Complex Roots

In this example, we use the functions on screen 9 to find complex roots of the equation

$$f(z) = z^8 - 3z^5 + 2z - 1 = 0.$$

Graphing this polynomial in real mode shows only 2 real roots, so there must be another 6 complex roots. Since the coefficients are real, whenever $z = a + b\,i$ is a root, its complex conjugate $z = a - b\,i$ will also be a root. This means the 6 complex roots consist of 3 conjugate pairs and whenever we find a complex root we have really found two roots.

We will show two different ways to search for complex roots.

(a) Secant Method

This is an iterative method. At each step we have two approximations to a root, z_1 and z_2. The next approximation is then z_3, defined by

$$z_3 = z_2 - f(z_2)\left(\frac{z_2 - z_1}{f(z_2) - f(z_1)}\right)$$

Then compute $f(z_3)$ and reset for the next iteration by updating so that the new z_1, z_2 values are the old z_2, z_3 from the previous iteration, with their function values also updated to match.

For most roots there will be some neighborhood of the root where z_3 will be closer to the root than z_1 or z_2, so the sequence of z_3 values will converge to that root. In practice, this neighborhood might be small, meaning that we pick two initial guesses and start iterating without any guarantee of which root the sequence may ultimately converge to. It can even happen that there is no convergence at all from some starting points.

Define function f1 to evaluate $f(z)$, and f2 to do one secant step.

f1: 9, func, 11, sto, 8, $\mathrm{y^x}$, 11, rcl, 5, $\mathrm{y^x}$, 3, $*$, $-$, 11, rcl, 2, $*$, $+$, 1, $-$

f2: 9, func, 33, rcl, 31, rcl, $-$, 43, rcl, 41, rcl, $-$, /, 43, rcl, $*$, chs, 33, rcl, $+$, 35, sto,
1, $\mathrm{f_n}$, 45, sto, 43, rcl, 41, sto, 45, rcl, 43, sto, 33, rcl, 31, sto, 35, rcl, 33, sto

Complex numbers take two storage registers each, one for the real part and one for the imaginary part. When f1 is being defined, pressing "11, sto" while on screen 9 will show "11, zsto" in the function definition. That indicates a complex number is being stored from the complex stack into registers 11 and 12.

z_1 will be kept in registers 31 and 32, with $f(z_1)$ in 41 and 42.

z_2 will be kept in registers 33 and 34, with $f(z_2)$ in 43 and 44.

z_3 will be kept in registers 35 and 36, with $f(z_3)$ in 45 and 46.

f2 makes a secant step to define z_3, calls f1 for $f(z_3)$, then updates $z_2 \to z_1$, $z_3 \to z_2$, and their corresponding function values to be ready for the next iteration, then shows the latest z_3 in the display.

There is a root near $-0.8 + 1.25\,i$. Secant needs 2 starting points, so let

$z_1 = -0.79 + 1.24\,i$ (31, sto from screen 9 will store this in registers 31 and 32)

$z_2 = -0.80 + 1.25\,i$ (33, sto from screen 9 will store this in registers 33 and 34)

Call f1 for each of these and store the function values in 41,42 and 43,44.

9, func, 12, fix, −0.79 + 1.24 i, enter, 31, sto, 1, $\mathrm{f_n}$, 41, sto, −0.80 + 1.25 i, enter, 33, sto, 1, $\mathrm{f_n}$, 43, sto

2, $\mathrm{f_n}$ will do one secant step and display the next iteration. The next few steps:

−.799535632977 + 1.248559460928 i
−.799568705068 + 1.248622622072 i
−.799568751536 + 1.248623007537 i
−.799568751542 + 1.248623007432 i
−.799568751542 + 1.248623007432 i

Now check the function value with 1, $\mathrm{f_n}$

3.044926889808333696e−24 − 1.048583364786161721e−24 i

So we seem to have about 24 digits correct. Doing 2 more iterations gives $f(z) =$

−1.100000000000000000e−55 − 2.200000000000000000e−55 i

The root is in registers 33 and 34 and should have over 50 digits correct.

Complex secant iteration often takes more steps to converge than it does for real-valued functions. Try starting at $1 + i$ and $2 + 2\,i$. It takes about 20 iterations to converge to

.627078280238 + .047441070579 i

Doing 1, $\mathrm{f_n}$ as before shows this is accurate to over 50 digits.

Secant iteration will not always converge to the root closest to the starting points.

