Project 1

Due on May 06 midnight

1 Introduction

The goal of this project is to apply Brent's method for solving root-finding problem $f(x) = 0$. We know that given an interval $[a, b]$ with $f(a)f(b) < 0$, the Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to a zero on $[a, b]$, say p. We also know that for Newton's method (also for its variant Secant method), only if the initial guess x_0 is "sufficiently close" to p, it converges. If it converges, and $f'(p) \neq 0$, it converges quadratically (super-linearly for Secant method)– much faster than Bisection method (usually linear convergence). However in practice it is very difficult to find a "good" initial guess. Brent's method attempts to solve this problem. In every iteration, it chooses between Secant method (or in some cases inverse quadratic interpolation) and Bisection method. The convergence is guaranteed given the same initial condition as Bisection method. While in the worst case, the performance of Brent's method is worse than Bisection method, it is usually as good as Secant method. See Figure 1 for an illustration of convergence rate.

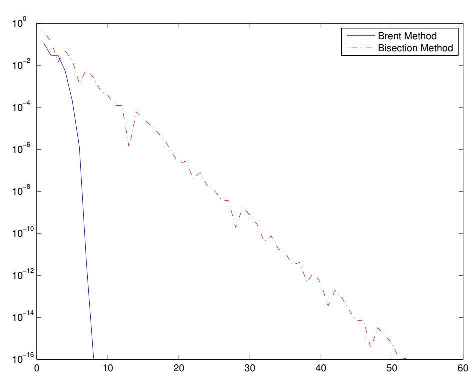

Figure 1: Comparison of convergence rate between Brent's method and Bisection method for finding a root of $f(x) = \cos(x^2) - x^3$ on $[0, 1]$. x axis is the number of iterations, and y axis is the quantity $|p_n - p|$ plotted in log-scale.

2 Description of Brent's Method

Like Bisection method, Brent's method initially requires an interval $[l, r]$ such that $f(l)f(r) < 0$. Brent's methods attempts to find a root p of f on $[l, r]$ in an iterative way. During the iterations, it keeps a record of three variables a, b, c: b is the current approximation of p, a is a point such that $f(a)f(b) < 0$, and c is the previous approximation of p. In each iteration, the algorithm chooses from the following three candidates (denoted by s) the next approximation of p:

1. (inverse quadratic interpolation)
$$\begin{aligned} s = & \frac{af(b)f(c)}{(f(a)-f(b))(f(a)-f(c))} + \frac{bf(a)f(c)}{(f(b)-f(a))(f(b)-f(c))} \\ & + \frac{cf(a)f(b)}{(f(c)-f(a))(f(c)-f(b))}; \end{aligned} \tag{1}$$
2. (secant method)
$$s = b - \frac{f(b)(b-a)}{f(b)-f(a)}; \tag{2}$$
3. (bisection method)
$$s = b + (b-a)/2. \tag{3}$$

The condition for choosing which candidate is technical. Roughly speaking, choosing inverse quadratic interpolation requires that $f(a), f(b), f(c)$ are all distinct, and both inverse quadratic interpolation and secant method require that s is between a and b but not too close to b. The specific algorithm is described in the table of Algorithm 1.

3 Application

We test Brent's method on two root finding problems:

1. $f(x) = (x+3)(x-1)^4$ on $[-4, 4/3]$;
2. $f(x) = \cos(x^2) - \frac{1}{2}x$ on $[0, 2]$.

In both these two cases, we choose $\epsilon = 10^{-15}$.

4 What to submit

Submit a zip file containing both MATLAB/Octave codes for Brent's method and a PDF report including the following

1. Documentation for MATLAB/Octave functions, including a description of what they do, what the variables in the input and output represent, and how to use the code (calling syntax).

Algorithm 1 Brent's method

input: function f, end points a, b such that $f(a) \cdot f(b) < 0$, some small threshold ϵ
output: solution p
if $|f(a)| < |f(b)|$ **then** swap a, b
end if
$c \leftarrow a$
flag $\leftarrow$ true ▷ flag denotes whether the previous iteration is bisection
$\delta \leftarrow \epsilon \max(1, |b|)$;
while $|f(b)| > \epsilon$ **or** $|b - a| > \delta$ **do**
    if $|f(a) - f(c)| > \epsilon$ **and** $|f(b) - f(c)| > \epsilon$
    then
        calculate s by inverse quadratic interpolation ▷ see Equation (1)
    else
        calculate s by secant method ▷ see Equation (2)
    end if
    if $(s - \frac{3a+b}{4})(s - b) > 0$ **or**
     (flag = true **and** $|s - b| \geq |b - c|/2$) **or**
     (flag = false **and** $|s - b| \geq |c - d|/2$) **or**
     (flag = true **and** $|b - c| < \delta$) **or**
     (flag = false **and** $|c - d| < \delta$)
    then
        calculate x by bisection method ▷ see Equation (3)
        flag = true
    else
        flag = false
    end if
    calculate $f(s)$
    $d \leftarrow c$
    $c \leftarrow b$
    if $f(a)f(s) < 0$ **then** $b \leftarrow s$
    else
        $a \leftarrow s$
    end if
    if $|f(a)| < |f(b)|$ **then** swap a, b
    end if
    $\delta \leftarrow \epsilon \max(1, |b|)$;
end while
$p \leftarrow b$
return p

2. Report the results for solving the two problems in "Applications" section. In particular, list the values of a, b, c in each iteration in a table. You may need to modify the algorithm a little to record a, b, c in each iteration.

3. Use MATLAB/Octave function `fzero` to compute a "ground truth" solution $\hat{p}$ for each problem. For each root-finding problem, use function `semilogy` to plot $|b - \hat{p}|$ versus the iteration number. You will observe a curve similar to the blue line in Fig. 1.

4. An implementation of Bisection method (`bisection`) is provided. The output of `bisection` is a sequence $\{p_n\}$ of approximate values for p. Solve each root-finding problem using `bisection` and plot by `semilogy` the graph $|p_n - \hat{p}|$ versus the iteration number n. You will see a curve similar to the red dots in Fig. 1.