

find

Find indices and values of nonzero elements

Syntax

<code>k = find(X)</code>	example
<code>k = find(X,n)</code>	example
<code>k = find(X,n,direction)</code>	example
 <code>[row,col] = find(__)</code>	 example
<code>[row,col,v] = find(__)</code>	example

Description

`k = find(X)` returns a vector containing the [linear indices](#) of each nonzero element in array X. [example](#)

- If X is a vector, then find returns a vector with the same orientation as X.
- If X is a multidimensional array, then find returns a column vector of the linear indices of the result.
- If X contains no nonzero elements or is empty, then find returns an empty array.

`k = find(X,n)` returns the first n indices corresponding to the nonzero elements in X. [example](#)

`k = find(X,n,direction)`, where direction is 'last', finds the last n nonzero elements in X. The default for direction is 'first', which finds the first n nonzero elements. [example](#)

`[row,col] = find(__)` returns the row and column subscripts of each nonzero element in array X using any of the input arguments in previous syntaxes. [example](#)

`[row,col,v] = find(__)` also returns vector v, which contains the nonzero elements of X. [example](#)

Examples

[collapse all](#)

Zero and Nonzero Elements in Matrix

Find the nonzero elements in a 3-by-3 matrix.

[Open This Example](#)

```
X = [1 0 2; 0 1 1; 0 0 4]
```

X =

```
1     0     2
0     1     1
0     0     4
```

```
k = find(X)
```

k =

```
1
5
7
8
9
```

Use the logical not operator on X to locate the zeros.

```
k2 = find(~X)
```

k2 =

```
2
3
4
6
```

Elements Satisfying a Condition

Find the first five elements that are less than 10 in a 4-by-4 magic square matrix.

[Open This Example](#)

```
X = magic(4)
```

```
X =
```

```
16     2     3    13
  5    11    10     8
  9     7     6    12
  4    14    15     1
```

```
k = find(X<10,5)
```

```
k =
```

```
2
3
4
5
7
```

View the corresponding elements of X.

```
X(k)
```

```
ans =
```

```
5
9
4
2
7
```

Elements Equal to Specific Values

To find a specific integer value, use the == operator. For instance, find the element equal to 13 in a 1-by-10 vector of odd integers.

[Open This Example](#)

```
x = 1:2:20
```

```
x =
```

```
     1     3     5     7     9    11    13    15    17    19
```

```
k = find(x==13)
```

```
k =
```

```
     7
```

To find a noninteger value, use a tolerance value based on your data. Otherwise, the result is sometimes an empty matrix due to floating-point roundoff error.

```
y = 0:0.1:1
```

```
y =
```

```
Columns 1 through 7
```

```
     0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
```

```
Columns 8 through 11
```

```
    0.7000    0.8000    0.9000    1.0000
```

```
k = find(y==0.3)
```

```
k =
```

```
Empty matrix: 1-by-0
```

```
k = find(abs(y-0.3) < 0.001)
```

k =

4

Last Several Nonzero Elements

Create a 6-by-6 magic square matrix with all of the odd-indexed elements equal to zero.

[Open This Example](#)

```
X = magic(6);  
X(1:2:end) = 0
```

X =

0	0	0	0	0	0
3	32	7	21	23	25
0	0	0	0	0	0
8	28	33	17	10	15
0	0	0	0	0	0
4	36	29	13	18	11

Locate the *last* four nonzeros.

```
k = find(X,4,'last')
```

k =

30
32
34
36

Elements Satisfying Multiple Conditions

Find the first three elements in a 4-by-4 matrix that are greater than 0 and less than 10. Specify two outputs to return the row and column subscripts to the elements.

[Open This Example](#)

```
X = [18 3 1 11; 8 10 11 3; 9 14 6 1; 4 3 15 21]
```

X =

18	3	1	11
8	10	11	3
9	14	6	1
4	3	15	21

```
[row,col] = find(X>0 & X<10,3)
```

row =

2
3
4

col =

1
1
1

The first instance is X(2,1), which is 8.

Subscripts and Values for Nonzero Elements

Find the nonzero elements in a 3-by-3 matrix. Specify three outputs to return the row subscripts, column subscripts, and element values.

