Close to Normal

It seems I have always used probabilities from the normal cumulative density function, early on merely as “areas under the curve” taken from tables at the end of statistics books. While in Calculus II, about 30 years ago, I tried every method taught to find an analytical solution to the integral of the normal probability density (npd) function, $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{X} e^{-\frac{t^2}{2}}\,dt$, which gives cumulative density. This was important to me since I had been effectively using the results of something I was now studying, results that I should be able to derive. But I failed at every attempt. Then Dr. Hardy introduced Taylor series for function and integral approximation. I wasted no time in applying this to my problem (actually, I used a MacLaurin series, a Taylor series centered at 0) and found a good compromise – although the results were approximate, their accuracy was controllable and they were verifiable, and I knew the theory behind their origin. You might ask, “Why go to such lengths?” Well, back then, we didn’t have SAS and R running on laptops as we sipped latte at Starbucks. We had to write much more of the code we used (yes, in the snow, uphill, both ways, in the freezing heat). Even today, you might ask, “Why? We have cdf() in SAS,” and, yes when in SAS we should use that. But, also, why not check their work? I have used my npd series integration algorithm in testing new systems and learning new programming environments. I recall searching for simpler or alternate solutions to npd areas, with no success. Recently, however, I returned to this old friend and within minutes found a simpler method using integration by parts. I enjoy finding alternate solutions to problems and find that, however different, each reflects a central concept or truth so that, in the end, all agree. Understanding the similarities and differences of alternate solutions, their strengths or weaknesses, helps to understand more fully the central principles being established and aids in solving real world problems creatively and effectively. Further, I believe that having two versions of a problem (an isomorphism) is a fundamental requirement of establishing mathematical truth – two statements share a common, hidden third, the one to be established, and through mathematical and logic operations the two statements are reduced to a single one which reveals and is identical to the one sought, the new truth, the discovery. Incidentally, the MacLaurin series method derives from integrating the series for e^t:

$$e^t = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots \;\Rightarrow\; e^{-t^2/2} = 1 - \frac{t^2}{2} + \frac{t^4}{2^2(2!)} - \frac{t^6}{2^3(3!)} + \cdots$$

$$\Rightarrow\; \frac{1}{\sqrt{2\pi}}\int_0^X e^{-t^2/2}dt \;=\; \frac{1}{\sqrt{2\pi}}\left(t - \frac{t^3}{3(2)} + \frac{t^5}{5(2^2)(2!)} - \frac{t^7}{7(2^3)(3!)} + \cdots\right)\Bigg|_0^X \;=\; \frac{1}{\sqrt{2\pi}}\sum_{i=0}^{\infty}\frac{X^{2i+1}}{(2i+1)(2^i)(i!)}$$

Integration by parts requires fewer steps to reveal the series and, because $e^{-X^2/2}$ factors out, the series is simpler:

$$\int_0^X e^{-\frac{t^2}{2}}dt \;=\; te^{-\frac{t^2}{2}}\Bigg|_0^X - \int_0^X -\frac{t^2}{2}e^{-\frac{t^2}{2}}dt \;=\; Xe^{-\frac{X^2}{2}} + \frac{t^3}{3}e^{-\frac{t^2}{2}}\Bigg|_0^X - \int_0^X -\frac{t^4}{3}e^{-\frac{t^2}{2}}dt$$

$$=\; Xe^{-\frac{X^2}{2}} + \frac{X^3}{3}e^{-\frac{X^2}{2}} + \frac{X^5}{5(3)}e^{-\frac{X^2}{2}} + \frac{X^7}{7(5)(3)}e^{-\frac{X^2}{2}} + \cdots \;=\; e^{-\frac{X^2}{2}}\sum_{i=0}^{\infty}\frac{X^{2i+1}}{i(i-2)\ldots1}$$

Dividing by $\sqrt{2\pi}$ gives the final integral. Note the identical numerator in corresponding terms of both series, while the denominators in that of the MacLaurin increase, term by term, much more than do those of integration by parts ($i!$ alone in the MacLaurin denominator is greater than the entire *parts* denominator, being $i!$ divided by all odd numbers < i). This causes the MacLaurin series to converge in fewer terms, giving it an advantage in efficiency.

Of interest is the following SQL implementation of cumulative normal probabilities and their inverse. SQL performs very efficient data retrieval and aggregation and offers fundamental math operators, such as trig, exponential, and so forth, but typically does not provide probability functions, making the following very useful. Note that just a few commands implement the actual algorithm (they are highlighted).

```
-- Randomly generate a normally distributed service time (between avg+-3.5s to avoid
-- extremes)
-- Adjust to right half of distribution and compress to [.001,.499] to eliminate extreme
-- values
-- Retain indicator of whether random p-value above or below that for mean (.5 for
-- normal cdf)
select @p=rand()-.5
while(@p not between -.499 and .499)
  select @p=rand()-.5
if(@p<0)
  select @p=-@p, @abovemean=0
else
  select @abovemean=1
-- Adjust random cumulative density (CD) by npd constant so that following MacLaurin series
-- is simply for the integral of e**-(x**2/2)
-- Use initial quantile estimate of .05 (near where most normally distributed values
-- should be on unit npd)
-- Note that initial CD estimate (Fe) value (as with all following) has been adjusted
-- for the npd constant
select @p=@p*sqrt(8*atan(1)), @z=.05, @Fe=0.04997917
-- Use Newton's method of roots to locate z corresponding to normal cdf within .0001 of
-- generated p
while(abs(@Fe-@p)>.0001)
  begin
    -- Calculate CD (F) at current quantile estimate (z)
    select @Fe=@z, @Fterm=@z, @k=0, @j=1
    while(@k=0 or abs(@Fterm)>.0001)
      begin
        -- This may appear cryptic, but it implements the Mac series by multiplying the
        -- current term by the proper factor to generate the next term as
        -- z**(2k+1)/[(2k+1)*(2**k)*k!] , as required by Mac, while improving efficiency
        -- and avoiding large numerators and denominators (consider z**51/50!)
        select @Fterm=@Fterm*@z*@z*(@k+@k+1)/(@k+@k+2)/(@k+@k+3), @j=-@j
        select @Fe=@Fe+@j*@Fterm, @k=@k+1
      end
    -- Estimate new z by subtracting from current estimate (difference in
    -- F1 and F0)/(cdf derivative), this Newton's method of roots
    -- Note that the derivative of a cdf is its pdf
    if(abs(@Fe-@p)>.0001) select @z=@z-(@Fe-@p)*exp(@z*@z/2)
  end
```