Finding Zeros of Single-Variable, Real Functions

Gautam Wilkins
University of California, San Diego

General Problem

- Given a single-variable, real-valued function, *f,* we would like to find a real number, *x*, such that $f(x)=0$.

- If we have an interval, [*a, b*] where $f(a)f(b)<0$ and *f* is continuous on [*a, b*], then there is guaranteed to be a zero of *f* in [*a, b*].

- The interval [*a, b*] is called a straddle, and finding one can be part of the problem.

Zero-Finding Methods

- Bisection
- Newton's Method
- Secant
- Inverse Quadratic Interpolation (IQI)

- Hyperbolic, Bi-Confluent Hyperbolic
- Halley's Method

Bisection Method

- Requires a straddle, $[a, b]$.

- Compute $f((a+b)/2)$. If $f(a)f((a+b)/2)<0$ then new straddle is $[a, (a+b)/2]$, otherwise it's $[(a+b)/2, b]$. Stops when size of interval is smaller than some $\delta>0$.

- Guaranteed to converge, but only linearly.

Newton's Method

- Tracks a single iterate, x_n.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Converges super-linearly in general.

Secant Method

- Tracks two iterates, x_n and x_{n-1} .

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

- Converges super-linearly in general.

Inverse Quadratic Interpolation

- Tracks three iterates, x_n, x_{n-1} , x_{n-2} .

$$x_{n+1} = \frac{f_{n-1}f_n}{(f_{n-2}-f_{n-1})(f_{n-2}-f_n)}x_{n-2} + \frac{f_{n-2}f_n}{(f_{n-1}-f_{n-2})(f_{n-1}-f_{n-2})}x_{n-1} + \frac{f_{n-2}f_{n-1}}{(f_n-f_{n-2})(f_n-f_{n-1})}x_n$$

- Converges super-linearly in general.

What we want

- Given a function, *f*, and a straddle, construct a method that converges super-linearly in general, but gives the same guarantees as bisection.

- If we do not have a straddle, begin searching for a zero around a given starting point. If we find a straddle, then maintain it.

First Attempt: Dekker's Method

- Maintains straddle, $[a,b]$.
- Uses secant method whenever possible.
- Uses bisection method if secant method returns an iterate, x_{n+1}, that is not between x_n and $(a+b)/2$.
- Terminates when it finds a zero, or when $|b-a| < \delta$ for some $\delta > 0$.

Problem with Dekker's Method

- Although the method is guaranteed to converge, it does not place a reasonable bound on the complexity of the search.

- For poorly-behaved functions, the method can take a very large number of extremely small steps with the secant method.

Brent's Method (Zero-In)

- Uses IQI when possible, defaults to secant if it cannot.

- Let b_j be j^{th} iterate, computed with IQI. Forces a bisection unless:

  1) $|b_{j+1} - b_j| < 0.5|b_{j-1} - b_{j-2}|$, and

  2) $|b_{j+1} - b_j| > \delta$

Brent's Method

- Terminates when it finds a zero, or when $|b\text{-}a|<\delta$.

- The two inequalities ensure that in the worst-case, a bisection will be forced every $2\log_2((b\text{-}a)/\delta)$ steps.

- This places an $O(n^2)$ complexity bound on Brent's Method, where n is the number of steps that the Bisection Method would take.

Brent's Method: Proof of $O(n^2)$ Time

If the first condition is never violated, then at the j^{th} step, the second condition will be violated after at most k more steps, where:

$$\frac{|b_{j-1} - b_{j-2}|}{2^{k/2}} = \delta$$

$$k = 2\log_2\left(\frac{|b_{j-1} - b_{j-2}|}{\delta}\right)$$

Brent's Method: Proof of $O(n^2)$ Time

$$k = 2\log_2\left(\frac{|b_{j-1} - b_{j-2}|}{\delta}\right)$$

Thus, a bisection step is performed at least every *k* steps following an interpolation step.

So the interval size decreases by a factor of *2* every *k* steps, meaning that given an initial interval [*a, b*], the method will terminate in no more than *m* steps, where:

Brent's Method: Proof of $O(n^2)$ Time

$$k = 2\log_2\left(\frac{|b_{j-1} - b_{j-2}|}{\delta}\right)$$

$$\frac{|b-a|}{2^{m/k}} = \delta$$

$$m = k\log_2\left(\frac{|b-a|}{\delta}\right)$$

$$m = 2\log_2\left(\frac{|b-a|}{\delta}\right)^2$$

The running time of the bisection method is $O(\log_2(|b\text{-}a|/\delta))$, so Brent's Method is $O(n^2)$

Worst-Case Function

- We want to show that Brent's Method can take Θ (n^2) time. We do so by explicitly constructing a worse case function.

