Jordan algebras in R

Robin K. S. Hankin

Jordan algebras

A *Jordan algebra* is a non-associative algebra over the reals with a bilinear multiplication that satisfies the following identities:

$$xy = yx$$

$$(xy)(xx) = x(y(xx))$$

(the second identity is known as the Jordan identity). In literature, multiplication is usually indicated by juxtaposition but one sometimes sees $x \circ y$. Package idiom is to use an asterisk, as in `x*y`. There are five types of Jordan algebras:

- Real symmetric matrices, class `real_symmetric_matrix`, abbreviated in the package to `rsm`
- Complex Hermitian matrices, class `complex_herm_matrix`, abbreviated to `chm`
- Quaternionic Hermitian matrices, class `quaternion_herm_matrix`, abbreviated to `qhm`
- Albert algebras, the space of 3×3 Hermitian octonionic matrices, class `albert`
- Spin factors, class `spin`

(of course, the first two are special cases of the next). The `jordan` package provides functionality to manipulate jordan objects using natural R idiom. Objects of all these classes are stored in matrix form with columns being elements of the jordan algebra. The first four classes are matrix-based in the sense that the algebraic objects are symmetric or Hermitian matrices (the S4 class is `jordan_matrix`). The fifth class, spin factors, is not matrix based.

Matrix-based Jordan algebras

The four matrix-based Jordan algebras have elements which are square matrices. The first three classes are real (symmetric) matrices, complex (Hermitian) matrices, and quaternionic (Hermitian) These all behave in the same way from a package idiom perspective. Consider:

```
x <- rrsm()  # "Random Real Symmetric Matrix"
y <- rrsm()
z <- rrsm()
x
```

```
## Vector of real symmetric matrices with entries
##         [1]   [2]   [3]
##  [1,] -0.14  0.36 -1.61
##  [2,] -0.32  1.20  0.40
##  [3,]  1.29 -0.07  2.53
##  [4,] -0.82 -0.58 -0.23
##  [5,]  0.02  0.99  0.56
## ......................
```

```
## [11,]  0.26 -0.73  0.23
## [12,]  1.91 -0.27 -0.19
## [13,] -0.75  0.01  0.20
## [14,]  0.78  1.77  1.07
## [15,] -1.51  1.09 -0.16
```

Object `x` is a three-element vector, with each element corresponding to a $x \times 5$ symmetric matrix (because `rrsm()` has `d=5` by default, specifying the size of the matrix). Thus each element has $5*(5+1)/2 = 15$ degrees of freedom, as indicated by the row labelling. Addition and multiplication of a Jordan object with a scalar are defined:

```
x*100
```

```
## Vector of real symmetric matrices with entries
##        [1] [2]  [3]
##  [1,]  -14  36 -161
##  [2,]  -32 120   40
##  [3,]  129  -7  253
##  [4,]  -82 -58  -23
##  [5,]    2  99   56
## ...................
## [11,]   26 -73   23
## [12,]  191 -27  -19
## [13,]  -75   1   20
## [14,]   78 177  107
## [15,] -151 109  -16
```

```
x + y*3
```

```
## Vector of real symmetric matrices with entries
##         [1]   [2]   [3]
##  [1,] -6.14  0.84  1.27
##  [2,]  0.94  0.27 -5.03
##  [3,]  3.48 -4.03 -1.46
##  [4,] -2.26  5.72 -0.38
##  [5,]  6.89  1.29 -1.96
## .......................
## [11,] -1.30 -0.64 -0.94
## [12,]  0.11 -2.91 -0.43
## [13,] -2.07  1.09 -1.63
## [14,] -0.03  0.81  0.11
## [15,] -5.23  0.07  0.38
```

```
x + 100
```

```
## Vector of real symmetric matrices with entries
##          [1]    [2]    [3]
##  [1,]  99.86 100.36  98.39
##  [2,]  99.68 101.20 100.40
##  [3,] 101.29  99.93 102.53
##  [4,]  99.18  99.42  99.77
##  [5,] 100.02 100.99 100.56
## ..........................
## [11,] 100.26  99.27 100.23
## [12,] 101.91  99.73  99.81
## [13,]  99.25 100.01 100.20
## [14,] 100.78 101.77 101.07
```

```
## [15,]  98.49 101.09  99.84
```

