

GSL 2.1 Bessel K0/K1 issues

- (Major) Cancellation error near $x=2$.

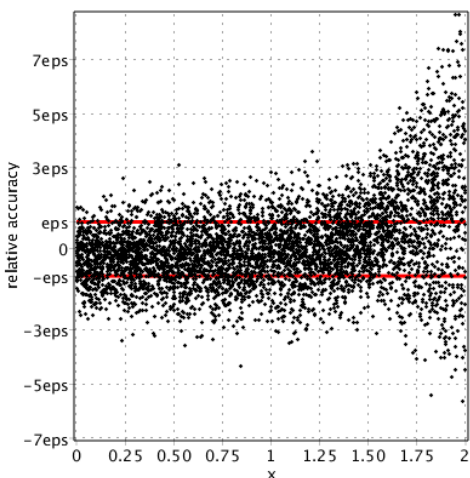
bessel_k0.c, line 145: `result->val = ex * ((-lx+M_LN2)*I0.val - 0.25 + c.val);`

bessel_k1.c, line 149: `result->val = ex * ((lx-M_LN2)*I1.val + (0.75 + c.val)/x);`

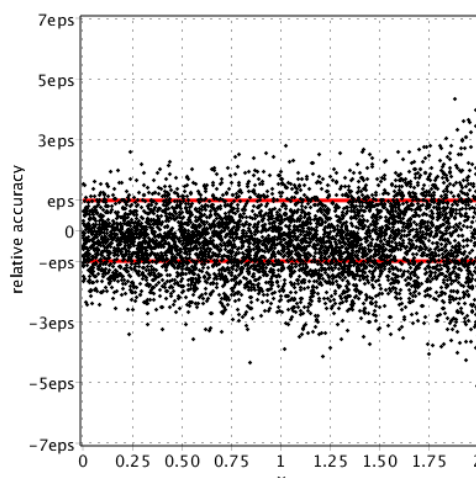
Expressions in brackets suffer from cancellation near $x=2$, since $\ln x = \log(x) \sim M_LN2$

It is better to use $-\log(x/2)$ and $\log(x/2)$ instead of subtracting the logarithms.

For example, if we measure accuracy of `bessel_k0` against quadruple precision (Maple):



`-lx+M_LN2`, max error = 8.7eps



`-log(x/2)`, max error = 5.11eps

Max error is computed based on 50K random samples in $[0, 2]$.

- (Minor) Somehow last digits of Chebyshev coefficients in all Bessel routines are not properly rounded, but just truncated.

I used original coefficients from double precision SLATEC (e.g. `dbesk0.f`) as true values and compared with GSL.

For example, `ak0_data` for `bessel_k0.c`:

GSL/truncated:	Rounded:
-0.01201869826307592,	-0.01201869826307592,
-0.00917485269102569,	-0.00917485269102570,
0.00014445509317750,	0.00014445509317750,
-0.00000401361417543,	-0.00000401361417544,
0.00000015678318108,	0.00000015678318109,
-0.00000000777011043,	-0.00000000777011044,
0.00000000046111825,	0.00000000046111826,
-0.00000000003158592,	-0.00000000003158593,
0.00000000000243501,	0.00000000000243502,
-0.00000000000020743,	-0.00000000000020743,
0.00000000000001925,	0.00000000000001926,
-0.00000000000000192,	-0.00000000000000193,
0.00000000000000020,	0.00000000000000021,
-0.00000000000000002,	-0.00000000000000002,

Please see the patch for `bessel_k0.c`.