A Proposal for Fortran 90 BLAS

Jeremy Du Croz
NAG Ltd, Oxford
Email: jeremy@nag.co.uk

Introduction

Proposal for a set of Fortran 90 dense BLAS

Straightforward interface to allow ready incorporation in higher level software

LAPACK Working note # 101 describes a Fortran 90 interface to LAPACK

Fortran 90 Features

- Generic interfaces
- Assumed shape arrays
- Optional arguments
- Modules

Level 1 BLAS

Many Level 1 BLAS operations can be handled naturally in Fortran 90, so only a small set of Level 1 BLAS is proposed.

Level 2 & 3

Combined a number of Level 2 & 3 BLAS into a generic interface. E.g. `SYMV` and `SYMM`; `GEMV`, `GEMM` and `GER1`

Have kept band and packed routines separate. Introduced complex symmetric Level 2 BLAS

Introduced consistency between Level 2 & 3 BLAS

$$A \leftarrow \alpha x y^T + A \quad \text{becomes} \quad C \leftarrow \alpha a b^T + \beta C$$

Some character arguments specifying options (e.g. `transa`) are optional, and the arguments are ordered as:

algebraic, band width, character, optional

BLAS v. MATMUL

$$C \leftarrow \alpha AB + \beta C$$

```
C = alpha*MATMUL( A, B ) + beta*C
```

or

```
CALL GEMM( alpha, A, B, beta, C )
```

$$C \leftarrow \alpha A^H B^H + \beta C$$

```
C = alpha*MATMUL( CONJG( TRANSPOSE(A) ),  &
                  CONJG( TRANSPOSE(B) ) ) + beta*C
```

or

```
CALL GEMM( alpha, A, B, beta, C      &
           transa='C', transb='C' )
```

BLAS v. MATMUL (continued)

- Efficiency?
  - arithmetic
  - storage
- Triangular, symmetric and band matrices? Packed storage?

Interfaces

```
subroutine gemm( alpha, a, b, beta, c [, transa, transb] )
```

rank of `a`	rank of `b`	rank of `c`	`transa`	`transb`	operation	F77 BLAS
2	2	2	'N'	'N'	$C \leftarrow \alpha AB + \beta C$	`_GEMM`
2	2	2	'N'	'T'	$C \leftarrow \alpha AB^T + \beta C$	`_GEMM`
2	2	2	'T'	'N'	$C \leftarrow \alpha A^T B + \beta C$	`_GEMM`
2	2	2	'T'	'T'	$C \leftarrow \alpha A^T B^T + \beta C$	`_GEMM`
2	1	1	'N'		$c \leftarrow \alpha Ab + \beta c$	`_GEMV`
2	1	1	'T'		$c \leftarrow \alpha A^T b + \beta c$	`_GEMV`
1	1	2			$C \leftarrow \alpha ab^T + \beta C$	`_GER_`

Interfaces (continued)

```
subroutine syrk( alpha, a, beta, c, uplo [, transa] )
subroutine herk( alpha, a, beta, c, uplo [, transa] )
```

rank of `a`	`transa`	operation	F77 BLAS
2	'N'	$C \leftarrow \alpha AA^T + \beta C$	_SYRK
2	'T'	$C \leftarrow \alpha A^T A + \beta C$	_SYRK
1		$C \leftarrow \alpha a a^T + \beta C$	_SYR1

where C is a symmetric matrix.

Generic Names

nrm2, swap, rotg, rot

gemm		gbmv
hemm, symm	hpmv, spmv	hbmv, sbmv
herk, syrk	hpr1, spr1	
her2k, syr2k	hpr2, spr2	
trmm	tpmv	tbmv
trsm	tpsv	tbsv

Extra Functionality

- Householder generation
- $C \leftarrow \alpha ADA^T + \beta C$ (D diagonal, block diagonal, tridiagonal?)
- $C \leftarrow \alpha AB + \beta C$ (updating triangular part only)

Error Handling

Fortran 77 BLAS type error handling, with additional checks:

- invalid value
- invalid shape
- inconsistent shapes

What Software Exists?

- Fortran 90 model implementation
- Fortran 77 interface
- Real test software
- To come – complex test software, timing software

Questions to the Community

- Longer names?
- Make `uplo` and `side` optional?
- Packed storage in same generic interface as conventional storage?
- Level 3 routines for packed storage?
- Level 3 routines for band matrices?