

Una introduzione a

**LAPACK**

# Cos'è **LAPACK**

?

● **LAPACK** è una libreria di **routines** scritta in FORTRAN 77.

● Tali **routines** risolvono i più comuni problemi di Calcolo Matriciale

**infatti**

● **LAPACK** è un acronimo per **Linear Algebra PACKage**

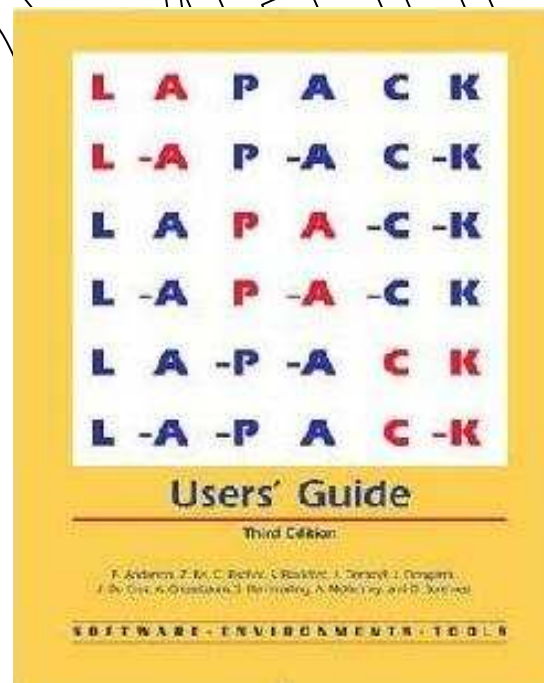
Il pacchetto **LAPACK** è disponibile sul **web** all'indirizzo

[www.netlib.org/lapack/lug/index.html](http://www.netlib.org/lapack/lug/index.html)

Altre informazioni su **LAPACK** sono nel manuale

**LAPACK Users' Guide - SIAM 1999**

A



# Quali problemi può risolvere

## LAPACK

?

**LAPACK** può risolvere

- sistemi lineari di equazioni
- problemi di minimi quadrati
- calcolo degli autovalori
- problemi ai valori singolari

Inoltre **LAPACK** può occuparsi anche di

- fattorizzazioni di matrici
- stima dell'indice di condizionamento

# Classificazione delle **Routines** di **LAPACK**

In **LAPACK** ci sono **tre livelli** di **Routines**

- **driver routines**
- **computational routines**
- **auxiliary routines**

A.Murli

# DRIVER ROUTINES

**Risolvono un problema “completo”.**

- **risolvono un sistema di equazioni lineari**
- **calcolano gli autovalori di una matrice reale simmetrica**
- ...
- ...

**Se per una dato problema esiste una apposita  
DRIVER ROUTINE è consigliato utilizzarla.**

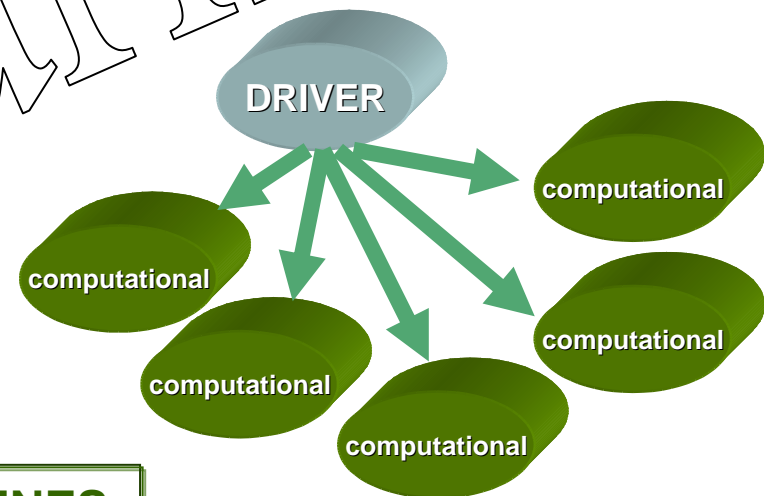
# COMPUTATIONAL ROUTINES

Eseguono un distinto “nucleo computazionale”.

- calcolo della fattorizzazione LU
- riduzione di una matrice reale simmetrica alla forma tridiagonale
- ...
- ...

Ogni DRIVER ROUTINE chiama una sequenza di COMPUTATIONAL ROUTINES

Per alcuni problemi le COMPUTAZIONAL ROUTINES sono più indicate delle DRIVER ROUTINES