(the last line is motivated by analogy with `M + x`, for `M` a matrix and `x` a scalar). Jordan objects may be multiplied using the rule $x \circ y = (xy + yx)/2$:

```
x*y
```

```
## Vector of real symmetric matrices with entries
##             [1]      [2]      [3]
##  [1,]  0.95500 -1.43700 -3.31680
##  [2,] -0.78720  0.94370 -2.21795
##  [3,] -0.51910 -3.14070 -4.87870
##  [4,]  0.87060  1.59905 -1.33845
##  [5,]  1.91690 -0.14530 -0.01870
## ...............................
## [11,]  1.15750 -0.07805  1.33450
## [12,] -1.36480 -1.60330  0.77455
## [13,]  2.42455 -0.30930  1.19285
## [14,] -2.18455 -0.44865  0.48600
## [15,]  0.71060 -0.71770 -0.56770
```

We may verify that the distributive law is obeyed:

```
x*(y+z) - (x*y + x*z)
```

```
## Vector of real symmetric matrices with entries
##                  [1]           [2]           [3]
##  [1,]  0.000000e+00  0.000000e+00  2.220446e-16
##  [2,] -1.110223e-16  0.000000e+00 -8.881784e-16
##  [3,]  0.000000e+00  5.551115e-17  0.000000e+00
##  [4,] -2.220446e-16  0.000000e+00  0.000000e+00
##  [5,] -8.881784e-16  0.000000e+00 -6.661338e-16
## ...............................................
## [11,]  0.000000e+00 -2.220446e-16 -1.387779e-16
## [12,] -3.885781e-16  0.000000e+00  1.110223e-16
## [13,]  0.000000e+00 -5.551115e-17  3.330669e-16
## [14,] -2.220446e-16 -1.110223e-16  0.000000e+00
## [15,]  0.000000e+00 -2.220446e-16  1.110223e-16
```

Further, we may observe that the resulting algebra is not associative:

```
LHS <- x*(y*z)
RHS <- (x*y)*z
LHS-RHS
```

```
## Vector of real symmetric matrices with entries
##               [1]         [2]        [3]
##  [1,]  0.9296130 -0.93112450  6.1084250
##  [2,] -1.3778915  1.18377850  5.4380920
##  [3,] -4.5292620  1.57170150  1.7479775
##  [4,] -0.4012045  0.51619625 -1.1421862
##  [5,] -0.8066505 -1.35590250 -1.0021397
## ......................................
## [11,] -1.2442917 -2.22589600 -1.0637272
## [12,] -1.3330803 -0.05856825 -0.8128517
## [13,] -2.8670260  0.36045525 -0.2632850
## [14,]  0.3951660 -2.67255125  1.1186615
## [15,]  2.1164525 -1.64049200 -0.2514230
```

However, the Jordan identity $(xy)(xx) = x(y(xx))$ is satisfied:

```
LHS <- (x*y)*(x*x)
RHS <- x*(y*(x*x))
LHS-RHS
```

```
## Vector of real symmetric matrices with entries
##                  [1]           [2]           [3]
##  [1,]  6.661338e-16 -1.110223e-15  0.000000e+00
##  [2,]  0.000000e+00  1.776357e-15  3.552714e-15
##  [3,] -1.595946e-15 -3.552714e-15  0.000000e+00
##  [4,]  2.220446e-15  0.000000e+00  8.881784e-16
##  [5,] -3.552714e-15  1.776357e-15 -1.776357e-15
## ..............................................
## [11,]  8.881784e-16 -8.881784e-16  0.000000e+00
## [12,] -3.552714e-15  0.000000e+00 -8.881784e-16
## [13,]  0.000000e+00  8.881784e-16  0.000000e+00
## [14,] -1.776357e-15  0.000000e+00  0.000000e+00
## [15,]  3.552714e-15  1.776357e-15 -4.440892e-16
```

