

Bessel Functions in other CRAN Packages

Martin Maechler
R Core Development Team
maechler@stat.math.ethz.ch

February 2009 (*typeset on March 21, 2012*)

Abstract

Why do I write yet another R package, when R itself has Bessel functions and several CRAN packages also have versions of these?

Loading C code of R package 'Rmpfr': GMP using 64 bits per limb

1 Introduction

R itself has had the function `besselI()`, `besselJ()`, `besselK()` and `besselY()`, from very early on.

However, they had shown deficiencies: First, they did only work for real (`double`) but not for *complex* arguments, even though the Bessel functions are well-defined on the whole complex plane. Second, for $x \approx 1500$ and larger, `besselI(x, nu, expon.scaled=TRUE)` jumped to zero, as I found, because of an overflow in the backward recursion (via difference equation), which I found elegantly to resolve (by re-scaling), for R2.9.0. However, the algorithm complexity is proportional to $\lfloor x \rfloor$, and for large x , a better algorithm has been desired for years. Hence, I had started experimenting with the two asymptotic expansions from Abramowitz and Stegun (1970).

The following R packages on CRAN (as of Jan.29, 2009) also provide Bessel functions:

gsl

fAsianOptions

QRMLib Uses many **gsl** C functions in its own code; or, rather, seems to have copy-pasted large parts of **gsl** in its own 'src/' directory

2 gsl

The R package **gsl** by Robin Hankin provides an R interface on a function-by-function basis to much of the GSL, the GNU Scientific Library. You get a first overview with

```
> library(gsl)
> ?bessel_Knu
```

What can I say ...

- only real 'x', not complex
- For fractional nu, the (only) interesting functions are

```

bessel_Inu      (nu, x, give=FALSE, strict=TRUE)
bessel_Inu_scaled(nu, x, give=FALSE, strict=TRUE)
bessel_Jnu      (nu, x, give=FALSE, strict=TRUE)
bessel_Jnu_scaled(nu, x, give=FALSE, strict=TRUE)
bessel_Knu      (nu, x, give=FALSE, strict=TRUE)
bessel_Knu_scaled(nu, x, give=FALSE, strict=TRUE)
bessel_Ynu      (nu, x, give=FALSE, strict=TRUE)
bessel_Ynu_scaled(nu, x, give=FALSE, strict=TRUE)

```

where the `*_scaled()` version of each corresponds to our functions `expon.scaled=TRUE`.

- `bessel_Inu_scaled()` works for large `x`, comparably to our `BesselI(.)` which give warnings about accuracy loss here :

```

> x <- (1:500)*50000; b2 <- BesselI(x, pi, expo=TRUE)
> b1 <- bessel_Inu_scaled(pi, x)
> all.equal(b1,b2,tol=0) ## "Mean relative difference: 1.544395e-12"

[1] "Mean relative difference: 1.849828e-12"

> ## the accuracy is *as* limited (probably):
> b1 <- bessel_Inu_scaled(pi, x, give=TRUE)
> summary(b1$err)

      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
8.299e-08 9.580e-08 1.173e-07 1.606e-07 1.655e-07 1.856e-06

```

where the GSL (info) manual says that `err` is an *absolute* error estimate, hence for *relative* error estimates, we look at

```

> range(b1$err/ b1$val)

[1] 0.001040159 0.001040161

```

So, we see that either the error estimate is too conservative, or the results only have 3 digit accuracy.

3 Session Info

```

> toLatex(sessionInfo())

• R version 2.15.0 beta (2012-03-20 r58793), x86_64-unknown-linux-gnu

• Locale: LC_CTYPE=de_CH.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C,
  LC_MONETARY=en_US.UTF-8, LC_MESSAGES=de_CH.UTF-8, LC_PAPER=C, LC_NAME=C, LC_ADDRESS=C,
  LC_TELEPHONE=C, LC_MEASUREMENT=de_CH.UTF-8, LC_IDENTIFICATION=C

• Base packages: base, datasets, grDevices, graphics, methods, stats, utils

• Other packages: Bessel 0.5-4, Rmpfr 0.4-7, gmp 0.5-1, gsl 1.9-9

• Loaded via a namespace (and not attached): tools 2.15.0

```

References

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, N. Y., 1970.