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 $x' \le t \le x$ 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 $x^{100000} + f_{99999}x^{99999} + \cdots + f_1 x + f_0$.

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

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

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 $f(r) = 0$.
$(r + \epsilon)^p = r^p + pr^{p-1}\epsilon$,
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!

1	x^{10}	1	x^{10}
-5.5	x^{9}	-5.5	x^{9}
$+13.2$	x^{8}	$+13.2$	x^{8}
-18.15	x^{7}	-18.15	x^{7}
$+15.7773$	x^{6}	$+15.7773$	x^{6}
-9.02055	x^{5}	-9.0205508	x^{5}
$+3.41693$	x^{4}	$+3.41693$	x^{4}
-0.84095	x^{3}	-0.84095	x^{3}
$+0.12753576$	x^{2}	$+0.12753576$	x^{2}
-0.01062864	x^{1}	-0.01062864	x^{1}
$+0.00036288$	x^{0}	$+0.00036288$	x^{0}
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 $31415x^4 + 92653x^3 + 58979x^2 + 32384x + 62643$:

$$
\begin{array}{lr}
\text{Square the integer} & \underline{31415} \\
\multicolumn{2}{l}{0000000\underline{92653}0000000\underline{58979}} \\
\multicolumn{2}{l}{0000000\underline{32384}0000000\underline{62643}} \\
\text{obtaining} & \underline{986902225} \\
\multicolumn{2}{l}{00\underline{5821387990}0\underline{12290228979}} \\
\multicolumn{2}{l}{0\underline{12963849294}0\underline{13415331635}} \\
\multicolumn{2}{l}{0\underline{15428075630}00\underline{8437966450}} \\
\multicolumn{2}{l}{00\underline{4057261824}00\underline{3924145449}.}
\end{array}
$$

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:

$$\begin{array}{r} 31415x^4 + \\ 92653000x^3 + \\ 58979000000x^2 + \\ 32384000000000x + \\ 62643000000000000. \end{array}$$

Graeffe step 0:

1

1026304387454526304387454526304387453500

4926342314520431780315047137218315563989

1469172841785558516581243229077871909653

3049793998145731970935350531942605378152

4678583049605233641812184066704653878697

5495739479306375978064407821892708598048

5053567831869041394299687824838500144334

3687475028117119336337230717998140161065

2151216902803610936008504774761602461192

1006059699945547063856454947154746883469

3765479350789982037862642273574823393957

1121108299859616344148760140499924047364

2626354956285699194392372110504574412403

4759577959466161791358479210818881916466

6507712266354746012225054791781261073583

6472278132632392177135048127086300090810

4435025237476068394269475012916965460465

1926238552301669297174672993918026564879

4618429638542809062480636878212457009687

4552349963162939008824745404735952692662

Step 1:

1

680322328043240935430685901548533327201 5

2120659270549223180593580576662955808180

4018266903018697428259567298350821648125

5177644319785720788591329703355367801403

4808640931052078476523078251803195210482

3328896513783230343204885484193585035214

1752277196247368128001121548661286397016

7091184217911875234866686475434702022257

2216161598383136950126855335846499978454

5343404045226796186797080536939468727549

9877066179142425716061769362198075297216

1382672354700684012360436234268499958056

1437140013859196601984536865467128796395

1075534091136640058059890015241887648793

5520337390927466848519027912872026504601

1789594958003660336510449768284432522867

3128824874978480724187008007231705948078

2031060618576511927069201243562257131915

3792068320943587587558965767492400375635

2072389018710961215061572195406505899130

Step 2:

1
3870661592433049559735749352088358199250
6529121740193935323352251604492559952475
6350233429333218493851890387579518731638
3970288532302431067407950953601666377652
1683520280985798525326343580158175155418
4976623062409991619238933196803825907720
1038265447354092330048047583130414453602
1531804373903973700765329765572151920259
1589109114744763075299902011808544712842
1145647597494998541572872964613632326672
5643751317027029698605207811062225797829
1858550573974420359557674505349429226846
3974324942231510010241240018733122505204
5305281722672248283150943990970858125842
4176465059906340177998942474360641806096
1742691540608675216705027160119213833248
2934205494217149831465762841793540415429
1826438437073313505588259714107877681260
59614867061554087254433281039183288544
42947962448737807531405426724710

Step 3:

1

1923777682748480414086378085136792585854

1410798487068942837182567426825603182501

5178202357298338319809706045327056394724

1071743105988882663159218790320556363737

1320616502292091587094372802144980840861

9839587961167705014646642834810899115210

4369198261454179520078690764166760416731

1111442474650658310996011266721783141490

1537378605288108179948525070444821388281

1080141445701179174476118677561423772748

3579910839400308073805965710461402108677

5632740220239798243696041311857562373414

3991938760664045457003237316985887057413

1095736316715475164419137601403549908459

1277430072066991992304006462416159686758

60542726317057201432565708929360146

797347603279652881250831843100

2986182513582425772292

19850961866

Step 4:

1

8793235985032272822620732374198246283115

2123589659685713784201017363093495213047

1457861953950773974848414349203560186203

4199446064459411647110706384623057403429

5664661564821594834233930584529061636499

3678569034870597608847452801912009273739

1050361367515788215783339532268481204435

9510986911811466383798821712132392069288

2643500506401563684268029520153919287383

1877124257260083624858045371491740094289

1850628087525542875298380394086679158624

5474149764737652519466146804185073

4493141752234489525076903487

248386955918139524536

3687092299071

16284

Step 5:

1

3484920589475220963139298611291901447545

2029757125075392325841707698067574887704

4406671270210570326016966655595259161608

2662659197786142639525883749158184801947

4214701501684465006212092240110300358369

2423104594477963738989501126026590485476

4333140312184391989112154037414048270113

363051859548639909498142333998376820407

34562703748067287192429710222840

255557169457812841165147

13934429763477

134

Step 6:

1

80851572647975369070795231786924033210 34

38133085298217642454319105623280785949 42

86390196808655743682632461117935714132 67

34735292746396084045770096547787071223 90

52416523942231044132066026871728300730 70

22381879009685202254298982742549 17

17019593000760762362841 22

10186602856 32

Step 7:

1

57743150936864859840936443701220324264441

145273524465353147068434577478051189675 4

48141468182408474833970900224789672905 26

1116145783782698221855460236106712 2

119555039415579862 3

32

Step 8:

1

3305216775224486511350417107494117715546

2110439135090543243984669899396302147832

19933080951047150905310904802

1245

Step 9:

1

1092403584339833568255485031367147386290

44539533429215884697313168531066228529

Step 10:

1

119334559018772520338277011201464048228 3

19837700381

Step 8, slanted by 36 digits:

330521677522449

2110439135090543243984669899396302147832

1993308095104715090531090480225132015372

1244630309001809684306717843581960377279

7093535924237200782180760673944471338303

9711286779791290560768400291024480906069

8364268117972788374667460609281940929594

1006074038416021562978627291931003246712

1010012808594384084170660036288016749357

17141241769

Cut off vertically after 60 digits:

330
21104391350905432439846698939
1993308095104715090531090480225132015372
1244630309001809684306717843581960377279
7093535924237200782180760673944471338303
9711286779791290560768400291024480906069
8364268117972788374667460609281940929594
1006074038416021562978627291931003246712
1010012808594384084170660036

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.