

## Re: Secant method vs Newton's method

---

*Source:* <http://sci.tech--archive.net/Archive/sci.math.num--analysis/2006-09/msg00293.html>

---

- *From:* Jentje Goslinga <[goslinga@xxxxxxxxxxx](mailto:goslinga@xxxxxxxxxxx)>
  - *Date:* Sat, 23 Sep 2006 17:27:46 GMT
- 

Schizoid Man wrote:

Hi,

Under what circumstances would I prefer the secant method to Newton's method?

Newton's method converges much faster and the number of function calls seem to be the same (if we assume the derivative is a function call).

The natural generalization of the Secant Method to higher order is obtained by using Normalized Inverse Hermite interpolation (NIH).

It is found that the Secant Method is obtained using zero order NIH and higher order methods with increasingly higher order error terms are obtained by using First or Second Order NIH. The resulting higher order methods have the desirable bracketing property.

The simple innovation which I am using in NIH, and which is indicated by the word Normalized, is that I am using for interpolation the two point Hermite Interpolating Polynomial interpolating the function and some of its derivatives on a normalized interval  $[-1/2, 1/2]$  by simply mapping the bracketing interval to the normalized interval each iteration. This allows use of the same interpolation formula every step of the iterative search for the root by evaluating the root of the Hermite polynomial interpolating the inverse function and its first or even second derivatives on two points within the normalized interval.

I have described this on my website [www.numerical--algorithms.com](http://www.numerical--algorithms.com) under the section "Root Finding Using Inverse Normalized Hermite Interpolation" and present an example where I compute the roots of the Legendre polynomials. I have found NIH to be somewhat better than Brent's algorithm (which is not bad) in the cases which I have tested.

[The other example where I compute the roots of the Secular Function for the Symmetric Eigenvalue Problem should be disregarded since this function is of a non polynomial nature for which the existing methods have been found to be more reliable.]

I have C code for the algorithm and can send it to whomever is interested.

Jen

.