughout this paper, yet I have not said what an algorithm is. Certainly the algorithms discussed here are not Turing machines; and to force them into the Turing machine framework would be detrimental to their analysis. It must be added that the idealizations I have suggested do not eliminate the study of round-off error. Dealing with such loss of precision is a necessary part of the program.

Problem 11 is not a clear-cut problem for various reasons. Factors which could affect the answer include dependence on the machine, whether one wants to solve one or many problems, time taken to write the program, whether polynomials have large or small degree, how the problem is presented, etc.

So in [Smale, 1981] he has launched into a discussion of the total cost of an algorithm without a precise notion of cost or algorithm available; these will come later! To have enough conviction that such a long-range project will work out is not uncharacteristic of Smale.

When Smale was awarded the Fields Medal in 1966 for his work in differential topology, René Thom wrote (translation my own):

... Smale is a pioneer, who takes his risks with calm courage, in a completely unexplored domain, in a geometric jungle of inextricable richness he is the first to have cleared a path and planted beacons. [Thom, 1966]

Returning to the 1981 paper, Smale restricted the class of functions for which roots are to be found to complex polynomials of one variable and degree d, normalized as $f(z) = \sum_{i=0}^{d} a_i z^i$ with $a_i \in \mathbb{C}$ $a_d = 1$ and $|a_i| \leq 1$ for $1 \leq i \leq d$; call this space $P_d(1)$. For complex polynomials f, the Jacobian is always ≥ 0, so global Newton can be taken as

$$\frac{dx}{dt} = -Df(x)^{-1}f(x).$$

FIGURE 3

Now the solution curve through z_0 is the branch through z_0 of the inverse of image of the ray from $f(z_0)$ to 0, and because f is proper, it does lead to a zero of f with the exception of a finite number of rays which contain the critical values of f. Incidentally, this argument proves the fundamental theorem of algebra. To turn the proof into an algorithm, Smale considers the Euler approximation to the solution of the differential equation, for h a positive real,

$$N_h(f)(z) = z - hDf^{-1}(z)f(z).$$

When $h = 1$, this is Newton's method; we denote this also by $N(f)$. Newton's method gives quadratic convergence near simple zeros, so Smale defines an approximate zero of f as a point where Newton's method is converging quadratically.

Definition. z_0 is an approximate zero for f iff $|z_k - z_{k-1}| \le (1/2)^{2^{k-1}-1}|z_1 - z_0|$, where $z_i = N(f)z_{i-1}$. The goal becomes to find an approximate zero of f.

Smale uses $x_0 = 0$ as a starting point for his algorithm and then iterates $N_h(f)$. The first main theorem using Lebesgue measure on $P_d(1)$ in [Smale, 1981] is:

Main Theorem. *There is a universal polynomial $S(d, 1/\mu)$, and a function $h = h(d, \mu)$ such that for degree d and μ, $0 < \mu < 1$, the following is true with probability $1 - \mu$. Let $x_0 = 0$. Then $x_n = N_h(f)(x_{n-1})$ is well-defined for all $n > 0$ and x_s is an approximate zero for f where $s = S(d, 1/\mu)$.*

More specifically, we can say, if $s \ge [100(d + 2)]^9/\mu^7$, then with probability $1 - \mu$, x_s is well-defined by the algorithm for suitable h and x_s is an approximate zero of f.

Borrowing the notion of polynomial cost computation from computer sciences, Smale has proven that approximate root-finding is "polynomial" in the degree d and one over the probability of failure.

The theorem reflects Smale's notion that fast algorithms fail sometimes (as Newton's method does) and are slow near where they fail, thus making a statistical analysis appropriate.

The proof of the main theorem is very long involving all the ingredients quoted above. Moreover, there are some outstanding questions on mean value theorems for polynomials. Smale proves:

Theorem 1. *Suppose $f(z)$ is a complex polynomial with $f(0) = 0$ and $f'(0) \neq 0$, then*

(a) *there is a critical point Θ (i.e., $f'(\Theta) = 0$) such that*

$$\frac{|f^{(k)}(0)|}{k!}\frac{|f(\Theta)|^{k-1}}{|f'(0)|^k} \le 4^{k-1};$$

(b) *there is a critical point* Θ *with*

$$\frac{|f(\Theta)|}{|\Theta|}\frac{1}{|f'(0)|} \leq 4.$$

Smale raises the problem [Smale, 1981]:

Problem

(a) Can the 4 in (a) of Theorem 1 be reduced, perhaps to 1?
(b) Can the 4 in (b) of Theorem 1 be reduced to 1, in fact, to $1 - 1/d$?

I do not think there has been much progress on (a). There is some progress on (b) by Tischler [1989].

It was these problems that got me involved in analyzing Newton's method. On one of my many trips to Berkeley, Steve was working on Theorem 1 for his complexity analysis which was to become the 1981 paper. He asked me if I could prove something like Theorem 1 above for some constant; in fact, he thought the constant should be 1. I was able to see the first case of the inequality (here my memory is a little different than Steve's) for the second derivative by applying Bieberbach's estimate for a_2 to the inverse of f, which must be defined and injective on a disc of radius at least the smallest modules of a critical value. I told this to Steve and returned to New York. I had no idea what he wanted it for. The next year I was in Berkeley on a sabbatical, [Smale, 1981] was already written and Steve was teaching a course on it, which I took. Steve and I began collaborating during the semester; our work was finally published in [Shub-Smale, 1985] and [Shub-Smale, 1986a]. We extended the analysis that Steve did to higher-order methods which we called generalized Euler iterations. It was our impression at the time that using methods of order $\ln d$ for d degree polynomials might be the most efficient among "incremental" algorithms. We gave a lower bound estimate for the area of the approximate zeros in $P_d(1)$ across the unit disc which was later improved; see [Friedman, 1990; Smale, 1986]. By choosing starting points far from the roots, we were able to improve the cost estimates of [Smale, 1981] for finding approximate zeros. We also studied the problem of finding small values of polynomials. Examining the foliation of the complex plane $\mathbb{C}$ by inverse images of rays for a polynomial f, we found that the inverse images of rays with small angle with respect to critical rays occupy about the same proportion (i.e., angle) of a large circle. For algorithms of Newton–Euler type we proved:

Theorem A. *For each* f, ε, *there is a Newton–Euler Algorithm which terminates with probability* 1 *and produces a* z *with* $|f(z)| < \varepsilon$. *The average number of iteration is less than* $O(d + |\log \varepsilon|)$.

Theorem B. *There is a Newton–Euler algorithm which produces an approximate zero for $f \in P_d(1)$ with probability 1 and average number of iterations $O(d \log d)$.*

Theorem B improved the main theorem in [Smale, 1981] in the sense that the estimate given there does not produce a finite average cost algorithm. Theorem A was improved in [Smale, 1985a; Kim, 1988a; and [Renegar, 1987b], where $|\log \varepsilon|$ is replaced by $\log|\log \varepsilon|$. Also Schonhage and others have zero finding algorithms of different flavor. Most recently, Neff [1990] has made a good contribution. Renegar [1987a] proved and an n-variable analogue of Smale's 1981 main theorem, and Canny [1988, 1990], Renegar [1989], and others have studied more algebraic approaches to the n-variable root-finding problem. Recently, Sutherland [1989] has an interesting result about the convergence of Newton's method itself on large circles.

Already in [Smale, 1981], the average case analysis of the Dantzig's simplex method for linear programming is cited as Problem 6. Consonant with Smale's perspective that fast algorithms are not always fast the simplex method was known to be a worst-case exponential but practically highly efficient. The linear programming problem (LPP) is: Given $m \times n$ matrices A and vectors $b \in R^m$, $c \in R^n$, determine if the function cx has a minimum on $Ax \geq b$ and $x \geq 0$. If it does, find a point x which minimizes it.

Taking a Gaussian distribution on $R^{mn} \times R^m \times R^n$ and letting $\rho(m,n)$ be the number of steps of Dantzig's self-dual method to solve LPP, Smale [1983a] proves that ρ is sublinear in the number of variables; see also [Smale, 1983b].

Theorem 2. *Let p be a positive integer. Then depending on p and m there is a positive constant c_m such that for all n*

$$\rho(m,n) \leq c_m n^{1/p}.$$

In [Smale, 1985a] which won the Chauvenet Prize of the Mathematical Association of America in 1988, Smale confronts some new issues. First is the problem of ill-posed problems. In the fall of 1983, Lenore Blum was visiting New York and we studied the problem of the average loss of precision (or significance) in evaluating rational functions of real variables. Let δ be the input accuracy necessary for desired output accuracy ε. Then $|\ln \delta| - |\ln \varepsilon|$ is the loss of precision (or significance for relative accuracy). We showed this loss was tractable on the average [Blum–Shub, 1986]; Smale focused on linear algebra. There the condition number, $K_A = \|A\| \|A^{-1}\|$, of a matrix A measures the worst-case relative error of the solution x of the equation $Ax = b$ divided by the relative error of the input b. Thus, $\log K_A$ measures the worst-case loss of significance. Smale wrote an explicit integral for the average of $\log K_A$ and Ocneanu, Kostlan, Renegar, and others made progress toward its estimation. Finally, Edelman [1988] has shown that up to an additive constant the average is $\ln n$. This result helps explain the success of fixed precision computers in solving fairly large linear systems. Demmel [1987a; 1987b] interprets the condition number as the inverse of the distance to the determinant zero variety, i.e., the singular matrices. Thus, there is an analogy between the success of fast algorithms and the intrinsic difficulty of robust computation. They are both measured in terms of distance to a subvariety of bad problems. This theme surfaces again in Smale's work, although the precise relationship remains somewhat mysterious.

The same 1985 paper dealt with two other problems. The efficiency of approximation of integrals: I will not say much about this except that Smale showed that the trapezoid rule is more efficient than Riemann integration on the average for fixed error on $\mathscr{H}^1$ functions, similarly Simpson's rule is more efficient for $\mathscr{H}^2$ functions. Newton's method is an example of a purely iterative algorithm for solving polynomial equations. A purely iterative algorithm is given as a rational endomorphism of the Riemann sphere which depends rationally on the coefficients of the polynomials (of fixed degree d) which are to be solved. A purely iterative algorithm is generally convergent if for almost all (f, x) iterating the algorithm on x, the iterates converge to a root of f. Smale [1986] conjectures that there are no purely iterative generally convergent algorithms for general d. McMullen [1988] proved this for $d \geq 4$ and produced a generally convergent iterative algorithm for $d = 3$. For $d = 2$, Newton's method is generally convergent. Doyle and McMullen [1979] have gone on to add to this examining $d = 5$ in terms of a Galois theory of purely iterative algorithms. In contrast, Steve and I showed in [Shub–Smale, 1986b] that if complex conjugation is allowed, then there are generally convergent purely iterative algorithms even for systems of n complex polynomials of fixed degree in n variables.

Smale has devoted a lot of effort to understanding Newton's method. These are recounted in [Smale, 1986], but let me mention a few of the results of this paper and [Smale, 1985b].

