MATRAN

A Fortran 95 Wrapper for Matrix Operations

G. W. Stewart

Department of Computer Science
University of Maryland

Mathematical and Computational Sciences Division
NIST

Helpers: John Reid, Bill Mitchell, Che Rung Lee

WHAT MATRAN IS

- MATRAN (pronounced MAY-tran) is an open wrapper written in Fortran 95 that implements matrix operations using LAPACK and the BLAS. It also computes common matrix functions and decompositions.

- MATRAN currently has two derived types.
  - The `Rmat` implementing a general rectangular matrix.
  - The `Rdiag` implementing a diagonal matrix.
  - The paucity of types is misleading.
  - The final wrapper will include types `Cmat` and `Cdiag`.

- MATRAN can be compiled into single or double precision versions.
  - It would be easy to produce a quad version.

WHAT MATRAN IS II

MATRAN computes common matrix operations.

- The usual operations: `A + B`; `A.xhy.B` $\equiv A^{\mathrm{T}}B$.
- Joining and Bordering: `BorderE(A, B)` $\equiv (A\ \ B)$.

- MATRAN implements a number of matrix functions and constructors.
  - Norms, identity matrices, random matrices.
- MATRAN is conservative in storage allocation.
  - It attempts to reuse storage.
  - Subroutines corresponding to the matrix operations are provided to reduce temporary matrix objects in expressions.
  - Unfortunately, the user must perform some explicit deallocations.
- MATRAN computes decompositions
  - LU, Cholesky, QR, QRP, Spectral, SVD, Schur, and Eig.
- MATRAN is modularized at a fine-grained level.

WHAT MATRAN IS NOT

- MATRAN is not object oriented — at least in the pure sense of the term.
- Nothing is private. This has two advantages.
  - The programmer can use the resources Fortran 95 to manipulate matrices in ways not provided by MATRAN.
  - MATRAN can be distributed as a library of object code. Yet people can still extend it.
- Operations are not associated with a particular object.
  - They are implemented by suites; e.g, the Product suite and the Solve suite.
  - This is a natural way to organize binary operations between different types.
- On the other hand, MATRAN makes full use of the (static) polymorphism available in Fortran 95.
  - It has some ad-hoc dynamic polymorphism.

GENERALITIES ON MATRIX TYPES

- In MATRAN a matrix type represents a storage class.
  - The `Rmat` represents a real matrix stored in a rectangular array.
  - The `Rdiag` represents a real diagonal matrix with its diagonal stored in a linear array.
- The array can be bigger than the matrix.
  - In an assignment or other operations that change a `Rmat` (or `Rdiag`), MATRAN attempts to use available storage.
  - When a matrix is built up by bordering, on can initialize the array to the maximum size and avoid a sequence of reallocations.
- A matrix is fully represented in its array.
  - A symmetric matrix has both its upper and lower parts present.
  - The array elements outside the matrix are zero.
- A packed symmetric matrix would be a new MATRAN type.

THE TYPE RMAT

```
type Rmat
   real(wp), pointer &                  ! The matrix array
           :: a(:,:) => null()          !
   integer :: nrow = 0                  ! Number of rows in the matrix
   integer :: ncol = 0                  ! Number of columns in the matrix
   integer :: narow = 0                 ! Number of rows in the array
   integer :: nacol = 0                 ! Number of columns in the array
   character(2) &                       ! Type of matrix
           :: tag = 'GE'                !
   logical :: adjustable =.true.        ! Adjustable array
   integer, pointer                     ! Intermediate value
           :: temporary => null()       !
end type Rmat
```

- The array containing the matrix is a pointer array.
- The name is short to make element access readable. Compare `X%a(i,j)` with `X%array(i,j)`.
- It can be larger than the matrix. Hence, `nrow`, `narow`, `ncol`, and `nacol`.

