# *Hilbert's 13th Problem*
# *Great Theorem; Shame about the Algorithm*

Bill Moran

# Structure of Talk

- Solving Polynomial Equations
- Hilbert's 13th Problem
- 'Kolmogorov-Arnold Theorem
- Neural Networks

# Quadratic Equations

- $$\boxed{ax^2 + bx + c = 0}$$

$$\boxed{x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

- How do we do it?
- Eliminate the $x$ term by replacing $x$ by $y = x + \frac{b}{2a}$

$$ay^2 + c - \frac{b^2}{2a} = 0$$

# What about Cubics?

$$\boxed{ax^3 + bx^2 + cx + d = 0} \tag{1}$$

- Eliminate $x^2$ term — replace $x$ by $y = x + \frac{b}{3a}$:

$$y^3 + c'y + d' = 0$$

- Write $y = u + v$

$$u^3 + v^3 + (3uv + c')(u + v) + d' = 0$$

- Set $3uv + c' = 0$

$$u^3 - \left(\frac{c'}{3u}\right)^3 + d' = 0$$

- Quadratic in $u^3$ — solve quadratic and take cube roots
- This gives $u$, then get $v$, then $y$ and finally $x$.
- del Ferro, Tartaglia, Cardano, 1530

# Let's be a little more adventurous

$$\boxed{ax^4 + bx^3 + cx^2 + dx + e = 0}$$

- Similar trick to cubic case to remove cubic term:

$$y^4 + py^2 + qy + r = 0$$

- Complete the square:

$$(y^2 + \frac{p}{2})^2 = \frac{p^2}{4} - qy - r$$

- Introduce new variable $z$: $(y^2 + \frac{p}{2} + z)^2$ — this is: $(y^2 + \frac{p}{2})^2 + pz + 2y^2 z + z^2$
- Then

$$(y^2 + \frac{p}{2} + z)^2 = 2zy^2 - qy + \left(z^2 + zp + \frac{p^2}{4} - r\right)$$

# Quartic Continued

- Choose $z$ to make RHS a perfect square — so discriminant $0$:

$$q^2 = 8z\left(z^2 + zp + \frac{p^2}{4} - r\right)$$

- Solve this cubic for $z$ then we have $A^2 = B^2$ where $A = (y^2 + \frac{p}{2} + z)$ and $B^2 = 2zy^2 - qy + \left(z^2 + zp + \frac{p^2}{4} - r\right)$
- $A = \pm B$ gives two quadratics in $y$
- Lodovico de Ferrari, Cardano

# Quintic

$$\boxed{ax^5 + bx^4 + cx^3 + dx^2 + ex + f = 0} \tag{2}$$

- Tschirnhaus transformations:
  - $y = \frac{g(x)}{h(x)}$
  - $g$ and $h$ polynomials $h$ non-vanishing at roots of quintic
- Can use Tschirnhaus transformations to reduce (2) to the Bring-Jerrard form:

$$x^5 - x + q = 0 \tag{3}$$

- $q$ is some rational function of the coefficients in (2)
- Can obtain solutions of (2) as rational functions of roots of (3)
- (Hermite) Elliptic modular functions involving $q$ are used to solve

# Lest you think this is useless nonsense!

## Root solver and associated method for solving finite field polynomial equations

US 20020170018 A1

### ZUSAMMENFASSUNG

An error correction algebraic decoder uses a key equation solver for calculating the roots of finite field polynomial equations of degree up to six, and lends itself to efficient hardware implementation and low latency direction calculation. The decoder generally uses a two-step process. The first step is the conversion of quintic equations into sextic equations, and the second step is the adoption of an invertible Tschirnhausen transformation to reduce the sextic equations by eliminating the degree 5 term. The application of the Tschirnhausen transformation considerably decreases the complexity of the operations required in the transformation of the polynomial equation into a matrix. The second step defines a specific Gaussian elimination that separates the problem of solving quintic and sextic polynomial equations into a simpler problem of finding roots of a quadratic equation and a quartic equation.