To generalize the one-variable theory, Smale considers the zero-finding problem for $f: E \to F$, where f is an analytic map of Banach spaces. Newton's method is the same $z' = N(f)z = z - Df(z)^{-1}f(z)$, and the definition of approximate zero is the same.

Definition. z_0 is an approximate zero for f iff $\|z_k - z_{k-1}\| \leq (1/2)^{2^{k-1}-1} \|z_1 - z_0\|$, where $z_i = N(f)z_{i-1}$.

Let $\beta(z, f) = \beta(z) = \|Df(z)^{-1}f(z)\|$, i.e., β is the norm of the Newton step $z' - z$. Let $\gamma(f, z) = \gamma(z) = \sup_{k \geq 2} \|(1/k!)Df(z)^{-1}D^k f(z)\|^{1/k-1}$ and let $\alpha(z, f) = \beta(z, f)\gamma(z, f)$; from [Smale, 1985a].

Theorem A. *There is a naturally defined number α_0 approximately equal to 0.130707 such that if $\alpha(z,f) < \alpha_0$ then z is an approximate zero of f.*

Thus, we have a test for approximate zeros in terms of data computed at the point z alone, and which is, hence, quite different from Kantorovich-type estimates for the domain of convergence of Newton's method that require estimates of derivatives on a neighborhood, and it is very useful. Smale uses it to get an estimate on global Newton. Given $f: E \to F$ and $z_0 \in E$, suppose $Df^{-1}(z)$ is defined on the whole ray $tf(z_0)$ for $0 < t \leq 1$, so that we may invert the ray. Call the inverse image σ. Let $M(z_0, f) = \max_{z \in \sigma} \alpha(z,f)/\|f(z)\|$ and ∞ if σ is not defined.

Theorem 3 [Smale, 1986] (The Speed of a Global Newton Method). *There exist (small) positive constants c real; ℓ an integer with this property. Let f: $E \to F$ be analytic, $z_0 \in E$ with $M(z_0, f) < \infty$. Suppose n is an integer*

$$n > c\|f(z_0)\| M(z_0, f), \quad \Delta = 1/n.$$

Let $w_i = (1 - i\Delta)f(z_0)$, $i = 0, \ldots, n$. Then, inductively, $z_i = N^{\ell}_{f-w_i}(z_{i-1})$ is well-defined and z_n is an approximate zero of f.

Renegar and I [Renegar–Shub, 1992] use a version of these theorems to give a simplified and unified proof of the convergence properties of several of the recent polynomial time linear programming algorithms. For one variable, Kim [1988a] also considered the algorithms of Theorem 3.

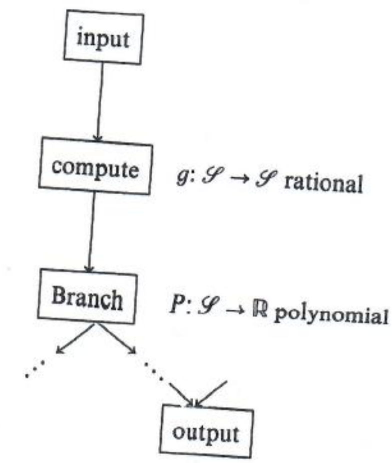
Smale [1986] also estimates $\gamma(f, z)$ for a polynomial f in terms of the norms of the coefficients, the norm of z, and the norm of the derivative of f at z. With this estimate, he was able to prove:

Theorem 4 [Smale, 1986]. *The average area of approximate zeros for $f \in P_d(1)$ is greater than a constant $c > 0$, where c is independent of d.*

Smale [1986] also dealt with a regularized version of the linear programming problem which I will not consider here. Smale [1985a] asserts:

> A study of total cost for algorithms of numerical analysis yields side benefits. It forces one to consider global questions of speed of convergence, and in so doing one introduces topology and geometry in a natural way into that subject. I believe that this will have a tendency to systematize numerical analysis. This development could turn out to be comparable to the systematizing effect of dynamical systems on the subject of ordinary differential equations over the last twenty-five years.

We have already seen geometry. In [Smale, 1987], he turns his attention to topology. To begin with, Smale defines a (uniform) algorithm for a problem. An algorithm is a rooted tree, the root at the top for input. Leaves are at the bottom for output. There is an input space $\mathscr{I}$ a state space $\mathscr{S}$, and an output space $\mathscr{O}$ which are finite-dimensional real vector spaces.



There are two additional types of nodes computation and branching nodes. Computations are rational functions from $\mathscr{S}$ to $\mathscr{S}$. The input and computation nodes have one existing edge. The branch nodes have two and which is chosen is determined by a polynomial inequality $p(x) \geq 0$ or $p(x) < 0$. The input and output functions are rational.

These algorithms are also called tame machines; they are limited, for example, by the fact that there are no loops.

A problem is a subset $X \subset \mathscr{I} \times \mathcal{O}$. Let $f: X \to Y$ be the restriction of the projection $\pi: \mathscr{I} \times \mathcal{O} \to \mathcal{O}$ to X and assume f is surjective. A solution of a problem is a section $\sigma: Y \to X$ such that $f \circ \sigma$ is the identity. A tame machine whose input–output map on inputs $y \in Y$ is a solution to a problem is said to solve the problem.

The minimum number of branch nodes of a tame machine which solves a problem is the branching complexity or topological complexity of the problem.

