Stand: April 29, 2013 20:10

Notes from testing implementation of special functions in CPPPAW

Robert Schade

Contents

1 functions to be tested

function	maple function
SPFUNCTION$BESSEL(L,X,Y,DYDX)	sqrt(Pi/(2*x))*BesselJ(l+1/2,x)
SPFUNCTION$BESSEL0(L,X,Y,DYDX)	$\frac{x^l}{1\cdot3\cdot5\cdot\ldots\cdot(2l+1)} = x^l/product(2*i-1, i=1..l+1)$
SPFUNCTION$NEUMANN(L,X,Y,DYDX)	sqrt(Pi/(2*x))*BesselY(l+1/2,x)
SPFUNCTION$NEUMANN0(L,X,Y,DYDX)	$-\frac{1\cdot3\cdot5\cdot\ldots\cdot(2l-1)}{x^{l+1}} = -product(2*i-1, i=1..l)/x^{l+1}$
SPFUNCTION$MODBESSEL(L,X,Y,DYDX)	sqrt(Pi/(2*x))*BesselI(l+1/2,x)
SPFUNCTION$MODNEUMANN(L,X,Y,DYDX)	sqrt(Pi/(2*x))*BesselK(l+1/2,x)
SPFUNCTION$MODHANKEL(L,X,Y,DYDX)	$\frac{\pi}{2}(-1)^{l+1}\frac{\pi}{2x}$(BesselI(l+1/2,x)-BesselK(l+1/2,x))
SPECIALFUNCTION$BESSEL(L,X,Y)	sqrt(Pi/(2*x))*BesselJ(l+1/2,x)
SPECIALFUNCTION$ERF(X,Y)	erf(x)

2 method of testing

- CPPAW-version to be tested (old implementation): version from pbloechl/devel, last change in src/paw_specialfunctions.f90 was in revision 1102 (2011-08-25 20:28:37 +0200)
- CPPAW-version to be tested (new implementation): patched version (see attached version of paw_-specialfunctions.f90)
- the numerical results from each implementation in CPPAW are tested against results from arbitrary precision numerics
- the calculation in maple 16.02 is done with a precision of 200 digits (evalf(sqrt(Pi/(2*x))*BesselJ(order+1/2,x),200)), benchmark results agree with results from Mathematica 8.0 (N[SphericalBesselJ[order, x],200])

  $$J_{15,maple}(0.1) = 0.52102909410089786555364182079129904470930627550604412445307627192822346825366384785 \tag{2.1}$$

  $$J_{15,mathe}(0.1) = 0.52102909410089786555364182079129904470930627550604412445307627192822346825366384785 \tag{2.2}$$

  $$\partial_x J_{15,maple}(x)|_{x=0.1} = 0.78152785225421803473682623024639654754275656616328310177500653623772682943875 \tag{2.3}$$

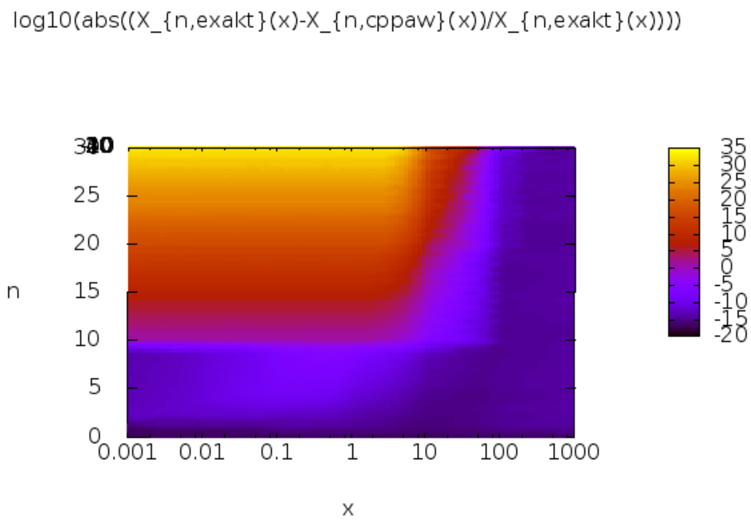
  $$\partial_x J_{15,mathe}(x)|_{x=0.1} = 0.78152785225421803473682623024639654754275656616328310177500653623772682943875 \tag{2.4}$$

  (since maple has the nicer commandline interface, we use that)
- Derivatives are calculated in maple with limit(diff(...,x),x=val)
- the CPPAW-codes are compiled with gcc version 4.7.2 (Debian 4.7.2-5) (FIXME)
- the logarithm of the relative difference is computed using bc with scale=1000
- when $log10(abs((X_{n,exakt}(x) - X_{n,cppaw}(x))/X_{n,exakt}(x))) = 0$ or when values are missing then an error (double overflow or underflow) happened or $x < x_{min} = 10^{-30}$