[Open This Example](#)

```
X = [3 2 0; -5 0 7; 0 0 1]
```

X =

3	2	0
-5	0	7
0	0	1

```
[row,col,v] = find(X)
```

row =

1
2
1
2
3

col =

1
1
2
3
3

v =

3
-5
2
7
1

Subscripts of Multidimensional Array

Find the nonzero elements in a 4-by-2-by-3 array. Specify two outputs, `row` and `col`, to return the row and column subscripts of the nonzero elements. When the input is a multidimensional array ($N > 2$), `find` returns `col` as a linear index over the $N-1$ trailing dimensions of `X`.

[Open This Example](#)

```
X = zeros(4,2,3);  
X([1 12 19 21]) = 1
```

```
X(:, :, 1) =
```

1	0
0	0
0	0
0	0

```
X(:, :, 2) =
```

0	0
0	0
0	0
1	0

```
X(:, :, 3) =
```

0	1
0	0
1	0
0	0

```
[row,col] = find(X)
```



```
row =
```

```
1  
4  
3  
1
```

```
col =
```

```
1  
3  
5  
6
```

Related Examples

- [Find Array Elements That Meet a Condition](#)

Input Arguments

[collapse all](#)

X — Input array

scalar | vector | matrix | multidimensional array

Input array, specified as a scalar, vector, matrix, or multidimensional array. If X is an empty array or has no nonzero elements, then k is an empty array.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64 | logical | char

Complex Number Support: Yes

n — Number of nonzeros to find

positive integer scalar

Number of nonzeros to find, specified as a positive integer scalar. By default, `find(X,n)` looks for the first n nonzero elements in X.

direction — Search direction

'first' (default) | 'last'

Search direction, specified as the string 'first' or 'last'. Look for the *last* n nonzero elements in X using `find(X,n,'last')`.

Output Arguments

[collapse all](#)

k — Indices to nonzero elements

vector

Indices to nonzero elements, returned as a vector. If X is a row vector, then k is also a row vector. Otherwise, k is a column vector. k is an empty array when X is an empty array or has no nonzero elements.

You can return the nonzero values in X using `X(k)`.

row — Row subscripts

vector

Row subscripts, returned as a vector. Together, row and col specify the `X(row,col)` subscripts corresponding to the nonzero elements in X.

col — Column subscripts

vector

Column subscripts, returned as a vector. Together, row and col specify the `X(row,col)` subscripts corresponding to the nonzero elements in X.

If X is a multidimensional array with $N > 2$, then col is a linear index over the N-1 trailing dimensions of X. This preserves the relation `X(row(i),col(i)) == v(i)`.

v — Nonzero elements of X

vector

Nonzero elements of X, returned as a vector.

More About

[collapse all](#)

Linear Indices

A linear index allows use of a single subscript to index into an array, such as `A(k)`. MATLAB® treats the array as a single column vector with each column appended to the bottom of the previous column. Thus, linear indexing numbers the elements in the columns from top to bottom, left to right.

For example, consider a 3-by-3 matrix. You can reference the `A(2,2)` element with `A(5)`, and the `A(2,3)` element with `A(8)`. The linear index changes depending on the size of the array; `A(5)` returns a differently located element for a 3-by-3 matrix than it does for a 4-by-4 matrix.

The `sub2ind` and `ind2sub` functions are useful in converting between subscripts and linear indices.

Tips

- To find array elements that meet a condition, use `find` in conjunction with a relational expression. For example, `find(X<5)` returns the linear indices to the elements in `X` that are less than 5.
- To directly find the elements in `X` that satisfy the condition `X<5`, use `X(X<5)`. Avoid function calls like `X(find(X<5))`, which unnecessarily use `find` on a logical matrix.
- When you execute `find` with a relational operation like `X>1`, it is important to remember that the result of the relational operation is a logical matrix of ones and zeros. For example, the command `[row,col,v] = find(X>1)` returns a column vector of logical 1 (true) values for `v`.
- The row and column subscripts, `row` and `col`, are related to the linear indices in `k` by `k = sub2ind(size(X),row,col)`.
- [Matrix Indexing](#)
- [Relational Operations](#)
- [Sparse Matrix Manipulation](#)

See Also

[ind2sub](#) | [ismember](#) | [Logical Operators: Short-Circuit](#) | [nonzeros](#) | [strfind](#) | [sub2ind](#)

Introduced before R2006a