An example of a problem is the ε-all root problem for $f \in P_d$, the space of complex univariate monic polynomials.

$$X_\varepsilon \subset P_d \times \mathbb{C}^d = \{(f, (a_1, \ldots, a_d)) | f = \prod_{i=1}^{d} (z - r_i) \text{ and } |a_i - r_i| < \varepsilon\} \qquad \text{and}$$

$$Y = P_d.$$

These definitions of problem and algorithm are a large step in the program called for in [Smale, 1985a] for the definition of an algorithm. Given a problem $f: X \to Y$, let $K(f)$ be the kernel of $f^*: H^*(Y) \to H^*(X)$ and $K(f) = \{\gamma \in H^*(Y) | f^*(\gamma) = 0\}$, where $H^*(Y)$ is the singular cohomology ring of Y. The cup length of $K(f)$ is the maximum number of element $\gamma_1, \ldots, \gamma_k$ of $K(f)$ s.t. the cup product $\gamma_1 \cup \cdots \cup \gamma_k \neq 0$. Smale [1987] proves:

Theorem 5. *Let $f: X \to Y$ be a problem. The topological complexity of f is bigger than or equal to the cup length of $K(f)$.*

He applied this result to prove, via a complex algebraic topology computation, the main theorem of Smale [1987].

Theorem 6. *There is a* $\varepsilon(d) > 0$ *such that for all* $0 < \varepsilon < \varepsilon(d)$ *the topological complexity of the* ε*-all root problem for* P_d *is greater than* $(\log_2 d)^{2/3}$.

This result has been vastly improved by Vasiliev, [1988]. Levine [1989] has also worked on n-dimensional analogues.

In the fall of 1987, Lenore Blum and Steve Smale were visiting at Watson. Steve began extending his model of computation from tame machines to allow loops and to work over ordered rings. Soon all three of us were involved.

First, we specify a ring, the functions we compute on the ring, and the branching structure. Our functions are polynomial or rational, involving only a fixed finite number of variables, and having a finite number of nontrivial coordinates. We branch on $\neq 0$ or $= 0$ and ≥ 0 or < 0. Examples are:

1. the integers $\mathbb{Z}$ with polynomial functions (i.e., with $+, -, \times$) and branching on ≥ 0 or < 0;
2. the integers $\mathbb{Z}$ with polynomial functions and branching on $\neq 0$ or $= 0$;
3. the reals $\mathbb{R}$ with rational functions and branching on ≥ 0 or < 0;
4. the complexes $\mathbb{C}$ with rational functions and branching on $\neq 0$ or $= 0$.

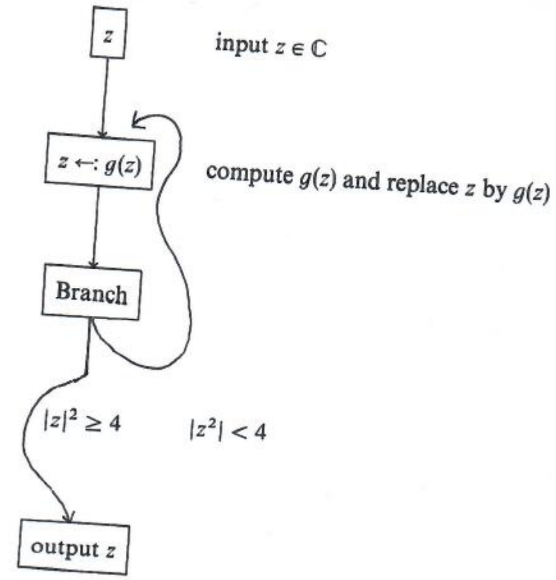
Frequently, we suppress the functions or the branching structure. If we discuss machines over $\mathbb{Z}$, $\mathbb{R}$, or $\mathbb{C}$ without further qualification, we mean 1, 3, and 4, respectively.

Our machines now are finite-directed graphs with one input node, computation nodes, branch nodes, output nodes, and a certain fifth node. The input and output spaces are the infinite direct product of the ring with itself, R^∞, and the state space is $\mathbb{Z}_+ \times \mathbb{Z}_+ \times R^\infty$. All computations only involve and affect a finite number of the coordinates of R^∞. The two $\mathbb{Z}_+$'s are like counters and a fifth node will copy the contents of the jth coordinate to the ith of R^∞ if the first two coordinates are (i, j).

This is all worked out formally in [Blum–Shub–Smale, 1989] except that there we always assume the ring ordered and branching done on ≥ 0 or < 0.

By doing things in an integrated way, we hope that the various settings will illuminate one another, that the theory of recursive functions and complexity over $\mathbb{R}$ or $\mathbb{C}$, for example, will benefit from the more developed discrete theory. On the other hand, we do not want to be so general that we lose the basic contacts with algebra and the geometry and topology of the reals and complexes.

A simple example of a machine over $\mathbb{R}$ is given by the iterates of the complex polynomial $g(z) = z^2 + 1$.



This machine halts on the basin of infinity and fails to halt on its complement, the Julia set.

The computable functions over the integers are the usual computable functions, while over the reals, most Julia sets are not halting sets. Universal machines are constructed over any ring.

To study the complexity of a problem, two additional pieces of data are required: the input size of the problem and the cost of the computation.

The *size* of an element is the *length* plus the *height*. The *length* of an element is the least k such that $x = (x_1, \ldots, x_{k-1}, 0, \ldots, 0, \ldots)$ and $x_i = 0$ for $i \geq k$. The *height* of $x = (x_1, \ldots, x_i, \ldots)$ is max *height* x_i over all i. Now we need to define the height of an element of the ring R. For the integers $\mathbb{Z}$, height $x = \log(|x| + 1)$. With this notion of height, we will call the size the *bit size*. For the reals $\mathbb{R}$ or complexes $\mathbb{C}$, the height is perhaps most naturally taken as one for any element. Then the size is the dimension. But one might also want the height to reflect the size of the number and then one might take height $x = \log(|x| + 1)$ in analogy with the integers. I will call this logarithmic height and the corresponding size logarithmic size. Given a notion of height, there is a natural notion of cost of a computation. Suppose on input $y \in \mathscr{I}$, machine M visits $(m - 1)$ nodes in succession and finally arrives at an output node. Then we say the time of the computation $Y_M(y) \equiv m$ and the cost $C_M(y)$ is the time $_M(y) \times$ max height generated in the computation. For the integers and logarithmic height, we call the cost the bit cost. For height identically one, we call the cost the algebraic cost.

