Gaussian elimination with elimination matrices

In [56]:

```
#keep
import numpy as np
import numpy.linalg as la
```

In [57]:

```
#keep
n = 3

np.random.seed(15)
A = np.round(5*np.random.randn(n, n))

A
```

Out[57]:

```
array([[-2.,  2., -1.],
       [-3.,  1., -9.],
       [-5., -5., -2.]])
```

`U` is the copy of `A` that we'll modify:

In [58]:

```
#keep
U = A.copy()
```

Now eliminate `U[1,0]`:

In [59]:

```
M1 = np.eye(n)
M1[1,0] = -U[1,0]/U[0,0]
M1
```

Out[59]:

```
array([[ 1. ,  0. ,  0. ],
       [-1.5,  1. ,  0. ],
       [ 0. ,  0. ,  1. ]])
```

In [60]:

```
#keep
U = M1.dot(U)
U
```

Out[60]:

```
array([[-2. ,  2. , -1. ],
       [ 0. , -2. , -7.5],
       [-5. , -5. , -2. ]])
```

Now eliminate `U[2,0]`:

In [61]:

```
M2 = np.eye(n)
M2[2,0] = -U[2,0]/U[0,0]
```

In [62]:

```
#keep
U = np.dot(M2, U)
U
```

Out[62]:

```
array([[ -2. ,   2. ,  -1. ],
       [  0. ,  -2. ,  -7.5],
       [  0. , -10. ,   0.5]])
```

Now eliminate `U[2,1]`:

In [63]:

```
M3 = np.eye(n)
M3[2,1] = -U[2,1]/U[1,1]
```

In [64]:

```
#keep
U = M3.dot(U)
U
```

Out[64]:

```
array([[ -2. ,   2. ,  -1. ],
       [  0. ,  -2. ,  -7.5],
       [  0. ,   0. ,  38. ]])
```

Try inverting one of the `M`s:

In [65]:

```
#keep
print(M2)
print(la.inv(M2))
```

```
[[ 1.   0.   0. ]
 [ 0.   1.   0. ]
 [-2.5  0.   1. ]]
[[ 1.  -0.  -0. ]
 [ 0.   1.   0. ]
 [ 2.5  0.   1. ]]
```

So we've built `M3*M2*M1*A=U`. Test:

In [66]:

```
#keep
U2 = M3.dot(M2.dot(M1.dot(A)))
U2
```

Out[66]:

```
array([[ -2. ,   2. ,  -1. ],
       [  0. ,  -2. ,  -7.5],
       [  0. ,   0. ,  38. ]])
```

In [67]:

```
#keep
U
```

Out[67]:

```
array([[ -2. ,   2. ,  -1. ],
       [  0. ,  -2. ,  -7.5],
       [  0. ,   0. ,  38. ]])
```

Now define `L`:

In [68]:

```
L = la.inv(M1).dot(la.inv(M2).dot(la.inv(M3)))
L
```

Out[68]:

```
array([[ 1. ,  0. ,  0. ],
       [ 1.5,  1. ,  0. ],
       [ 2.5,  5. ,  1. ]])
```

Observations? (Shape? Diagonal values?)

In [69]:

```
#keep
np.dot(L, U) - A
```

Out[69]:

```
array([[ 0.,  0.,  0.],
       [ 0.,  0.,  0.],
       [ 0.,  0.,  0.]])
```