(the entries are zero to numerical precision). If we wish to work with the matrix itself, a single element may be coerced with `as.1matrix()`:

```
M1 <- as.1matrix(x[1])
(M2 <- as.1matrix(x[2]))
```

```
##       [,1]  [,2]  [,3] [,4]  [,5]
## [1,]  0.36  1.20 -0.58 0.23 -0.73
## [2,]  1.20 -0.07  0.99 2.09 -0.27
## [3,] -0.58  0.99  0.58 1.70  0.01
## [4,]  0.23  2.09  1.70 0.77  1.77
## [5,] -0.73 -0.27  0.01 1.77  1.09
```

(in the above, observe how the matrix is indeed symmetric). We may verify that the multiplication rule is indeed being correctly applied:

```
(M1 %*% M2 + M2 %*% M1)/2 - as.1matrix(x[1]*x[2])
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

It is also possible to verify that symmetry is closed under the Jordan operation:

```
jj <- as.1matrix(x[1]*x[2])
jj-t(jj)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

The other matrix-based Jordan algebras are similar but differ in the details of the coercion. Taking quaternionic matrices:

```
as.1matrix(rqhm(n=1,d=2))
```

```
##     [1,1] [2,1] [1,2] [2,2]
## Re -0.64 -0.14 -0.14  0.19
## i   0.00 -0.07  0.07  0.00
## j   0.00 -1.07  1.07  0.00
## k   0.00 -1.35  1.35  0.00
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

above we see the matrix functionality of the `onion` package being used. See how the matrix is Hermitian (elements `[1,2]` and `[2,1]` are conjugate; elements `[1,1]` and `[2,2]` are pure real). Verifying the Jordan identity would be almost the same as above:

```
x <- rqhm()
y <- rqhm()
(x*y)*(x*x) - x*(y*(x*x))
```

```
## Vector of quaternionic Hermitian matrices with entries
##                  [1]           [2]           [3]
##  [1,] -7.105427e-15 -1.421085e-14  7.105427e-15
##  [2,] -7.105427e-15  1.776357e-15  7.105427e-15
##  [3,]  3.552714e-15  0.000000e+00  7.105427e-15
##  [4,] -1.243450e-14 -5.329071e-15  0.000000e+00
##  [5,]  0.000000e+00  8.881784e-15  1.953993e-14
## ..............................................
## [41,] -4.440892e-15  3.552714e-15  0.000000e+00
## [42,]  1.421085e-14  7.105427e-15 -8.881784e-16
## [43,] -7.105427e-15  7.105427e-15  0.000000e+00
## [44,]  0.000000e+00  0.000000e+00  0.000000e+00
## [45,]  0.000000e+00  0.000000e+00 -1.421085e-14
```

Spin factors

The fifth class is slightly different from a package idiom perspective. Mathematically, elements are of the form $\mathbb{R}\oplus\mathbb{R}^n$, with addition and multiplication defined by

$$\alpha(a,\mathbf{a}) = (\alpha a, \alpha\mathbf{a})$$

$$(a,\mathbf{a}) + (b,\mathbf{b}) = (a+b, \mathbf{a}+\mathbf{b})$$

$$(a,\mathbf{a}) \times (b,\mathbf{b}) = (ab + \langle\mathbf{a},\mathbf{b}\rangle, a\mathbf{b} + b\mathbf{a})$$

where $a,b\in\mathbb{R}$, and $\mathbf{a},\mathbf{b}\in\mathbb{R}^n$. Here $\langle\cdot,\cdot\rangle$ is an inner product defined on $\mathbb{R}^n$ (by default we have $\langle(x_1,\ldots,x_n),(y_1,\ldots,y_n)\rangle = \sum x_i y_i$ but this is configurable in the package).

So if we have $\mathcal{I},\mathcal{J},\mathcal{K}$ spin factor elements it is clear that $\mathcal{I}\mathcal{J}=\mathcal{J}\mathcal{I}$ and $\mathcal{I}(\mathcal{J}+\mathcal{K}) = \mathcal{I}\mathcal{J}+\mathcal{I}\mathcal{K}$. The Jordan identity is not as easy to see but we may verify all the identities numerically:

```
I <- rspin()
J <- rspin()
K <- rspin()
I
```

```
## Vector of spin objects with entries
##       [1]   [2]   [3]
##  r   1.32 -1.01  0.21
## [1]  0.33  1.63 -1.13
## [2]  0.56 -0.75 -0.49
## [3]  1.22  0.98 -0.40
## [4]  2.83 -0.91  0.58
## [5] -1.07  0.76  0.32
```