Thus, we have various settings in which to discuss complexity. I list a few.

Input size	Ring	Functions	Branching	Cost
1) bit	$\mathbb{Z}$	polynomial	≥ 0 or <0	bit
2) bit	$\mathbb{Z}$	polynomial	≥ 0 or <0	algebraic
3) bit	$\mathbb{Z}$	polynomial	$=0$ or $\neq 0$	bit
4) bit	$\mathbb{Z}$	polynomial	$=0$ or $\neq 0$	algebraic
5) dimension	$\mathbb{R}$	rational	≥ 0 or <0	algebraic
6) logarithmic	$\mathbb{R}$	rational	≥ 0 or <0	logarithmic
7) dimension	$\mathbb{C}$	rational	$\neq 0$ or $=0$	algebraic
8) logarithmic	$\mathbb{C}$	rational	$\neq 0$ or $=0$	logarithmic
9) bit	$\mathbb{Z}$	linear	$\neq 0$ or $=0$	bit
10) dimension	$\mathbb{R}$ or $\mathbb{C}$	linear	$=0$ or $\neq 0$	algebraic

Given a problem $X \subset \mathscr{I} \times \mathscr{O}, f: X \to Y$, we say that f is in class P if there is a machine M which solves the problem, a real constant $c > 0$, and a positive integer q such that for input $y \in Y$, $C_M(y) \leq c(\text{size } y)^q$.

A special class of problems are decision problems (Y, Y_{yes}): Given a set of inputs Y and a subset Y_{yes}, determine if $y \in Y$ is in Y_{yes}.

A decision problem (Y, Y_{yes}) is in NP if there are constants $c > 0$, $q \in \mathbb{Z}_+$ and a machine M with input space $\mathscr{I} \times \mathscr{I}'$ such that on inputs $(y, y') \in Y \times \mathscr{I}'$

(a) M outputs 1 (yes) or 0 (no);
(b) M outputs 1 only if $y \in Y_{\text{yes}}$;
(c) if $y \in Y_{\text{yes}}$, then there is a $y' \in \mathscr{I}'$ such that M outputs 1 on (y, y') and $C_M(y, y') \leq c(\text{size } y)^q$.

This is the formalism of NP completeness theory (see [Garey–Johnson, 1979]). In analogy with the standard problem in setting (1), one may ask for decision problems:

Problem. Does $P \neq NP$?

A decision problem (Y, Y_{yes}) is called NP complete if given any other decision problem (Y', Y'_{yes}) there is a machine M which maps (Y', Y'_{yes}) to (Y, Y_{yes}) faithfully (i.e., M takes an input $y' \in Y'$ into Y_{yes} iff $y' \in Y'_{\text{yes}}$) and $\exists c > 0$, $q \in \mathbb{Z}_+$ such that $C_M(y') \leq c(\text{size } y')^q$ for all $y' \in Y'$.

Thus, $P \neq NP$ iff any NP-complete problem is not in P.

For each complexity setting (1)–(10) above, we may consider the existence of NP complete problems and the question: Is $P \neq NP$? In [Blum–Shub–Smale, 1989], we mainly considered case (5) which we simply refer to as over $\mathbb{R}$. Similarly, the standard setting (1) is called over $\mathbb{Z}$ and (7) is called over $\mathbb{C}$. The next theorem is proved in [Blum–Shub–Smale, 1989] over $\mathbb{R}$, but the arguments apply more generally; so I state it over $\mathbb{C}$ as well.

Theorem 7. (1) *Let 4-satisfiability be the decision problem $(\mathscr{F}, \mathscr{F}_{\text{yes}})$ where $\mathscr{F}$ is the set of all 4th degree polynomials $f: \mathbb{R}^n \to \mathbb{R}$ for all $n \in \mathbb{Z}_+$ and $\mathscr{F}_{\text{yes}}$ are* *those f such that* $\exists x \in \mathbb{R}^n$ *with* $f(x) = 0$. *Then 4-satisfiability is NP complete over R.*

(2) *Let Hilbert Nullstellensatz (HN) be the decision problem* $(\mathcal{F}, \mathcal{F}_{\text{yes}})$ *where* $\mathcal{F}$ *is all sets of k polynomials of arbitrary degree*

$$f_i\colon \mathbb{C}^n \to \mathbb{C}, \quad i = 1, \ldots, k,$$

for all n and $k \in \mathbb{Z}_+$ *and* $\mathcal{F}_{\text{yes}}$ *are those for which the algebraic set they define is nonempty, i.e.,* $\exists x \in \mathbb{C}^n$ *s.t.* $f_i(x) = 0$, $\forall i = 1, \ldots, k$, *or what is the same set of those for which 1 is not in the ideal generated. Then HN is NP-complete over* $\mathbb{C}$.

The $\mathcal{F}$ in HN could also be replaced by systems of degree $\leq d$ where $d \geq 2$.

In Rio de Janeiro, last January, Smale discussed the problem of the existence of NP-complete problems and the question $P \neq NP$ in various settings. Megiddo [1993] proves a general NP-completeness theorem. The question $P \neq NP$ does not make much sense in all of our settings because in settings (2) and (4), there are problems in NP which are not even decidable. In Rio, Smale proved that $P \neq NP$ in settings (9) and (10) and presented some ideas on (3), (6), and (8). In [Shub, 1993], I give a simple argument for setting (3).