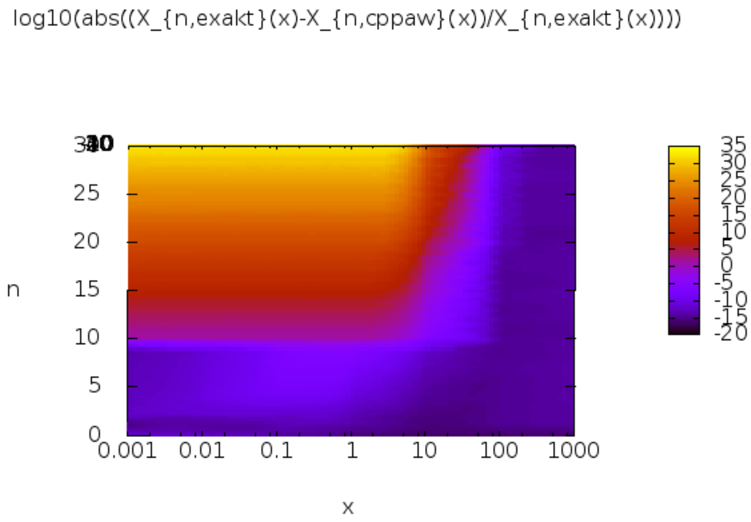
3 results for old implementation

3.1 SPFUNCTION$BESSEL

3.1.1 Value

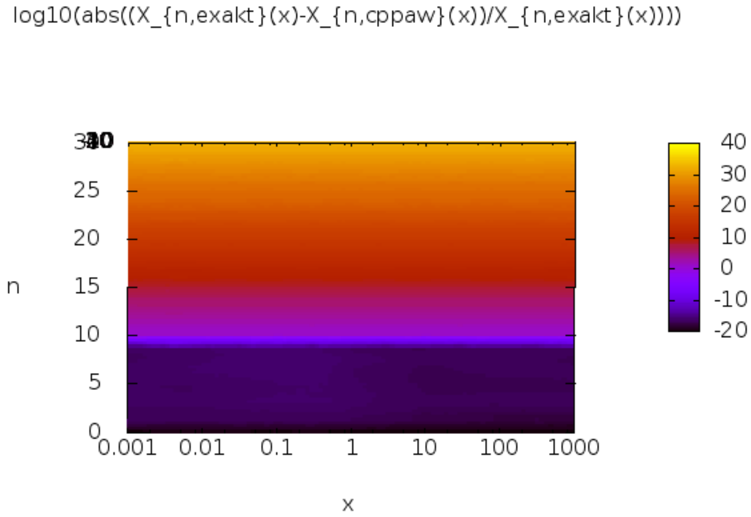


3.1.2 Derivative

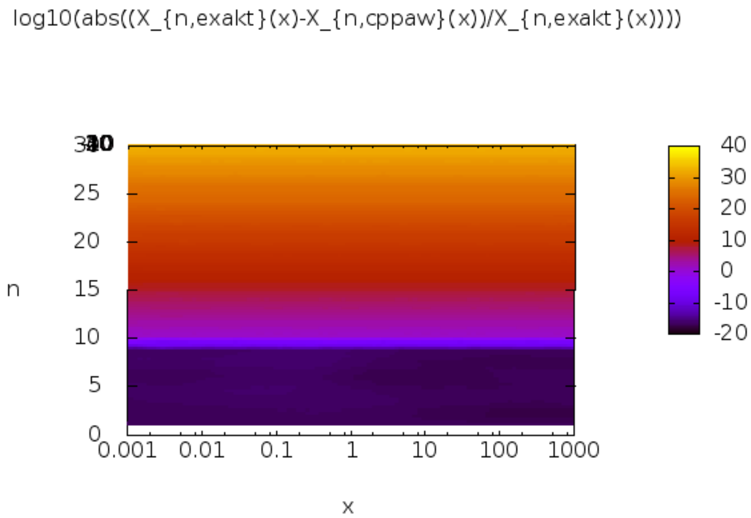


3.2 SPFUNCTION$BESSEL0

3.2.1 Value

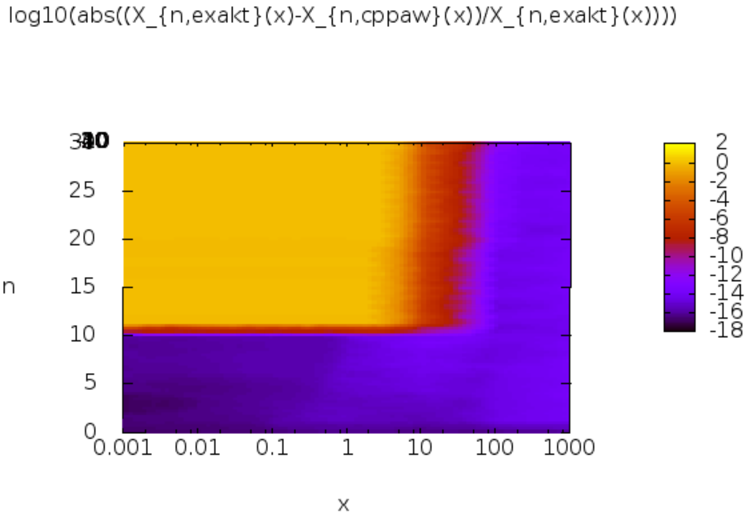


3.2.2 Derivative

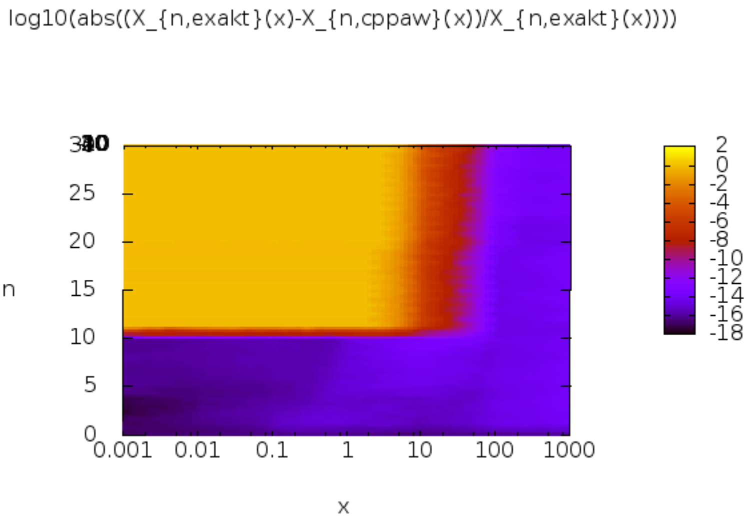


3.3 SPFUNCTION$NEUMANN

3.3.1 Value

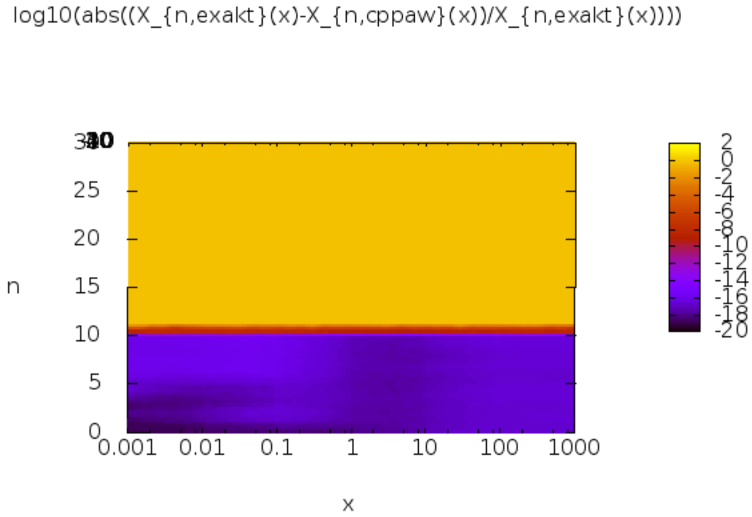


3.3.2 Derivative

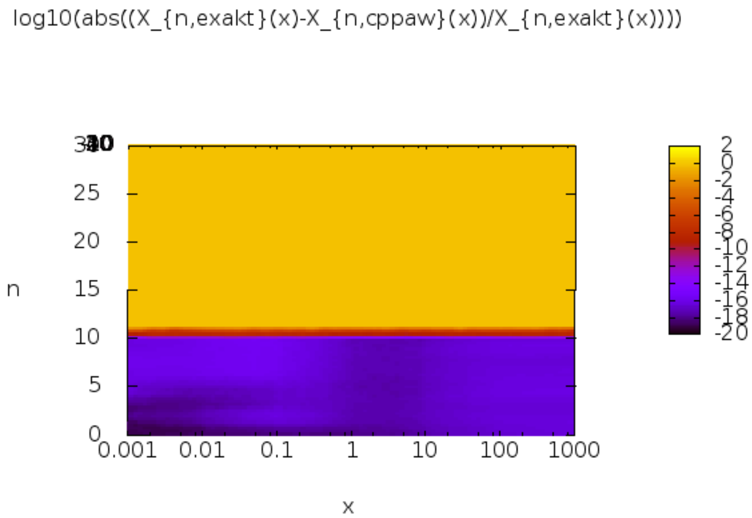


3.4 SPFUNCTION$NEUMANN0

3.4.1 Value

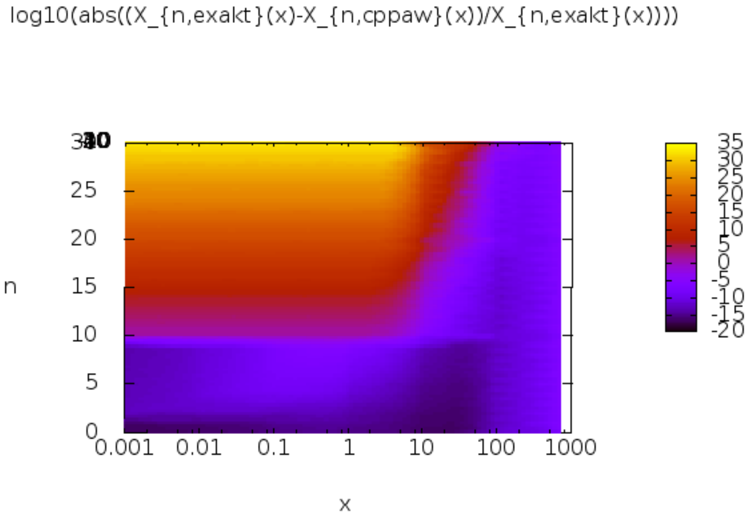


3.4.2 Derivative

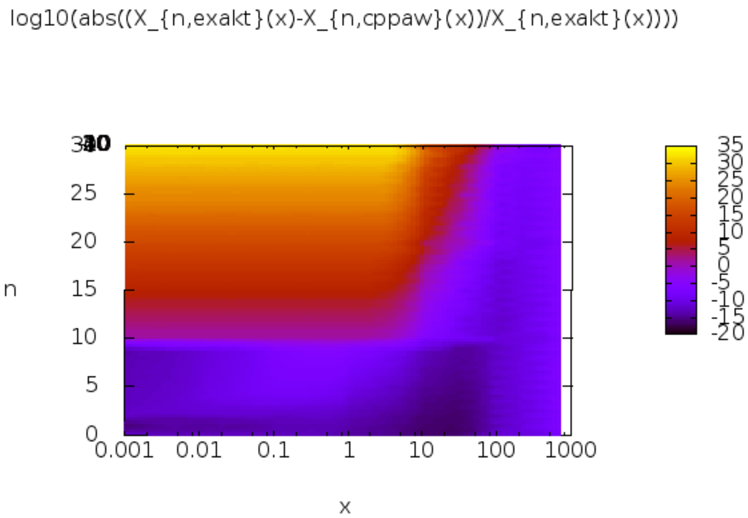


3.5 SPFUNCTION$MODBESSEL

3.5.1 Value

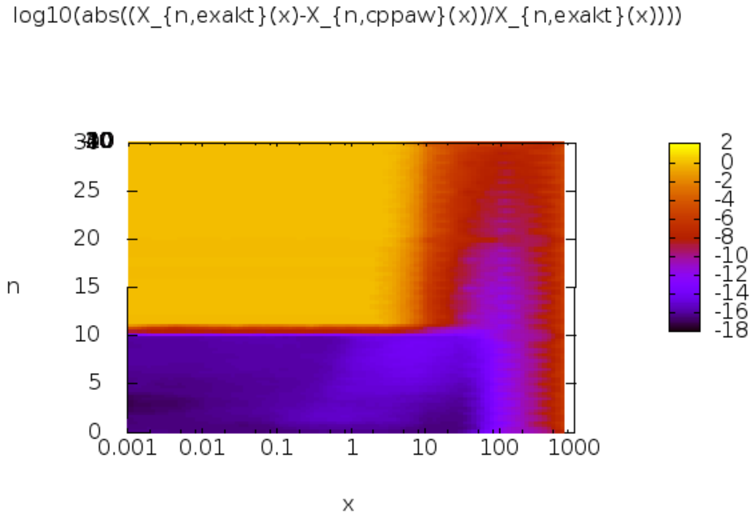


3.5.2 Derivative

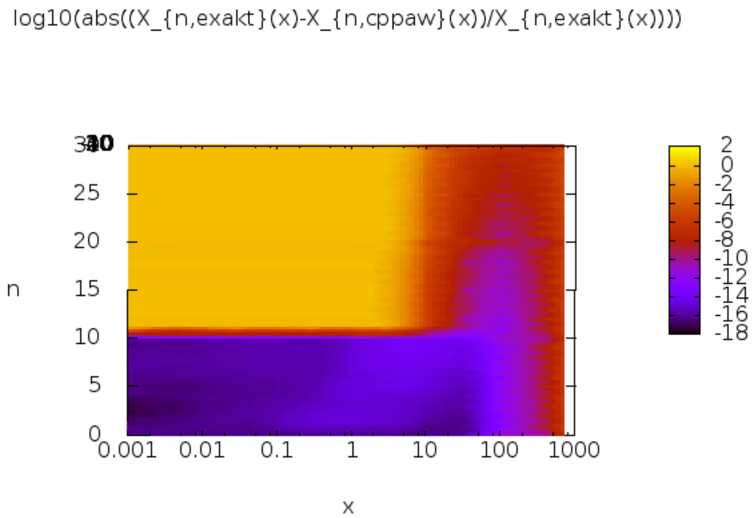


3.6 SPFUNCTION$MODNEUMANN

3.6.1 Value

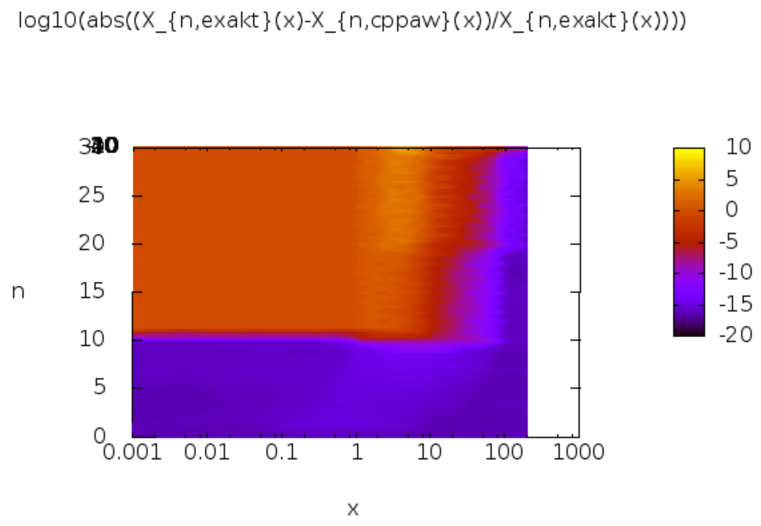


3.6.2 Derivative

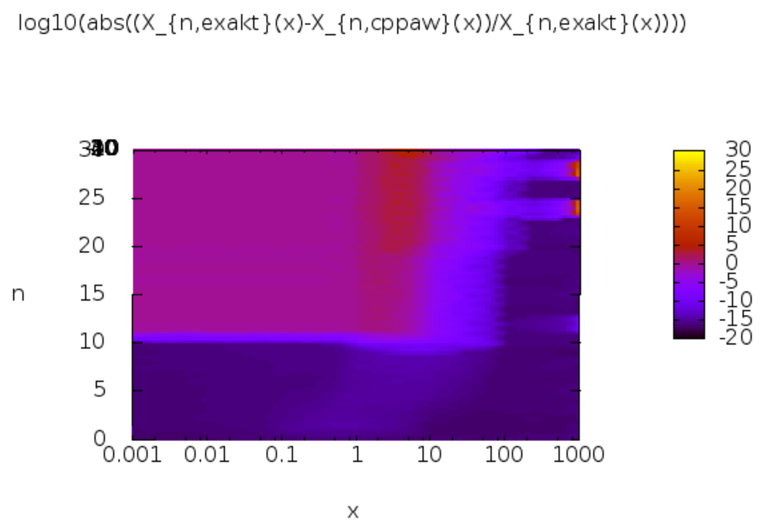


3.7 SPFUNCTION$MODHANKEL

3.7.1 Value

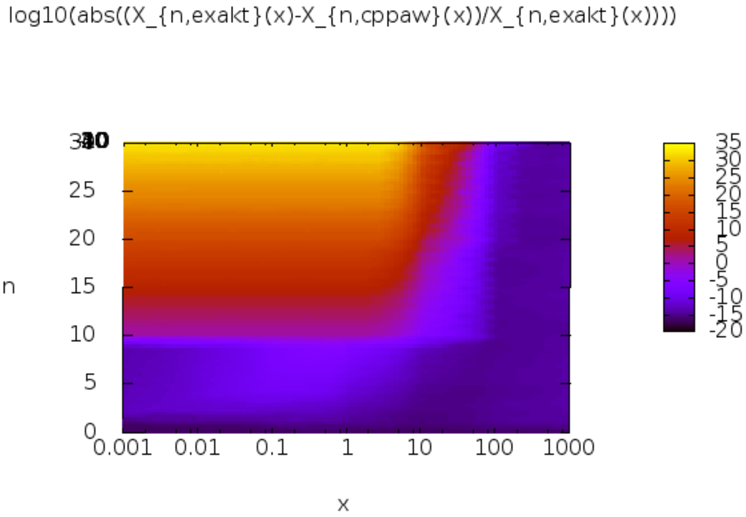


3.7.2 Derivative



3.8 SPECIALFUNCTION$BESSEL

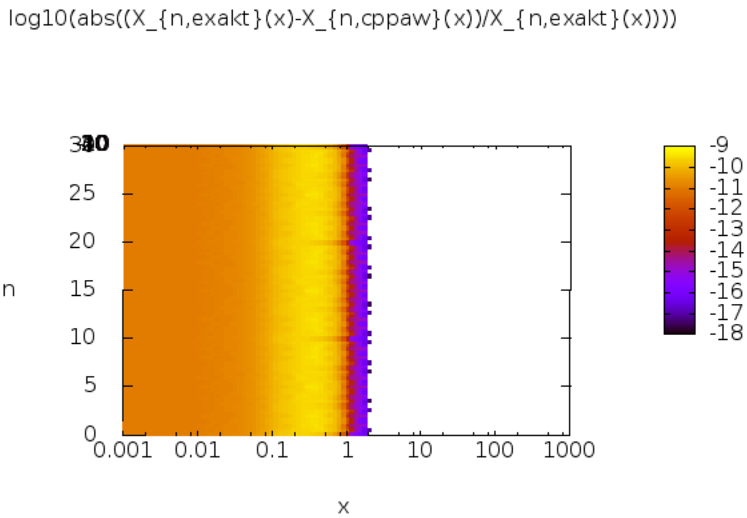
3.8.1 Value



3.9 SPECIALFUNCTION$ERF

3.9.1 Value

note: $erf(x)$ has no l-parameter



4 results for new implementation

4.1 SPFUNCTION$BESSEL

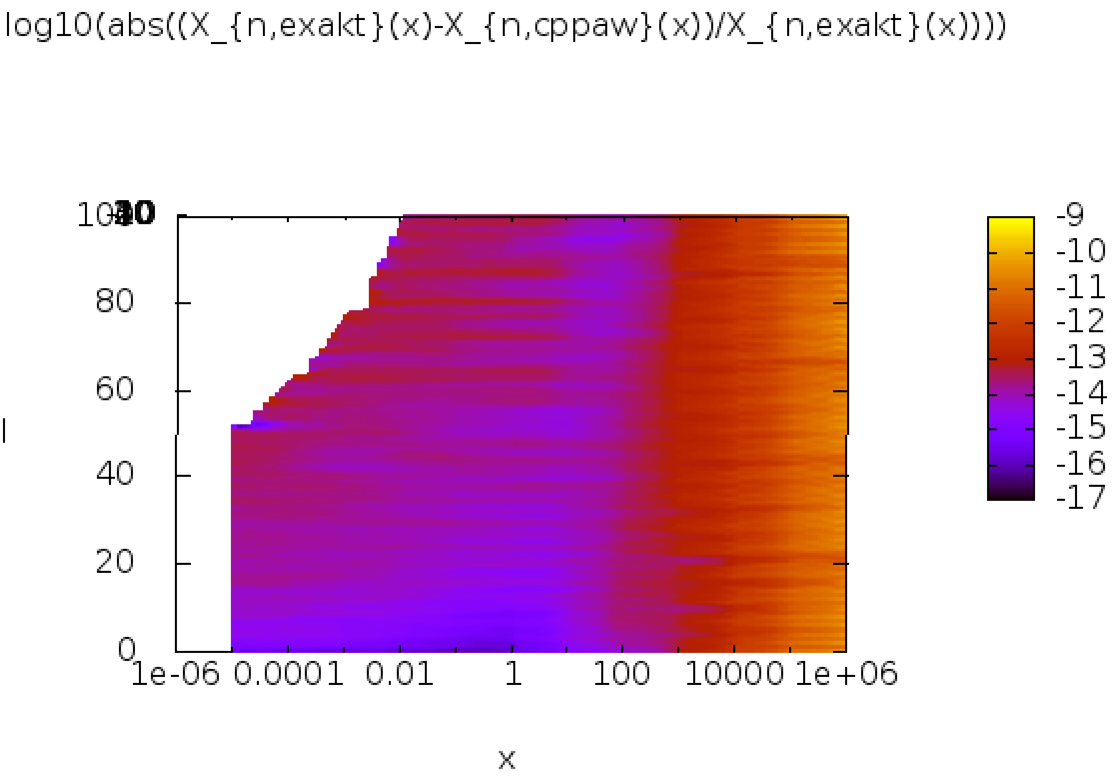
- uses DBESJ from slatec for $J_n(x)$ $(n \geq 0)$