# AUXILIARY ROUTINES

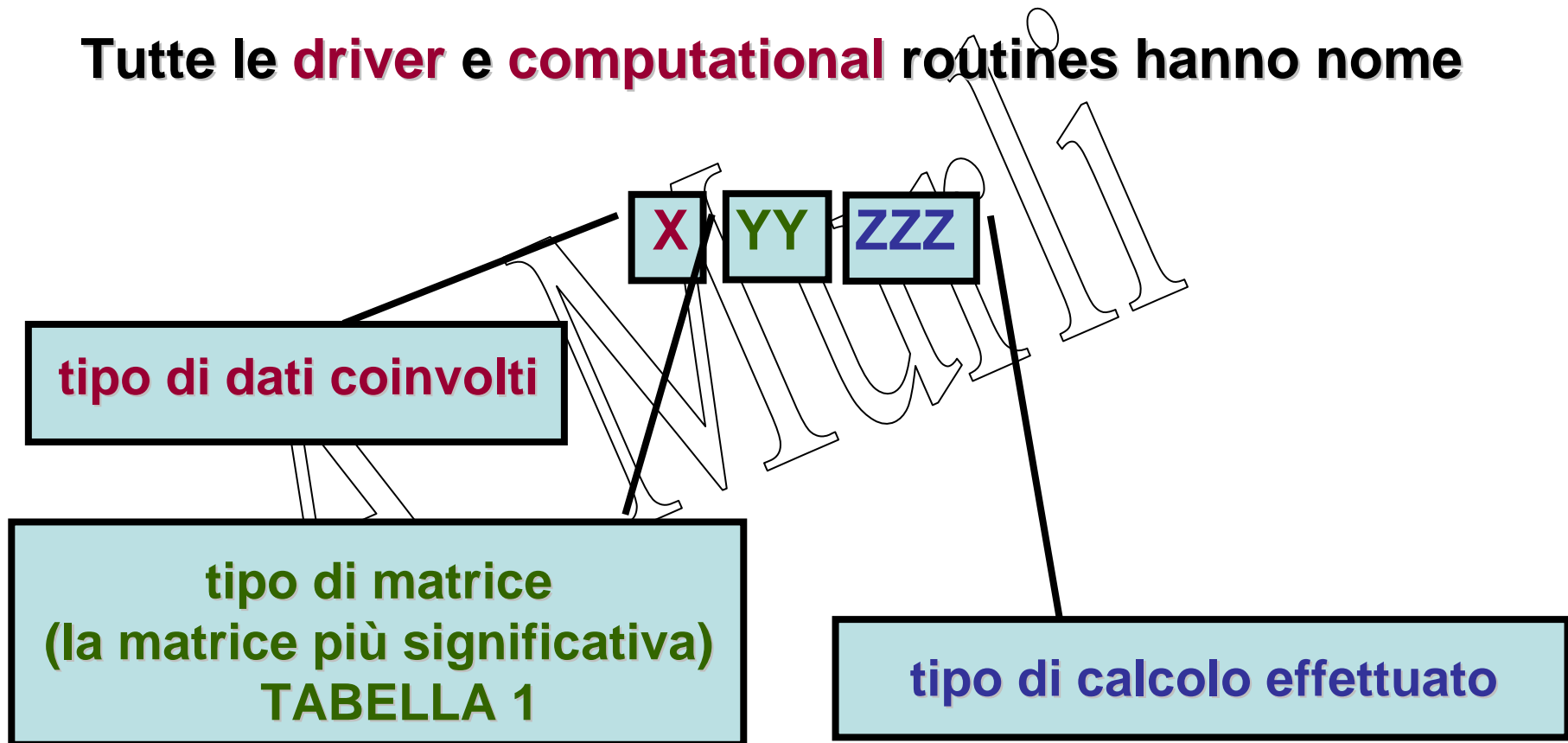
Classificabili in tre tipi

- routines che **eseguono sotto-operazioni di algoritmi a blocchi** (ad esempio lo stesso algoritmo non a blocchi)
- routines che **eseguono operazioni di basso livello** (ad esempio scalare una matrice, calcolare la norma ecc ecc)
- routines che **sono estensioni delle routines di BLAS** (ad esempio routines che calcolano il prodotto matrice vettore con matrice complessa simmetrica)



Schema di denominazione delle  
**Routines** di **LAPACK**

Tutte le **driver** e **computational** routines hanno nome



## tipo di dati coinvolti (X)

**S** real  
**D** double precision  
**C** complex  
**Z** complex\*16 o double complex

le routines lavorano sia con  
dati reali che con dati complessi

infatti

**LAPACK** fornisce lo stesso “range di funzionalità”  
sia per dati reali che per dati complessi

# tipo di matrice (YY)

## tabella 1

BD	bidiagonal
GB	general band
GE	general (i.e., unsymmetric, in some cases rectangular)
GG	general matrices, generalized problem (i.e., a pair of general matrices) ( <i>not implemented in Release 1.0</i> )
GT	general tridiagonal
HB	(complex) Hermitian band
HE	(complex) Hermitian
HG	upper Hessenberg matrix, generalized problem (i.e a Hessenberg and a triangular matrix) ( <i>not implemented in Release 1.0</i> )
HP	(complex) Hermitian, packed storage
HS	upper Hessenberg
OP	(real) orthogonal, packed storage
OR	(real) orthogonal

# tipo di matrice (YY)

## tabella 1 (segue)

PB	symmetric or Hermitian positive definite band
PO	symmetric or Hermitian positive definite
PP	symmetric or Hermitian positive definite, packed storage
PT	symmetric or Hermitian positive definite tridiagonal
SB	(real) symmetric band
SP	symmetric, packed storage
ST	(real) symmetric tridiagonal
SY	symmetric
TB	triangular band
TG	triangular matrices, generalized problem (i.e., a pair of triangular matrices) ( <i>not implemented in Release 1.0</i> )
TP	triangular, packed storage
TR	triangular (or in some cases quasi-triangular)
TZ	trapezoidal
UN	(complex) unitary
UP	(complex) unitary, packed storage

# tipo di calcolo effettuato (ZZZ)

(dato il tipo di dati **x** ed il tipo di matrice **yy**)

## DRIVER ROUTINES

**xyySVX**

risolve un sistema lineare

**xyySV**

risolve un sistema lineare

**xyyLS**

risolve minimi quadrati utilizzando fattorizzazioni QR o LQ

**xyyLSS**

risolve minimi quadrati utilizzando SVD

**xyyEVX**

determina autovalori per matrici simmetriche

...

Amministratore

# tipo di calcolo effettuato (ZZZ)

dato il tipo di dati **x** ed il tipo di matrice **yy**

## COMPUTATIONAL ROUTINES

**xyyTRF**

esegue la fattorizzazione LU

**xyyTRS**

utilizza la fattorizzazione per la Forward (Back)-Substitution

**xyyCON**

stima il reciproco dell'indice di condizionamento (Metodo di Hager)

**xyyRFS**

stima l'errore nella soluzione calcolata (ritornata da **xyyTRF**)

**xyyTRI**

utilizza la fattorizzazione (ritornata da **xyyTRF**) per calcolare  $A^{-1}$

**xyyEQU**

calcola il fattore di scala per equilibrare A  
(lo scaling non viene effettuato dalle **xyyEQU** ma ad esempio dalle auxiliary routines **xLAQyy**.)

...

**Risoluzione di sistemi lineari**

con **LAPACK**

A.