To extend the notion of the polynomial time algorithm to the numerical analysis context requires incorporating approximate solutions and round-off error into the problem. It is not exactly clear in all cases how to do this. A beginning was made in [Blum–Shub–Smale, 1989; Smale, 1990] went further.

In [Blum–Shub–Smale, 1989, we consider ε as a variable and the effect of scaling on approximate solutions. We take as a new size the logarithmic size $+|\ln \varepsilon|$, and as cost function the algebraic cost. In [Smale, 1990], the round-off error is incorporated and probabilistic algorithms. For round-off error, the size is taken as dimension $+\ln|\varepsilon| + \ln W$, where W is a weight which might reflect the logarithmic size and the inverse of the distance to the ill-posed problems. The cost reflects the admissible round-off error and the polynomial class is called numerically stable. Whereas the notion of numerical stability is good for computations, it does not seem good for algorithms because according to the definition there is no numerically stable algorithm for the square root. So here I will make another attempt which is slightly different in the definition of admissible error and uses a different cost function. The computation of a machine M on input y is described by the computing endomorphism $(n_i, x_i) = H_M(n_{i-1}, x_{i-1})$, where essentially (n_{i-1}, x_{i-1}) is the (node, state) at time $i - 1$ and (n_i, x_i) is the (next node, next state) with the further conditions that n_0 is the input node, $x_0 = y$, $n_{T_M(y)}$ is an output node, and M output $x_{T_M(y)}$.

We will say that the sequence (n_i, x_i) is a *δ-pseudo-computation* if n_0 is the input node, and for $i \geq 1$ $|(n_i, x_i) - H_M(n_{i-1}, x_{i-1})| \leq \delta$.

Now given a problem $f\colon X \to Y$, a machine M, an input $y \in Y$, and an $\varepsilon > 0$, then δ is an *admissible error (round-off and input error)* for (ε, y, M) if for any δ-pseudo-computation x_i with $|y - x_0| \le \delta$ there is a first time I such that n_I is an output node and x_I is within ε of a solution to problem instance y, i.e., $\exists x \in f^{-1}(y)$ such that the distance from x_I to x is less than or equal to ε.

Let $\delta(\varepsilon, y) = \delta(\varepsilon, y, M)$ be the maximum admissible round-off error. The round-off cost

$$C_R(\varepsilon, y) = \min_{0 < \delta \le \delta(\varepsilon, y)} \text{ max over } \delta\text{-pseudo-computations of}$$

$$I \times \left(\max_{0 \le i \le I} \text{ logarithmic height } (x_i) + |\log^- \delta(\varepsilon, y)| \right),$$

where $\log^-$ is $\min(\log, 0)$.

The input size of input (ε, y) as in [Smale, 1990] is taken as

$$S(\varepsilon, y) = \text{dimension} + \log|\varepsilon| + \log W,$$

where W is a weight representing the size of y, the inverse of the distance of y to the ill-posed problems, or something like the condition of the problem instance y; see [Smale, 1990] for a discussion of this. The polynomial class are those problems for which there are algorithms (machines) for which $\exists c > 0$ and $q \in \mathbb{Z}_+$ such that

$$C_R(\varepsilon, y) < cS(\varepsilon, y)^q.$$

For univariate polynomial root-finding, Myong-Hi Kim has found such algorithms [Kim, 1988b]. When round-off error is not taken into account, $|\log|\log(\varepsilon)||$ is perhaps more appropriate than $|\log \varepsilon|$; see [Renegar, 1987b]. The role of the distance to the ill-posed problems is less clear, although size of coefficients must play a role because of scaling.

The work of Smale in conjunction with Blum [Blum–Smale, 1993] on Gödels theorem is in this volume so I will not comment on it. Also to be mentioned are the expository articles [Smale, 1988; 1989] where some of Smale's philosophy on real number machines are exposited for a general audience.

Appendix: Personal Reminiscences by Mike Shub

I first met Steve Smale during the 1961–62 academic year. I was a sophomore at Columbia College. Some of my older roommates had a copy of one of Steve's papers on structural stability and they couldn't make out some of the subtleties of the definition. I think it had to do with epsilons and deltas. Somehow they put me up to going to see Steve and asking about it. Steve answered my question rapidly and succinctly but at a depth I didn't even know existed. Apparently I had phrased my elementary question in terms close to problems he was thinking about. He must have thought me quite presumptuous. I am still frequently shocked when Steve answers some question of mine or a colloquium question at a depth I didn't imagine.

The next year my friends advised me to try to take Steve's graduate course in differential topology since he was a famous topologist who had proven a great theorem. In those days Columbia College didn't have much of an undergraduate mathematics curriculum. Many math majors vied in taking graduate courses which we were frequently hopelessly unprepared for. I enrolled for Steve's course. The very first day, he arrived and announced that the course would be about infinite-dimensional differential topology because that was where the most interesting work was to be done. In the class the first day was Sammy Eilenberg and other luminaries of the Columbia math department. Steve began by defining the derivative in Banach space. He didn't quite get it right, and the class degenerated as various of the luminaries shouted out suggested corrections. I ran out and bought Lang's infinite-dimensional *Introduction to Differential Topology* downtown at the publisher's office (it wasn't in the bookstores yet), and began struggling with Lang and Ralph Abraham's course notes which were trailing the lectures. I was always behind, but some of the seniors could follow. Sometimes as I was sitting through a lecture which I couldn't understand I broke into giggles as Steve would get confused at the blackboard only to be saved by an undergraduate. One day, Serge Lang took David Frank who was also enrolled in Steve's course and me to lunch. He explained that while the undergraduates were locally correct, Steve was almost always locally wrong but globally correct. Actually, in the many courses I have taken from Steve since I haven't noticed so many local errors, but Serge's hyperbole was comforting at the time. Steve left Columbia after the spring semester 1964 for Berkeley. David Frank and I decided to go to Berkeley for graduate school. Steve and Clara offered us their car to drive across the country. David and Kathy Simon were getting married. So the three of us first drove to Pittsburgh for their wedding and then on to California. Beth Pessen and I got married that September.

