MACM 316 – D100
February 4, 2019

Kai Sackville-Hii (301310336)
ksackvil@sfu.ca

Computing Assignment 3

Root Finding 2D Contour

The goal of this assignment is to compare the bisection and secant root finding methods when computing the zero contour of 2D functions.

Bisection and Secant Methods

Both bisection and secant methods require an initial interval. In this case the interval for the angle θ_n is $[\theta_{n-1} - \delta, \theta_{n-1} + \delta]$, where δ yields opposite signs at the endpoints of the zero contour function $HI(x,(\theta_n), y(\theta_n))$. δ alongside Δs, which controls the graphical quality of the contour, are the numerical parameters for this experiment. For our bisection method, let $\Delta s = 0.6$ and $\delta = 3\pi/4$. The computed contour produced by this bisection method has a nearly identical graph as the one produced by Matlab's *fzero*. For our secant method, let $\Delta s = 0.1$ and $\delta = \pi/2$. The computed contour produced by this secant method traces the zero contour with a better graphical quality than the previous bisection method. Decreasing Δs causes the search circles to be closer together thus producing a better graphical representation.

Obtaining A Smooth Computed Contour

For both the bisection and secant methods, as Δs decreases we expect the computed contour curve to become more visually smooth. At $\Delta s = 0.1$ the computed curve for both methods is adequate but could be improved. At $\Delta s = 0.07$ the computed curves become visually smooth. The smaller Δs becomes, the better the graph will look; however, at a certain value, the differences become visually unnoticeable. To obtain a smooth computed curve printed on a full letter-size page, let $\Delta s = 0.07$.

The parameter δ was experimented on using half the Δs value from above. For bisection, our method fails when δ is decreased to $\pi/13$ and below. The interval at which we search for the root is too small and does not contain a sign change; bisection will not work in this case. When δ is increased to π and above, our bisection method successfully finds roots. However, in this case the search interval is too large, and will cause this method to reverse itself. The average number of function evaluations remains constant at 31 regardless of the δ value. For the secant method, decreasing δ below our initial value has no effect on our computed curve, but increasing δ to π and above causes the secant method to fail. When δ is large, the two initial points for the secant method are too far apart, which causes a convergence failure. The average number of function evaluations decreases as δ decreases. Figure 1 shows the resulting computed curve using the secant method with $\Delta s = 0.035$ and $\delta = \pi/2$. Note that Nsteps was increased to 415, ensuring that the computed curve fully traces the zero contour.

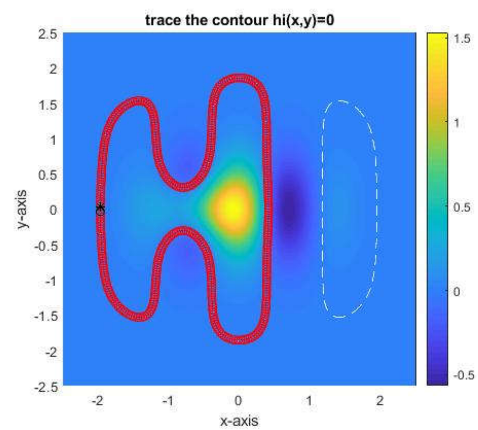

Figure 1: Computed Contour Using Secant Method

Bisection Verse Secant Method

Experimenting with bisection and secant methods yielded the following observations. Secant method is more efficient in that the number of function evaluations per point decreases as the two initial guesses become closer together, whereas the bisection method's efficiency remains constant. Bisection method is more robust in that it guarantees to find a root if the interval is valid. Secant method does not promise convergence.