| | |
|---|---|
| **Veröffentlichungsnummer** | US20020170018 A1 |
| **Publikationstyp** | Anmeldung |
| **Anmeldenummer** | US 09/842,244 |
| **Veröffentlichungsdatum** | 14. Nov. 2002 |
| **Eingetragen** | 24. Apr. 2001 |
| **Prioritätsdatum** ⓘ | 24. Apr. 2001 |
| **Auch veröffentlicht unter** | US6792569 |
| **Erfinder** | Charles Cox, Martin Hassner, Barry Trager, Shmuel Winograd |
| **Ursprünglich Bevollmächtigter** | International Business Machines Corporation |
| **Zitat exportieren** | BiBTeX, EndNote, RefMan |

Patentzitate (3), Referenziert von (2), Klassifizierungen (5), Juristische Ereignisse (4)

**Externe Links:** USPTO, USPTO-Zuordnung, Espacenet

# Sextic

- $$\boxed{ax^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g = 0} \quad (4)$$
- Tschirnhaus transformations:

$$x^6 + px^2 + qx + 1 = 0. \quad (5)$$

- Its solution is $\phi(p, q)$.
- Solution uses derivatives of generalized hypergeometric functions wrt their parameters called Kampé de Fériet functions

# Septic

- $$\boxed{ax^7 + bx^6 + cx^5 + dx^4 + ex^3 + fx^2 + gx + h = 0} \quad (6)$$
- Tschirnhaus transformations:

$$x^7 + px^3 + qx^2 + rx + 1 = 0. \quad (7)$$

- Its solution is $\phi(p, q, r)$.
- Hilbert: Can we express $\phi(p, q, r)$ in terms of functions of $2$ variables?
- Measure of complexity of problem

# What this means

- A function $f(x_1, x_2, \ldots, x_n)$ of $n$ variables is a superposition of functions $g_k(y_{k,1}, y_{k,2}, \ldots, y_{k,r_k})$, $(k = 0, 1, \ldots, m)$ if each $y_{k,i}$ is one of the variables $x_j$ and there is a function $h$ so that

$$\begin{aligned} &f(x_1, x_2, \ldots, x_n) \\ &\quad = h(g_1(y_{1,1}, y_{1,2}, \ldots, y_{1,r_1}), g_2(y_{2,1}, y_{2,2}, \ldots, y_{2,r_2}), \ldots \\ &\qquad \ldots, g_m(y_{m,1}, y_{1,2}, \ldots, y_{m,r_m})) \end{aligned}$$

# Solutions of Polynomial Equations and Superposition

- Every solution of a polynomial equation of degree $< 7$ can be written as a superposition of functions of $\leq 2$ variables
- Every solution of a polynomial equation of degree $n$ can be written as a superposition of functions of $\leq n - 4$ variables
- What about degree $7$?

# Hilbert's 13th Problem

A solution of the general equation of degree 7 cannot be represented as a superposition of continuous functions of two variables

What he meant to say was "algebraic" or "analytic" instead of "continuous" as we shall see!

# Why this might be a useful idea

- Most functions we want to compute are composed of functions of at most two variables $(x, y) \to x + y$, $(x, y) \to x.y$, $x \to \frac{1}{x}$, $(x, y) \to \sqrt[y]{x}$, $x \to e^x$, $x \to \log x$, $x \to \sin x$, etc.
- To compute gradients of such functions one can use chain rule
- This approach computes partial derivatives of functions of $n$ variables more efficiently
- Kim, Nesterov, and Cherkasskii (1984)

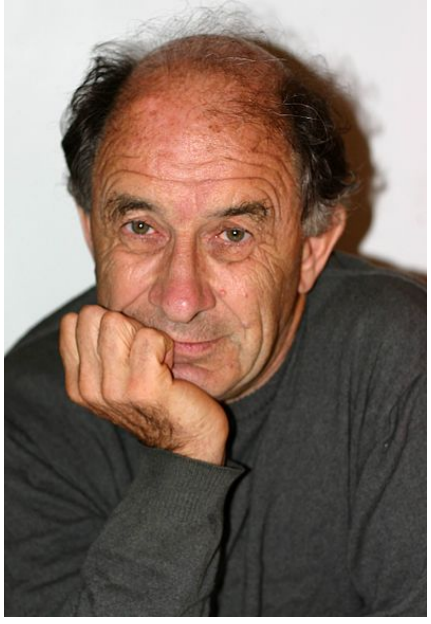
> Given such a computable function of $n$ variables, can compute the function and its gradient in only $4$ times as many operations — for large $n$

# Enter Kolmogorov

- Every continuous function of $n$-variables on the unit cube is a superposition of continuous functions of 3 variables

