Finding roots of high-degree polynomials

D. J. Bernstein
University of Illinois at Chicago
NSF DMS–9970409

ADVERTISEMENT

Riemann $\int_a^b f = c$ means:

for every $\epsilon > 0$,

there is a $\delta > 0$ so that

for every δ-partition p,

$$|p(f) - c| < \epsilon.$$

δ-partition: nondecreasing sequence $a = x_0, t_1, x_1, t_2, \ldots, x_n = b$ with $[x_{i-1}, x_i] \subset (t_i - \delta, t_i + \delta)$.

$p(f) = \sum (x_i - x_{i-1}) f(t_i)$.

ADVERTISEMENT

Generalized Riemann $\int_a^b f = c$:

for every $\epsilon > 0$,

there is a gauge δ so that

for every δ-partition p,

$$|p(f) - c| < \epsilon.$$

Gauge: function $[a, b] \to (0, \infty)$.

δ-partition: ... with

$[x_{i-1}, x_i] \subset (t_i - \delta(t_i), t_i + \delta(t_i))$.

ADVERTISEMENT

Kurzweil-Henstock integral.

Extension of Lebesgue.

Equivalent to Denjoy-Lusin-Perron.

If g is differentiable on $[a, b]$
then $\int_a^b g' = g(b) - g(a)$.

δ flexibility makes proofs easy.

ADVERTISEMENT

For every δ there is a p:
i.e., $[a, b]$ is compact.

Computational proof: Choose
$x' \leq t \leq x$ with $x - x'$ "large"
and $[x', x] \subset (t - \delta(t), t + \delta(t))$.
Repeat for $[a, x'] \cup [x, b]$.

All points are covered after
a finite number of steps.
(And not much δ overlap.)

The problem

Given monic $f \in \mathbf{C}[x]$,
compute all roots of f.

Represent f as coeff sequence.

e.g. Given $f_0, f_1, \ldots, f_{99999} \in \mathbf{C}$,
compute all roots of $x^{100000} +$
$f_{99999}x^{99999} + \cdots + f_1x + f_0$.

Focus on simplest case:

real roots, in $[0, 1]$, not too close.

Standard example:

$$(x - 0.1)(x - 0.2) \cdots (x - 1.0) =$$
$$x^{10} - 5.5x^9 + 13.2x^8 - 18.15x^7$$
$$+ 15.7773x^6 - 9.02055x^5$$
$$+ 3.41693x^4 - 0.84095x^3$$
$$+ 0.12753576x^2 - 0.01062864x$$
$$+ 0.00036288.$$

Algebraic algorithms

Computation: sequence of complex additions, subtractions, multiplications, divisions, using constants and input coeffs.

Identify subsequences that converge (quickly!) to the roots.

The Durand-Kerner method

(Newton, Weierstrass, etc.)

Given monic $f \in \mathbf{C}[x]$, $\deg f = n$,
and distinct $\alpha_1, \ldots, \alpha_n$, define
$W(\alpha_1, \ldots, \alpha_n) = (\beta_1, \ldots, \beta_n)$
where $\beta_k = \alpha_k - f(\alpha_k)/g'(\alpha_k)$
where $g = (x - \alpha_1) \cdots (x - \alpha_n)$.

Choose sensible $\alpha \in \mathbf{C}^n$.

Compute $W(\alpha)$, $W^2(\alpha)$, etc.

$O(n^2)$ operations per step.

Pray that $W^k(\alpha)$ shares
$\approx 2^k$ digits with roots of f.

Graeffe's method

(Dandelin, Lobachevsky, Graeffe)

If $f = (x - r_1) \cdots (x - r_n)$ then
$Q(f) = (x - r_1^2) \cdots (x - r_n^2)$ where
$Q(f) = (-1)^n f(\sqrt{x}) f(-\sqrt{x})$.

Compute $Q(f), Q^2(f), \ldots$.
$O(n^2)$ operations per step.

If $r_1^p \gg r_2^p \gg \cdots$ then

$$\begin{aligned}(x - r_1^p) \cdots (x - r_n^p) \approx \quad & x^n \\ & - r_1^p x^{n-1} \\ & + r_1^p r_2^p x^{n-2} \\ & - r_1^p r_2^p r_3^p x^{n-3} \\ & + \cdots \end{aligned}.$$

Divide: r_1^p, r_2^p, r_3^p, etc.

Somehow extract pth roots.

Tangent Graeffe

(Ostrowski, Malajovich, Zubelli)

Replace $\mathbf{C}$ with $\mathbf{C}[\epsilon]/\epsilon^2$:
polynomials $c + d\epsilon$ with $\epsilon^2 = 0$.

Apply Graeffe's method to $f - \epsilon f'$.
$(f - \epsilon f')(r + \epsilon) = 0$ if $f(r) = 0$.
$(r + \epsilon)^p = r^p + p r^{p-1} \epsilon$,
so simply divide to extract r.