- $j_l(x) = \sqrt{\frac{\pi}{2x}} J_{l+\frac{1}{2}}(x)$
- $\partial_x j_0(x) = -j_1(x) = \frac{cos(x)}{x} - \frac{sin(x)}{x^2}$ (Abramowitz/Stegun 10.1.11)
- for $l > 0$ (Abramowitz/Stegun 9.1.30):

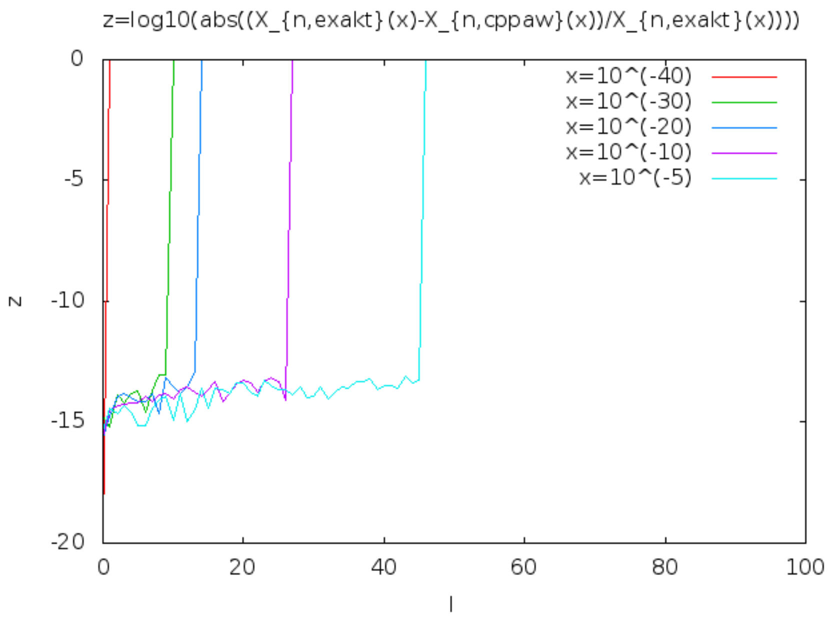
$$\partial_x j_l(x) = \partial_x \sqrt{\frac{\pi}{2x}} J_{l+\frac{1}{2}}(x) \tag{4.1}$$

$$= -\frac{\sqrt{2\pi x}}{4x^2} J_{l+\frac{1}{2}}(x) + \frac{\sqrt{2\pi}}{4\sqrt{x}} (J_{l-\frac{1}{2}}(x) - J_{l+\frac{3}{2}}(x)) \tag{4.2}$$

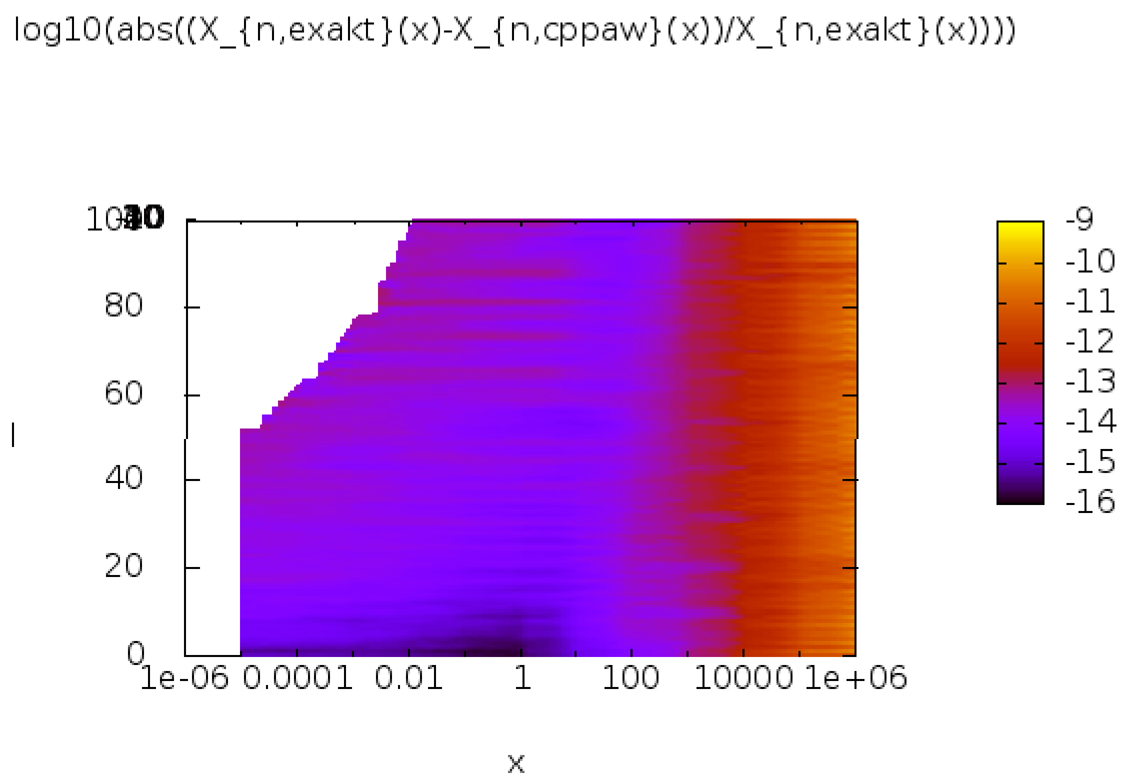
- DBESJ has internal overflow and undeflow detection
- for arguments in the white area above $l > 55$ and small arguments the routine DBESJ throws an error because of a double underflow (values smaller than $< 10^{-308}$)

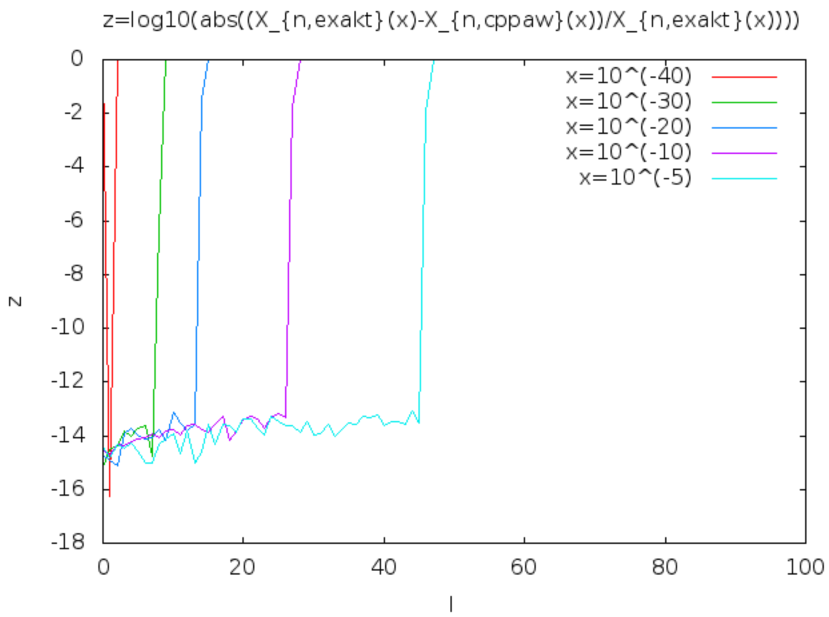
4.1.1 Value


log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
0
20
40
60
80
100
l
1e-06
0.0001
0.01
1
100
10000
1e+06
x
-9
-10
-11
-12
-13
-14
-15
-16
-17


z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
0
-5
-10
-15
-20
z
0
20
40
60
80
100
l

4.1.2 Derivative





4.2 SPFUNCTION$BESSEL0

- (Abramowitz/Stegun 10.1.2)

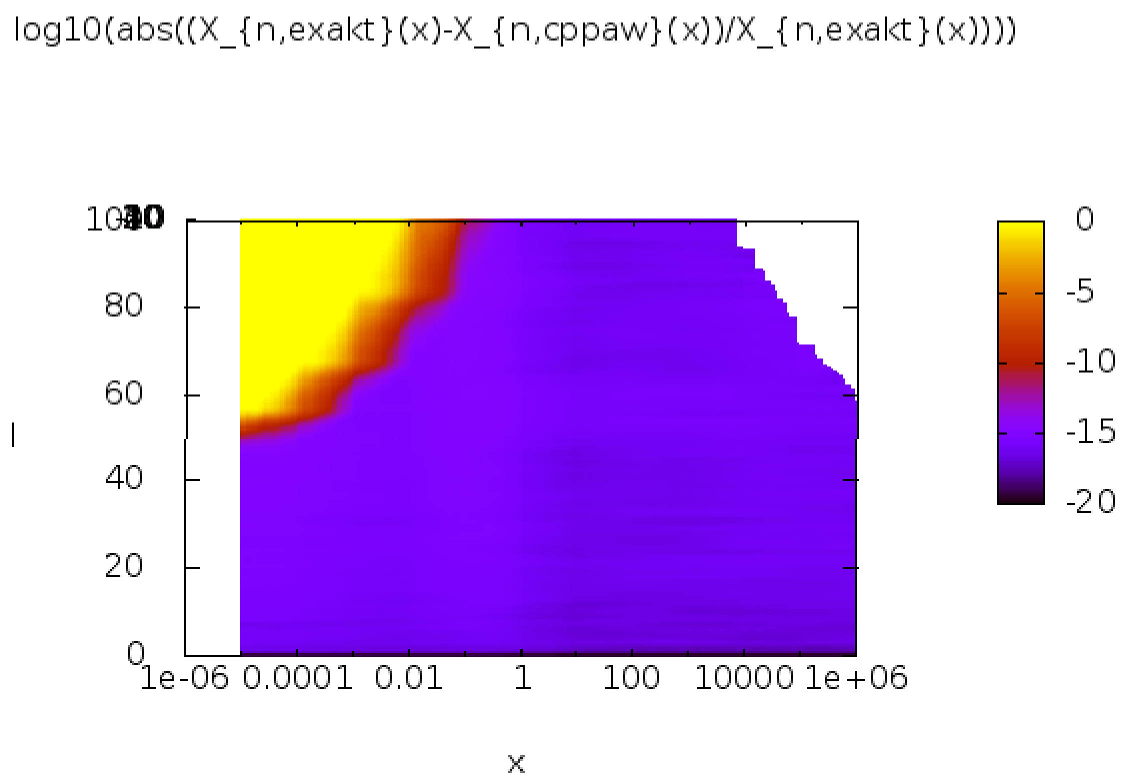
$$j_{l,0}(x) = \frac{x^l}{1 \cdot 3 \cdot 5 ... (2l+1)} \tag{4.3}$$

$$\partial_x j_{l,0}(x) = \frac{l x^{l-1}}{1 \cdot 3 \cdot 5 ... (2l+1)} \tag{4.4}$$