The fall semester at Berkeley that year was dominated by the Free Speech Movement. David, Kathy, Beth, and I were loyal foot soldiers in the movement. Steve Smale and Moe Hirsch were prominent faculty supporters. The Free Speech Movement had a very good effect on faculty–student relations in general on the Berkeley campus. The faculty and students were thrown together and made common cause on a political matter where they were more equals than in academic disciplines. The faculty became more aware of student concerns and reached out to accommodate a spirit of reform and even revolution. David, Kathy, Beth, and I were arrested in Sproul Hall. David and I were, I believe, the only math graduate students among the 800 or so students arrested there on December 8. David was taken to the Berkeley jail. I to Santa Rita. Steve actually went down to the Berkeley jail and bailed David out. I was released early the next morning when the faculty raised the funds to bail us out en masse.

That spring Steve was co-chairman of the Viet Nam Day Committee with Jerry Rubin. I was on the steering committee. Sometime that year I remember Steve telling Charles Pugh that he had proven that structurally stable systems were not dense. I was amazed by his ability to do research in the midst of all the turmoil. Beth and I were frequently at Steve and Clara's for dinners and parties that year and the next, and they were sometimes at ours. The last one that I remember was in our house shortly before Steve and Clara left for Europe in the summer of 1966. I remember telling Steve that I thought a teach-in should be organized at the International Congress of Mathematicians in Moscow that year. Given what actually happened on the steps of Moscow University as Steve recounts it, a teach-in in 1966 was quite far-fetched. By the spring of 1966 I had already become Steve's graduate student. The initial problem, which was quickly done, was to prove the Kupka–Smale theorem for endomorphisms. I used to stop by at Steve's office almost daily to say hello, and tell him anything new or ask questions. He was always happy to see me and to hear anything new. But he wasn't too interested in technical details or vague ideas. When he started biting his lower lip, I knew it was time to go. On one of these visits I told Steve that I thought that the squaring map on the circle was structurally stable. Thus, the more extensive part of my thesis research began. Steve was always helpful and encouraging, and good about the big picture. I remember some advice which took place in strange circumstances. Once we encountered each other running in different directions as the police were breaking up a demonstration on Telegraph Avenue. Steve stopped for a moment and said that he understood why expanding maps were stable; they were contracting. I never found out what he meant as we had to start running again almost immediately. Another time, we were in the Greek Theater where some sort of Vietnam War protest was taking place; in the midst of watching events on the stage, Steve asked me if I could prove the expanding map conjecture if I knew the fundamental group had a nilpotent subgroup of finite index. I already knew how to do it if the groups was nilpotent.

Those years in Berkeley were heady days not only for politics but for dynamical systems. Steve returned from infinite dumensions to dynamical systems theory. After the nondensity of structurally stable systems, he proved the omega stability theorem. He was writing his 1967 *Bulletin* paper which was a distillation and amplification of his previous work. The paper is a major restructuring of ordinary differential equations from the point of view of one of the leading topologists of the time. Steve's enthusiasm and the scope of his vision created a large group working on dynamics. Charles Pugh joined the faculty at Berkeley in 1964. Moe Hirsch got involved in dynamics, partly he has claimed because Steve went on leave and Steve's students came around to talk to Moe. Jacob Palis, Nancy Kopell, and I were the first bunch. There was enough interesting work for all of us and plenty more. Some of it was important for Steve's own work as well. In 1969, I was sharing an office with Steve at Warwick during the dynamical system year. I had been puzzled by a certain aspect of the stable manifold theorem for hyperbolic sets on and off for two years. Finally, I could put my finger on my objection. A technical point in the general theorem was not correct and the omega stability theorem depended on it. Steve tried to fill the gap for a while without success. A few days later, we were in our office and Steve was calmly sitting and working on something else. I asked him how come. He said he had been asking us guys to prove the theorem for some years and that we kept saying that we could. A little later that summer, Charlie, Jacob, Moe, and I indeed did prove a version which was correct and all that Steve needed.

Steve is always conscious of what he is doing and evaluating its position within science. He is willing to undertake enormous projects over the long term on subjects he finds important. He starts out full of energy and conviction that he will do something important and perhaps a bit naively, but he is extremely flexible and learns rapidly along the way. Partly, learning proceeds from talking to people a lot and taking what they say very seriously. Partly, it comes from going to the library a lot. His time in the library and confidence that he can learn what he needs to there remind me of the story I remember (I hope correctly) about Steve's education in a one-room schoolhouse where he looked up how to solve linear equations in the encyclopedia. By the time Steve's papers are finally written, they tend to be so clear and well-organized that it is difficult to detect the enormous effort that went into them. I have been partly involved in Steve's project on the theory of computation, and Steve and I have written a few joint papers by now. In 1981 while we were working on polynomial root-finding, we got a bit competitive as is both our wonts. Steve always works very hard. But I was lucky to be on sabbatical while he had to teach, so I could (barely) hold my own.

