

Description	Syntax	Remarks and examples	Conformability
Diagnostics	Reference	Also see	

Description

`schurd(X , T , Q)` computes the Schur decomposition of a square, numeric matrix, X , returning the [Schur-form](#) matrix, T , and the matrix of Schur vectors, Q . Q is orthogonal if X is real and unitary if X is complex.

`_schurd(X , Q)` does the same thing as `schurd()`, except that it returns T in X .

`schurddgroupby(X , f , T , Q , w , m)` computes the Schur decomposition and the eigenvalues of a square, numeric matrix, X , and groups the results according to whether a condition on each eigenvalue is satisfied. `schurddgroupby()` returns the Schur-form matrix in T , the matrix of Schur vectors in Q , the eigenvalues in w , and the number of eigenvalues for which the condition is true in m . f is a pointer of the function that implements the condition on each eigenvalue, as discussed [below](#).

`_schurddgroupby(X , f , Q , w , m)` does the same thing as `schurddgroupby()` except that it returns T in X .

`_schurd_la()` and `_schurddgroupby_la()` are the interfaces into the LAPACK routines used to implement the above functions; see [\[M-1\] LAPACK](#). Their direct use is not recommended.

Syntax

```
void      schurd( $X$ ,  $T$ ,  $Q$ )
void      _schurd( $X$ ,  $Q$ )

void  schurddgroupby( $X$ ,  $f$ ,  $T$ ,  $Q$ ,  $w$ ,  $m$ )
void  _schurddgroupby( $X$ ,  $f$ ,     $Q$ ,  $w$ ,  $m$ )
```

where inputs are

X : *numeric matrix*
 f : *pointer scalar* (points to a function used to group eigenvalues)

and outputs are

T : *numeric matrix* (Schur-form matrix)
 Q : *numeric matrix* (orthogonal or unitary)
 w : *numeric vector* of eigenvalues
 m : *real scalar* (the number of eigenvalues satisfy the grouping condition)

Remarks and examples

Remarks are presented under the following headings:

[Schur decomposition](#)
[Grouping the results](#)

Schur decomposition

Many algorithms begin by obtaining the Schur decomposition of a square matrix.

The Schur decomposition of matrix **X** can be written as

$$\mathbf{Q}' \times \mathbf{X} \times \mathbf{Q} = \mathbf{T}$$

where **T** is in Schur form, **Q**, the matrix of Schur vectors, is orthogonal if **X** is real or unitary if **X** is complex.

A real, square matrix is in Schur form if it is block upper triangular with 1×1 and 2×2 diagonal blocks. Each 2×2 diagonal block has equal diagonal elements and opposite sign off-diagonal elements. A complex, square matrix is in Schur form if it is upper triangular. The eigenvalues of **X** are obtained from the Schur form by a few quick computations.

In the example below, we define **X**, obtain the Schur decomposition, and list **T**.

```
: X=(.31,.69,.13,.56\ .31,.5,.72,.42\ .68,.37,.71,.8\ .09,.16,.83,.9)
: schurd(X, T=., Q=.)
: T
```

	1	2	3	4
1	2.10742167	.1266712792	.0549744934	.3329112999
2	0	-.0766307549	.3470959084	.1042286546
3	0	-.4453774705	-.0766307549	.3000409803
4	0	0	0	.4658398402

Grouping the results

In many applications, there is a stable solution if the modulus of an eigenvalue is less than one and an explosive solution if the modulus is greater than or equal to one. One frequently handles these cases differently and would group the Schur decomposition results into a block corresponding to stable solutions and a block corresponding to explosive solutions.

In the following example, we use `schurgroupby()` to put the stable solutions first. One of the arguments to `schurgroupby()` is a [pointer](#) to a function that accepts a complex scalar argument, an eigenvalue, and returns 1 to select the eigenvalue and 0 otherwise. Here `isstable()` returns 1 if the eigenvalue is less than 1:

```
: real scalar isstable(scalar p)
> {
>     return((abs(p)<1))
> }
```

Using this function to group the results, we see that the Schur-form matrix has been reordered.

```
: schurgroupby(X, &isstable(), T=., Q=., w=., m=.)
: T
```

	1	2	3	4
1	-.0766307549	.445046622	.3029641608	-.0341867415
2	-.3473539401	-.0766307549	-.1036266286	.0799058566
3	0	0	.4658398402	-.3475944606
4	0	0	0	2.10742167

Listing the moduli of the eigenvalues reveals that they are grouped into stable and explosive groups.

```
: abs(w)
```

	1	2	3	4
1	.4005757984	.4005757984	.4658398402	2.10742167

`m` contains the number of stable solutions

```
: m
3
```

Conformability

`schurd(X , T , Q):`

input:

X : $n \times n$

output:

T : $n \times n$

Q : $n \times n$

`_schurd(X , Q):`

input:

X : $n \times n$

output:

X : $n \times n$

Q : $n \times n$

`schurgroupby(X , f , T , Q , w , m):`

input:

X : $n \times n$

f : 1×1

output:

T : $n \times n$

Q : $n \times n$

w : $1 \times n$

m : 1×1

`_schurgroupby(X, f, Q, w, m):`

input:

X : $n \times n$

f : 1×1

output:

X : $n \times n$

Q : $n \times n$

w : $1 \times n$

m : 1×1

Diagnostics

`_schurd()` and `_schurgroupby()` abort with error if X is a view.

`schurd()`, `_schurd()`, `schurgroupby()`, and `_schurgroupby()` return missing results if X contains missing values.

`schurgroupby()` groups the results via a matrix transform. If the problem is very ill conditioned, applying this matrix transform can cause changes in the eigenvalues. In extreme cases, the grouped eigenvalues may no longer satisfy the condition used to perform the grouping.

Issai Schur (1875–1941) was born in Mogilev, which is now in Belarus. He studied mathematics and physics at the University of Berlin, where Frobenius was one of his teachers. Schur obtained his doctorate with a thesis on rational representations of the general linear group over the complex field. Thereafter, he taught and researched at Berlin University (with an interval in Bonn) until he was dismissed by the Nazis in 1935. He was a superb lecturer. Schur is now known mainly for his fundamental contributions to the representation theory of groups, but he also worked in other areas of algebra, including matrices, number theory, and analysis. In 1939, he emigrated to Palestine, where he later died in poverty.

Reference

Ledermann, W. 1983. Issai Schur and his school in Berlin. *Bulletin of the London Mathematical Society* 15: 97–106.
<https://doi.org/10.1112/blms/15.2.97>.

Also see

[M-1] **LAPACK** — Linear algebra package (LAPACK) routines

[M-5] **hessenbergd()** — Hessenberg decomposition

[M-4] **Matrix** — Matrix functions

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).