THE RMAT TAG

```
character(2) :: tag = 'GE'  ! Type of matrix
```

- The tag field specifies what kind of matrix the type contains.

`GE`	General	`HE`	Hermitian
`UT`	Upper triangular	`HP`	Hermitian positive definite
`LT`	Lower trangular		

- Triangular matrices (`UT, LT`) can be rectangular.

- Positive (semi) definite matrices may not be so owing to rounding error.

ADJUSTABLE AND TEMPORARY RMATS

```
logical :: adjustable =.true.      ! Adjustable array
integer, pointer                   ! Intermediate value
        :: temporary => null()
```

- A `Rmat` is adjustable if its array can be resized.
  - If a new matrix will not fit in the current array and the `Rmat` is not adjustable, an error occurs.
- A `temporary` `Rmat` is one occurring in the course of evaluating an expression (including a function evaluation).
  - The deallocation of `temporary` `Rmats` is done by `GuardTemp` and `CleanTemp`.
  - Any matrix object returned by a function must be marked temporary.

ASSIGNMENT

- The = operator is overloaded in four ways.

`Rmat A = Rmat B`	`B` is copied to `A`.
`Rmat A = Array B(:,:)`	The array `B` is copied to the `A%a`.
`Rmat A = (/m,n, ma,na/)`	`A` becomes an `m` x `n` zero matrix in an `ma` x `na` array.
`Rmat A = real(wp) s`	`A` becomes a 1 x 1 matrix whose element is `s`.

- Various natural things happen to `tag`, `temporary`, etc.

- These assignments have corresponding operator forms.
  - For example, `.rm.(/m,n, ma,na/)`

.

GENERALITIES ON OPERATIONS

- MATRAN implements matrix operations by overloading existing operators (`A*B`) or defining new ones (`A.xhy.B`).

- The matrix types in MATRAN are aranged in a hierarchy.
  - Each type is responsible for implementing operations with itself and lower members of the hierarchy.
  - The length of operator suites grows as one goes up in the hierarchy.

- An operation like `A*B` creates a temporary. An alternative is to use subroutine forms provided by MATRAN.
  - For example, `call Times(C, A, B)`.
  - One should use the subroutines only if truly necessary.

THE PRODUCT SUITE

Operation	Operator	Subroutine
$C = sA$	`C = s*A`	`call Times(C, s, A)`
$C = As$	`C = A*s`	`call Times(C, s, A)`
$C = AB$	`C = A*B`	`call Times(C, A, B)`
$C = A^{\mathrm{H}}B$	`C = A.xhy.B`	`call TimesXhy(C, A, B)`
$C = AB^{\mathrm{H}}$	`C = A.xyh.B`	`call TimesXyh(C, A, B)`
$C = A^{\mathrm{H}}A$	`C = .xhx.A`	`call TimesXhx(C, A)`
$C = AA^{\mathrm{H}}$	`C = .xxh.A`	`call TimesXxh(C, A)`

- In the above `s` is a scalar.
- The operations `s*A`, `A*s`, and `A*B` are defined for any combinations of `Rmats` and `Rdiags`.
- The operations `A.xhy.B`, `A.xyh.B`, `.xhx.A`, and `.xxh.A` are defined for `Rmats` only.

THE SOLVE SUITE

Operation	Operator	Subroutine
$C = A/s$	`C = A/s`	`call Solve(C, A, s)`
$C = A^{-1}B$	`C = A.xiy.B`	`call SolveXiy(C, A, B)`
$C = A^{-\mathrm{H}}B$	`C = A.xihy.B`	`call SolveXihy(C, A, B)`
$C = AB^{-1}$	`C = A.xyi.B`	`call SolveXyi(C, A, B)`
$C = AB^{-\mathrm{H}}$	`C = A.xyih.B`	`call SolveXyih(C, A, B)`

- The operations are performed by solving linear systems.
- The suite uses the `tag` field to determine how to solver the system.
  - `LT` and `UT` use BLAS routines.
  - The others compute a decomposition first.
  - In the subroutine form, the decomposition can be returned via an optional argument for reuse.

