In [1]:

```
import numpy as np
```

In [6]:

```
def relaxation(matrix, vector, initial, iterations):
    result = initial
    residual = vector - np.dot(matrix, initial)
    id = np.identity(len(matrix))
    for _ in range(iterations):
        corr = np.argmax(np.abs(residual))
        alpha = residual[corr]/matrix[corr][corr]
        result = result + id[corr]*alpha
        residual = residual - alpha*np.transpose(matrix)[corr]
    return result
```

In [7]:

```
matrix = np.array([[1, 0.42, 0.54, 0.66],
                   [0.42, 1, 0.32, 0.44],
                   [0.54, 0.32, 1, 0.22],
                   [0.66, 0.44, 0.22, 1]])
vector = np.array([0.3, 0.5, 0.7, 0.9])
initial = np.array([0, 0, 0, 0])
```

In [14]:

```
# ошибка уменьшается с увеличением количества итераций
print(relaxation(matrix, vector, initial, 100))
print(relaxation(matrix, vector, initial, 1000))
```

```
[-1.25779361  0.04348739  1.03916615  1.48239278]
[-1.25779375  0.0434873   1.03916625  1.48239288]
```

In [21]:

```
import numpy.linalg as la # проверяем решение
numpy_solution = np.dot(la.inv(matrix), vector)
my_solution = relaxation(matrix, vector, initial, 1000)
print(numpy_solution)
print(my_solution)
print(numpy_solution - my_solution)
```

```
[-1.25779375  0.0434873   1.03916625  1.48239288]
[-1.25779375  0.0434873   1.03916625  1.48239288]
[ 8.88178420e-16 -6.93889390e-18  2.22044605e-16 -2.22044605e-16]
```