

To print higher-resolution math symbols, click the **Hi-Res Fonts for Printing** button on the jsMath control panel.

tutorial 3

```
K.<j,b>=QQ[sqrt(-1),sqrt(2)] # extend field of rational numbers with j and b
```

j # behaves like I

```
I
```

```
j*j
```

```
-1
```

j*sqrt(2) # sage does not understand multiplication of j by sqrt(2)

```
Traceback (click to the left for traceback)
```

```
...
```

```
TypeError: unsupported operand parent(s) for '*': 'Number Field in I with defining polynomial x^2 + 1 over its base field' and 'Symbolic Ring'
```

b # behaves like sqrt(2)

```
sqrt2
```

```
b*b
```

```
2
```

j*b # replacing sqrt(2) by b fixes the problem!

```
sqrt2*I
```

```
sx=(1/2)*matrix([[0,1],[1,0]]);sx
```

```
[ 0 1/2]
[1/2  0]
```

sx.eigenmatrix_right() # no problem computing eigenvectors, because matrix elements are real

```
([ 1/2  0]
 [ 0 -1/2], [ 1  1]
 [ 1 -1])
```

```
sy=(1/2)*matrix([[0,-I],[I,0]]);sy
```

```
[ 0 -1/2*I]
```

```
[ 1/2*I    0]
```

sy.eigenmatrix_right() # sage has a bug: cannot compute eigenvectors for complex matrix

```
Traceback (click to the left for traceback)
```

```
...
```

```
TypeError: degree() takes exactly one argument (0 given)
```

sy=(1/2)*matrix([[0,-j],[j,0]]);sy # replacing I with j fixes the problem!

```
[    0 -1/2*I]
[ 1/2*I    0]
```

sy.eigenmatrix_right()

```
([ 1/2    0]
 [  0 -1/2], [ 1  1]
 [ I -I])
```

pp=matrix([[1,0,0,0],[0,0,1,0],[0,1,0,0],[0,0,0,1]]);pp # permutation matrix

```
[1 0 0 0]
[0 0 1 0]
[0 1 0 0]
[0 0 0 1]
```

```
def id(n):
    return identity_matrix(2^n)
```

id(2)

```
[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
[0 0 0 1]
```

```
def sz(n):
    if n==1: return (1/2)*matrix([[1,0],[0,-1]])
    result = sz(n-1).tensor_product(id(1)) + id(n-1).tensor_product(sz(1))
    return result
```

sz(1)

```
[ 1/2    0]
[  0 -1/2]
```

sz(2)

```
[ 1  0  0  0]
[ 0  0  0  0]
[ 0  0  0  0]
[ 0  0  0 -1]
```

```
var('x1,x2') # Useful for problem 3.4
```

```
(x1, x2)
```

```
eq1 = x1 + x2 == 5
```

```
eq2 = x1 - x2 == 1
```

```
solve([eq1,eq2],x1,x2)
```

```
[[x1 = 3, x2 = 2]]
```

```
v=[-0.3,1.2,3.1] # Useful for problem 3.5+
```

```
v
```

```
[-0.3000000000000000, 1.2000000000000000, 3.1000000000000000]
```

```
v[0]
```

```
-0.3000000000000000
```

```
v[1]
```

```
1.2000000000000000
```

```
srange(3)
```

```
[0, 1, 2]
```

```
srange(3,include_endpoint=True)
```

```
[0, 1, 2, 3]
```

```
srange(3,include_endpoint=1)
```

```
[0, 1, 2, 3]
```

```
srange(1,8)
```

```
[1, 2, 3, 4, 5, 6, 7]
```

```
sum((x-v[a])^2 for a in srange(3))
```

```
(x - 3.1000000000000000)^2 + (x - 1.2000000000000000)^2 + (x +
```

0.3000000000000000)^2