MATRIX OPERATIONS

Transpose	A^{H}, A^{T}
Sum	$A + B$, $A - B$, $-A$
Product	σA, AB, $A^{\mathrm{T}}B$, ...
Solve	$A^{-1}B$, AB^{-1}, $A^{-\mathrm{T}}B$, ...
Join	$(A \ \ B)$, $\begin{pmatrix} A \\ B \end{pmatrix}$
Border	$A = (A \ \ B)$, $A = (B \ \ A)$, ...
Submatrix	$A(i_1{:}i_2, j_1{:}j_2)$, $A(:, j)$, ...

- Join concatenates two `Rmat`s.
  - `A.jwe.B` gives `[A, B]`.
  - `A.jns.B` gives `[A; B]`.
- Border has three sided bordering.
  - `BorderSE(A, S, E, SE)` gives `A = [A, E; S, SE]`.

MATRIX MISCELLANEA

Diag	Extracts diagonals
Eye	Identity matrices
Inverse	Computes inverses
Norm	1-, ∞-, and Frobenius norms
Norm2	2-norm
Pivot	Performs interchanges
Print	Prints `Rmats`
Rand	Random Matrices

- This class of suites is small now, but watch it grow.

GENERALITIES ABOUT DECOMPOSITIONS

- A decomposition is a factorization of a matrix into simpler matrices.
- In MATRAN a decomposition is contained in a derived type and is computed by a constructor.

Decomposition	Container	Constructor
LU with partial pivoting	`RmatLudpp`	`Ludpp`
Cholesky decomposition	`RmatChol`	`Chol`
QR decomposition	`RmatQR`	`QR`
QR decomposition with pivoting	`RmatQRP`	`QRP`
Spectral decomposition	`RmatSpec`	`Spec`
Singular value decomposition	`RmatSVD`	`SVD`
Real Schur decomposition	`RmatRSchur`	`Schur`
Eigendecomposition	`RmatEig`	`Eig`

- Decompositions are computed by LAPACK.
- The calling sequence for constructors is

```
call <constructor>(<container>, <matrix>, <opt args>)
```

THE SPECTRAL DECOMPOSITION

```
type RmatSpec
   type(Rdiag) :: D          ! Contains the eigenvalues
   type(Rmat)  :: V          ! Contains the eigenvectors
   logical     :: companion  ! True if the decomposition is
                             ! associated with a matrix of interest
end type RmatSpec
```

—————◇—————

- Given a symmetric matrix A, there is an orthogonal matrix V and a diagonal matrix D such that $A = VDV^{\mathrm{T}}$.
- The type `RmatSpec` is the container for this decomposition.
- The flag `companion` is used to signal whether or not the matrix from which the decomposition was computed has changed.
  - If it is true, the decomposition can be reused, otherwise the decomposition must be recomputed.
  - The recomputation must be initiated by the programmer.

THE CONSTRUCTOR

- The constructor for `RmatSpec` has the calling sequence

```
call Spec(S, A, wantv, info, mywork)
```

- `A` is the `Rmat` is the matrix and `S` is a `RmatSpec` containing it spectral decomposition.

- `wantv` is true if `V` is wanted.

- `info`, if present, contains the `info` parameter from the LAPACK routine `dgeev`.

- The routine `dgeev` requires a work array. If `mywork` is present, `Spec` uses it rather than allocating the space internally.

ENVOI

- MATRAN has been released and is available at

  `http://www.cs.umd.edu/~stewart/`

- Coding with MATRAN is akin to coding in a matrix oriented language (Matlab, Octave, etc.)
  - A significant difference is that there is no workspace.

- We are working on the types (`Cmat` and `Cdiag`).

- If you use MATRAN, please share your experience with me, while the package is still somewhat fluid.