Computation of $\log(\Phi(z))$ For Large Negative z

Andrew Schein*
Senior Algorithms Engineer
Adobe Systems Incorporated
aschein@adobe.com

April 4, 2012

We first detour into the erfc function:

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \tag{1}$$

for which we have convenient assymptotic expansion[1]:

$$\mathrm{erfc}(x) = \frac{e^{-x^2}}{x\sqrt{\pi}}[1 + \sum_{n=1}^{N-1} (-1)^n \frac{(2n-1)!!}{(2x^2)^n} + O(x^{-2N+1}e^{-x^2})]. \tag{2}$$

Now we exploit the identity:

$$\Phi(z) = \frac{1}{2}\mathrm{erfc}(-\frac{z}{\sqrt{2}}) \tag{3}$$

and by substituting the assymptotic expansion for erfc:

$$\Phi(z) = \frac{e^{-z^2/2}}{-z\sqrt{2\pi}}[1 + \sum_{n=1}^{N-1} (-1)^n \frac{(2n-1)!!}{(z^2)^n} + O(z^{-2N+1}e^{-z^2/2})] \tag{4}$$

$$= \frac{e^{-z^2/2}}{-z\sqrt{2\pi}}[1 + \sum_{n=1}^{N-1} (-1)^n \frac{(2n-1)!!}{(z^2)^n}] + O(z^{-2N+2}) \tag{5}$$

$$= [\mathrm{LHS}] * [\mathrm{RHS}] + \mathrm{error}. \tag{6}$$

In these equations, we drop the big-O portions that do not depend on N, since we need the approximation to work even more moderate z such as -10. We can control accuracy by bounding z and using a sufficiently large N to ensure a residual summation that falls below the machine-specific epsilon (DBL_EPSILON in C). Finally, we use the symbols LHS (left hand side) and RHS (right hand side) to highlight the multiplication which we will transform into a log of sums in what follows.

Now we can finally focus on the function of interest: $\log(\Phi(z))$. Taking the log we have:

$$\log(\Phi(z)) \simeq \left(-\frac{z * z}{2} - \log(-z) - \frac{\log(2\pi)}{2}\right) + \log[1 + \sum_{n=1}^{N-1} (-1)^n \frac{(2n-1)!!}{(z^2)^n}]. \tag{7}$$

*Special thanks to Charles Blake for pointing me towards some useful links.

[1] http://en.wikipedia.org/wiki/Error_function

1 Validation of the Method: Modest values of z

In order to evaluate the method, we first explore z in the range $-120\ldots0$ to determine if it produces similar values to R's implementation. Below is a small test program utilizing our patch.

```
import math, numpy as np
from scipy.stats import norm
import rpy2.robjects as robjects
__r_pnorm = robjects.r['pnorm']
norm = norm()

def test(x) :
    py = norm.logcdf(x)
    r  = __r_pnorm(x,log_p=True)[0]
    print "%10.2f %10.2f %10.2f" %(x, py, r)

print "%10s %10s %10s"%("z","our_method","R_method")
for i in range(0,120,4) :
    test(-i)
```

The output follows.

```
aschein@aschein-dev:logcdf$ python temp.py
         z our_method   R_method
      0.00      -0.69      -0.69
     -4.00     -10.36     -10.36
     -8.00     -35.01     -35.01
    -12.00     -75.40     -75.41
    -16.00    -131.69    -131.70
    -20.00    -203.91    -203.92
    -24.00    -292.10    -292.10
    -28.00    -396.25    -396.25
    -32.00    -516.38    -516.39
    -36.00    -652.50    -652.50
    -40.00    -804.61    -804.61
    -44.00    -972.70    -972.70
    -48.00   -1156.79   -1156.79
    -52.00   -1356.87   -1356.87
    -56.00   -1572.94   -1572.94
    -60.00   -1805.01   -1805.01
    -64.00   -2053.08   -2053.08
    -68.00   -2317.14   -2317.14
    -72.00   -2597.20   -2597.20
    -76.00   -2893.25   -2893.25
    -80.00   -3205.30   -3205.30
    -84.00   -3533.35   -3533.35
    -88.00   -3877.40   -3877.40
    -92.00   -4237.44   -4237.44
    -96.00   -4613.48   -4613.48
   -100.00   -5005.52   -5005.52
   -104.00   -5413.56   -5413.56
   -108.00   -5837.60   -5837.60
   -112.00   -6277.64   -6277.64
```

```
   -116.00   -6733.67   -6733.67
aschein@aschein-dev:logcdf$
```

The numbers look identical.

2 Validation of Method: Range of Acceptable Values

A separate test explored the maximum $-z$ for which we can produce a result from this function:

```
aschein@aschein-dev:logcdf$ python range.py
        z our_method  R_method
-1.00e+00 -1.84e+00 -1.84e+00
-2.00e+00 -3.78e+00 -3.78e+00
-8.00e+00 -3.50e+01 -3.50e+01
-1.28e+02 -8.20e+03 -8.20e+03
-3.28e+04 -5.37e+08 -5.37e+08
-2.15e+09 -2.31e+18 -2.31e+18
-9.22e+18 -4.25e+37 -4.25e+37
-1.70e+38 -1.45e+76 -1.45e+76
-5.79e+76 -1.68e+153 -1.68e+153
-6.70e+153 -2.25e+307 -2.25e+307
Warning: overflow encountered in log_ndtr
-8.99e+307  -inf  -inf
 -inf  -inf  -inf
```

We can conclude that both our patch and R share similar ranges for acceptable inputs.