M. J. J.

## **DRIVER ROUTINES** per la **risoluzione di sistemi lineari**

- **simple driver**: nome che termina per **SV**  
risolvono il sistema  $AX=B$  fattorizzando  $A$  e sovrascrivendo  $X$  su  $B$
- **expert driver**: nome che termina per **SVX**  
risolvono il sistema  $AX=B$  fattorizzando  $A$  e sovrascrivendo  $X$  su  $B$   
inoltre possono compiere anche le seguenti funzioni:
  - risolvono il sistema  $A^tX=B$  o  $A^hX=B$
  - stimano l'indice di condizionamento di  $A$
  - raffinano le soluzioni e calcolano FEA e BEA
  - equilibrano il sistema se  $A$  è male scalata



# DRIVER ROUTINES per la risoluzione di sistemi lineari

## Tabella 2

Type of matrix and storage scheme	Operation	Single precision		Double precision	
		real	complex	real	complex
general	simple driver	SGESV	CGESV	DGESV	ZGESV
	expert driver	SGESVX	CGESVX	DGESVX	ZGESVX
general band	simple driver	SGBSV	CGBSV	DGBSV	ZGBSV
	expert driver	SGBSVX	CGBSVX	DGBSVX	ZGBSVX
general tridiagonal	simple driver	SGTSV	CGTSV	DGTSV	ZGTSV
	expert driver	SGTSVX	CGTSVX	DGTSVX	ZGTSVX
symmetric/Hermitian positive definite	simple driver	SPOSV	CPOSV	DPOSV	ZPOSV
	expert driver	SPOSVX	CPOSVX	DPOSVX	ZPOSVX
symmetric/Hermitian positive definite (packed storage)	simple driver	SPPSV	CPPSV	DPPSV	ZPPSV
	expert driver	SPPSVX	CPPSVX	DPPSVX	ZPPSVX
symmetric/Hermitian positive definite band	simple driver	SPBSV	CPBSV	DPBSV	ZPBSV
	expert driver	SPBSVX	CPBSVX	DPBSVX	ZPBSVX
symmetric/Hermitian positive definite tridiagonal	simple driver	SPTSV	CPTSV	DPTSV	ZPTSV
	expert driver	SPTSVX	CPTSVX	DPTSVX	ZPTSVX
symmetric/Hermitian indefinite	simple driver	SSYSV	CHESV	DSYSV	ZHESV
	expert driver	SSYSVX	CHESVX	DSYSVX	ZHESVX
complex symmetric	simple driver		CSYSV		ZSYSV
	expert driver		CSYSVX		ZSYSVX
symmetric/Hermitian indefinite (packed storage)	simple driver	SSPSV	CHPSV	DSPSV	ZHPSV
	expert driver	SSPSVX	CHPSVX	DSPSVX	ZHPSVX
complex symmetric (packed storage)	simple driver		CSPSV		ZSPSV
	expert driver		CSPSVX		ZSPSVX

# COMPUTATIONAL ROUTINES per la risoluzione di sistemi lineari

## Tabella 3

Type of matrix and storage scheme	Operation	Single precision		Double precision	
		real	complex	real	complex
general	factorize	SGETRF	CGETRF	DGETRF	ZGETRF
	solve using factorization	SGETRS	CGETRS	DGETRS	ZGETRS
	estimate condition number	SGECON	CGECON	DGECON	ZGECON
	error bounds for solution	SGERFS	CGERFS	DGERFS	ZGERFS
	invert using factorization	SGETRI	CGETRI	DGETRI	ZGETRI
	equilibrate	SGEQU	CGEQU	DGEEQU	ZGEEQU
general band	factorize	SGBTRF	CGBTRF	DGBTRF	ZGBTRF
	solve using factorization	SGBTRS	CGBTRS	DGBTRS	ZGBTRS
	estimate condition number	SGBCON	CGBCON	DGBCON	ZGBCON
	error bounds for solution	SGBRFS	CGBRFS	DGBRFS	ZGBRFS
	equilibrate	SGBEQU	CGBEQU	DGBEQU	ZGBEQU
	general tridiagonal	factorize	SGTTRF	CGTTRF	DGTTRF
solve using factorization		SGTTRS	CGTTRS	DGTTRS	ZGTTRS
estimate condition number		SGTCON	CGTCON	DGTCON	ZGTCON
error bounds for solution		SGTRFS	CGTRFS	DGTRFS	ZGTRFS

# COMPUTATIONAL ROUTINES per la risoluzione di sistemi lineari

## Tabella 3 (segue)

Type of matrix and storage scheme	Operation	Single precision		Double precision	
		real	complex	real	complex
symmetric/Hermitian positive definite	factorize	SPOTRF	CPOTRF	DPOTRF	ZPOTRF
	solve using factorization	SPOTRS	CPOTRS	DPOTRS	ZPOTRS
	estimate condition number	SPOCON	CPOCON	DPOCON	ZPOCON
	error bounds for solution	SPORFS	CPORFS	DPORFS	ZPORFS
	invert using factorization	SPOTRI	CPOTRI	DPOTRI	ZPOTRI
	equilibrate	SPOEQU	CPOEQU	DPOEQU	ZPOEQU
symmetric/Hermitian positive definite (packed storage)	factorize	SPPTRF	CPPTRF	DPPTRF	ZPPTRF
	solve using factorization	SPPTRS	CPPTRS	DPPTRS	ZPPTRS
	estimate condition number	SPPCON	CPPCON	DPPCON	ZPPCON
	error bounds for solution	SPPRFS	CPPRFS	DPPRFS	ZPPRFS
	invert using factorization	SPPTRI	CPPTRI	DPPTRI	ZPPTRI
	equilibrate	SPPEQU	CPPEQU	DPPEQU	ZPPEQU
symmetric/Hermitian positive definite band	factorize	SPBTRF	CPBTRF	DPBTRF	ZPBTRF
	solve using factorization	SPBTRS	CPBTRS	DPBTRS	ZPBTRS
	estimate condition number	SPBCON	CPBCON	DPBCON	ZPBCON
	error bounds for solution	SPBRFS	CPBRFS	DPBRFS	ZPBRFS
	equilibrate	SPBEQU	CPBEQU	DPBEQU	ZPBEQU
symmetric/Hermitian positive definite tridiagonal	factorize	SPTTRF	CPTTRF	DPTTRF	ZPTTRF
	solve using factorization	SPTTRS	CPTTRS	DPTTRS	ZPTTRS
	estimate condition number	SPTCON	CPTCON	DPTCON	ZPTCON
	error bounds for solution	SPTRFS	CPTRFS	DPTRFS	ZPTRFS

## Come si chiamano le routines di LAPACK in C ?

COME LE ALTRE ROUTINES, CON ALCUNI ACCORGIMENTI

- le routines di LAPACK sono scritte in F77 quindi matrici e vettori devono essere comunque passati in vettori (puntatori)
- tutte le variabili devono essere passate per indirizzo
- le routines sono chiamate utilizzando il simbolo "underscore" \_