We can use the sum key from screen 7 to iterate the complex secant method. Define f3(z_1, z_2, n) to do n secant steps with starting points z_1 and z_2 by summing f2 from 1 to n by steps of 1.

We are just using the sum function as a loop. All the secant iterates are kept in registers and we don't care about the sum of the terms. We will run f3 from screen 9 and after the sum, f3 will recall register 33 to display the last secant approximation to the root.

f3: 9, func, 38, sto, roll, 33, sto, roll, 31, sto, 1, $\mathrm{f_n}$, 41, sto, 33, rcl, 1, $\mathrm{f_n}$, 43, sto,
7, func, 1, enter, 38, rcl, 1, enter, 2, sum,
9, func, 33, rcl

Use f3 to start at $z_1 = 1 + i$ and $z_2 = 2 + 2\,i$ and do all 20 iterations at once.

9, func, 1+i, enter, 2+2i, enter, 20, enter, 3, $\mathrm{f_n}$

That gives .627078280238 + .047441070579 i as before. Checking with 1, $\mathrm{f_n}$ shows

−1.864542950000000000e−54 − 5.338554041400000000e−53 i.

We don't need the secant method to find the two real roots. After plotting f1 to get starting points for the solv key on screen 6, starting solv at −1 gives −0.944550401394456448164395912277 and starting at 1.5 gives 1.384170071335253499009642716959 as the real roots.

(b) Complex Line Integrals

From the theory of complex functions, the argument principle says that the number of roots for a function $f(z)$ in a region of the complex plane bounded by a curve B is

$$\frac{1}{2\pi i}\int_B \frac{f'(z)}{f(z)}\,dz$$

assuming $f(z)$ has no singularities on or inside B. The integral around the curve B is to be done in a counter-clockwise direction.

A convenient curve B is the circle centered at $a + b\,i$ with radius r.
We can express this circle in parametric form as

$$z = z(t) = (a + b\,i) + r\,e^{i\,t} = (a + r\,\cos(t)) + (b + r\,\sin(t))\,i$$

as t goes from 0 to 2π.

Then we will be integrating with respect to t in parametric form, so the f' term in the integral is

$$\frac{df}{dt} = \frac{df}{dz}\,\frac{dz}{dt}.$$

Here $f(z) = z^8 - 3z^5 + 2z - 1$, so

$$f'(z) = (8z^7 - 15z^4 + 2)\;r\;i\;e^{i\,t}$$

The integration function from screen 6 deals only with real-valued functions, but since we know this integral is a real integer, we just need to compute the imaginary part of the complex integral. That will cause the i multiplied by the imaginary part of the complex integral to cancel with the i in the denominator outside the integral.

$$\frac{1}{2\pi}\int_B g(z)\,dz = \frac{1}{2\pi}\int_0^{2\pi} g(z(t))\,dt$$

where $g(z) = \mathrm{Im}(f'(z)/f(z))$ is the imaginary part of $f'(z)/f(z)$, so it is a real-valued function.

In general, to do a complex line integral of $f(z)$ where both real and imaginary parts of the integral can be non-zero, we make two calls to the real integration function, one for real($f(z)$) and one for imag($f(z)$).

Define f4(t) to compute $g(z(t))$. We will define circle B before using f4 by storing $a + b\,i$ in registers 1,2 and $r + 0\,i$ in registers 3,4. f1 is the function for $f(z)$ from the secant example above. f5 will compute $f'(z)$. It is called from f4 and uses the saved values of $r\,e^{i\,t}$ and $z(t)$ that f4 computes before it calls f5.

f4: 9, func, 0+i, $*$, e^{x}, 3, rcl, $*$, 15, sto,	(save $r\,e^{i\,t}$ in registers 15,16)
1, rcl, +, 13, sto,	(save $z(t)$ in registers 13,14)
1, f_{n}, 17, sto,	(save $f(z)$ in registers 17,18)
13, rcl, 5, f_{n}, 19, sto,	(save $f'(z)$ in registers 19,20)
17, rcl, /, imag, 2, /, π, /	(compute $f'(z)/(2\pi\,f(z))$

f5: 9, func, 7, $\mathrm{y^x}$, 8, $*$, 13, rcl, 4, $\mathrm{y^x}$, 15, $*$, $-$, 2, +, 15, rcl, $*$, 0+i, $*$

f6 takes 2 inputs, $a + b\,i$ and $r + 0\,i$, stores them and then integrates f4.

f6: 9, func, 3, sto, roll, 1, sto,
6, func, 0, π, 2, $*$, 4, $\int_a^b$

To count the roots of $f(z)$ within the circle with center $0 + 0\,i$ and radius 1:

9, func, 0, enter, 1, enter, 6, $\mathrm{f_n}$ 5.000000000000 + .000000000000 i

This means there are 5 roots for this $f(z)$ inside the unit circle. Subtracting 5 from this shows the error in the integration was about 6.1e$-$41.

