

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

## How to select a submatrix (not in any particular pattern) in Matlab



How to select a submatrix (not in any pattern) in Matlab? For example, for a matrix of size 10 by 10, how to select the submatrix consisting of intersection of the 1st 2nd and 9th rows and the 4th and 6th columns?

Thanks for any helpful answers!

matlab matrix matrix-indexing

edited May 23 '13 at 13:04



Eitan T

27.1k 11 36 74

asked Oct 26 '12 at 16:55



v32

66 1 1 3

### 2 Answers

#### TLDR: Short Answer

As for your question, suppose you have an arbitrary 10-by-10 matrix `A`. The simplest way to extract the desired sub-matrix would be with an index vector:

```
B = A([1 2 9], [4 6]);
```

#### Indexing in MATLAB

There's an [interesting article](#) in the official documentation that comprehensively explains indexing in MATLAB. Basically, there are several ways to extract a subset of values, I'll summarize them for you:

##### 1. Indexing Vectors

Indexing vectors indicate the indices of the element to be extracted. They can either contain a single index or several, like so:

```
A = [10 20 30 40 50 60 70 80 90]
## Extracts the third and the ninth element
B = A([3 9]) ## B = [30 90]
```

Indexing vectors can be specified for each dimension separately, for instance:

```
A = [10 20 30; 40 50 60; 70 80 90];
## Extract the first and third rows, and the first and second columns
B = A([1 3], [1 2]) ## B = [10 30; 40 60]
```

There are also two special subscripts: `end` and the colon (`:`):

- `end` simply indicates the last index in that dimension.
- The colon is just a short-hand notation for `"1:end"`.

For example, instead of writing `A([1 2 3], [2 3])`, you can write `A(:, 2:end)`. This is especially useful for large matrices.

##### 2. Linear Indexing

Linear indexing treats any matrix as if it were a column vector by concatenating the columns into one column vector and assigning indices to the elements respectively. For instance, we have:

```
A = [10 20 30; 40 50 60; 70 80 90];
```

and we want to compute  $b = A(2)$ . The equivalent column vector is:

```
A = [10;
     40;
     70;
     20;
     50;
     80;
     30;
     60;
     90]
```

and thus  $b$  equals 40.

The special colon and end subscripts are also allowed, of course. For that reason,  $A(:)$  converts any matrix  $A$  into a column vector.

**Linear indexing with matrix subscripts:** It is also possible to use another matrix for linear indexing. The subscript matrix is simply converted into a column vector, and used for linear indexing. The resulting matrix is, however always of the same dimensions as the subscript matrix.

For instance, if  $I = [1 \ 3; \ 1 \ 2]$ , then  $A(I)$  is the same as writing  $\text{reshape}(A(I(:)), \text{size}(I))$ .

**Converting from matrix subscripts to linear indices and vice versa:** For that you have `sub2ind` and `ind2sub`, respectively. For example, if you want to convert the subscripts  $[1, \ 3]$  in matrix  $A$  (corresponding to element 30) into a linear index, you can write `sub2ind(size(A), 1, 3)` (the result in this case should be 7, of course).

### 3. Logical Indexing

In logical indexing the subscripts are binary, where a logical 1 indicates that the corresponding element is selected, and 0 means it is not. The subscript vector must be either of the same dimensions as the original matrix or a vector with the same number of elements. For instance, if we have:

```
A = [10 20 30; 40 50 60; 70 80 90];
```

and we want to extract  $A([1 \ 3], [1 \ 2])$  using logical indexing, we can do either this:

```
Ir = logical([1 1 0]);
Ic = logical([1 0 1]);
B = A(Ir, Ic)
```

or this:

```
I = logical([1 0 1; 1 0 1; 0 0 0]);
B = A(I)
```

or this:

```
I = logical([1 1 0 0 0 0 1 1 0]);
B = A(I)
```

Note that in the latter two cases is a one-dimensional vector, and should be reshaped back into a matrix if necessary (for example, using `reshape`).

edited May 30 '13 at 16:36

answered Oct 26 '12 at 19:31



Eitan T

27.1k

11

36

74

2 +1 liked the TLDR part :-> – Shai Dec 31 '12 at 12:45

Work on work you love. From home.



 **stackoverflow**  
CAREERS

Let me explain with an example:

Let's define a 6x6 matrix

```
A = magic(6)
```

```
A =
35    1    6   26   19   24
 3   32    7   21   23   25
31    9    2   22   27   20
 8   28   33   17   10   15
30    5   34   12   14   16
 4   36   29   13   18   11
```

From this matrix you want the elements in rows 1, 2 and 5, and in the columns 4 and 6

```
B = A([1 2 5],[4 6])
```

```
B =
```

3/29/2016

matrix - How to select a submatrix (not in any particular pattern) in Matlab - Stack Overflow

26 24  
21 25  
12 16

Hope this helps.

answered Oct 26 '12 at 17:21



[R. Schifini](#)

2,207 8 13

---

2 How to do that in numpy? – [linello](#) Jun 5 '14 at 10:32

---