- Start with straddle [*a, b*], and tolerance, δ. We will force Brent's Method to take $k = \log_2(|b-a|/\delta)$ steps before it performs a bisection.

- In order to satisfy the first condition the distance between successive iterates must also decrease by less than a factor of 0.5 every two steps.

Worst-Case Iterates

- Choose a factor, $p > \sqrt{2}$. We will make the distance between two successive iterates decrease by a factor of $1/p$.

- The last step before a bisection is performed will decrease the size of the interval by δ, violating the second condition.

Worst-Case Iterates

- If the last step decreases the interval by *δ*, then the first step must decrease the interval by ($p^{(k-1)}$ *δ*).

- So we get the series:

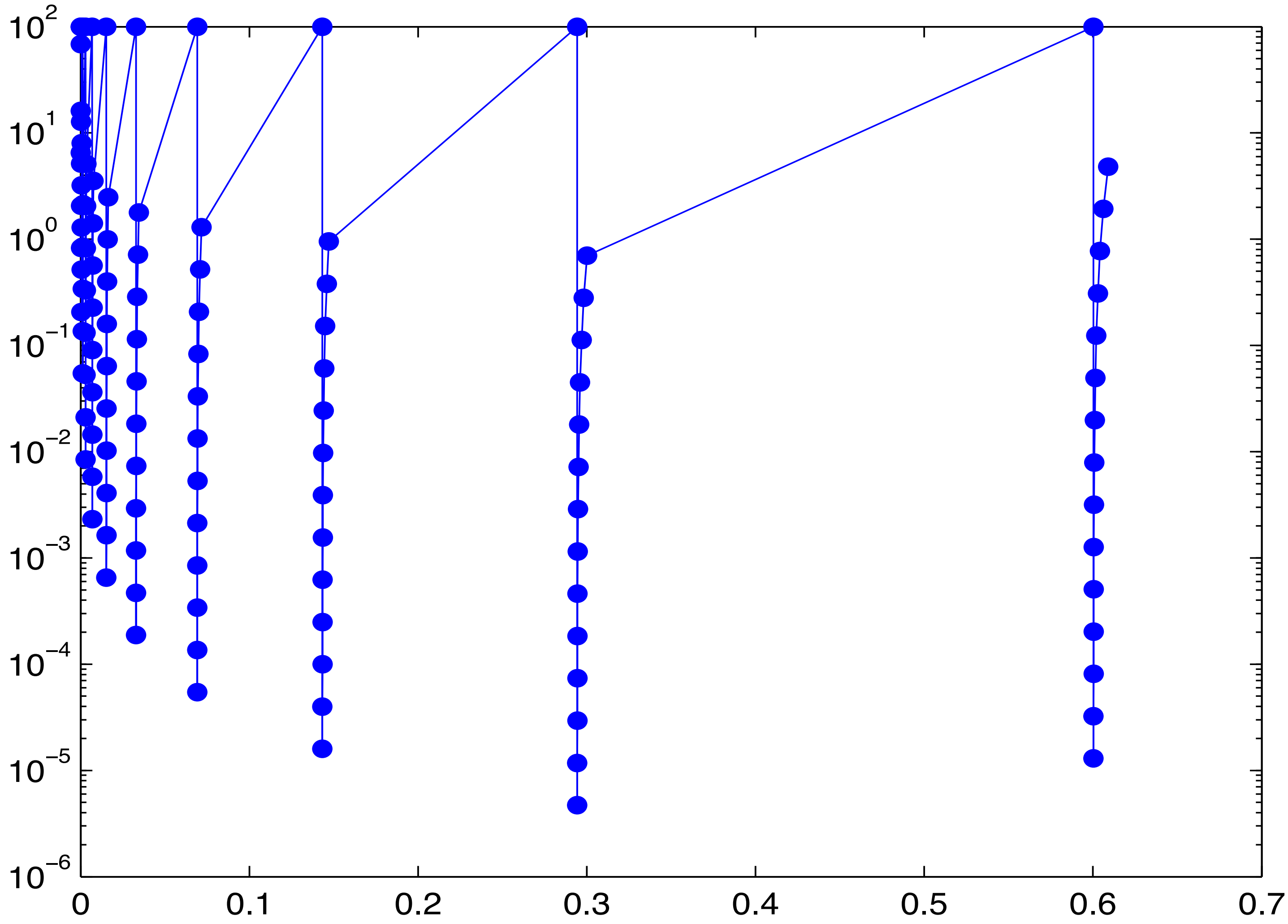
$$[b, b - p^{k-1}\delta, b - p^{k-1}\delta - p^{k-2}\delta, \ldots, b - \sum_{j=1}^{k} p^{k-j}\delta]$$

Worst-Case Iterates

- We will force Brent's Method to evaluate the function at this set of worst-case iterates, and then perform a bisection.