ESEMPIO: chiamata alla routine

```
dgesv_(&n,&ncol,A,&n,IPIV,b,&n,&info);
```

**A, b** puntatori a double  
**IPIV** puntatore a int  
**n,ncol,info** int

**Come si compila un programma in C che chiama funzioni di libreria scritte in F77 ?**

**Con il comando:**

```
cc -o ESE file.c -L/pathlibreria -lnomelibreria -lg2c
```

**oppure**

```
cc -o ESE file.c /pathlibreria/nomelibreria -lg2c
```

**ESEMPIO: nel caso che si richiamano funzioni di LAPACK si compila**

**con il comando:**

```
cc -o ESE file.c -L/usr/bin -llapack -lblas -lg2c
```

**oppure**

```
cc -o ESE file.c /usr/bin/liblapack.a /usr/bin/libblas.so -lg2c
```

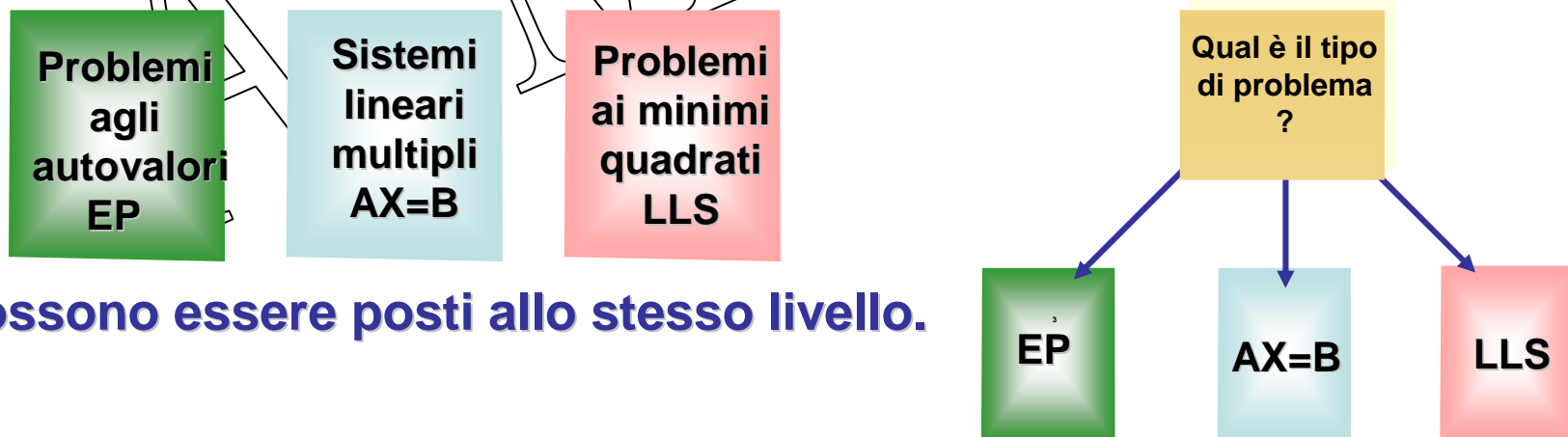
# L'albero di LAPACK (routines driver)

Le routines di **LAPACK** risolvono essenzialmente tre tipi di problema :

- ❑ Problemi agli autovalori
- ❑ Sistemi lineari multipli  $AX=B$
- ❑ Problemi ai minimi quadrati

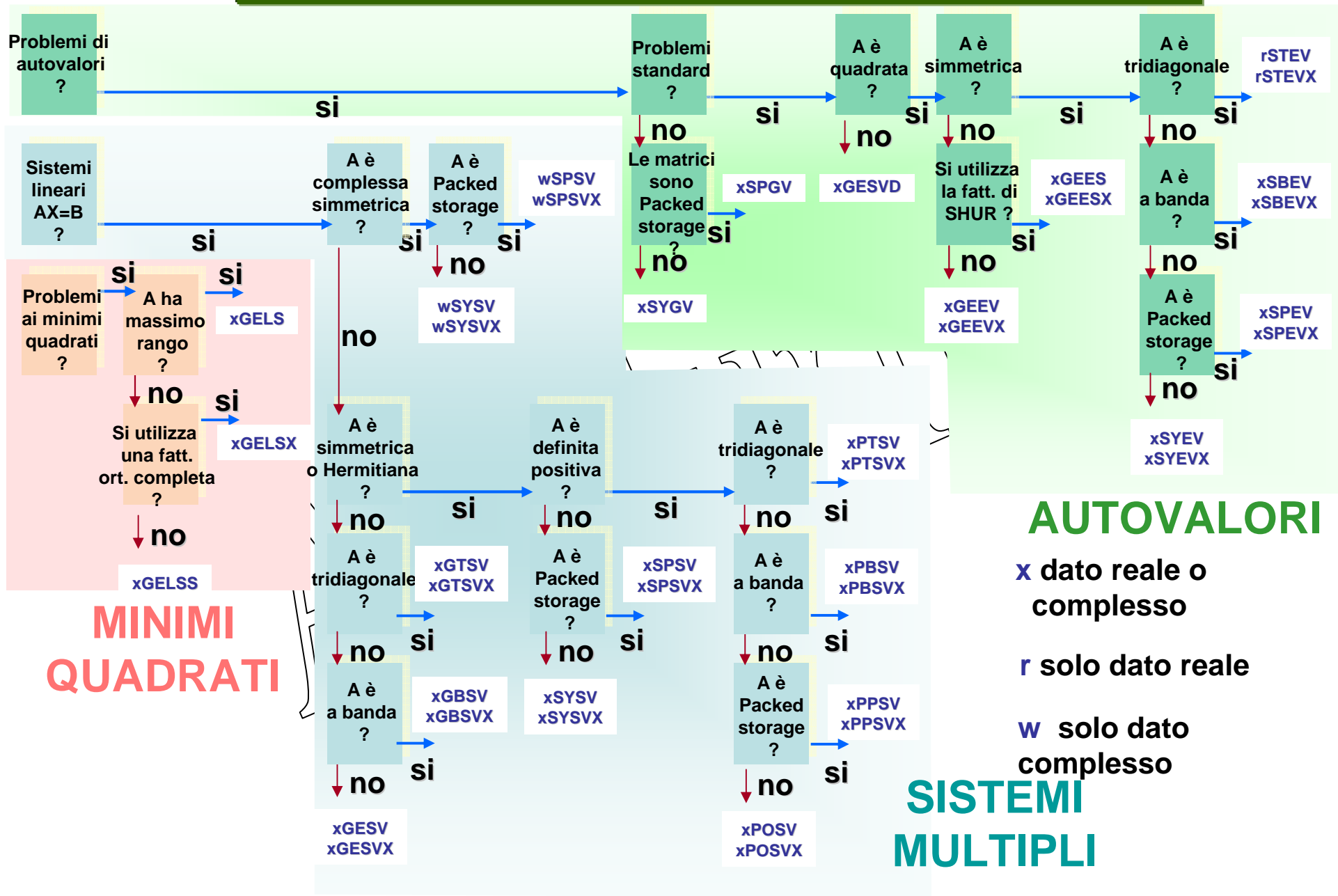
Una guida alla scelta della routine di **LAPACK** da utilizzare è fornita dall'albero decisionale

L'albero decisionale si compone quindi di tre "rami" principali



che possono essere posti allo stesso livello.