Fast multiplication

FFT multiplies polynomials
with $O(n \log n)$ operations.
Makes Graeffe much faster.

Also speeds up Durand-Kerner:
use divide-and-conquer
to multiply $x - \alpha_1, \ldots, x - \alpha_n$
and to evaluate f, g' at $\alpha_1, \ldots, \alpha_n$.
$O(n(\log n)^2)$ operations per step.

The real world

Switch to more realistic models
of computation:
multitape Turing machine;
1.734GHz Athlon XP; etc.

Handling many digits is slow.

Represent real number
as nearby digit string.
Usually not exact!

$$
\begin{aligned}
&1 && x^{10}\\
&-5.5 && x^{9}\\
&+13.2 && x^{8}\\
&-18.15 && x^{7}\\
&+15.7773 && x^{6}\\
&-9.02055 && x^{5}\\
&+3.41693 && x^{4}\\
&-0.84095 && x^{3}\\
&+0.12753576 && x^{2}\\
&-0.01062864 && x^{1}\\
&+0.00036288 && x^{0}
\end{aligned}
$$

$$
\begin{aligned}
&1 && x^{10}\\
&-5.5 && x^{9}\\
&+13.2 && x^{8}\\
&-18.15 && x^{7}\\
&+15.7773 && x^{6}\\
&-9.0205508 && x^{5}\\
&+3.41693 && x^{4}\\
&-0.84095 && x^{3}\\
&+0.12753576 && x^{2}\\
&-0.01062864 && x^{1}\\
&+0.00036288 && x^{0}
\end{aligned}
$$

0.1	0.099...
0.2	0.200...
0.3	0.299...
0.4	0.401...
0.5	0.492...
0.6	0.629...
0.7	0.663...
0.8	0.825...
0.9	0.885...
1.0	1.002...

For $\deg f = n$ can't expect
any accuracy in roots without
$\approx n$ digits of coeffs of f.

Reasonable problem: Given
$2n$ digits of each coeff,
find n digits of each root.

(Much worse for repeated roots:
consider $(x-1)^n - (0.1)^{n(n-1)}$.)

Can multiply n-digit integers
in time $n^{1+o(1)}$
by FFT-type techniques.
Intuition: time $\approx$ data volume.

Can multiply degree-n polynomials
with n-digit coeffs
in time $n^{2+o(1)}$
if coeffs are on the same scale.

To square $31415x^4 + 92653x^3 + 58979x^2 + 32384x + 62643$:

Square the integer 31415

0000000926530000000589790000000323840000000626430

obtaining 986902225

0058213879900122902289790129638492940134153316350154280756300084379664500405726182400392414544 9.

Read off result coeffs, round.

Just as easy to square

$$0.31415x^4 + 0.92653x^3 + 0.58979x^2 + 0.32384x + 0.62643.$$

Or slanted:

$$31415x^4 + 92653000x^3 + 58979000000x^2 + 323840000000000x + 626430000000000000.$$

Graeffe step 0:

1

1026304387454526304387454526304387453500

4926342314520431780315047137218315563989

14691728417855585165812432290778719096 53

30497939981457319709353505319426053781 52

46785830496052336418121840667046538786 97

54957394793063759780644078218927085980 48

50535678318690413942996878248385001443 34

36874750281171193363372307179981401610 65

21512169028036109360085047747616024611 92

10060596999455470638564549471547468834 69

37654793507899820378626422735748233939 57

11211082998596163441487601404999240473 64

26263549562856991943923721105045744124 03

47595779594661617913584792108188819164 66

65077122663547460122250547917812610735 83

64722781326323921771350481270863000908 10

44350252374760683942694750129169654604 65

19262385523016692971746729939180265648 79

46184296385428090624806368782124570096 87

45523499631629390088247454047359526926 62

Step 1:

1

68032232804324093543068590154853332720152

212065927054922318059358057666295580818040

401826690301869742825956729835082164812550

517764431978572078859132970335536780140300

480864093105207847652307825180319521048200

332889651378323034320488548419358503521400

175227719624736812800112154866128639701600

709118421791187523486668647543470202225700

221616159838313695012685533584649997845400

534340404522679618679708053693946872754900

987706617914242571606176936219807529721600

138267235470068401236043623426849995805600

143714001385919660198453686546712879639500

107553409113664005805989001524188764879300

552033739092746684851902791287202650460100

178959495800366033651044976828443252286700

312882487497848072418700800723170594807800

203106061857651192706920124356225713191500

379206832094358758755896576749240037563500

207238901871096121506157219540650589913000

Step 2:

1

387066159243304955973574935208835819925

652912174019393532335225160449255995247 5

635023342933321849385189038757951873163 8

397028853230243106740795095360166637765 2

168352028098579852532634358015817515541 8

49766230624099916192389331968038259077 20

10382654473540923300480475831304144536 02

153180437390397370076532976557215192025 9

158910911474476307529990201180854471284 2

1145647597494998541572872964613632326672

5643751317027029698605207811062225797829

185855057397442035955767450534942922684 6

39743249422315100102412400187331225052 04

530528172267224828315094399097085812584 2

4176465059906340177998942474360641806096

1742691540608675216705027160119213833248

2934205494217149831465762841793540415429

18264384370733135055882597141078776812 60

5961486706155408725443328103918328

42947962448737807531405426 72

Step 3:

1

192377682748480414086378085136792585854

1410798487068942837182567426825603182501

5178202357298338319809706045327056394724

1071743105988882663159218790320556363737

132061650229209158709437280214498084086 1

98395879611677050146466428348108991152 10

436919826145417952007869076416676041673 1

1111442474650658310996011266721783141490

153737860528810817994852507044821388281

10801414457011791744761186775614237727 48

3579910839400308073805965710461402108677

563274022023979824369604131185756237341 4

399193876066404545700323731698588705741 3

109573631671547516441913760140354990845 9

1277430072066991992304006462416159 68

6054272631705720143256570892 936

797347603279652881250831 84

29861825135824257 7

198509 6

Step 4:

1

879323598503227282262073237419824628311 5

212358965968571378420101736309349521304 7

14578619539507739748484143492035601862 03

4199446064459411647110706384623057403429

566466156482159483423393058452906163649 9

3678569034870597608847452801912009273739

1050361367515788215783339532268481204435

95109869118114663837988217121323920692 88

2643500506401563684268029520153919287383

1877124257260083624858045371491740094289

185062808752554287529838039408667915

54741497647376525194661468041 8

44931417522344895250769 0

248386955918139 52

3687092 29

1

Step 5:

1

348492058947522096313929861129190144754 5

2029757125075392325841707698067574887704

44066712702105703260169666555952591616 08

266265919778614263952588374915818480194 7

42147015016844650062120922401103003583 69

24231045944779637389895011260265904854 76

433314031218439198911215403741404827011 3

363051859548639909498142333998376 82

345627037480672871924297102 2

25555716945781284116

139344297 6

Step 6:

1

808515726479753690707952317869240332103 4

38133085298217642454319105623280785949 42

8639019680865574368263246111793571413267

34735292746396084045770096547787071223 90

5241652394223104413206602687172830073 0

22381879009685202254298982742 5

17019593000760762362 8

10186602 8

Step 7:

1

577431509368648598409364437012203242644 1

1452735244653531470684345774780511896754

4814146818240847483397090022478967290526

11161457837826982218554602361060

119555039415579

Step 8:

1

330521677522448651135041710749411771554621104391350905432439846698993963021478321993308095104715090531090

Step 9:

1

$$1092403584339833568255485031367147386290$$

$$44539533429215884697313168531066222$$

Step 10:

1

119334559018772520338277011201464048228319837700

Step 8, slanted by 36 digits:

3305216
2110439135090543243984669899396302
199330809510471509053109048022513201537 2
12446303090018096843067178435819603772 79
709353592423720078218076067394447133830 3
971128677979129056076840029102448090606 9
836426811797278837466746060928194092959 4
1006074038416021562978627291931003246712
10100128085943840841706600362880
171

Cut off vertically after 60 digits:

$$
\begin{vmatrix}
330 \\
211043913509054324398466989939 \\
199330809510471509053109048022513201537 \\
124463030900180968430671784358196037727 \\
709353592423720078218076067394447133830 \\
971128677979129056076840029102448090606 \\
836426811797278837466746060928194092959 \\
4 \\
100607403841602156297862729193100324671 \\
2 \\
101001280859438408417066003 \\
6
\end{vmatrix}
$$

Apply Q to these nine coeffs.

Gives $\approx$ 40-digit accuracy for inner four coeffs of $Q(f)$.

More generally:

Assume f shape is convex.

Slant to put selected coeff on left.

Inner coeffs: $\leq n/2$ digits from left.

Outer coeffs: $\leq 3n$ digits from left.

Apply Q to outer box.

Gives $\approx 2n$-digit accuracy for inner coeffs of $Q(f)$.

Time: $\approx 3n \cdot \#\{\text{outer coeffs}\}$.

By convexity, $\#\{\text{outer coeffs}\}$
is at most $6 \cdot \#\{\text{inner coeffs}\}$.

What about other coeffs of $Q(f)$?
Repeat with different slants.

How to put reasonable bound on
$\sum_{\text{slants}} \#\{\text{inner coeffs}\}$?

Recall compactness proof.
Select slant producing largest
range of new inner coeffs; repeat.
Overlap is bounded.

Total time $n^{2+o(1)}$ to multiply
degree-n convex-shaped polynomials
with $2n$-digit accuracy.

In particular: For typical f,
Graeffe time $\approx$ input size.