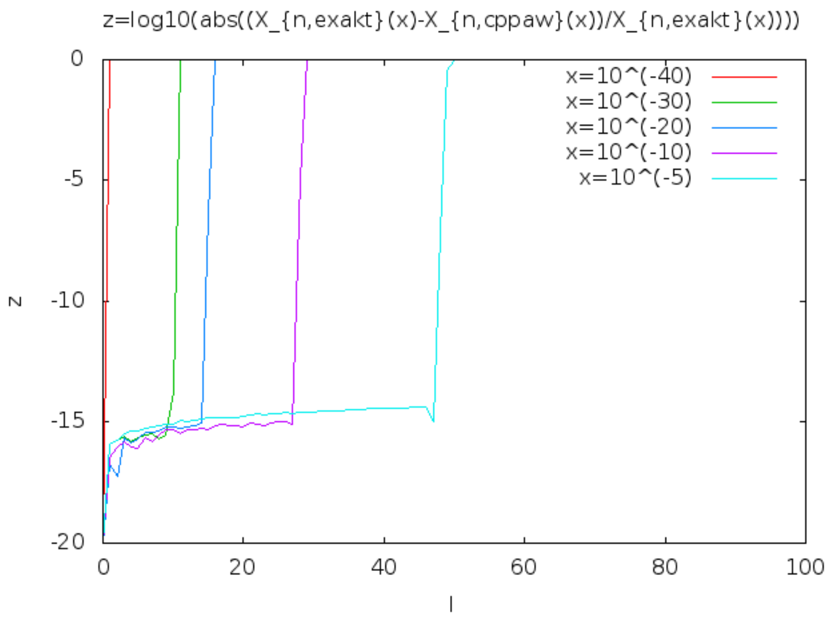
$$\partial_x j_{l=0,0}(x) = 0 \tag{4.5}$$

- no underflow/overflow handling implemented yet (FIXME)

4.2.1 Value

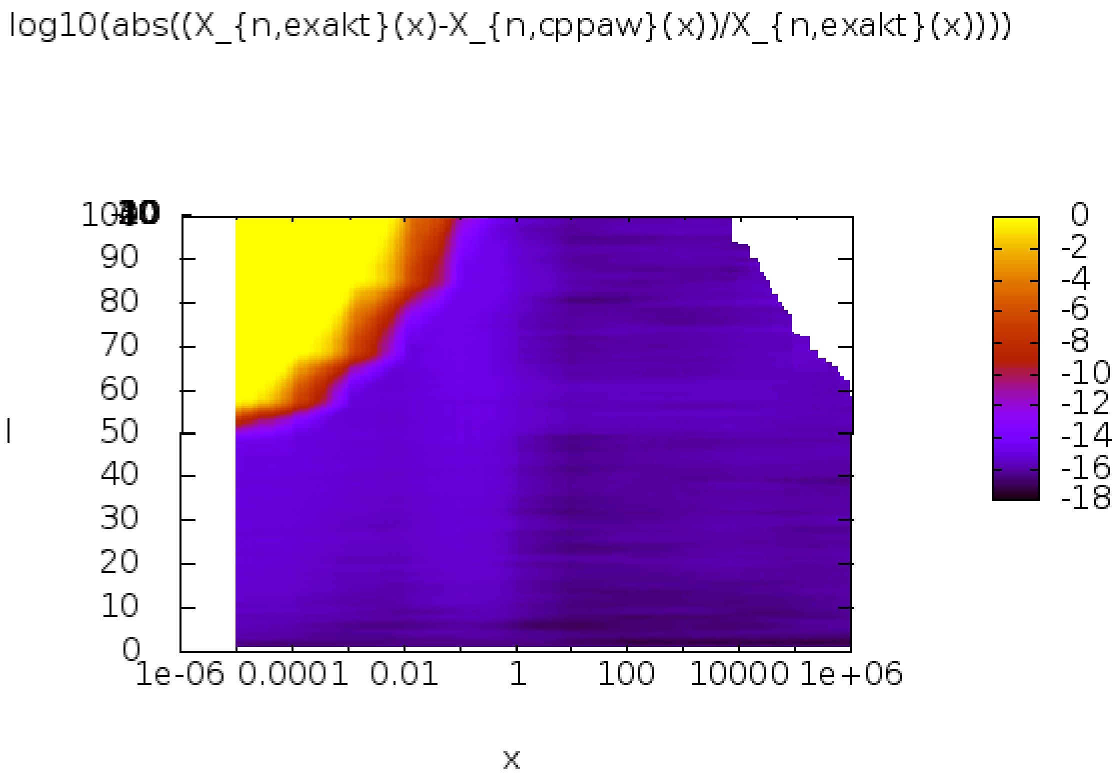


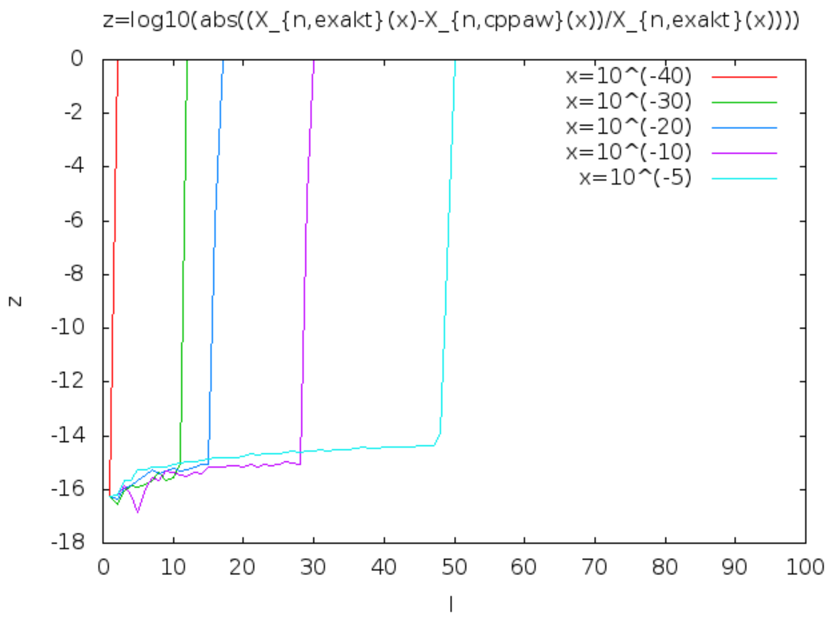
log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
100
80
60
40
20
0
l
1e-06
0.0001
0.01
1
100
10000
1e+06
x
0
-5
-10
-15
-20



z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
0
-5
-10
-15
-20
z
0
20
40
60
80
100
l

4.2.2 Derivative





4.3 SPFUNCTION$NEUMANN

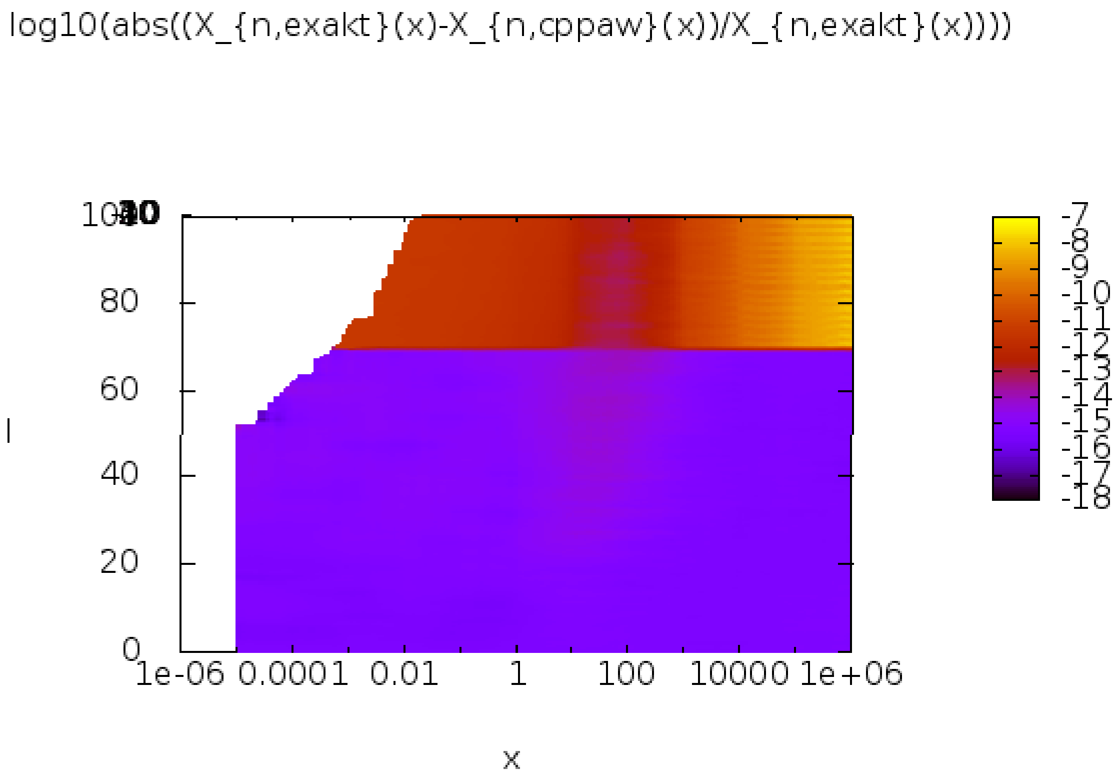
- uses DBESY from slatec for $Y_n(x)$ $(n \geq 0)$
- $y_l(x) = \sqrt{\frac{\pi}{2x}} Y_{l+\frac{1}{2}}(x)$
- $\partial_x y_0(x) = -y_1(x) = \frac{cos(x)}{x^2} + \frac{sin(x)}{x}$ (Abramowitz/Stegun 10.1.12)
- for $l > 0$ (Abramowitz/Stegun 9.1.30):

$$\partial_x y_l(x) = \partial_x \sqrt{\frac{\pi}{2x}} Y_{l+\frac{1}{2}}(x) \tag{4.6}$$

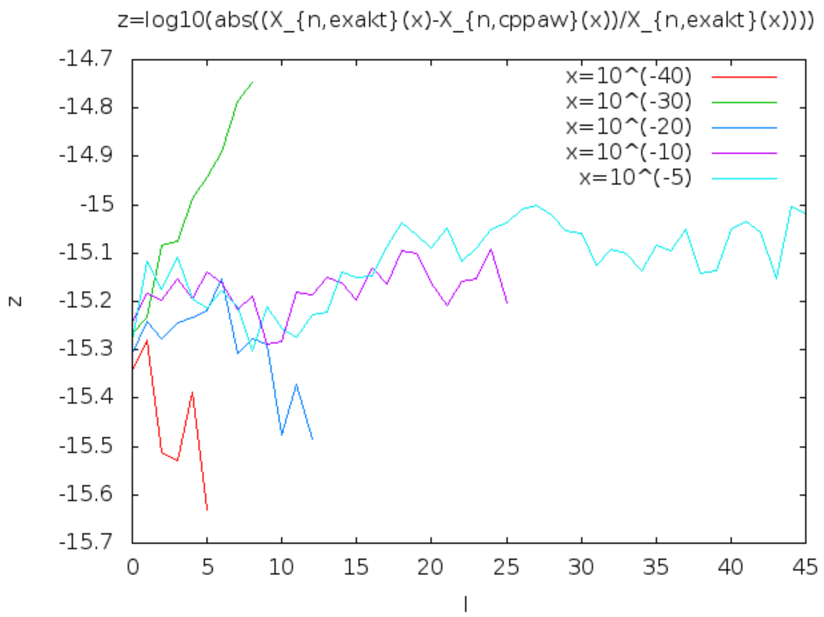
$$= -\frac{\sqrt{2\pi x}}{4x^2} Y_{l+\frac{1}{2}}(x) + \frac{\sqrt{2\pi}}{4\sqrt{x}} (Y_{l-\frac{1}{2}}(x) - Y_{l+\frac{3}{2}}(x)) \tag{4.7}$$

