Muller's method

From Wikipedia, the free encyclopedia

Muller's method is a root-finding algorithm, a numerical method for solving equations of the form $f(x) = 0$. It was first presented by David E. Muller in 1956.

Muller's method is based on the secant method, which constructs at every iteration a line through two points on the graph of *f*. Instead, Muller's method uses three points, constructs the parabola through these three points, and takes the intersection of the *x*-axis with the parabola to be the next approximation.

Contents [hide]

1 Recurrence relation
2 Speed of convergence
3 Generalizations and related methods
4 References
5 External links

Recurrence relation [edit]

Muller's method is a recursive method which generates an approximation of the root ξ of *f* at each iteration. Starting with the three initial values x_0, x_{-1} and x_{-2}, the first iteration calculates the first approximation x_1, the second iteration calculates the second approximation x_2, the third iteration calculates the third approximation x_3, etc. Hence the k^{th} iteration generates approximation x_k. Each iteration takes as input the last three generated approximations and the value of *f* at these approximations. Hence the k^{th} iteration takes as input the values x_{k-1}, x_{k-2} and x_{k-3} and the function values $f(x_{k-1})$, $f(x_{k-2})$ and $f(x_{k-3})$. The approximation x_k is calculated as follows.

A parabola $y_k(x)$ is constructed which goes through the three points $(x_{k-1}, f(x_{k-1}))$, $(x_{k-2}, f(x_{k-2}))$ and $(x_{k-3}, f(x_{k-3}))$. When written in the Newton form, $y_k(x)$ is

$$y_k(x) = f(x_{k-1}) + (x - x_{k-1}) f[x_{k-1}, x_{k-2}] + (x - x_{k-1})(x - x_{k-2}) f[x_{k-1}, x_{k-2}, x_{k-3}],$$

where $f[x_{k-1}, x_{k-2}]$ and $f[x_{k-1}, x_{k-2}, x_{k-3}]$ denote divided differences. This can be rewritten as

$$y_k(x) = f(x_{k-1}) + w(x - x_{k-1}) + f[x_{k-1}, x_{k-2}, x_{k-3}]\,(x - x_{k-1})^2$$

where

$$w = f[x_{k-1}, x_{k-2}] + f[x_{k-1}, x_{k-3}] - f[x_{k-2}, x_{k-3}].$$

The next iterate x_k is now given as the solution closest to x_{k-1} of the quadratic equation $y_k(x) = 0$. This yields the recurrence relation

$$x_k = x_{k-1} - \frac{2f(x_{k-1})}{w \pm \sqrt{w^2 - 4f(x_{k-1})f[x_{k-1}, x_{k-2}, x_{k-3}]}}.$$

In this formula, the sign should be chosen such that the denominator is as large as possible in magnitude. We do not use the standard formula for solving quadratic equations because that may lead to loss of significance.

Note that x_k can be complex, even if the previous iterates were all real. This is in contrast with other root-finding algorithms like the secant method, Sidi's generalized secant method or Newton's method, whose iterates will remain real if one starts with real numbers. Having complex iterates can be an advantage (if one is looking for complex roots) or a disadvantage (if it is known that all roots are real), depending on the problem.

Speed of convergence [edit]

The order of convergence of Muller's method is approximately 1.84. This can be compared with 1.62 for the secant method and 2 for Newton's method. So, the secant method makes less progress per iteration than Muller's method and Newton's method makes more progress.

More precisely, if ξ denotes a single root of *f* (so $f(\xi) = 0$ and $f'(\xi) \neq 0$), *f* is three times continuously differentiable, and the initial guesses x_0, x_1, and x_2 are taken sufficiently close to ξ, then the iterates satisfy

$$\lim_{k\to\infty} \frac{|x_k-\xi|}{|x_{k-1}-\xi|^\mu} = \left|\frac{f'''(\xi)}{6f'(\xi)}\right|^{(\mu-1)/2},$$

where μ ≈ 1.84 is the positive solution of $x^3 - x^2 - x - 1 = 0$.

Generalizations and related methods [edit]

Muller's method fits a parabola, i.e. a second-order polynomial, to the last three obtained points $f(x_{k-1})$, $f(x_{k-2})$ and $f(x_{k-3})$ in each iteration. One can generalize this and fit a polynomial $p_{k,m}(x)$ of degree m to the last m+1 points in the k^{th} iteration. Our parabola y_k is written as $p_{k,2}$ in this notation. The degree m must be 1 or larger. The next approximation x_k is now one of the roots of the $p_{k,m}$, i.e. one of the solutions of $p_{k,m}(x)$=0. Taking m=1 we obtain the secant method whereas m=2 gives Muller's method.

Muller calculated that the sequence $\{x_k\}$ generated this way converges to the root ξ with an order μ_m where μ_m is the positive solution of $x^{m+1} - x^m - x^{m-1} - \ldots - x - 1 = 0$.

The method is much more difficult though for m>2 than it is for m=1 or m=2 because it is much harder to determine the roots of a polynomial of degree 3 or higher. Another problem is that there seems no prescription of which of the roots of $p_{k,m}$ to pick as the next approximation x_k for m>2.

These difficulties are overcome by Sidi's generalized secant method which also employs the polynomial $p_{k,m}$. Instead of trying to solve $p_{k,m}(x)$=0, the next approximation x_k is calculated with the aid of the derivative of $p_{k,m}$ at x_{k-1} in this method.

References [edit]

- Muller, David E., "A Method for Solving Algebraic Equations Using an Automatic Computer," *Mathematical Tables and Other Aids to Computation*, 10 (1956), 208-215. JSTOR 2001916
- Atkinson, Kendall E. (1989). *An Introduction to Numerical Analysis*, 2nd edition, Section 2.4. John Wiley & Sons, New York. ISBN 0-471-50023-2.
- Burden, R. L. and Faires, J. D. *Numerical Analysis*, 4th edition, pages 77ff.
- Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 9.5.2. Muller's Method". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.

External links [edit]

- Module for Muller's Method by John H. Mathews

Categories: Root-finding algorithms