# Enter Kolmogorov

- Every continuous function of $n$-variables on the unit cube is a superposition of continuous functions of 3 variables
- And Arnold: Every continuous function of $n$-variables on the unit cube is a superposition of continuous functions of 2 variables (Resolves Hilbert's 13th Problem)



# Sprecher's Version

- Sprecher: For each $N \geq 2$ there is a Lipschitz function $\psi$ in $\mathrm{Lip}^{\left(\frac{\log 2}{\log(2N+2)}\right)}(I)$ with the following property: for each $\delta > 0$, there is a rational $\epsilon$ in interval $(0, \delta)$ s.t. for all integers $n$ $(2 \leq n \leq N)$, and for every continuous function $f(x_1, x_2, \ldots, x_n)$ on $I^n$,

$$f(x_1, x_2, \ldots, x_n) = \sum_{0 \leq q \leq 2n} g\left(\sum_{p=0}^{n} \lambda^p \psi(x_p + \epsilon q) + q\right) \quad (8)$$

where $g$ is continuous and $\lambda > 0$ is independent of $f$.

# Idea of Proof — First use discontinuous functions

- $\tau_k(x)$ is $k$th decimal place of $x$ so $x = \sum_{k=1}^{\infty} \frac{\tau_k(x)}{10^k}$ (assume none ends $00000\ldots$, except $0$ itself)
- Write $\psi_r(x) = \sum_{k=1}^{\infty} \frac{\tau_k(x)}{10^{kn+r}}$ for $r = 0, 1, \ldots, n-1$
- Now

$$(x_1, x_2, \ldots, x_n) \to \sum_{r=0}^{n-1} \chi_r(x_{r+1}) = \kappa(x_1, x_2, \ldots, x_n)$$

  is $1-1$ and onto $[0,1]$ but not continuous!
- Interlacing decimals
- Define $g(y) = f\left(\kappa^{-1}(x_1, x_2, \ldots, x_n)\right)$
- And

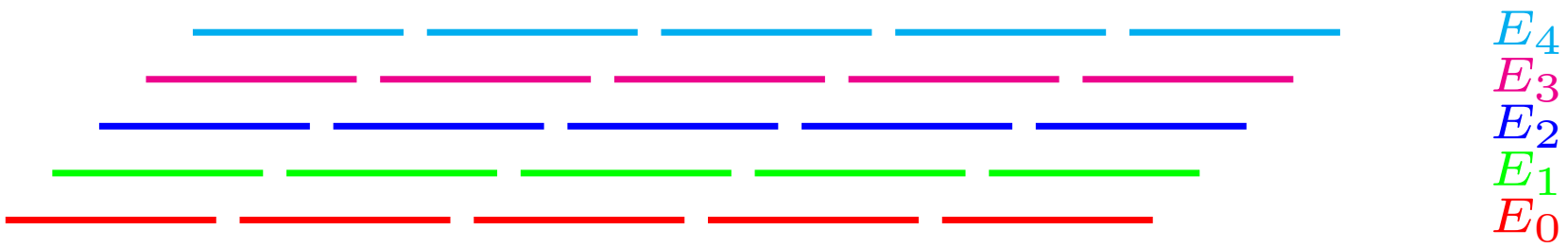
$$\boxed{f(x_1, x_2, \ldots, x_n) = g\Big(\sum_{r=0}^{n-1} \psi_r(x_r)\Big)} \tag{9}$$

# How does it work?

- Two ideas:
  - The map $(x_1, x_2, \ldots, x_n) \to \sum_{p=1}^{n} \psi_r(x_r)$ is $1-1$ — ontoness not needed — but we will need them continuous
  - Then use $g$ to "approximate" values of $f$ on inverse of that map
- Key issue: a continuous version of $1-1$-ness — cannot map $I^n$ in a $1-1$ continuous way into one dimension