```
I*J - J*I   # commutative:
```

```
## Vector of spin objects with entries
##     [1] [2] [3]
##  r    0   0   0
## [1]   0   0   0
## [2]   0   0   0
## [3]   0   0   0
## [4]   0   0   0
## [5]   0   0   0
```

```
I*(J+K) - (I*J + I*K)  # distributive:
```

```
## Vector of spin objects with entries
##               [1]           [2]           [3]
##  r  -8.881784e-16  1.776357e-15 -2.220446e-16
## [1]  0.000000e+00 -8.881784e-16  4.440892e-16
## [2]  4.440892e-16  2.220446e-16  0.000000e+00
## [3] -1.110223e-16 -1.110223e-16  0.000000e+00
## [4]  0.000000e+00  2.220446e-16 -1.110223e-16
## [5] -1.110223e-16 -2.775558e-16 -2.220446e-16
```

```
I*(J*K) - (I*J)*K  # not associative:
```

```
## Vector of spin objects with entries
##               [1]           [2]       [3]
##  r   2.775558e-16 -1.776357e-15  0.000000
## [1] -1.736220e-01  1.671900e-02 -3.280523
## [2] -1.293880e+00 -2.657379e+00 -1.420755
## [3]  1.082492e+00  1.191294e+00 -1.166128
## [4]  3.101635e+00 -1.885407e+00  1.623246
## [5] -1.352639e+00  4.818600e-01  0.949435
```

```
(I*J)*(I*I) - I*(J*(I*I))  # Obeys the Jordan identity
```

```
## Vector of spin objects with entries
##               [1]           [2]           [3]
##  r   0.000000e+00  7.105427e-15 -2.220446e-16
## [1]  0.000000e+00 -7.105427e-15  4.440892e-16
## [2]  0.000000e+00 -3.552714e-15  0.000000e+00
## [3] -3.552714e-15  0.000000e+00  0.000000e+00
## [4]  0.000000e+00  0.000000e+00 -3.330669e-16
## [5]  0.000000e+00  1.776357e-15  2.220446e-16
```

Albert algebras

Class 4 corresponds to 3×3 Hermitian matrices with octonions for entries. This is class `albert` in the package:

```r
x <- ralbert()
y <- ralbert()
x
```

```
## Vector of Albert matrices with entries
##         [,1]  [,2]  [,3]
##     d1  1.27 -0.13 -0.71
##     d2 -1.76  1.08 -0.39
##     d3 -0.38  0.39 -1.07
## Re(o1)  1.56  1.64  0.92
##  i(o1)  0.31  0.70 -0.04
##  j(o1) -0.15 -0.46  0.56
##  k(o1) -0.11 -2.04 -0.99
##  l(o1) -0.69  1.28  0.96
## il(o1) -1.04 -0.59  0.15
## jl(o1) -0.56 -1.22 -1.56
## kl(o1)  0.02  0.21  0.10
## Re(o2) -0.11  1.06 -0.94
##  i(o2) -0.71 -0.08 -0.59
##  j(o2)  1.84 -0.31  0.82
##  k(o2)  1.07 -1.41  0.94
##  l(o2)  0.95  0.95  0.40
## il(o2)  0.01  1.28 -0.28
## jl(o2)  0.28 -1.27  0.78
## kl(o2)  0.81  1.44 -0.63
## Re(o3)  0.43 -0.77  1.16
##  i(o3) -0.78 -0.71 -0.31
##  j(o3) -0.06 -0.10 -0.77
##  k(o3)  1.69  1.07 -2.01
##  l(o3)  0.58  0.26 -0.21
## il(o3)  0.05  0.68 -0.70
## jl(o3)  2.37 -0.96  1.16
## kl(o3) -1.97 -1.07  0.26
```