We could now try different (a, b, r) values to isolate the different complex roots within small circles. Try the circle centered at $0 + 1\,i$ with radius 1/4.

9, func, 0+i, enter, 0.25, enter, 6, $\mathrm{f_n}$ 1.000000000000 + .000000000000 i

meaning there is only one root within 1/4 unit of $0 + i$. Subtracting 1 from this shows the error in the integration was about 1.5e$-$57.

Try to find this root by using f3 to do 10 secant steps starting at $0 + i$ and $0.1 + 0.9\,i$.

9, func, 0+i, enter, 0.1+0.9i, enter, 10, enter, 3, $\mathrm{f_n}$

This gives $-.047319363666 + .936759058604\,i$, and then 1, $\mathrm{f_n}$ shows the function value has magnitude less than 1e$-$55, so this is the root within 1/4 unit of $0 + i$.

There is a possible pitfall if we do too many secant steps. If we had done 20 steps instead of 10 above, we would have seen “unknown + unknown i” in the display as our result.

When the accuracy of the secant approximation gets close to the full accuracy of the calculator, the function values can be so close to zero that z_3 rounds to the same value as z_2 in the secant update formula. That means we get $z_2 = z_1$ and $f(z_2) = f(z_1)$ for the next iteration, so the next secant calculation fails due to division by zero and Calc-50 gives “unknown” as the result.

It is possible to fix this problem by using the select function from screen 7 to check for $z_2 = z_1$ and return $z_3 = z_2$ if so, else return the z_3 from the secant formula. That version is left as an exercise for the reader.

It is also possible that $f(z_2) = f(z_1)$ when z_1 and z_2 are different, possibly not close together. Then the secant update will also give “unknown”, but that means we have had bad luck and landed on two points from which the secant method fails. Picking different starting points and trying again will usually fix it.

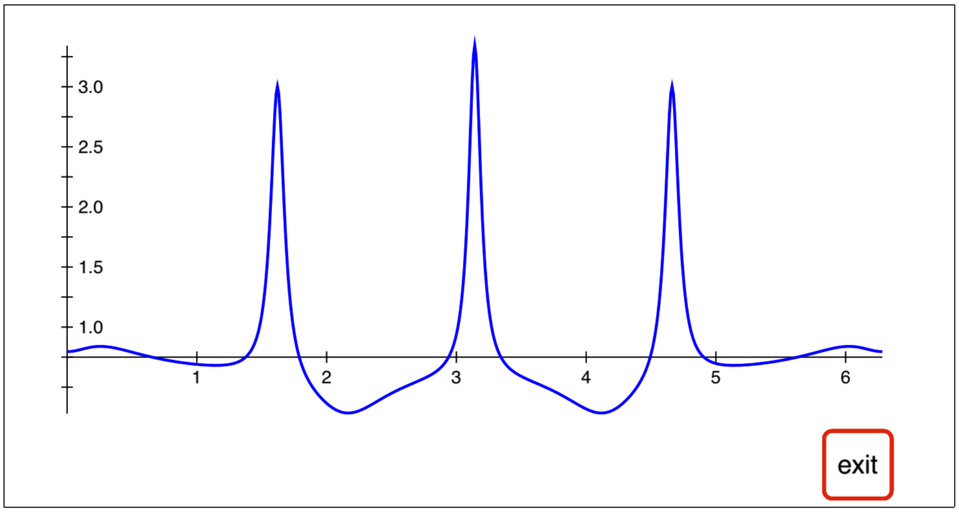
The second integration was more accurate as well as much faster than the first one.

The first result is accurate enough in this case, since we know that the mathematically exact value for the integration is an integer.

But if we are curious about why the first integral was more difficult for the Calc-50 integration function, we can look at the two graphs of f4.

