1 Introduction

The `expm` package provides an R function `expm` to compute the matrix exponential of a real, square matrix. The matrix exponential of a matrix $A$ is defined as

$$e^A = \mathbf{I} + A + \frac{A^2}{2!} + \ldots = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$ 

The actual computations are done in C by a function of the same name that is callable by other packages. Therefore, package authors can use these functions and avoid duplication of efforts.

2 Description of the functions

The R function `expm` takes as argument a real, square matrix and returns its exponential. Dimension names are preserved:

```r
> library(expm)
> m <- matrix(c(4, 1, 1, 2, 4, 1, 0, 1, 4), 3, 3)
> expm(m)
> dimnames(m) <- list(letters[1:3], LETTERS[1:3])
> m
```

```r
[,1] [,2] [,3]
[1,] 147.8666 183.7651 71.79703
[2,] 127.7811 183.7651 91.88257
[3,] 127.7811 163.6796 111.96811
```

```r
> dimnames(m) <- list(letters[1:3], LETTERS[1:3])
> m
```
The actual computational work is done in C by a routine defined as

\[
\text{void expm(double *x, int n, double *z)}
\]

where \(x\) is the vector underlying the \(R\) matrix and \(n\) is the number of lines (or columns) of the matrix. The matrix exponential is returned in \(z\). The routine uses the algorithm of Ward (1977) based on diagonal Padé table approximations in conjunction with three step preconditioning. The Padé approximation to \(e^A\) is

\[
e^A \approx R(A),
\]

with

\[
R_{pq}(A) = (D_{pq}(A))^{-1}N_{pq}(A)
\]

where

\[
D_{pq}(A) = \sum_{j=1}^{p} \frac{(p + q - j)!p!}{(p + q)!j!(p - j)!} A^j
\]

and

\[
N_{pq}(A) = \sum_{j=1}^{q} \frac{(p + q - j)!q!}{(p + q)!j!(q - j)!} A^j.
\]

See Moler and Van Loan (1978) for an exhaustive treatment of the subject.

The C routine is based on a translation made by \(?\) of the implementation of the corresponding Octave function (Eaton, 2002).
3 Calling the functions from other packages

Package authors can use facilities from `expm` in two (possibly simultaneous) ways:

1. call the R level function `expm` in R code;
2. if matrix exponential calculations are needed in C, call the routine `expm`.

Using R level function `expm` in a package simply requires the following two import directives:

```r
Imports: expm
```

in file `DESCRIPTION` and

```r
import(expm)
```

in file `NAMESPACE`.

Accessing the C level routine further requires to prototype `expm` and to retrieve its pointer in the package initialization function `R_init__pkg`, where `pkg` is the name of the package:

```c
void (*expm)(double *x, int n, double *z);

void R_init_pkg(DllInfo *dll)
{
    expm = (void (*)(double, int, double)) R_GetCCallable("expm", "expm");
}
```

The definitive reference for these matters remains the *Writing R Extensions* manual.

References