- DBESY has internal overflow detection
- for arguments in the white area above $l > 50$ and small arguments the routine DBESY throws an error because of a double overflow

4.3.1 Value

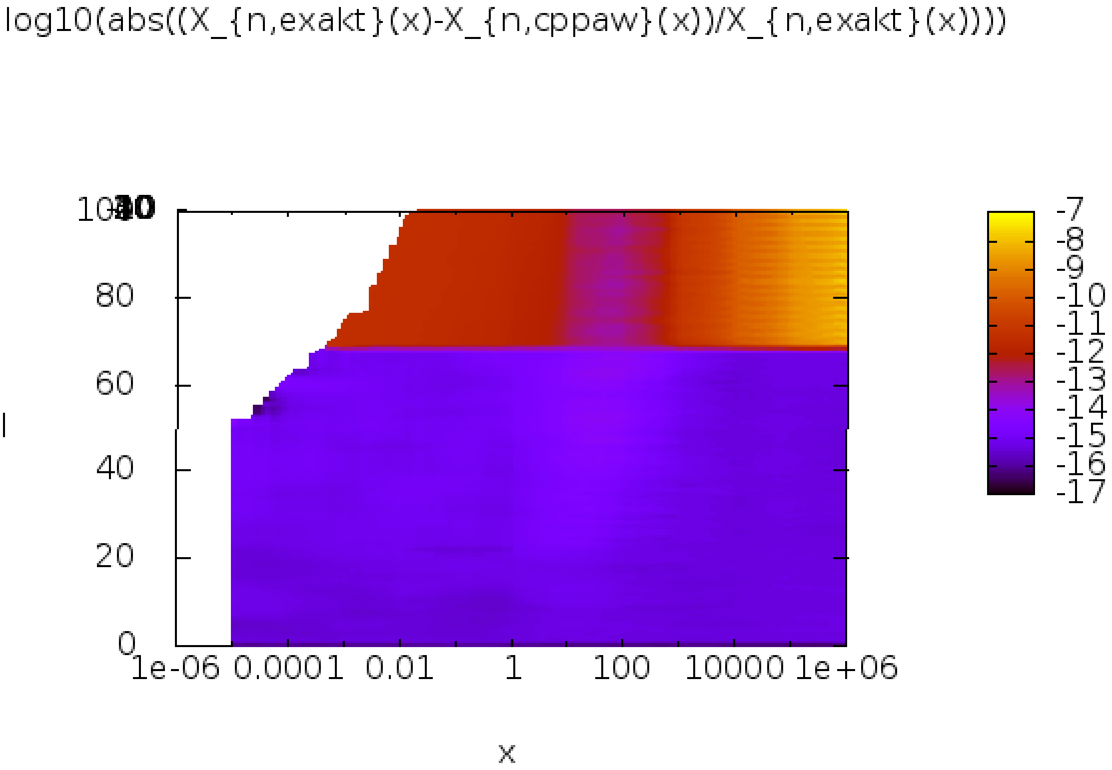


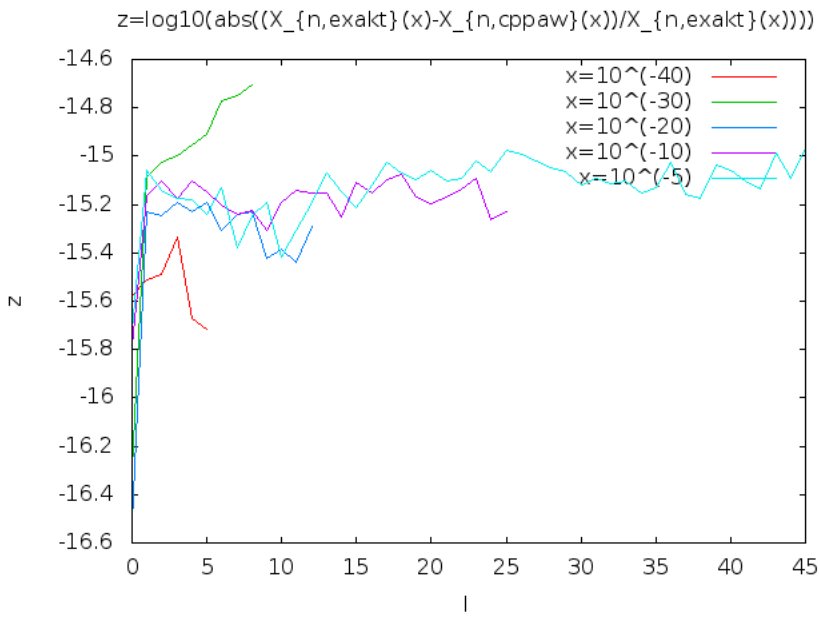
log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
100
80
60
40
20
0
l
1e-06 0.0001 0.01 1 100 10000 1e+06
x
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17
-18



z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x))))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
-14.7
-14.8
-14.9
-15
-15.1
-15.2
-15.3
-15.4
-15.5
-15.6
-15.7
z
0 5 10 15 20 25 30 35 40 45
l

4.3.2 Derivative





4.4 SPFUNCTION$NEUMANN0

- (Abramowitz/Stegun 10.1.3)

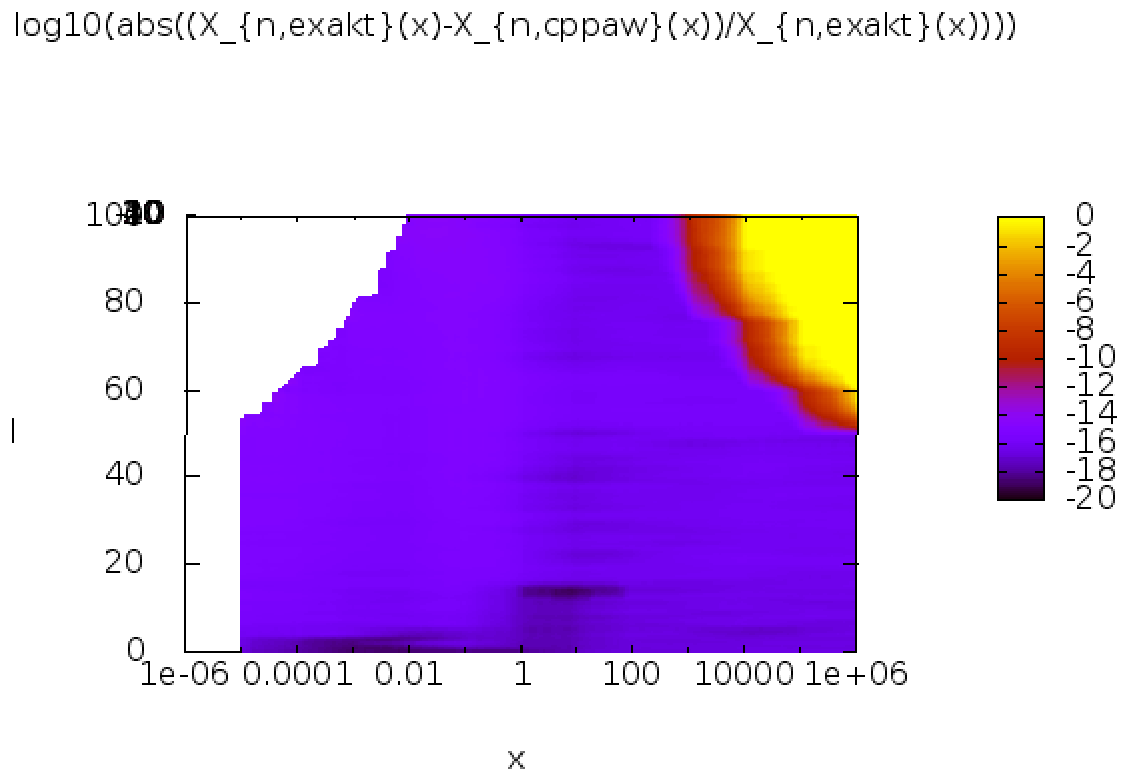
$$y_{l,0}(x) = -\frac{1 \cdot 3 \cdot 5 ... (2l-1)}{x^{l+1}} \tag{4.8}$$

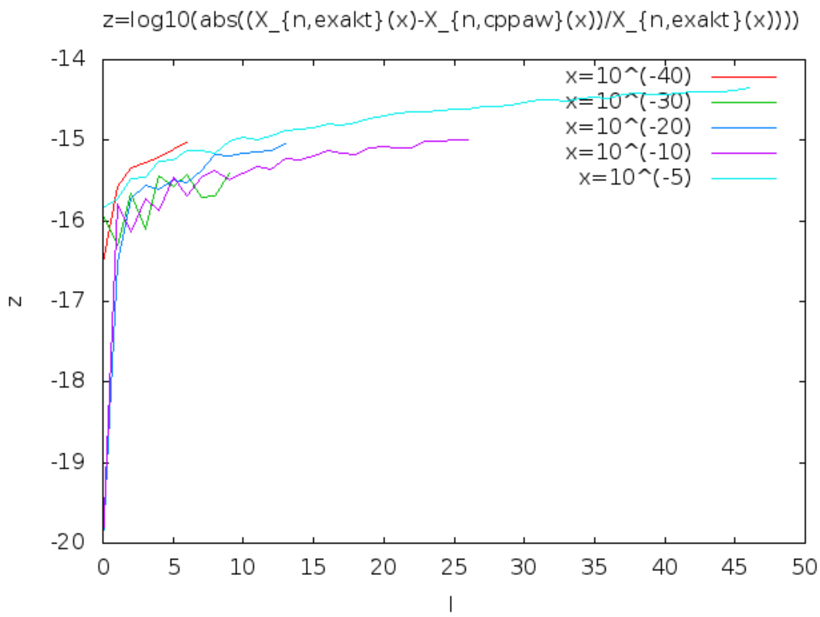
$$\partial_x y_{l,0}(x) = -(l+1)\frac{y_{l,0}(x)}{x} \tag{4.9}$$

$$\tag{4.10}$$

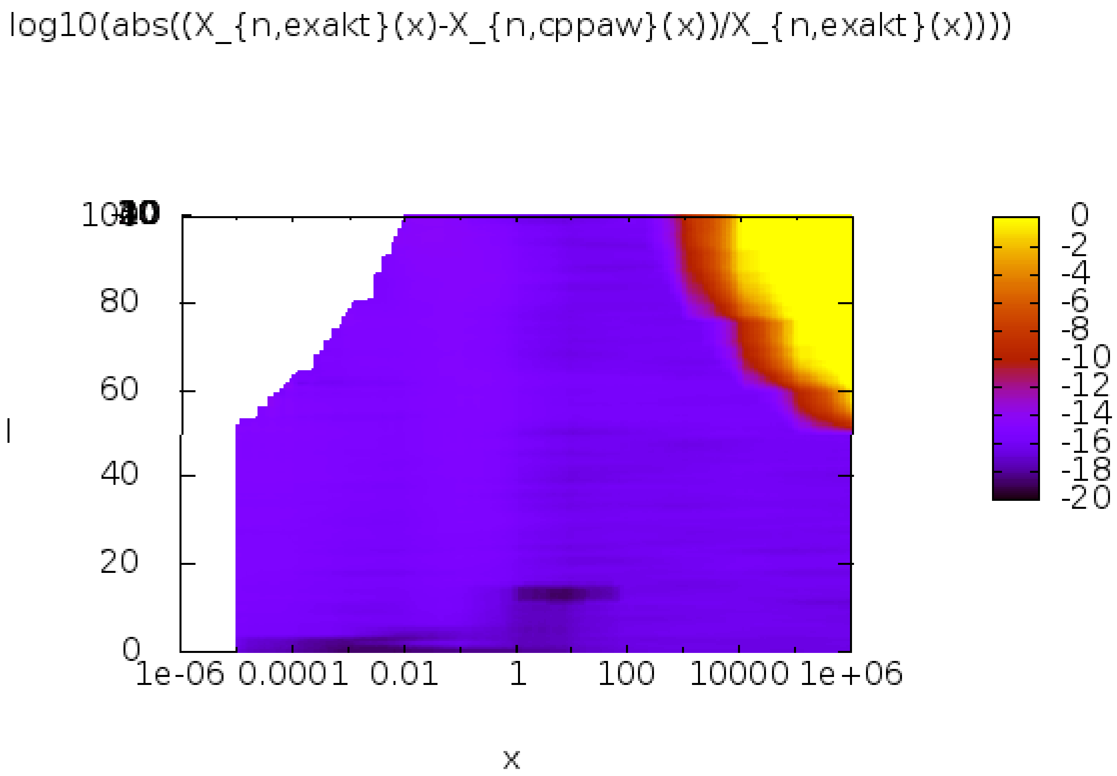
- no underflow/overflow handling implemented yet (FIXME)