# albero delle routines driver di LAPACK (1)



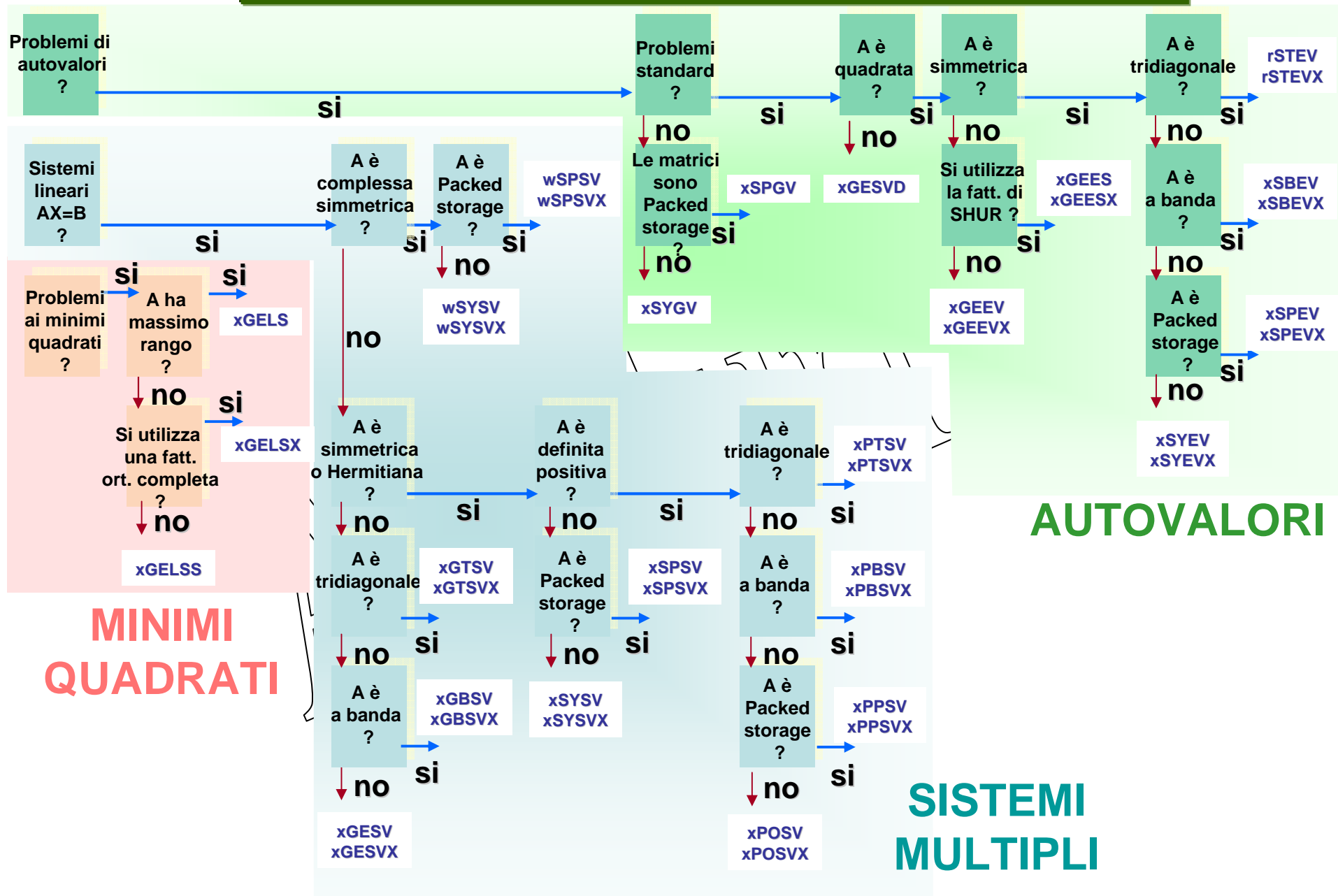
## AUTOVALORI

- x dato reale o complesso
- r solo dato reale
- w solo dato complesso

## MINIMI QUADRATI

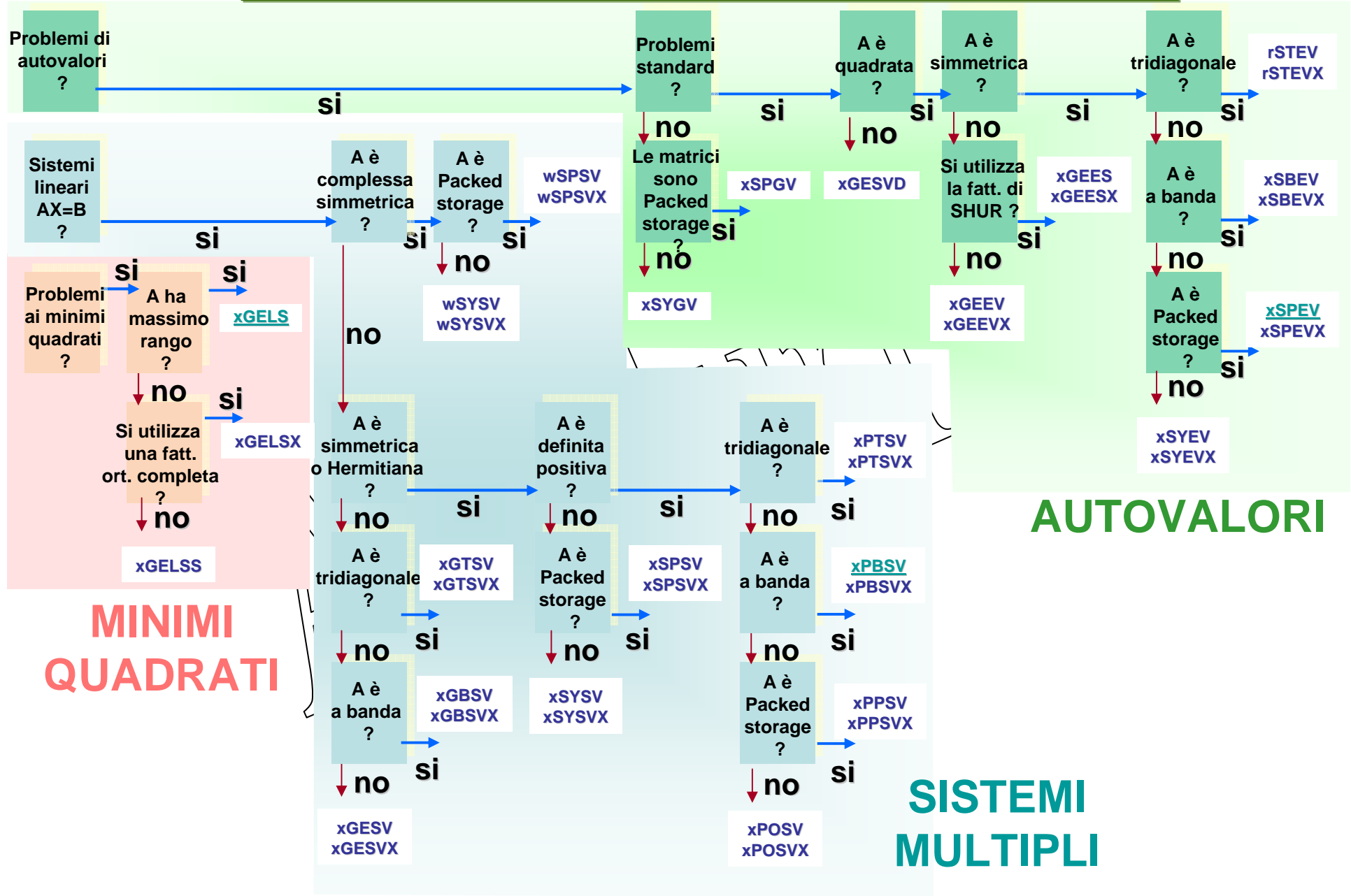
## SISTEMI MULTIPLI

# albero delle routines driver di LAPACK (2)





# albero delle routines