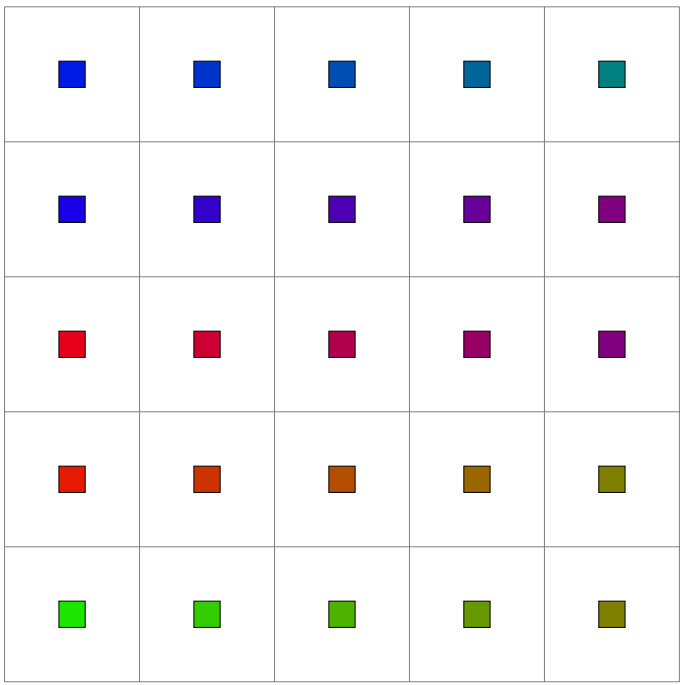
# Continuous Version

- Divide $I = [0, 1]$ into $10$ equal intervals and then shrink them slightly from their centres — call these $E_1(j)$ $(j = 0, 1, \ldots, 9)$
- Repeat this construction $2n + 1$ times ($n$ is number of variables in function)— call them $E_k(j)$
- Shift the new $E_k(j)$ $(k > 1)$ along so that every $x$ in $I$ appears in all but at most one $E_k$



# Done in Two Dimensions

- Take two copies $E_k^i(j)$ and consider $E_k^1(j_1) \times E_k^2(j_2)$
- For each fixed $k$ can find increasing continuous functions $\psi_{k,1}$ and $\psi_{k,2}$ on $I$ such that $\psi_{k,1}(E_k^{(1)}(j)) + \psi_{k,2}(E_k^{(2)}(k))$ are all disjoint for each fixed $k$ — and in 1-dim
- Note: enough to do for one $k$ and then shift to cover all of square — cover square in $2n+1$ shifts)



# Refine this

- Now divide up $I$ into $100$ equal pieces, shrink slightly (less this time) from centre to form $E_2(j)$
- Can adjust old $\psi_{k,1}$ and $\psi_{k,2}$ so that in refined version: $\psi_{k,1}(E_k^{(1)}(j)) + \psi_{k,2}(E_k^{(2)}(k))$ are all disjoint — moreover, adjustment needs only to be small because variation over $E_k(j)$s is small!
- Keep going ...
- We end up sequence of compact sets $E_k$ on each axis and $\psi_{k,i}$ so that $(x_1, x_2) \mapsto \psi_{k,1}(x_1) + \psi_{k,2}(x_2)$ is $1-1$ on each member of sequence and $E_k$ is most of the interval
- Union of $5$ shifts of $E_k$s cover $I^2$

# Approximate

- Fix a continuous function $f$ on $I^2$
- Approximate by a function of the form $g\big(\phi_{k,1}(x_1) + \phi_{k,2}(x_2)\big)$ over most of $I^2$
- Using shifted forms of $\psi$s we can cover all of square $I^2$
- Given $f$ continuous on $I^2$, there exists $g_1$ continuous on $\mathbf{R}$ with $\|g_1\|_\infty \leq \|f\|_\infty$ s.t.

$$\left| f(x_1, x_2) - \sum_{k=1}^{5} g_1\big(\psi_{k,1}(x_1) + \psi(x_2)\big) \right| < (1-\epsilon)\|f\|_\infty$$

- Induct — $f_1 = f$ and

$$f_{r+1}(x_1, x_2) = f_r(x_1, x_2) - \sum_{k=1}^{5} g_r\big(\psi_{k,1}(x_1) + \psi(x_2)\big)$$