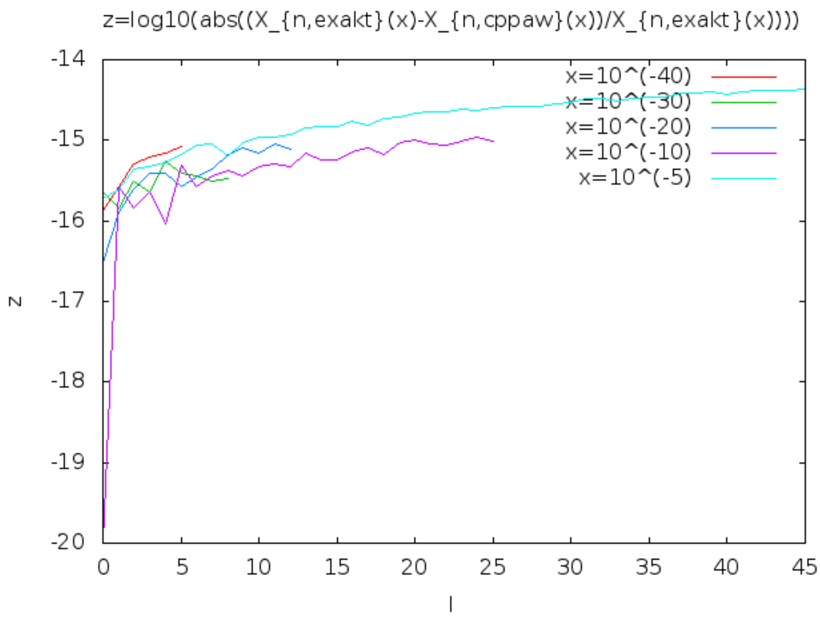
4.4.1 Value


log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
100
80
60
40
20
0
1e-06 0.0001 0.01 1 100 10000 1e+06
x
l
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20


z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
-14
-15
-16
-17
-18
-19
-20
z
0 5 10 15 20 25 30 35 40 45 50
l

4.4.2 Derivative





4.5 SPFUNCTION$MODBESSEL

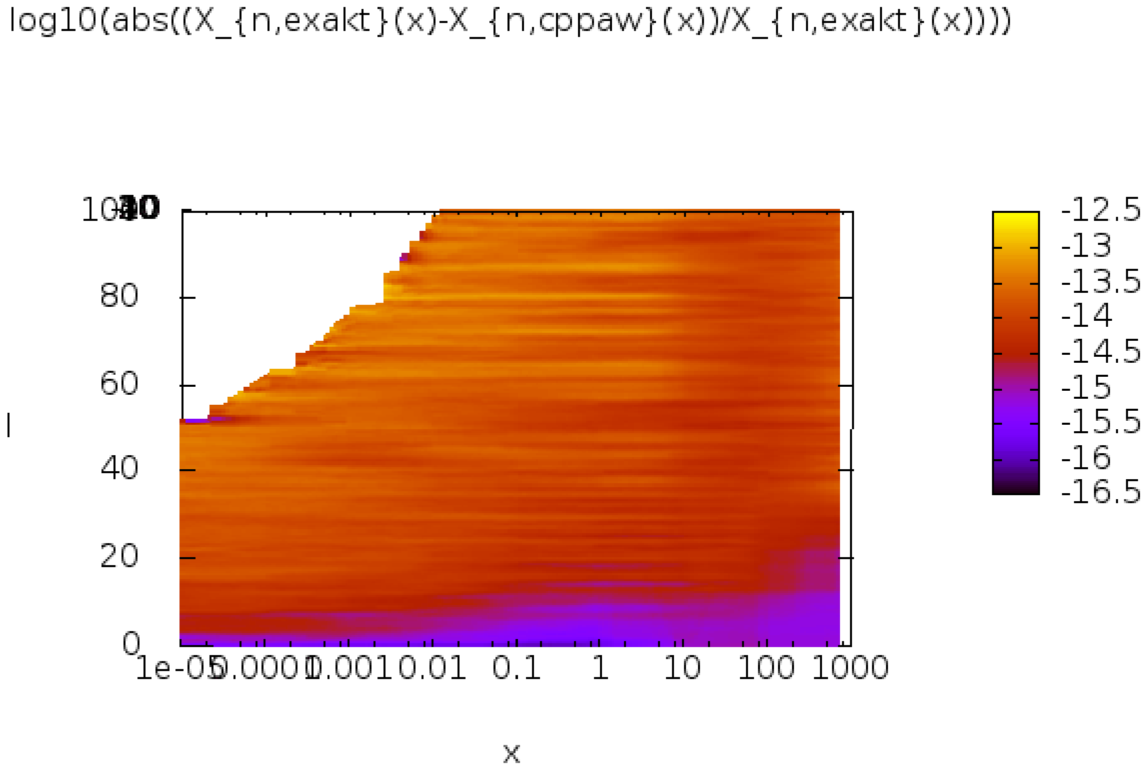
- uses DBESI from slatec for $I_n(x)$ $(n \geq 0)$
- $i_l(x) = \sqrt{\frac{\pi}{2x}} I_{l+\frac{1}{2}}(x)$
- $\partial_x i_0(x) = i_1(x) = \frac{cosh(x)}{x} - \frac{sinh(x)}{x^2}$ (Abramowitz/Stegun 10.2.13)
- for $l > 0$ (Abramowitz/Stegun 9.6.29):

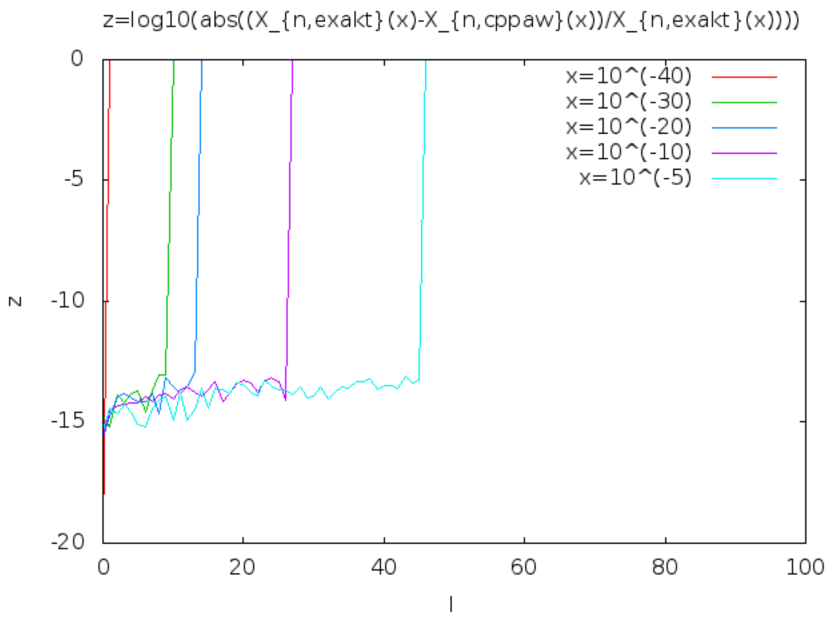
$$\partial_x i_l(x) = \partial_x \sqrt{\frac{\pi}{2x}} I_{l+\frac{1}{2}}(x) \tag{4.11}$$

$$= -\frac{\sqrt{2\pi x}}{4x^2} I_{l+\frac{1}{2}}(x) + \frac{\sqrt{2\pi}}{4\sqrt{x}} (I_{l-\frac{1}{2}}(x) + I_{l+\frac{3}{2}}(x)) \tag{4.12}$$

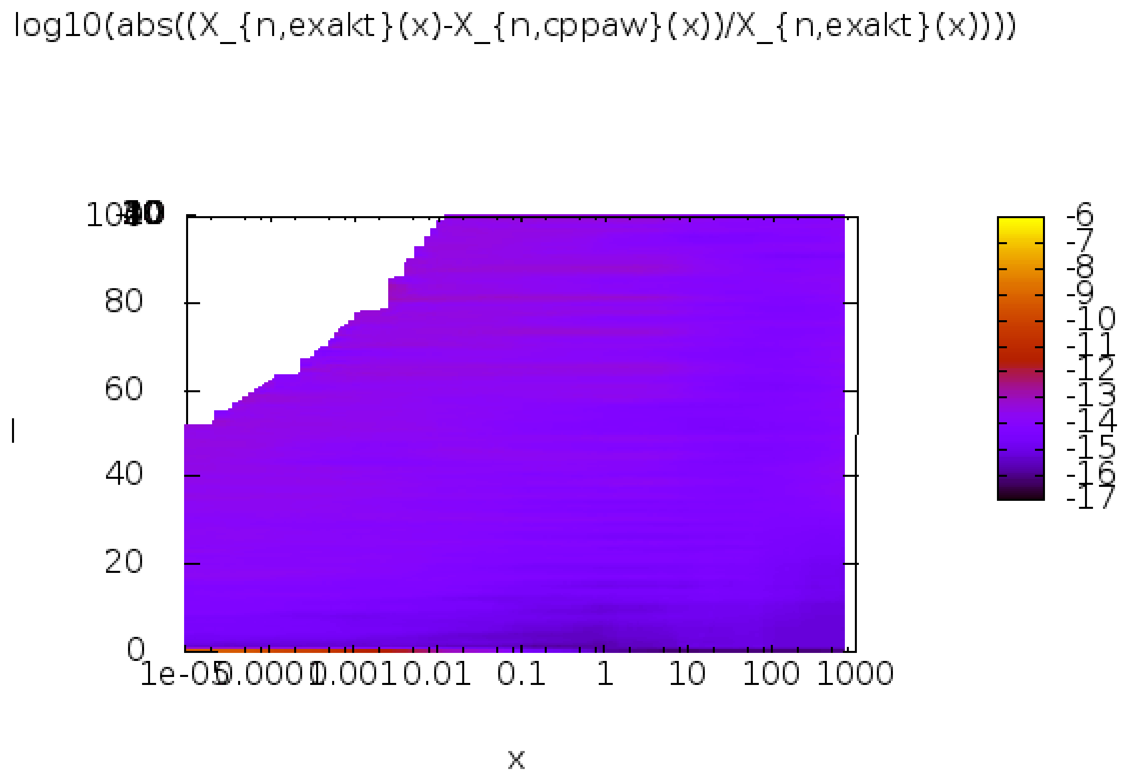
- DBESI has internal underflow and overflow detection
- for arguments in the white area above $l > 50$ and small arguments the routine DBESI throws an error because of a double underflow
- for large x the value of $i_l(x)$ becomes very large (double overflow)