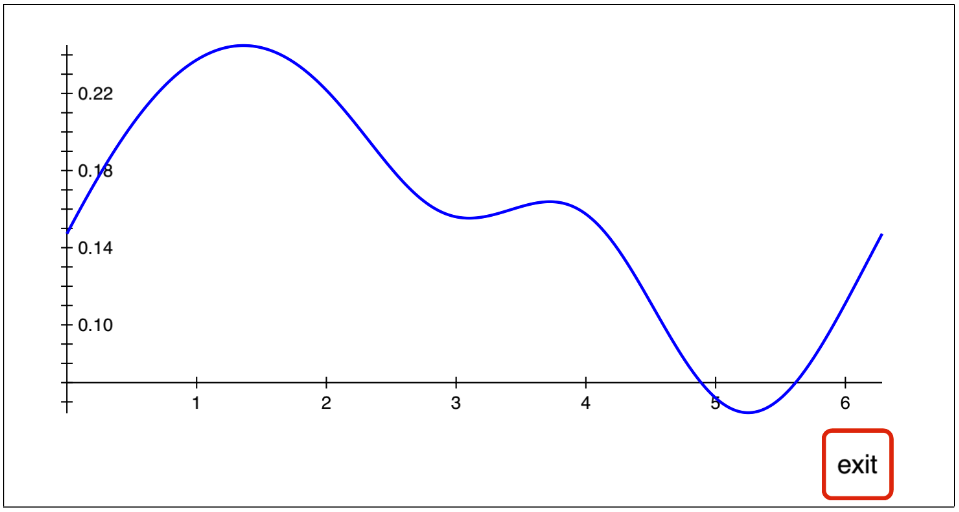
The first plot is the graph of f4 when integrating around the circle centered at $0 + 0\,i$ with radius 1.

> 9, func, 0, enter, 1, sto, 1, enter, 3, sto,
>
> 7, func, 0, π, 2, $*$, 4, plot



The second plot is the graph of f4 when integrating around the circle centered at $0 + 1\,i$ with radius 1/4.

> 9, func, 0+1i, enter, 1, sto, 0.25, enter, 3, sto,
>
> 7, func, 0, π, 2, $*$, 4, plot



The three tall narrow spikes in the first graph are similar to interior singularities. The "General Comments about Calc-50" help pages that are found at help, 0, help, 1, etc., mention that to increase speed and accuracy for numerical integration we could try breaking the integral up into pieces, making multiple calls to the integrate function with these spikes being endpoints.

Three calls to the maxf function, starting at 1.5, 3, 4.5 locate the top of the spikes. First re-set the center to $0 + 0\,i$ and radius to 1.

9, func, 0, enter, 1, sto, 1, enter, 3, sto

Then find the values of t where f4 has a local maximum. Store those values in registers 51, 52, 53.

6, func, 30, fix, 1.5, enter, 4, maxf, 51, sto	1.6209401313156835465847061275 97
3, enter, 4, maxf, 52, sto	3.141592653589793238462643383280
4.5, enter, 4, maxf, 53, sto	4.662245175863902930340580638962

Now break the original integral into 4 integrals and save the results in registers 61, 62, 63, 64.

0, enter, 51, rcl, 4, $\int_a^b$, 61, sto	1.381819564784612559867847566625
51, rcl, 52, rcl, 4, $\int_a^b$, 62, sto	1.118180435215387440132152433375
52, rcl, 53, rcl, 4, $\int_a^b$, 63, sto	1.118180435215387440132152433375
53, rcl, π, 2, $*$, 4, $\int_a^b$, 64, sto	1.381819564784612559867847566625

Each of these 4 integrals now takes only a second or two, and adding the four results gives more accuracy than before. Subtracting 5 from this sum gives −1.0e−56.

Finding max/min points or singularities on the graph and splitting an integral into pieces like this is a common method for trying to improve speed and/or accuracy of a difficult numerical integral.

Summary

The polynomial $f(z) = z^8 - 3z^5 + 2z - 1 = 0$ has 8 roots, consisting of 2 real roots and 6 complex roots. Since the coefficients of $f(z)$ are real, the complex roots are in the form of 3 conjugate pairs. Whenever we find a root $a + b\,i$, we get a second root $a - b\,i$ for free.

We found all 8 roots to an accuracy of over 50 digits. The complex roots show fewer digits on screen 9, but we can store them in registers and then move to a real screen to see more digits. Here are the 8 roots:

$-0.944550401394456448164395912277$	real root found using plot and solve
$1.384170071335253499009642716959$	real root found using plot and solve
$-0.799568751542 \pm 1.248623007432\,i$	2 complex roots found using secant
$0.627078280238 \pm 0.047441070579\,i$	2 complex roots found using secant
$-0.047319363666 \pm 0.936759058604\,i$	2 complex roots found using a line integral plus secant