Subject: Re: Modified Bessel functions
Posted by meron on Thu, 04 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

In article <340EC565.2FB7@ii.iac.es>, Stephen Dunne <sdunne@ii.iac.es> writes:
>
> Does anybody know of a straight forward way to calculate the
> modified Bessel functions K0 and K1 in such a way that the expression
> may be part of a function to be integrated using QROMB?
>
There is this BESSELK routine, that I wrote. Let me know if you're
interested.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: Modified Bessel functions
Posted by J.D. Smith on Thu, 04 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Stephen Dunne wrote:
>
> Does anybody know of a straight forward way to calculate the
> modified Bessel functions K0 and K1 in such a way that the expression
> may be part of a function to be integrated using QROMB?
>
> Time's a wastin'..
> Steve.

Here's some Numerical Recipes functions you can convert easily to
IDL... Note that the K calculation relies on the other modified
B-function I, for x<=2.

```
float bessi0(float x)
{
        float ax,ans;
        double y;

        if ((ax=fabs(x)) < 3.75) {
                y=x/3.75;
                y*=y;
                ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067492
                        +y*(0.2659732+y*(0.360768e-1+y*0.45813e-2)))));
        } else {
                y=3.75/ax;
                ans=(exp(ax)/sqrt(ax))*(0.39894228+y*(0.1328592e-1
```

```
                +y*(0.225319e-2+y*(-0.157565e-2+y*(0.916281e-2
              +y*(-0.2057706e-1+y*(0.2635537e-1+y*(-0.1647633e-1
                +y*0.392377e-2))))))));
        }
        return ans;
}

float bessk0(float x)
{
        float bessi0(float x);
        double y,ans;

        if (x <= 2.0) {
                y=x*x/4.0;
                ans=(-log(x/2.0)*bessi0(x))+(-0.57721566+y*(0.42278420
                        +y*(0.23069756+y*(0.3488590e-1+y*(0.262698e-2
                        +y*(0.10750e-3+y*0.74e-5))))));
        } else {
                y=2.0/x;
                ans=(exp(-x)/sqrt(x))*(1.25331414+y*(-0.7832358e-1
                        +y*(0.2189568e-1+y*(-0.1062446e-1+y*(0.587872e-2
                        +y*(-0.251540e-2+y*0.53208e-3))))));
        }
        return ans;
}

float bessi1(float x)
{
        float ax,ans;
        double y;

        if ((ax=fabs(x)) < 3.75) {
                y=x/3.75;
                y*=y;
                ans=ax*(0.5+y*(0.87890594+y*(0.51498869+y*(0.15084934

+y*(0.2658733e-1+y*(0.301532e-2+y*0.32411e-3))))));
        } else {
                y=3.75/ax;
                ans=0.2282967e-1+y*(-0.2895312e-1+y*(0.1787654e-1
                        -y*0.420059e-2));
                ans=0.39894228+y*(-0.3988024e-1+y*(-0.362018e-2
                        +y*(0.163801e-2+y*(-0.1031555e-1+y*ans))));
                ans *= (exp(ax)/sqrt(ax));
        }
        return x < 0.0 ? -ans : ans;
}
```

```
float bessk1(float x)
{
	float bessi1(float x);
	double y,ans;

	if (x <= 2.0) {
		y=x*x/4.0;
		ans=(log(x/2.0)*bessi1(x))+(1.0/x)*(1.0+y*(0.15443144
			+y*(-0.67278579+y*(-0.18156897+y*(-0.1919402e-1
			+y*(-0.110404e-2+y*(-0.4686e-4)))))));
	} else {
		y=2.0/x;
		ans=(exp(-x)/sqrt(x))*(1.25331414+y*(0.23498619

+y*(-0.3655620e-1+y*(0.1504268e-1+y*(-0.780353e-2
			+y*(0.325614e-2+y*(-0.68245e-3)))))));
	}
	return ans;
}
```