driver di LAPACK (3)



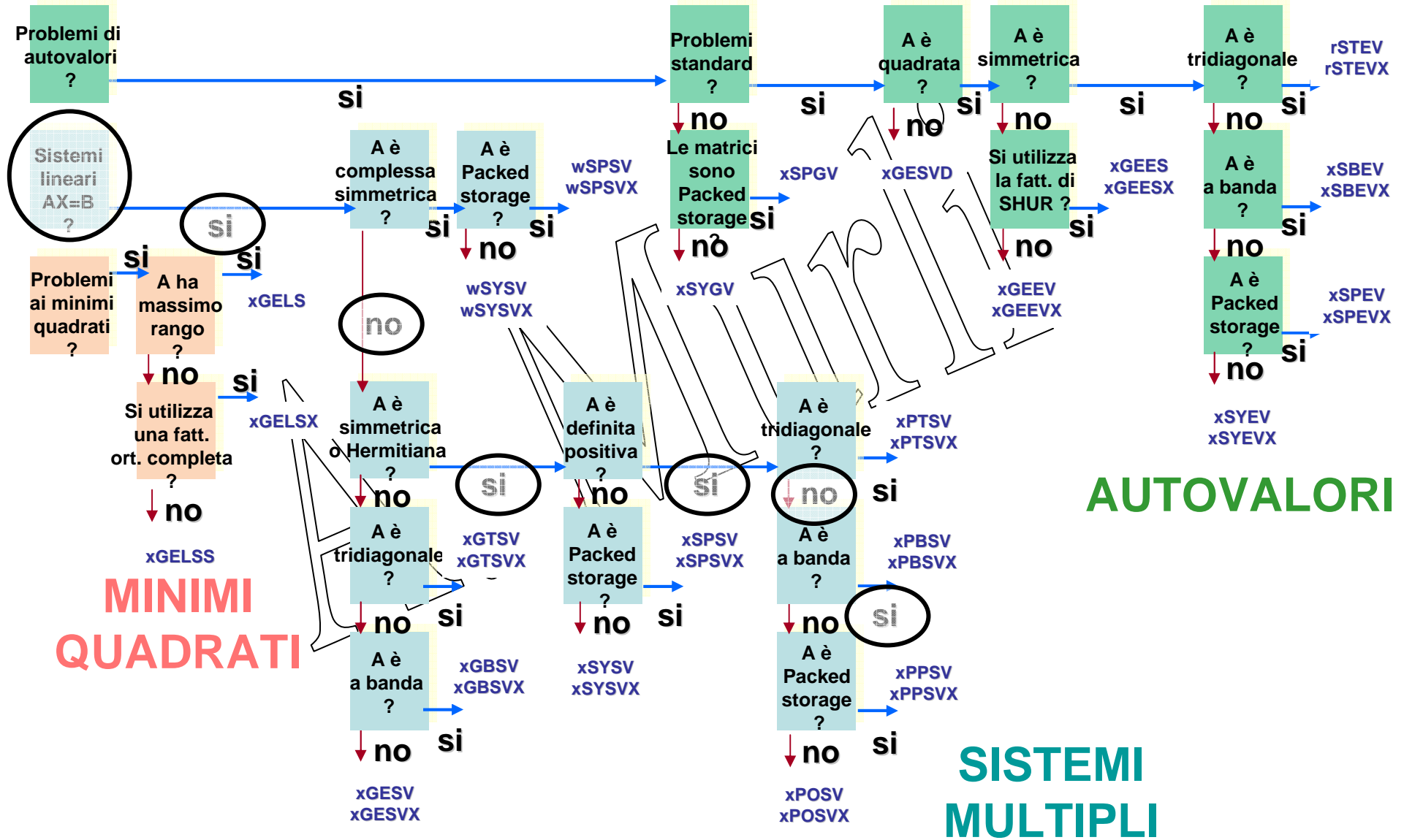
**AUTOVALORI**

**MINIMI QUADRATI**

**SISTEMI MULTIPLI**

## Esempio di utilizzo 1

Si vuole individuare la routine di **LAPACK** adatta per risolvere un sistema lineare multiplo  $AX=B$  con matrice dei coefficienti simmetrica definita positiva e a banda.



