

Math library

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-09/msg01245.html>

- *From:* "Jon Slaughter" <Jon_Slaughter@xxxxxxxxxxxxx>
 - *Date:* Wed, 12 Sep 2007 04:00:21 GMT
-

I wrote a routine to replace Math's Exp method but it turns out to be almost 2x slower ;/ (well, actually its about 1.5x in release)

I'm essentially using a lookup table and interpolate between integer values (but even just looking up its still as slow).

```
public static double Exp(double x)
{
    int n = (int)Math.Floor(x);
    //if ((n > 2351) || (n < -2351)) return Math.Sign(x) *
    double.PositiveInfinity;

    double f = x - n;
    double ff = 1;
    double e = 1;

    if (f != 0)
    for (int i = 1; i < 10; i++)
    {
        ff = ff * f;
        e += invnfac[i] * ff;
    }

    return ExpIp[n + 2351]*e;
}
```

What I did was precompute n! and ExpIp is exp(k) for integer k. exp(+/-2351) is the maximum that a double can hold so no reason to go beyond that.

If I comment out the loop I get almost there speed but still about 1% slower.

How the hell are they computing so fast? Normally these Math library routines are very slow but either my code/method sucks or Math .NET is fairly optimized?

Anyone know whats going on here?

Math library

Thanks,
Jon

.