- This gives a new straddle, [*a, b'*] that is roughly half the length of the original interval.

- We now repeat the same process for [*a, b'*].



124 Brent's Method Iterates for a root near zero, tolerance = 1e-5

Worst-Case Function

- In conclusion, we first constructed a sequence, *X*, containing $\Theta(n^2)$ points.

- Then we constructed a function that caused Brent's Method to evaluate it at every point in *X*, proving that Brent's Method is $\Theta(n^2)$ in the worst-case.

Modified Zero-In

- Brent's Method may be modified to ensure $O(n)$ time instead of $O(n^2)$.

- Force a bisection if:

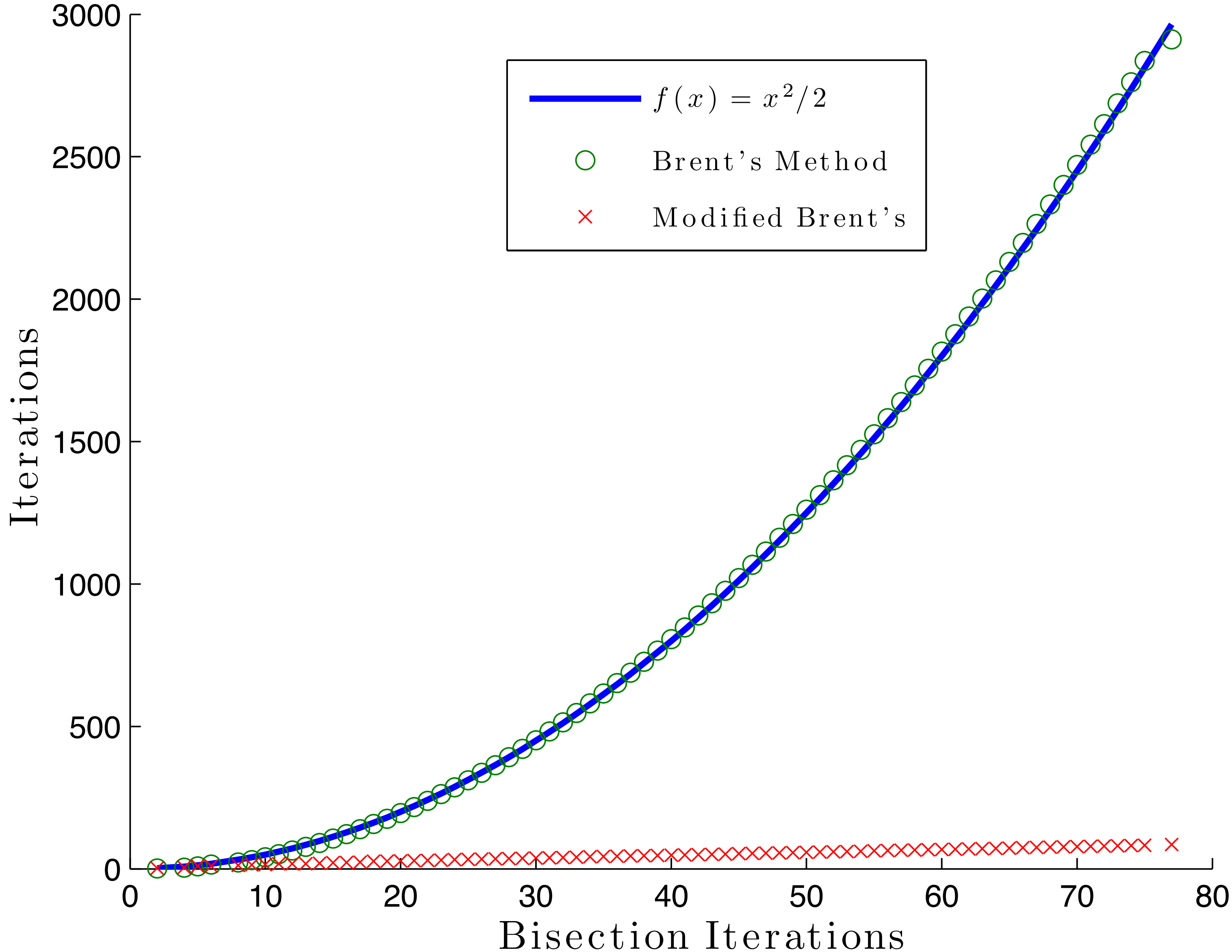
  1) If the size of the original interval is not reduced by a factor of 1/2 after five interpolation steps.

