Re: Finding the inverf

Source: http://sci.tech−archive.net/Archive/sci.math/2007−12/msg01716.html

- *From*: David W. Cantrell <DWCantrell@xxxxxxxxxxx>
- *Date*: 09 Dec 2007 01:50:43 GMT

Ray Koopman <koopman@xxxxxx> wrote:

> On Dec 2, 9:43 pm, morg...@xxxxxxxxx wrote:
>
>> Could someone tell me how to calculate inverf (x) if I am given
>> the value of erf (x).
>
> Here's a relatively simple approximation for inverf[x]:
>
> With[{t = −2*Log[1−|x|]}, Sign[x]*Sqrt[(t −
> Log[1 + t + (.01167845*t+.1066561)*t^2/
> ((.02118035*t+.3710243)*t+1)])/2]]
>
> The maximum absolute error is about 3.6*10^−5
> for all x in (−1,1) in 64−bit IEEE format.

Thanks for that approximation. I have some comments and what some might consider to be an improvement.

First, I believe your statement about the error to be correct. (N.B. I've forgotten much about floating−point arithmetic, so anything I say related to that here should be "taken with a grain of salt".) The reason that your statement is correct is that numbers only very slightly less than 1 are not representable in 64−bit IEEE format. But forgetting floating−point and just considering the interval (−1, 1) of real numbers, the maximum error of your approximation is about 1.2*10^−4, occurring when |x| is about 1 − 10^−79 .

Sometimes we prefer to consider relative, rather than absolute, error.
That's what I would normally choose to do for erf. Your approximation has relative error exceeding 8*10^−4 at x = 0.

With the constraint that relative error = 0 at x = 0 and then minimizing |relative error| on the real interval (−1, 1), for an approximation of erf(x) in the form

With[{t = −2*Log[1−|x|]},
Sign[x]*Sqrt[(t − Log[1 + t + (a*t + b)*t^2/((c*t + d)*t+1)])/2]]

we find that b = (4 − pi)/8, implied by the constraint, and
a = 0.01857224, c = 0.03453839 and d = 0.44739441 .

This gives |relative error| < 3.7*10^−5 . (If we were to drop the constraint, then worst relative error could be reduced a bit further. However, if for nothing but aesthetics, I prefer to keep the constraint.)

BTW, with my coefficients above, if we were to approximate the inverse of the complementary error function, inverfc(x), on (0, 2), my bound on |relative error| would remain unchanged, of course. But with your original coefficients, if we were to approximate inverfc(x) on (0, 2) in 64−bit IEEE format, your error bound would increase (since floating−point allows representation of numbers much closer to 0 than to 1).

Best regards,
David W. Cantrell
.