**AUTOVALORI**

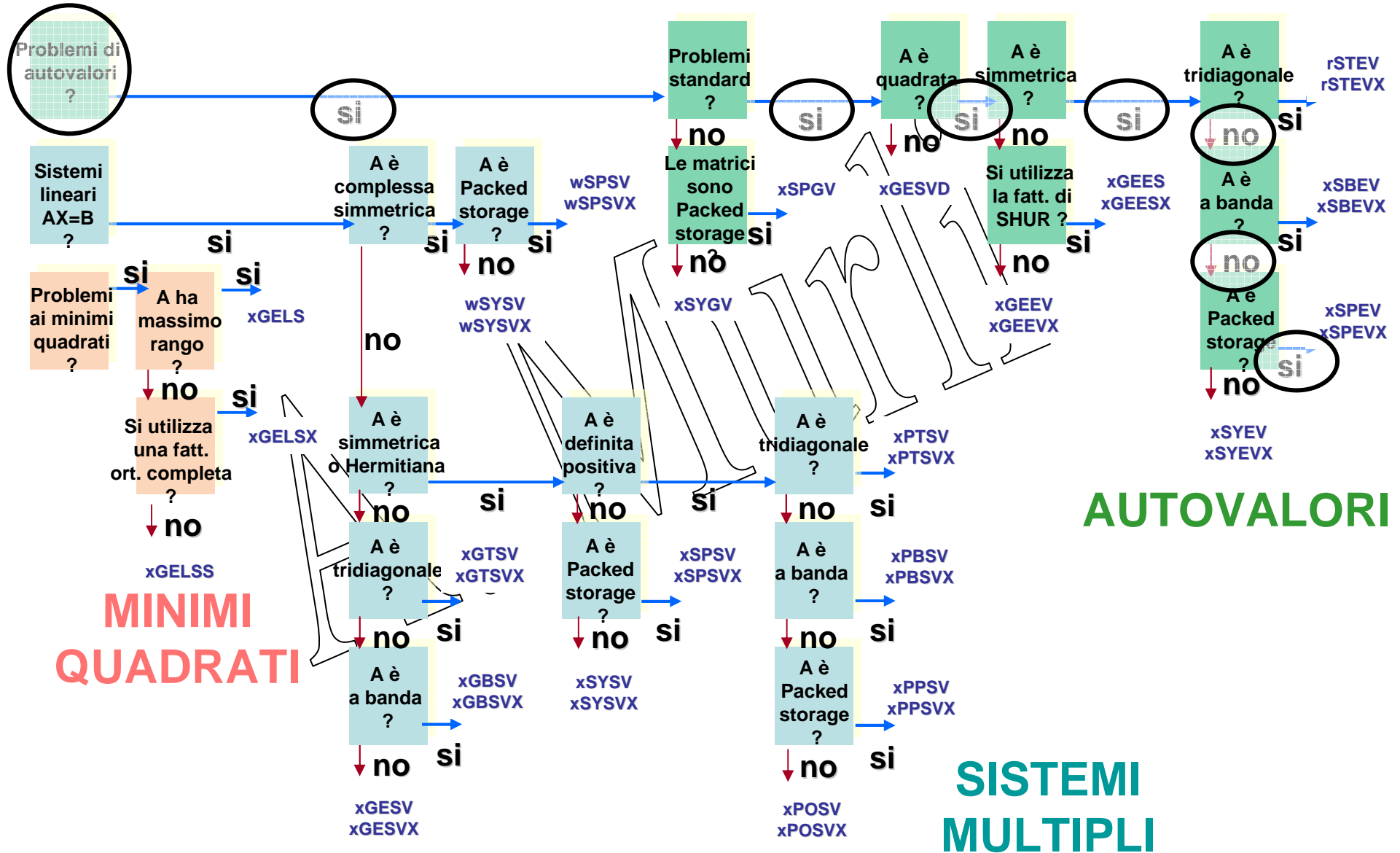
**MINIMI QUADRATI**

**SISTEMI MULTIPLI**

# albero delle routines driver di LAPACK

## Esempio di utilizzo 2

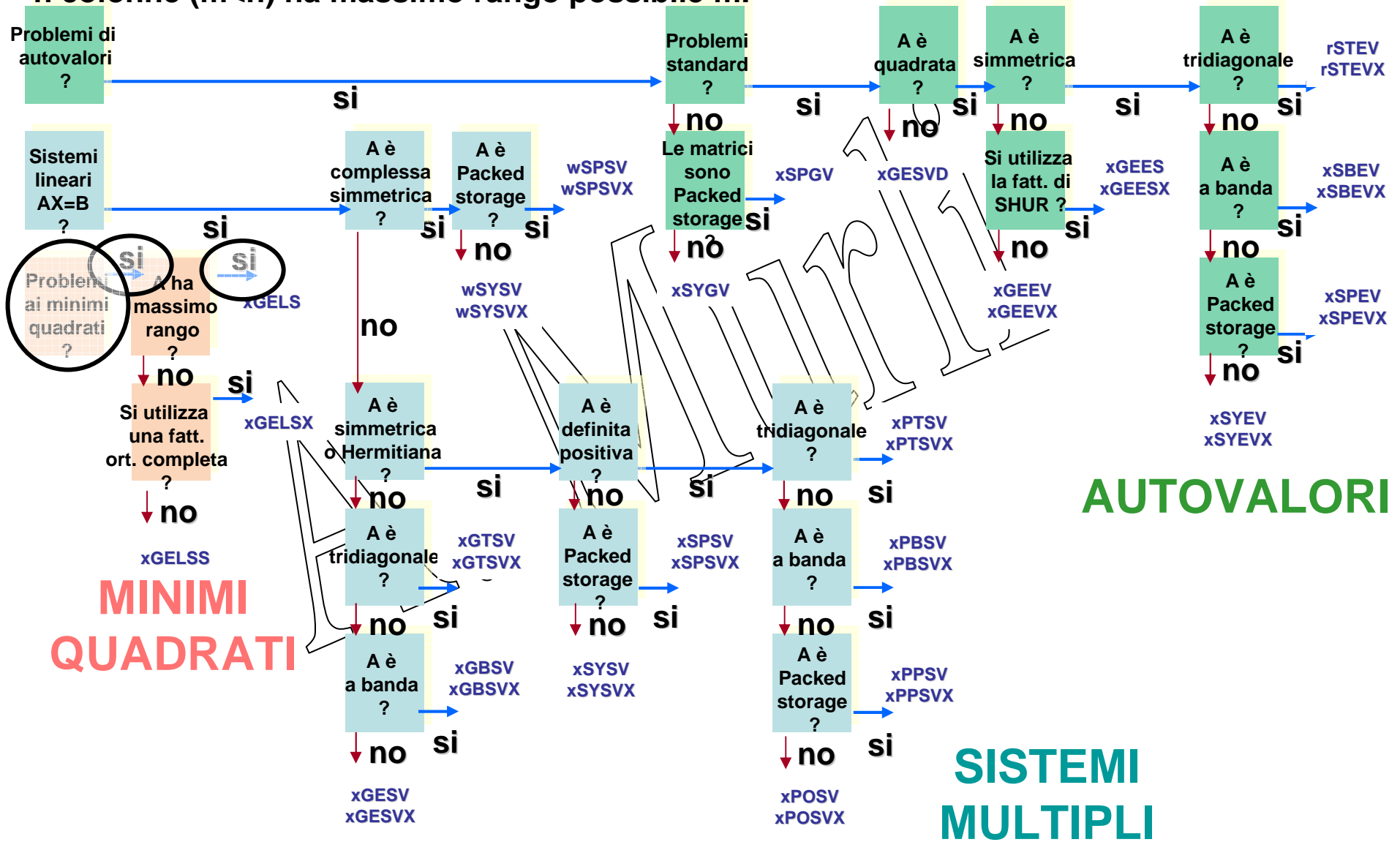
Si vuole individuare la routine di LAPACK adatta per determinare gli autovalori di una matrice Hermitiana memorizzata "packed storage".



# albero delle routines driver di LAPACK

## Esempio di utilizzo 3

Si vuole individuare la routine di LAPACK adatta per determinare la soluzione di minima norma del sistema lineare multiplo sottodeterminato  $AX=B$ , dove A matrice di m righe ed n colonne ( $m < n$ ) ha massimo rango possibile m.



**AUTOVALORI**

**MINIMI QUADRATI**

**SISTEMI MULTIPLI**

## SSPEV

SUBROUTINE **SSPEV**(JOBZ, UPLO, N, AP, W, Z, LDZ, WORK,INFO)

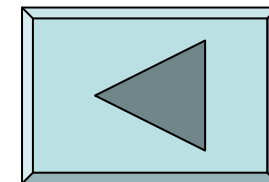
CHARACTER JOBZ, UPLO

INTEGER INFO,LDZ,N

REAL AP(\*), W(\*), WORK(\*), Z(LDZ,\*)

**SCOPO:** SSPEV calcola tutti gli autovalori e, se si desidera, gli autovettori di una matrice reale simmetrica memorizzata "packed".

- JOBZ** = 'N' calcola solo gli autovalori  
= 'V' calcola anche gli autovettori
- UPLO** = 'U' è memorizzato il triangolo superiore di A  
= 'L' è memorizzato il triangolo inferiore di A
- AP** la matrice A packed storage
- N** dimensione di A
- W** contiene gli autovettori in ordine decrescente
- Z** se JOBZ='N' contiene gli autovalori in ordine decrescente
- LDZ** leading dimension di Z
- WORK** variabile locale di dimensione 3 x n
- INFO** = 0 esito con successo  
= - i l'i-esimo argomento non è lecito  
= i determinati solo i-1 autovalori



**Torna all'albero**

## SPBSV

SUBROUTINE **SPBSV**(UPLO, N, KD, NRHS, AB, LDAB, B, LDB, INFO)

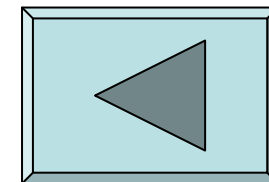