```r
(x*y)*(x*x)-x*(y*(x*x)) # Jordan identity:
```

```
## Vector of Albert matrices with entries
##                 [,1]          [,2]          [,3]
##     d1  0.000000e+00 -1.421085e-14  1.776357e-15
##     d2  0.000000e+00  0.000000e+00  1.421085e-14
##     d3  0.000000e+00  3.730349e-14  7.105427e-15
## Re(o1)  3.552714e-15  1.421085e-14 -5.329071e-15
##  i(o1)  0.000000e+00 -5.329071e-15  7.105427e-15
##  j(o1)  6.661338e-16  2.131628e-14 -7.105427e-15
##  k(o1)  1.421085e-14  7.105427e-15  7.105427e-15
##  l(o1)  0.000000e+00  0.000000e+00 -3.552714e-15
## il(o1)  0.000000e+00  3.552714e-15 -3.552714e-15
## jl(o1)  7.105427e-15  3.552714e-15 -8.881784e-16
## kl(o1) -1.243450e-14  8.881784e-15  3.552714e-15
## Re(o2) -3.552714e-15  7.105427e-15 -1.421085e-14
##  i(o2) -3.552714e-15 -7.105427e-15 -2.131628e-14
##  j(o2) -7.105427e-15  1.776357e-15  8.437695e-15
##  k(o2) -3.108624e-15  1.421085e-14  0.000000e+00
##  l(o2) -7.105427e-15 -7.105427e-15  0.000000e+00
## il(o2) -5.329071e-15  0.000000e+00  3.552714e-15
```

```
## jl(o2)  3.552714e-15 -1.421085e-14  3.552714e-15
## kl(o2)  3.552714e-15 -4.440892e-15  0.000000e+00
## Re(o3) -1.953993e-14  2.486900e-14 -7.105427e-15
##  i(o3)  3.552714e-15 -2.486900e-14  2.664535e-15
##  j(o3) -8.881784e-16 -7.105427e-15 -3.552714e-15
##  k(o3)  2.842171e-14  0.000000e+00 -1.776357e-15
##  l(o3)  3.552714e-15 -7.105427e-15 -7.105427e-15
## il(o3) -3.552714e-15  1.421085e-14 -1.776357e-15
## jl(o3)  0.000000e+00  0.000000e+00  0.000000e+00
## kl(o3)  0.000000e+00 -4.263256e-14  7.993606e-15
```

Special identities

In 1963, C. M. Glennie discovered a pair of identities satisfied by special Jordan algebras but not the Albert algebra. Defining

$$U_x(y) = 2x(xy) - (xx)y$$

$$\{x, y, z\} = 2(x(yz) + (xy)z - (xz)y)$$

(it can be shown that Jacobson's identity $U_{U_x(y)} = U_x U_y U_x$ holds), Glennie's identities are

$$H_8(x, y, z) = H_8(y, x, z) \qquad H_9(x, y, z) = H_9(y, x, z)$$

(see McCrimmon 2004 for details), where

$$H_8(x, y, z) = \{U_x U_y(z), z, xy\} - U_x U_y U_z(xy)$$

and

$$H_9(x, y, z) = 2U_x(z)U_{y,x}U_z(yy) - U_x U_z U_{x,y} U_y(z)$$

Numerical verification of Jacobson

We may verify Jacobson's identity:

```
U <- function(x){function(y){2*x*(x*y)-(x*x)*y}}
diff <- function(x,y,z){
     LHS <- U(x)(U(y)(U(x)(z)))
     RHS <- U(U(x)(y))(z)
     return(LHS-RHS)  # zero if Jacobson holds
}
```

Then we may numerically verify Jacobson for type 3-5 Jordan algebras:

```
diff(ralbert(),ralbert(),ralbert())  # Albert algebra obeys Jacobson:
```

```
## Vector of Albert matrices with entries
##                 [,1]          [,2]          [,3]
##     d1  6.821210e-13  3.637979e-12 -9.094947e-13
##     d2  2.273737e-13  0.000000e+00  0.000000e+00
##     d3  9.094947e-13  3.637979e-12  3.637979e-12
## Re(o1)  2.273737e-12  3.637979e-12  1.136868e-12
##  i(o1)  0.000000e+00 -3.637979e-12  1.136868e-12
```