  2) If an interpolation step produces a point, x, such that $|f(x)|$ is not a factor of 1/2 smaller than the previous best point.

Modified Zero-In

- The first condition ensures that the complexity of the search is $O(n)$.

- The second condition addresses the issue of very flat functions, for which Brent's Method converges rather slowly.



$f(x) = x^2/2$
Brent's Method
Modified Brent's
Iterations
Bisection Iterations
0
500
1000
1500
2000
2500
3000
0
10
20
30
40
50
60
70
80

Comparison

- For the worst-case function shown earlier, when Brent's Method took 2914 iterations, Modified Zero-In took 85 iterations.

- This reduction in complexity, as far as we can tell, comes at virtually no cost to performance in general.

- We compared the performance against a number of functions from Burden and Faires' 2009 Numerical Analysis textbook.

Function	Interval	Brent's Method	Modified Zero-in	Bisection
$\sqrt{x}-\cos(x)$	[0.0,1.0]	7	8	52
$3(x+1)(x-5)(x-1)$	[-2.0,1.5]	2	2	53
$3(x+1)(x-5)(x-1)$	[-1.2,2.5]	8	8	54
$x^3-7x^2+14x-6$	[0.0,1.0]	8	8	51
$x^3-7x^2+14x-6$	[3.2,4.0]	13	14	47
$x^4-2x^3-4x^2+4x+4$	[-2.0,-1.0]	9	9	51
$x^4-2x^3-4x^2+4x+4$	[0.0,2.0]	8	8	52
$x-2^(-x)$	[0.0,1.0]	5	6	52
e^x-x^2+3x-2	[0.0,1.0]	7	7	52
$2x\cos(2x)-(x+1)^2$	[-3.0,-2.0]	8	8	50
$2x\cos(2x)-(x+1)^2$	[-1.0,0.0]	9	9	52
$3x-e^x$	[1.0,2.0]	9	10	50
$x+3\cos(x)-e^x$	[0.0,1.0]	7	6	52
$x^2-4x+4-\log(x)$	[1.0,2.0]	9	8	49
$x^2-4x+4-\log(x)$	[2.0,4.0]	10	10	51
$x+1-2\sin(\pi x)$	[0.0,0.5]	9	8	51
$x+1-2\sin(\pi x)$	[0.5,1.0]	10	10	51
$e^x-2-\cos(e^x-2)$	[-1.0,2.0]	11	11	53
$(x+2)(x+1)^2x(x-1)^3(x-2)$	[-0.5,2.4]	13	13	53
$(x+2)(x+1)^2x(x-1)^3(x-2)$	[-0.5,3.0]	15	15	52
$(x+2)(x+1)^2x(x-1)^3(x-2)$	[-3.0,-0.5]	13	13	52
$(x+2)(x+1)x(x-1)^3(x-2)$	[-1.5,1.8]	15	15	53
x^4-3x^2-3	[1.0,2.0]	7	8	51
x^3-x-1	[1.0,2.0]	9	10	51
$\pi+5\sin(x/2)-x$	[0.0,6.3]	7	7	52
$2^{-x}-x$	[0.3,1.0]	6	6	51
$(2-e^{-x}+x^2)/3-x$	[-5.0,5.0]	11	15	53
$5x^{-2}+2-x$	[1.0,5.0]	8	8	52
$\sqrt{\frac{e^x}{3}-x}$	[2.0,4.0]	9	9	51
$5^{-x}-x$	[-2.0,5.0]	8	9	54

Finding a Straddle

- Methods that guarantee convergence need to maintain an interval, [*a, b*], such that

  $f(a)f(b)<0$.

- Given a function, *f,* and an initial guess, x_0, we want to either find a straddle, or, if we have monotonic convergence, a zero.

Matlab's Approach

- Matlab has a function, **fzero**, that tries find zeros of functions.

