Demonstration of the Elsinger (2009) Algorithm

Start by initializing a simple system of three nodes:

```
vecE = [0;50;10];
matL = [0 20 10;5 0 40;10 0 0];
matL(:,:,2) = [0 0 15;0 0 24;10 0 0];
matTheta = [0 0.1 0;0 0 0.15;0 0 0];
```

The equity has to be computed iteratively, we can get a first approximation, seeing that the first node is already in default, and the second very close to being in default:

```
vecEquity_approx = vecE + sum(sum(matL,3),1)' - sum(sum(matL,3),2);
vecEquity_approx = vecEquity_approx + max(0,matTheta'*vecEquity_approx);
disp([{'Approximated Equity' 'Defaulted'};
    num2cell([vecEquity_approx vecEquity_approx < 0])])
```

```
    'Approximated Equity'    'Defaulted'
    [                -20]    [        1]
    [                  1]    [        0]
    [            79.1500]    [        0]
```

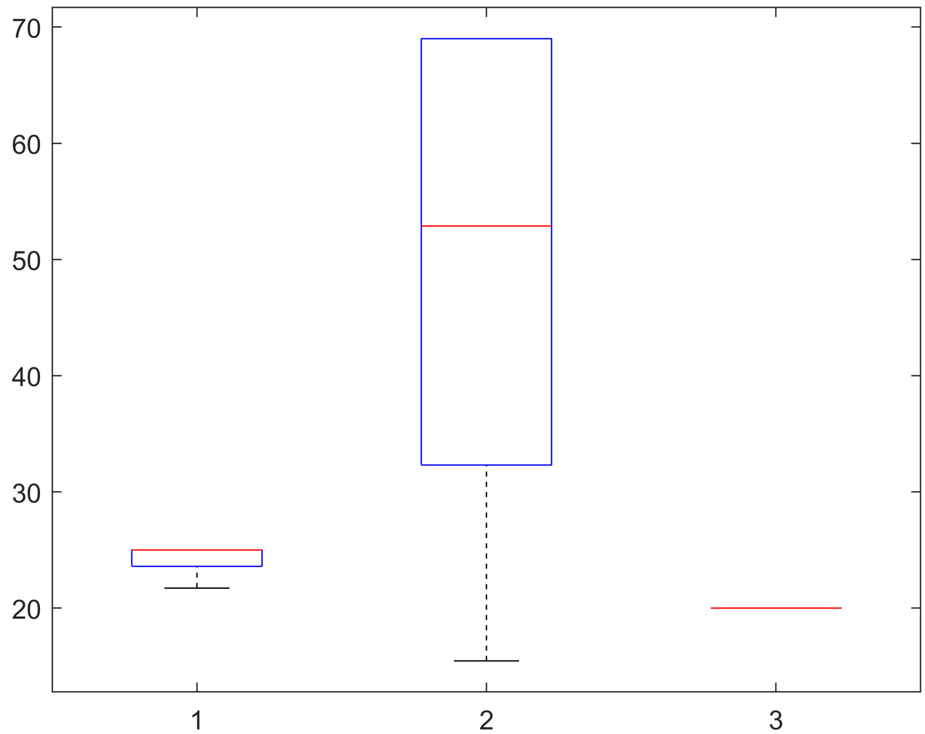
We compute the clearing payment vector and see that all nodes face losses due to contagion, and that the second node is pushed into default through these losses:

```
[matP, vecEquity, vecDefaultedBanks] = calcPayments(vecE,matL,matTheta);
disp([{'Equity' 'Defaulted'};
    num2cell([vecEquity vecDefaultedBanks])])
```

```
    'Equity'     'Defaulted'
    [    -20]    [        1]
    [-2.3333]    [        1]
    [     60]    [        0]
```

We can now also simulate payoffs under stochastic paths for external assets.

```
vecMu = [0.5;0.1;0.1];
matCovariance = [1 0 0;0 1 0;0 0 1];
numSimulations = 1000;
matValuations = calcValuations(vecE,vecMu,matCovariance,matL,matTheta,numSimulations);
boxplot(squeeze(sum(matValuations,2))')
```


70
60
50
40
30
20
1
2
3