```
##  j(o1)  0.000000e+00 -7.275958e-12  0.000000e+00
##  k(o1)  0.000000e+00  2.728484e-12  7.958079e-13
##  l(o1) -1.250555e-12 -1.818989e-12 -1.136868e-12
## il(o1) -4.547474e-13 -3.637979e-12 -2.273737e-13
## jl(o1)  1.364242e-12 -9.094947e-12  4.547474e-13
## kl(o1) -2.273737e-13 -3.637979e-12 -4.547474e-13
## Re(o2)  9.094947e-13 -3.637979e-12  4.547474e-13
##  i(o2) -2.273737e-13 -1.818989e-12  1.818989e-12
##  j(o2)  0.000000e+00  3.637979e-12  2.728484e-12
##  k(o2) -6.821210e-13 -2.728484e-12  1.136868e-12
##  l(o2)  3.637979e-12  1.818989e-12 -4.547474e-13
## il(o2) -9.094947e-13 -5.456968e-12 -2.273737e-12
## jl(o2)  5.684342e-13 -3.637979e-12 -2.273737e-13
## kl(o2)  1.364242e-12  7.730705e-12  1.250555e-12
## Re(o3)  9.094947e-13 -3.637979e-12  4.547474e-13
##  i(o3)  9.094947e-13 -1.818989e-12 -9.663381e-13
##  j(o3) -2.501110e-12 -3.637979e-12  0.000000e+00
##  k(o3)  9.094947e-13 -3.637979e-12  1.136868e-13
##  l(o3) -4.547474e-13  1.818989e-12  0.000000e+00
## il(o3) -1.364242e-12 -9.094947e-13 -9.094947e-13
## jl(o3)  1.818989e-12  1.818989e-12 -1.136868e-13
## kl(o3)  0.000000e+00 -5.456968e-12  2.728484e-12
```

```
diff(rqhm(),rqhm(),rqhm()) # Quaternion Jordan algebra obeys Jacobson:
```

```
## Vector of quaternionic Hermitian matrices with entries
##                  [1]           [2]           [3]
##  [1,]  9.094947e-13 -1.136868e-13  4.092726e-12
##  [2,]  2.728484e-12 -4.547474e-13 -1.818989e-12
##  [3,]  1.364242e-12 -4.547474e-13 -4.547474e-13
##  [4,]  0.000000e+00 -9.094947e-13 -4.547474e-13
##  [5,]  1.421085e-12 -9.094947e-13 -5.684342e-13
## ..............................................
## [41,]  4.547474e-13 -9.094947e-13  1.818989e-12
## [42,] -5.684342e-13  9.094947e-13  1.023182e-12
## [43,] -6.821210e-13  0.000000e+00  4.547474e-13
## [44,]  4.547474e-13  1.818989e-12  7.389644e-13
## [45,]  4.547474e-13  4.547474e-13  6.821210e-13
```

```
diff(rspin(),rspin(),rspin()) # spin factors obey Jacobson:
```

```
## Vector of spin objects with entries
##               [1]           [2]           [3]
##  r  -1.598721e-14  0.000000e+00 -1.210143e-14
## [1] -1.687539e-14 -2.273737e-13 -1.132427e-14
## [2] -9.059420e-14  6.821210e-13  3.108624e-15
## [3]  9.769963e-15  1.136868e-13 -8.881784e-16
## [4] -1.287859e-14 -1.563194e-13 -4.440892e-16
## [5] -6.306067e-14  5.684342e-14 -3.996803e-15
```

showing agreement to numerical accuracy (the output is close to zero). We can now verify Glennie's G_8 and G_9 identities.

Numerical verification of G_8

```
B <- function(x,y,z){2*(x*(y*z) + (x*y)*z - (x*z)*y)}  # bracket function
H8 <- function(x,y,z){B(U(x)(U(y)(z)),z,x*y) - U(x)(U(y)(U(z)(x*y)))}
G8 <- function(x,y,z){H8(x,y,z)-H8(y,x,z)}
```

and so we verify for type 3 and type 5 Jordans:

```
G8(rqhm(1),rqhm(1),rqhm(1))   # Quaternion Jordan algebra obeys G8:
```

```
## Vector of quaternionic Hermitian matrices with entries
##                 [1]
##  [1,]  6.366463e-12
##  [2,]  2.000888e-11
##  [3,] -1.818989e-11
##  [4,]  1.273293e-11
##  [5,]  1.273293e-11
## ...................
## [41,] -5.456968e-12
## [42,]  2.273737e-12
## [43,]  7.275958e-12
## [44,]  5.456968e-12
## [45,] -7.275958e-12
```

```
G8(rspin(1),rspin(1),rspin(1)) # Spin factors obey G8:
```

```
## Vector of spin objects with entries
##                 a
##  r   1.421085e-14
## [1]  7.105427e-15
## [2] -4.263256e-14
## [3] -1.776357e-14
## [4]  7.993606e-15
## [5] -7.105427e-15
```

again showing acceptable accuracy. The identity is *not* true for Albert algebras:

```
G8(ralbert(1),ralbert(1),ralbert(1)) # Albert algebra does not obey G8:
```

```
## Vector of Albert matrices with entries
##                [,1]
##     d1  19181.86172
##     d2 -12498.90857
##     d3  -6682.95315
## Re(o1)   -894.12105
##  i(o1) -15600.74289
##  j(o1)  16368.40446
##  k(o1)   2978.55713
##  l(o1)  -3828.19761
## il(o1)    -70.32181
## jl(o1)   7120.96624
## kl(o1)    973.34363
## Re(o2)  -2204.71438
##  i(o2)  13622.47416
##  j(o2)  18705.01997
##  k(o2)   2237.64834
##  l(o2)   4866.95255
```

```
## il(o2)  -8178.38760
## jl(o2)    252.58353
## kl(o2)  11472.72923
## Re(o3)   6108.13213
##  i(o3)   -740.22288
##  j(o3) -25739.57101
##  k(o3)   -188.59056
##  l(o3)   3727.52169
## il(o3)  37155.50197
## jl(o3)  -2220.85825
## kl(o3)  -6710.73819
```

Numerical verification of G_9

```
L <- function(x){function(y){x*y}}
U <- function(x){function(y){2*x*(x*y)-(x*x)*y}}
U2 <- function(x,y){function(z){L(x)(L(y)(z)) + L(y)(L(x)(z)) - L(x*y)(z)}}
H9 <- function(x,y,z){2*U(x)(z)*U2(y,x)(U(z)(y*y)) - U(x)(U(z)(U2(x,y)(U(y)(z))))}
G9 <- function(x,y,z){H9(x,y,z)-H9(y,x,z)}
```

Then we may verify the `G9()` identity for type 3 Jordans:

```
G9(rqhm(1),rqhm(1),rqhm(1))  # Quaternion Jordan algebra obeys G9:
```

```
## Vector of quaternionic Hermitian matrices with entries
##                 [1]
##  [1,] -1.164153e-10
##  [2,] -5.820766e-11
##  [3,] -1.164153e-10
##  [4,]  1.309672e-10
##  [5,] -5.093170e-11
## ...................
## [41,] -2.910383e-11
## [42,]  7.275958e-11
## [43,]  1.164153e-10
## [44,]  0.000000e+00
## [45,] -1.746230e-10
```

However, the Albert algebra does not satisfy the identity:

```
G9(ralbert(1),ralbert(1),ralbert(1)) # Albert algebra does not obey G9:
```

```
## Vector of Albert matrices with entries
##              [,1]
##     d1 -48992.063
##     d2 -71656.716
##     d3 -92390.794
## Re(o1)  21412.902
##  i(o1)  48793.736
##  j(o1) -12607.997
##  k(o1)   5559.046
##  l(o1)  90028.783
## il(o1)  -1557.534
## jl(o1)  34927.493
## kl(o1)    174.108
## Re(o2) -41396.590
```

```
##   i(o2) -34192.319
##   j(o2)   8281.492
##   k(o2)   6578.010
##   l(o2)  19380.850
##  il(o2) -17627.009
##  jl(o2)  25565.357
##  kl(o2) -32539.914
##  Re(o3) -29256.400
##   i(o3) -24614.816
##   j(o3) -18266.454
##   k(o3)  11670.589
##   l(o3) -25797.009
##  il(o3) -38376.574
##  jl(o3) -24264.600
##  kl(o3)   2369.255
```