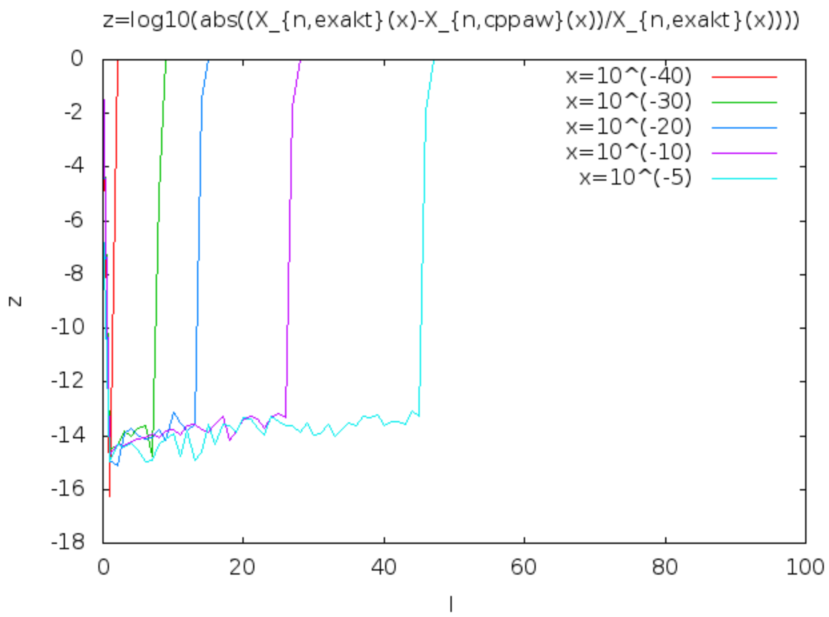
4.5.1 Value


log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
l
x
-12.5
-13
-13.5
-14
-14.5
-15
-15.5
-16
-16.5


z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
z
l

4.5.2 Derivative





4.6 SPFUNCTION$MODNEUMANN

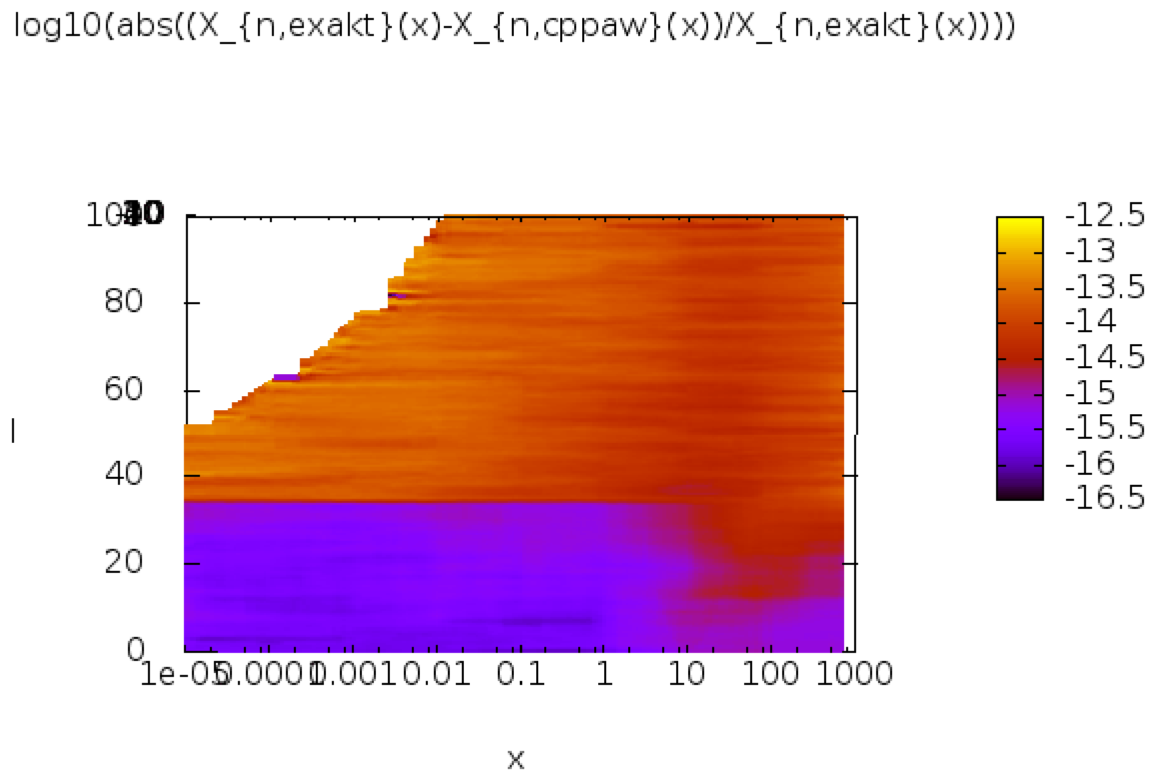
- uses MODBESSEL AND MODHANKEL (Abramowitz/Stegun 10.2.4)

$$i_{-l}(x) = \frac{2(-1)^{l+1}}{\pi} k_l(x) + i_l(x) \tag{4.13}$$

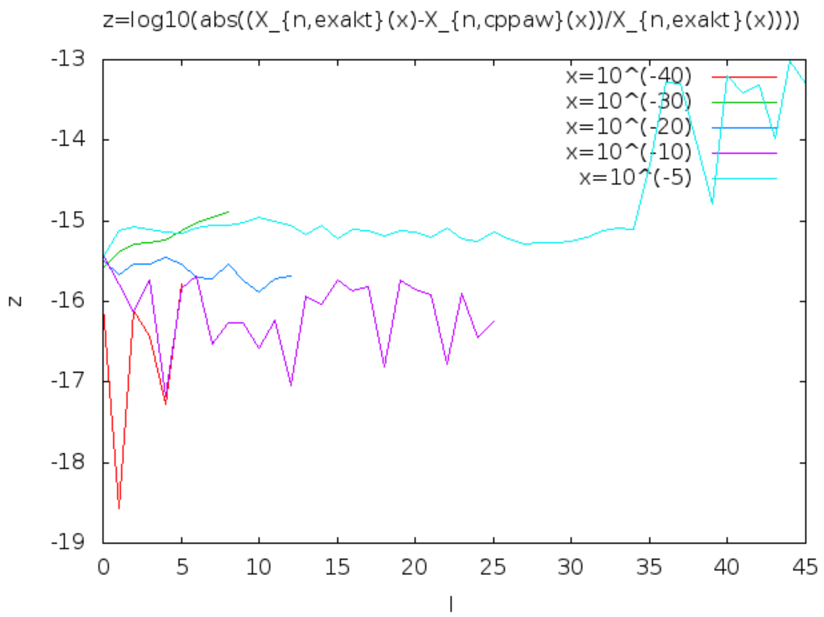
$$\partial_x i_{-l}(x) = \frac{2(-1)^{l+1}}{\pi} \partial_x k_l(x) + \partial_x i_l(x) \tag{4.14}$$

$$\tag{4.15}$$

4.6.1 Value

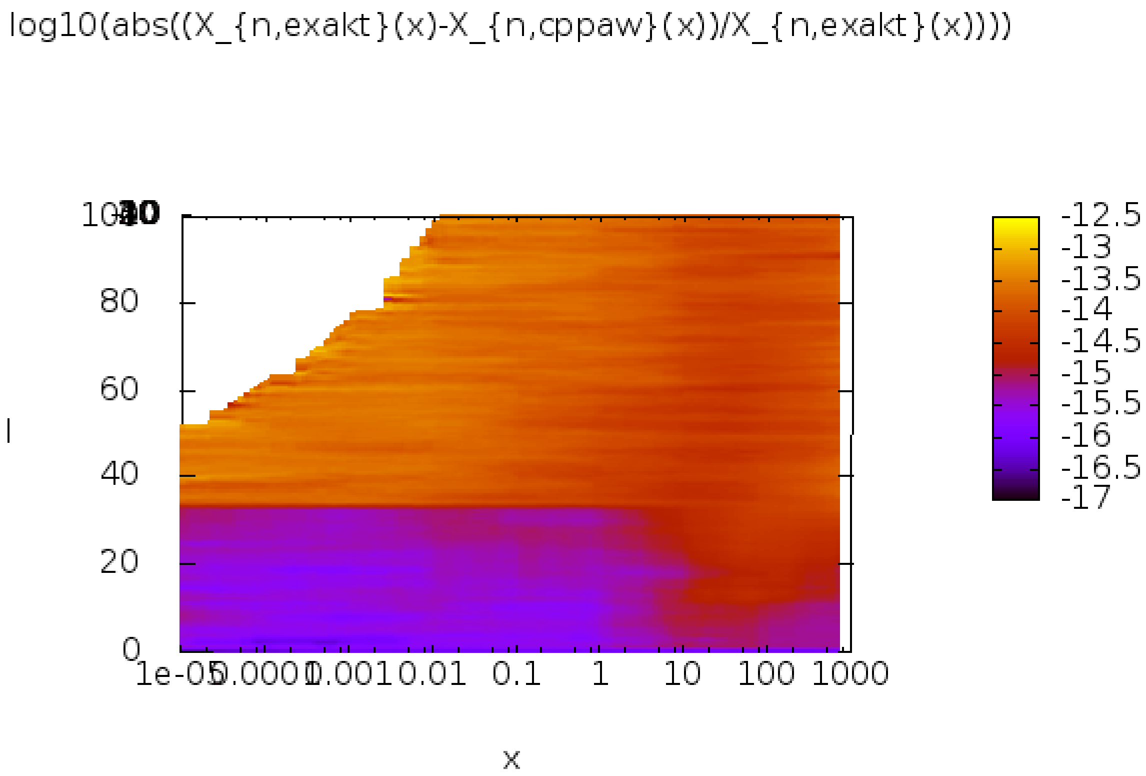


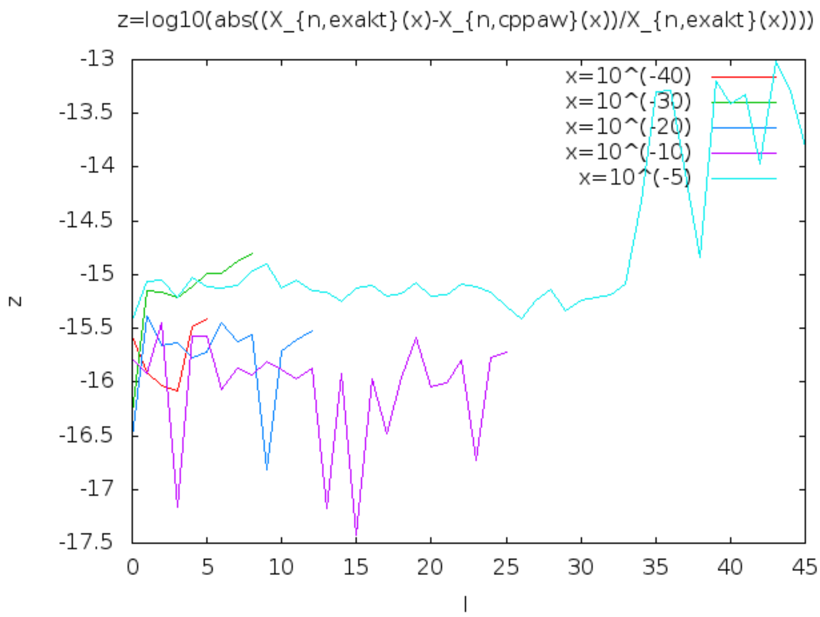
log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
l
x



z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
z
l

4.6.2 Derivative





4.7 SPFUNCTION$MODHANKEL

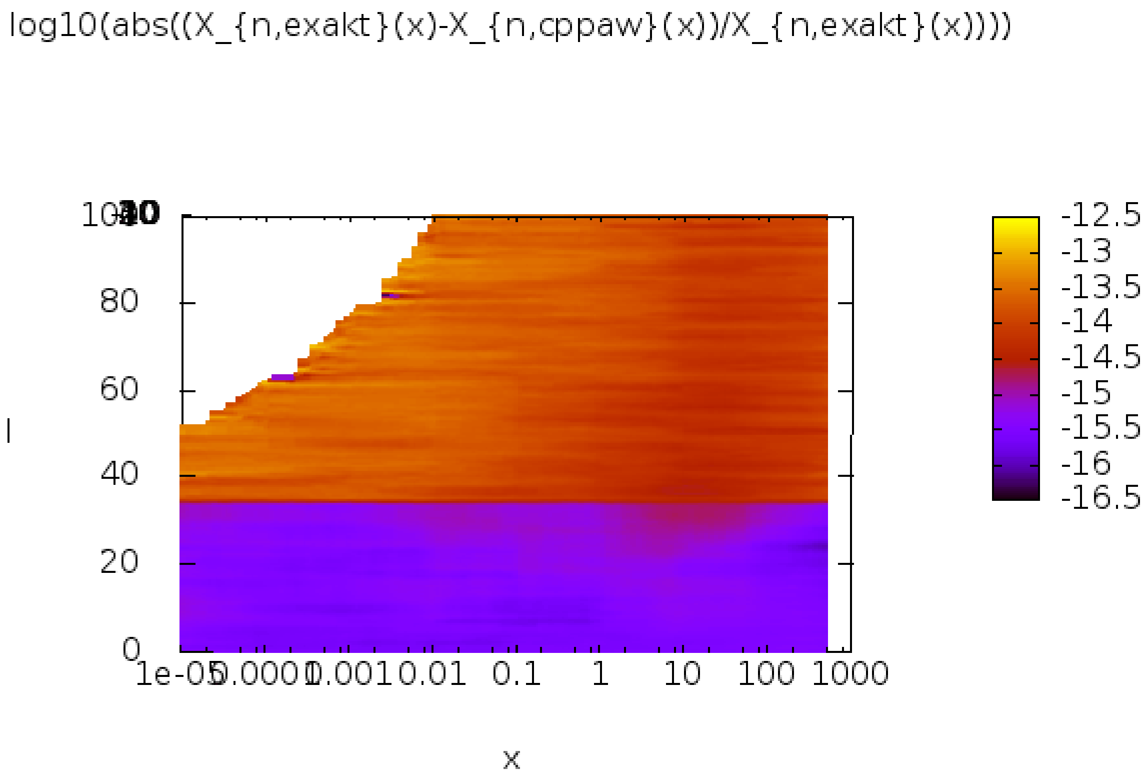
- uses DBESK from slatec for $K_n(x)$ $(n \geq 0)$
- $k_l(x) = \sqrt{\frac{\pi}{2x}} K_{l+\frac{1}{2}}(x)$
- $\partial_x k_0(x) = -\frac{\pi}{2} e^{-x} \left(\frac{1}{x} + \frac{1}{x^2}\right)$ (Abramowitz/Stegun 10.2.17)
- for $l > 0$ (Abramowitz/Stegun 9.6.29):

$$\partial_x k_l(x) = \partial_x \sqrt{\frac{\pi}{2x}} K_{l+\frac{1}{2}}(x) \tag{4.16}$$