- Get $g_r \to g$ and $f_r \to 0$ uniformly so

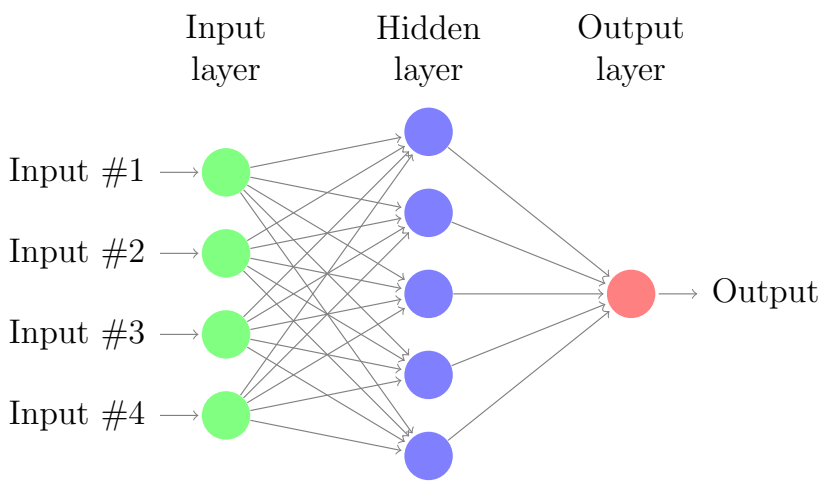
$$\boxed{f(x_1, x_2) = \sum_{k=1}^{5} g\big(\psi_{k,1}(x_1) + \psi(x_2)\big)}$$

# But what about differentiable?

- $$f(x_1, x_2, \ldots, x_n) = \sum_{0 \le q \le 2n} g\Big(\sum_{p=0}^{n} \psi_{p,q}(x_p)\Big) \qquad (*)$$
- (Hilbert) There is an analytic function of three variables that cannot be expressed as a superposition of analytic functions of 2 variables
- (Konrad, 1954) There is a continuously differentiable function of 3 variables that cannot be expressed as a superposition of continuously differentiable functions of 2 variables
- (Fridman, 1967) can replace $\psi$s by Lipschitz functions of exponent $1$
- (Vitushkin, 1964) There exist analytic functions not expressible by (*) when $\psi$s are chosen continuously differentiable

# Neural Networks

- A neuron is a node that takes as input a vector $(y_1, y_2, \ldots, y_M)$ and outputs a value $h(\sum_{m=1}^{M} w_m y_m - w_0)$ where $w_m$ are called weights
- (Hecht-Nielsen, 1987) Kolmogorov-Arnold can be seen as a 3-layer neural network



# Algorithmic Issues

- Functions involved are highly non-smooth and cannot be made smooth
- Only get equality in $(*)$ by letting iteration go to $\infty$

# Making it Computationally Feasible

- Can live with $\epsilon$ rather than equality provided we know how many iterations for a given level of accuracy
- Can use Lipschitz functions!
- (Kurkova "*Kolmogorov's Theorem is Relevant*" 1991-2) Can specify number of iterations in terms of $\epsilon$

# Making it Computationally Feasible II

- (Nakamura, Mines, Kreinovich, $\sim$ 1995) There is an algorithm $U$ that, for every $N \geq 2$, generates an increasing function $\psi \in \mathrm{Lip}^{(\frac{\log 2}{\log(2N+2)})}(I)$, with following property:

For all $\delta > 0$, there exists a real number $\lambda > 0$ and a rational number $\epsilon \in (0, \delta)$ (both computable from $\delta$), s.t., for $2 \leq n \leq N$, every continuous function $f : I^n \to \mathbf{R}$, has a representation as

$$f(x_1, x_2, \ldots, x_n) = \sum_{q=0}^{2n} g\Big(\sum_{p=1}^{n} (\lambda^p \psi(x_p + q\epsilon) + q\Big)$$

for some continuous function $g$ that is computable from $f$

# But ...

Not how NNs are used:

- Train to find weights that fit (perhaps approximate) finite set of input/output data
- Data often has uncertainties
- Overfitting is serious problem
- Evans and Jones — $\gamma$-test
  `http://users.cs.cf.ac.uk/O.F.Rana/Antonia.J.Jones/GammaArchive/Theses/DEvansThesis.pdf`

# But continued ...

- Continuity is about "robustness" but not enough — we really want something like Lipschitz for computability
- If we only want approximation rather than equality then simpler approaches — Projection Pursuit

$$f(x_1, x_2, \ldots, x_n) = \sum_q g_q\Big(\sum_{p=1}^{n} w_p x_p\Big)$$

- Performs well with noisy high dimensional data and nonparametric regression techniques
- Diaconis and Shahshahani (1984) discuss projection pursuit (as equality)

## The End

Questions