CHARACTER **UPLO**

INTEGER **INFO, KD, LDAB, LDB, N, NRHS**

REAL **AB(LDAB,\*), B(LDB,\*)**

**SCOPO:** SPBSV calcola la soluzione di un sistema reale multiplo dove la matrice dei coefficienti A è simmetrica, definita positiva e a banda.

- UPLO** = 'U' è memorizzato il triangolo superiore di A  
= 'L' è memorizzato il triangolo inferiore di A
- N** dimensione di A
- KD** numero di diagonal superiori di A
- NRHS** numero di colonne di B
- AB** la matrice A band storage
- LDAB** leading dimension di AB
- B** matrice dei termini noti; in output soluzione del sistema
- LDB** leading dimension di B
- INFO** = 0 esito con successo  
= -i l'i-esimo argomento non è lecito  
= i l'i-esimo minore principale non è definito positivo e la fattorizzazione non può andare avanti



**Torna all'albero**

# SGELS

SUBROUTINE **SGELS**(TRANS, M , N, NRHS, A, LDA, B, LDB, LWORK, INFO)

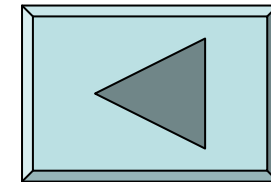
CHARACTER TRANS

INTEGER INFO,LDA,LDB,LWORK,M,N,NRHS

REAL A(LDA,\*), B (LDB,\*), WORK(LWORK)

**SCOPO:** SGELS risolve un sistema sovra- (o sotto-) determinato reale involvente una matrice dei coefficienti A rettangolare a pieno rango, utilizzando la fattorizzazione LQ o QR.

- TRANS** = 'N' il sistema utilizza A  
= 'T' il sistema utilizza  $A^T$   
= 'C' il sistema utilizza  $A^H$
- M** numero di righe di A
- NRHS** numero di colonne di B
- LDA** leading dimension di AB
- B** matrice dei termini noti; in output le soluzioni di minima norma
- LDB** leading dimension di B
- WORK** variabile locale
- LWORK** dimensione di WORK
- INFO** = 0 esito con successo  
= - i l'i-esimo argomento non è lecito



**Torna all'albero**

**Fine Esercitazione**

A. V. V. V. V. V.