Over the years, my friendship with Steve has deepened. Recently Steve, Clara, Beate, and I were on our terrace in New York having a drink. Beate and I have been married for two years now. Steve described his plans for his photography. During the three days Steve was in New York to give a lecture, he also scoured the city looking for the perfect photography paper. "You see, Beate," I said, "Steve is a man of no small ambitions" and, I should have added, successes. I think that is true and marvellous. Yet Steve is gentle, direct, unpretentious, and honest. His views are frequently novel and refreshing from mathematics to movies and politics. His reactions personally and politically have always been sympathetic and on the side of basic human rights and decency, as long as I've known him from the Free Speech Movement and Vietnam protests until now. I have great admiration, respect, and affection for Steve and feel very lucky to have been his student and to be his colleague and friend.

References

Aho, A., Hopcraft, J., and Ullman, J. [1979], *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA.

Blum, L. and Shub, M. [1986], Evaluating rational functions: Infinite precision is finite cost and tractable on average, *SIAM J. Comput.* **15**, 384–398.

Blum, L., Shub, M., and Smale, S. [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.

Blum, L. and Smale, S. [1993], The Gödel incompleteness theorem and decidability over a ring, this volume.

Branin, Jr., F.H. [1972], Widely convergent methods for finding multiple solutions of simultaneous non-linear equations, *IBM J. Res. Develop.* **16**, (5) 504–522.

Canny, J. [1988], Some algebraic and geometric computations is P-space, Proceedings of the 20th Annual ACM Symposium on the Theory of Computing, pp. 460–467.

Canny, J. [1990], Generalized characteristic polynomials, *J. Symbol. Comput.* **9**, 241–250.

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

Demmel, J. [1987b], The geometry of ill conditioning, *J. Complexity* **3**, 201–229.

Doyle, P. and McMullen, C. [1989], Solving the Quintic by iteration, *Acta Math.* **163**, 152–180.

Edelman, A. [1988], Eigenvalues and condition numbers of random matrices, *SIAM J. Matrix Anal. Appl.* **9**, 543–560.

Friedman, J. [1990] Random polynomials and the density of approximate zeros *SIAM J. Comput.*, **19**, 1068–1099.

Garey, M. and Johnson, D. [1979], *Computers and Intractability*, Freeman, New York.

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

Kim, M.-H. [1988a], On approximate zeros and root finding algorithms, *Math. Comput.* **51**, 707–719.

Kim, M.-H. [1988b], Error analysis and bit complexity: Polynomial root finding problem, Part I, preprint, Bellcore, Morristown, NJ.

Levine, H. [1989], A lower bound for the topological complexity of Poly(D, n), *J. Complexity* **5**, 34–44.

McMullen, C. [1988], Braiding of the attractor and the failure of iterative algorithms, *Invent. Math.* **91**, 259–272.

Megiddo, N. [1993], A general NP-completeness theorem, this volume.

Neff, C.A. [1990], Specified Precision Polynomial Root Isolation Is in NC, IBM Research Report, RC 15653, and to appear *J. Computer Syst. Sci.*

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

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

Renegar, J. [1989], On the worst case arithmetic complexity of approximating zeros of systems of polynomials, *SIAM J. Computing* **18**, 350–370.

Renegar, J. and Shub, M. [1992], Unified complexity analysis for Newton LP methods, *Math. Programming*, **53**, 1–16.

Schönhage, A. The fundamental theorem of algebra in terms of computational complexity, preliminary report, University of Tubingen.

Shub, M. and Smale, S. [1985], Computational complexity, on the geometry of polynomials and a theory of cost: Part I, *Ann. Scient. Ecole Norm. Sup.* **18**, 107–142.

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

Shub, M. and Smale, S. [1986b], On the existence of generally convergent algorithms, *J. Complexity* **2**, 2–11.

Shub, M. [1993], Some remarks on Bezout's theorem and complexity theory, this volume.

Smale, S. [1976], A convergent process of price adjustment and global Newton methods, *J. Math. Econ.*, **3**, 107–120.

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

Smale, S. [1983a], On the average number of steps in the simplex method of linear programming, *Math. Programming* **27**, 241–262.

Smale, S. [1983b], The problem of the average speed of the simplex method, *Proceedings of the XI International Symposium on Mathematical Programming, June 1983*, Bachem et al., eds., Springer-Verlag, Berlin, pp. 530–539.

Smale, S. [1985a], On the efficiency of algorithms of analysis, *Bull. Amer. Math. Soc.* **13**, 87–121.

Smale, S. [1985b], Newton's method estimates from data at one point, *The Merging of Disciplines: New Directions in Pure, Applied and Computational Methematics*, R.E. Ewing, et al. eds., Springer-Verlag, New York.

Smale, S. [1986], Algorithms for solving equations, Proceedings of the International Congress of Mathematicians, Berkeley, CA, American Mathematical Society, Providence, R.I., pp. 172–195.

Smale, S. [1987], On the topology of algorithms I, *J. Complexity* **3**, 81–89.

Smale, S. [1988], The Newtonian contribution to our understanding of the computer, *Queen's Quarterly* **95/1**, 90–95.

Smale, S. [1989], Newton's Contribution and the Computer Revolution, *Math. Medley* **2**, pp. 51–57.

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

Sutherland, S. [1989], Finding roots of complex polynomials with Newton's method, preprint, Institute for Math Science, SUNY, Stoney Brook.

Thom, R. [1966], Sur les Travaux de Stephen Smale, Proceedings of International Congress of Mathematicians, Moscow, pp. 25–28.

Tischler, D. [1989], Critical points and values of complex polynomials, *J. Complexity* **5**, 438–456.

Vasiliev, V.A. [1988], Braid group cohomologies and algorithm complexity, *Funct. Anal. Appl.* **22** (3), 182–190 (English translation).