- Given an initial guess, x_0 , it chooses $dx=x_0/50$ and constructs the interval $[x_0-dx, x_0+dx]$.

- If $[x_0-dx, x_0+dx]$ is a straddle, it returns it. Otherwise it sets $dx=\sqrt{2}*dx$ and tries again.

Problems with Matlab's Approach

- Can easily miss sign reversals since it takes increasingly large steps. Simple example: $f(x)=x^2 - 10^{-3}$, start with $x_0=1$.

- Discards the computed values of the function.

- In some cases, **fzero** takes longer to find a straddle than it does to find the zero.

Our Method

- If $f(x_0)<0$, then set $f(x) = -f(x)$.

- Choose a second number, x_1. Start performing iterations of Secant Method.

Termination Conditions

Terminate the search if:

1) We find a point, *x*, such that $f(x) <= 0$
2) Two successive iterates are the same
3) Five successive iterates fail to reduce function value by a factor of 0.5
4) After five successive iterates the step size has not decreased by a factor of 0.5

Edge Cases

- If $f(x_{n+1}) > f(x_n)$ then there is a local min between x_{n+1} and x_{n-1} .
- Start searching for this min using a modified Brent’s minimization method to ensure that it has $O(n)$ complexity.
- Stop search if we find a number, x, where $f(x) <= 0$, or we find a minimum.

Edge Cases

- If we find two successive iterates, x_{n+1} and x_n , where $f(x_{n+1})=f(x_n)$, perturb x_{n+1} .

- Fail if five successive points all have the same function value.

Edge Cases

- If complex, NaN, or Inf value is encountered, exclude that point, and do not allow search to continue in that direction.

- If two non-successive iterates have the same value then we entered a cycle. Use modified Brent's minimization method to find a local min.

Function	x_0	Our Method	**fzero**
$x^4 - 2x^3 - 4x^2 + 4x + 4$	-1.0	3	17
$x - 2^(- x)$	0.0	4	23
$e^x - x^2 + 3x - 2$	0.0	4	17
$2x\cos(2x) - (x+1)^2$	-3.0	10	17
$x\cos(x) - 2x^2 + 3x - 1$	0.2	3	21
$x - 2\sin(x)$	-1.0	14	23
$3x - e^x$	1.0	3	19
$x + 3\cos(x) - e^x$	0.0	3	25
$x^2 - 4x + 4 - \log(x)$	1.0	4	19
$x^2 - 4x + 4 - \log(x)$	2.0	3	17
$x + 1 - 2\sin(\pi x)$	0.0	4	15
$x + 1 - 2\sin(\pi x)$	0.5	3	19
$(x+2)(x+1)^2 x(x-1)^3(x-2)$	-0.5	3	25
$(x+2)(x+1)x(x-1)^3(x-2)$	-1.5	10	19
$x^3 - x - 1$	1.0	3	19
$\pi + 5\sin(x/2) - x$	0.0	5	33
$2^{-x} - x$	0.3	3	25
$(2 - e^{-x} + x^2)/3 - x$	-5.0	18	19
$5x^{-2} + 2 - x$	1.0	11	27
$\sqrt{e^x/3} - x$	2.0	3	21
$5^{-x} - x$	-2.0	18	25
$5(\sin(x) + \cos(x)) - x$	-2.0	4	27
$2\sin(\pi x) + x$	-2.0	5	13
$-x^3 - \cos(x)$	-3.0	18	23
$x^3 + 3x^2 - 1$	-3.0	4	7
$x - \cos(x)$	0.0	3	23
$x - 8 - 2\sin(x)$	0.0	3	25
$e^x + 2^{-x} + 2\cos(x) - 6$	1.0	3	23
$\log(x-1) + \cos(x-1)$	1.3	3	9
$2x\cos(2x) - (x-2)^2$	2.0	4	15

Conclusions

- Given a straddle, we have constructed a method that performs as well as Brent's Method, but only has $O(n)$ complexity.

- The method to bound the complexity may be applied to arbitrary zero-finding iterators as long as we have a straddle.

- Linear time to find local min, straddle, or zero given an initial point.

Future Work

- Modify Brent's Minimization Method to reduce complexity from $O(n^2)$ to $O(n)$.

- Continue to develop and stress test zero-finding method when we start with a single point instead of a straddle.

Acknowledgements

- This work was done jointly with Professor Ming Gu.

- We would also like to thank Professor William Kahan and Dr. Hanyou Chu for a number of enlightening discussions while conducting this research.