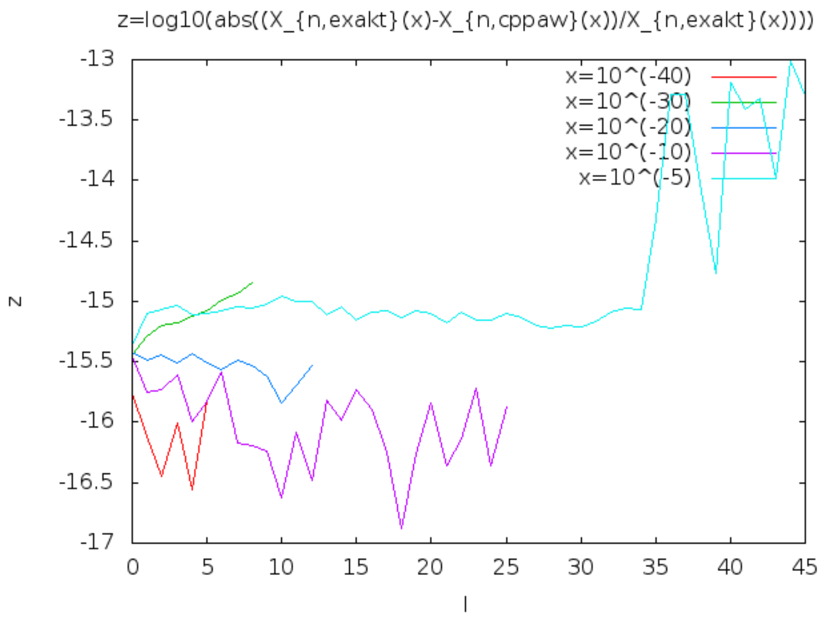
$$= -\frac{\sqrt{2\pi x}}{4x^2} I_{l+\frac{1}{2}}(x) + \frac{\sqrt{2\pi}}{4\sqrt{x}} (-K_{l-\frac{1}{2}}(x) - K_{l+\frac{3}{2}}(x)) \tag{4.17}$$

- DBESK has internal underflow and overflow detection
- for arguments in the white area above $l > 50$ and small arguments the routine DBESK throws an error because of a double underflow
- for large x the value of $k_l(x)$ becomes very small (double underflow)

4.7.1 Value

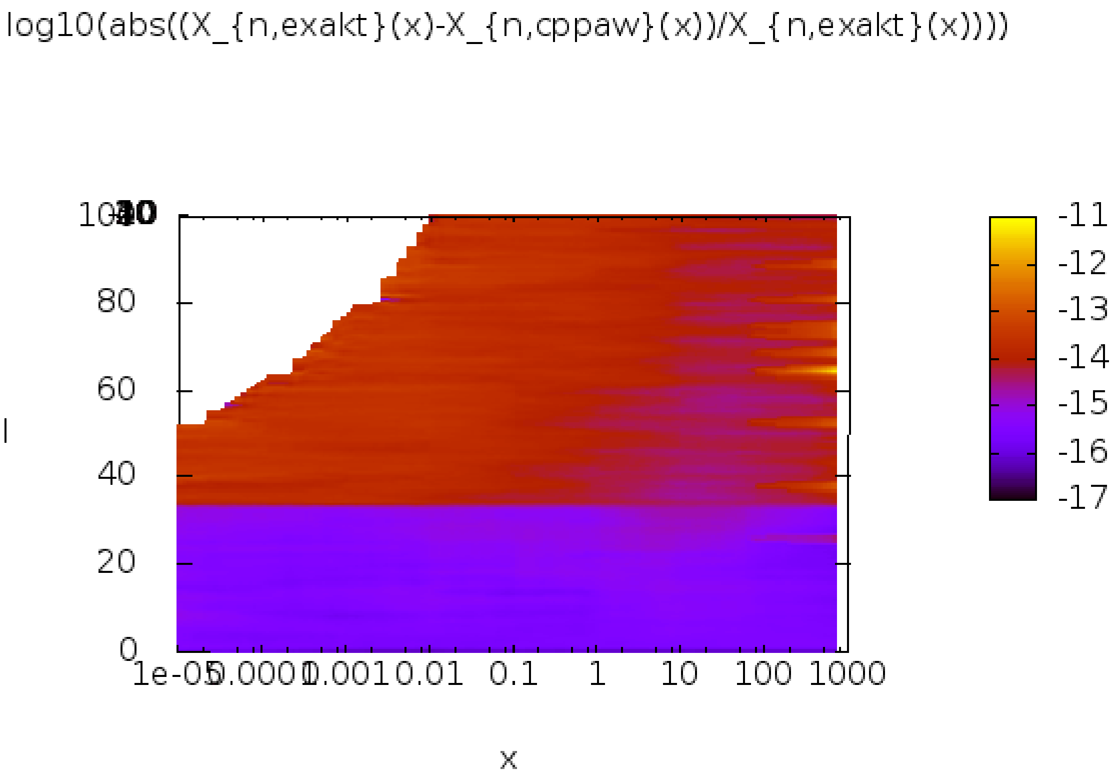


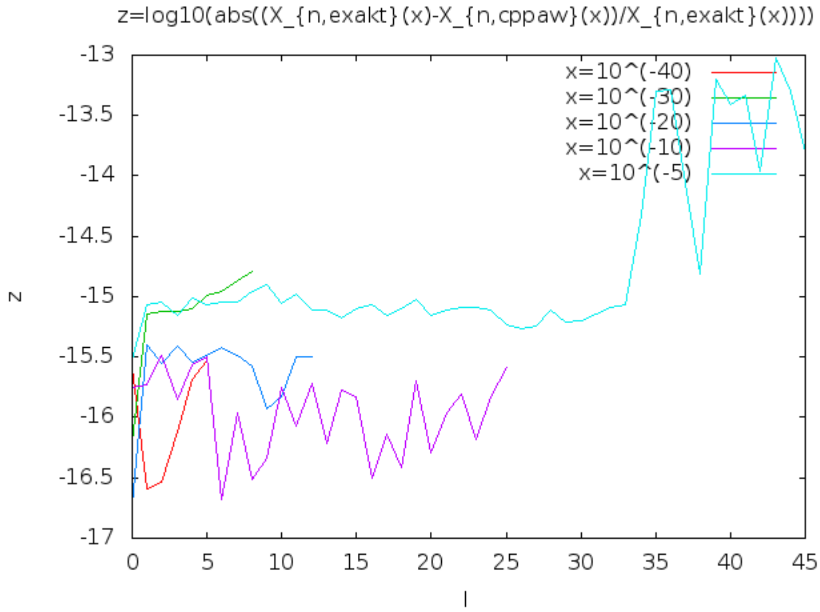
log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
100
80
60
40
20
0
l
1e-05
0.0001
0.001
0.01
0.1
1
10
100
1000
x
-12.5
-13
-13.5
-14
-14.5
-15
-15.5
-16
-16.5



z=log10(abs((X_{n,exakt}(x)-X_{n,cppaw}(x))/X_{n,exakt}(x)))
x=10^(-40)
x=10^(-30)
x=10^(-20)
x=10^(-10)
x=10^(-5)
-13
-13.5
-14
-14.5
-15
-15.5
-16
-16.5
-17
z
0
5
10
15
20
25
30
35
40
45
l

4.7.2 Derivative

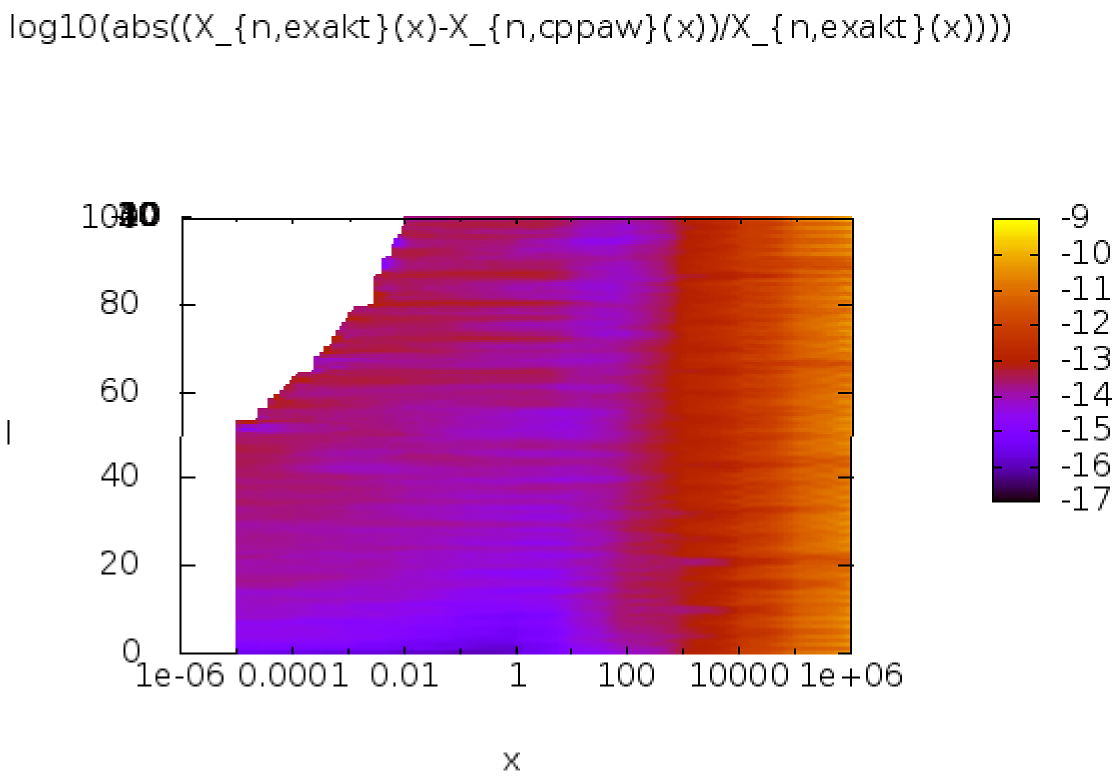




4.8 SPECIALFUNCTION$BESSEL

- uses DBESJ from slatec for $J_n(x)$ $(n \geq 0)$
- $j_l(x) = \sqrt{\frac{\pi}{2x}} J_{l+\frac{1}{2}}(x)$
- DBESJ has internal overflow and undeflow detection
- for arguments in the white area above $l > 55$ and small arguments the routine DBESJ throws an error because of a double underflow (values smaller than $< 10^{-308}$)

4.8.1 Value



4.9 SPECIALFUNCTION$ERF

- $erf(x)$ uses fortran intrinsic ERF

4.9.1 Value

note: $erf(x)$ has no l-parameter