NUMERICAL ANALYSIS I
Finding Roots

Surya Teja Gavva

BISECTION METHOD

Input:

1) A continuous function and two end points a and b such that $f(a)$ and $f(b)$ have opposite signs i.e., $f(a)f(b) < 0$.
2) The number of iterations or the accuracy needed.

Output: An approximation to the root of $f(x)$ inside $[a, b]$.

Set $i = 0, a_i = a$ and $b_i = b$.
Set $c_i = \dfrac{a_i + b_i}{2}$
If $f(c_i) = 0$, then c_i is a zero
Else check the signs of $f(a_i)f(c_i)$ and $f(c_i)f(b_i)$
If $f(a_i)f(c_i) < 0$, then $a_{i+1} = a_i, b_{i+1} = c_i$
If $f(c_i)f(b_i) < 0$, then $a_{i+1} = c_i, b_{i+1} = bi$
and repeat with interval $[a_{i+1}, b_{i+1}]$

Error: The error after n iterations $|x - x_n|$ is bounded by the length of the interval $|a_n - b_n|$
We have $|a_n - b_n| = \dfrac{|b - a|}{2^n}$. So, the number of iterations to get an accuracy of ε is n such that $\dfrac{|b - a|}{2^n} < \varepsilon$, that is $n > \log_2 \dfrac{|b - a|}{\varepsilon}$

Convergence: ALWAYS converges and the rate of convergence is linear.

$$|x - x_{n+1}| \sim \frac{1}{2}|x - x_n|$$

.

FALSE POSITION METHOD

Input:

1) A continuous function and two end points a and b such that $f(a)$ and $f(b)$ have opposite signs i.e., $f(a)f(b) < 0$
2) The number of iterations or the accuracy needed

Output: An approximation to the root of $f(x)$ inside $[a, b]$

Set $i = 0, a_i = a$ and $b_i = b$.
Set $c_i = \dfrac{a_i f(b_i) - b_i f(a_i)}{f(b_i) - f(a_i)}$
$Iff(c_i) = 0$, then c_i is a zero
Else check the signs of $f(a_i)f(c_i)$ and $f(c_i)f(b_i)$
If $f(a_i)f(c_i) < 0$, then $a_{i+1} = a_i, b_{i+1} = c_i$
If $f(c_i)f(b_i) < 0$, then $a_{i+1} = c_i, b_{i+1} = bi$
and repeat with interval $[a_{i+1}, b_{i+1}]$

Convergence: ALWAYS converges and the rate of convergence is superlinear (if the root is not a multiple root, if the function is smooth–slower convergence for multiple roots).

$$|x - x_{n+1}| \sim C|x - x_n|^{\frac{\sqrt{5}+1}{2}},\ C = |\frac{f"(x)}{2f'(x)}|^{\frac{\sqrt{5}-1}{2}}$$

.

.

NEWTON"S METHOD

Input:

1) A differentiable function and a point x_0 "close" enough to the root.
2) The number of iterations or the accuracy needed

Output: An approximation to the root of $f(x)$ near x_0

Convergence: Doesn't always converge. Converges if x_0 is close enough to the root. and the rate of convergence is quadratic (if the root is not a multiple root-slower convergence for multiple roots) if the function is smooth.

$$|x - x_{n+1}| \sim C|x - x_n|^2, \; C = |\frac{f"(x)}{2f'(x)}|$$

.

Algorithm:

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

SECANT METHOD

Input:

1) A continuous function and points x_0, x_1 "close" enough to the root.
2) The number of iterations or the accuracy needed

Output: An approximation to the root of $f(x)$ near x_0, x_1

Convergence: Doesn't always converge. Converges if x_0, x_1 are close enough to the root. and the rate of convergence is superlinear (if the root is not a multiple root-slower convergence for multiple roots) if the function is smooth.

$$|x - x_{n+1}| \sim C|x - x_n|^{\frac{\sqrt{5}+1}{2}},\ C = |\frac{f''(x)}{2f'(x)}|^{\frac{\sqrt{5}-1}{2}}$$

.

Algorithm:

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

STEFFENSEN"S METHOD

Input:

1) A differentiable function and points x_0 "close" enough to the root.
2) The number of iterations or the accuracy needed

Output: An approximation to the root of $f(x)$ near x_0, x_1

Convergence: Doesn't always converge. Converges if x_0 is close enough to the root. and the rate of convergence is quadratic (if the root is not a multiple root-slower convergence for multiple roots).

Algorithm:

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

FIXED POINT METHOD

Input: A differentiable function g and a point x_0

Output: An approximation to the fixed point of $g(x)$

Convergence: Doesn't always converge.
Converges if
a) g maps an interval $[a, b]$ to itself.
b) g is "contracting" that is $|g'(x)| < k, k < 1$

The convergence is linear.

$$|x - x_{n+1}| < k|x - x_n|,$$

Algorithm:

$$x_{n+1} = g(x_n)$$

AITKEN'S Δ^2 METHOD

Input: A linearly convergent sequence p_n with $\lim_{n\to\infty} \frac{p-p_n}{p-p_{n+1}} < 1$

Output: A sequence $\hat{p}_n$ which converges faster than p_n to p

Definition The forward difference

$$\Delta p_n = p_{n+1} - p_n$$

$$\Delta^2 p_n = \Delta(\Delta p_n) = p_{n+2} - 2p_{n+1} + p_n$$

Algorithm:

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}$$

$$\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$$