Computational Eigenvector Simulation

Ali Taqi

Example 1: A Symmetric Stochastic Matrixx

Step 0: Setup the matrix

```
# Set seed
set.seed(23)
# Set parameters
M <- 2
mu <- 0
sd <- 1
# Generate matrix
P <- RM_stoch(M, symm = T, sparsity = F)
```

The Matrix

```
##           [,1]      [,2]
## [1,] 0.7210461 0.2789539
## [2,] 0.2789539 0.7210461
```

Eigenvalues of the Symmetric Stochastic Matrix

```
eigen_frame(P)
```

```
##         Re Im row_i
## 1  0.70711  0     1
## 2 -0.70711  0     1
## 3  0.70711  0     2
## 4  0.70711  0     2
```

Step 1: Get the batch

```
# Set batch parameters
B <- 100
# Create batch
batch <- make_batch(M = M, B = B)
head(batch)
```

```
##           x1         x2
## 1  0.6388979 -0.1525589
## 2  0.9270891  0.9562608
## 3  0.6810438  0.9932225
## 4  0.7319181  0.4028434
## 5 -0.2190539 -0.3704606
## 6  0.6918946 -0.7214430
```

Step 2: Evolve the batch

```
# Set evolution parameters
steps <- 20
# Evolve batch
evolved_batch <- evolve_batch(batch, steps, with_steps = T)
head(evolved_batch)
```

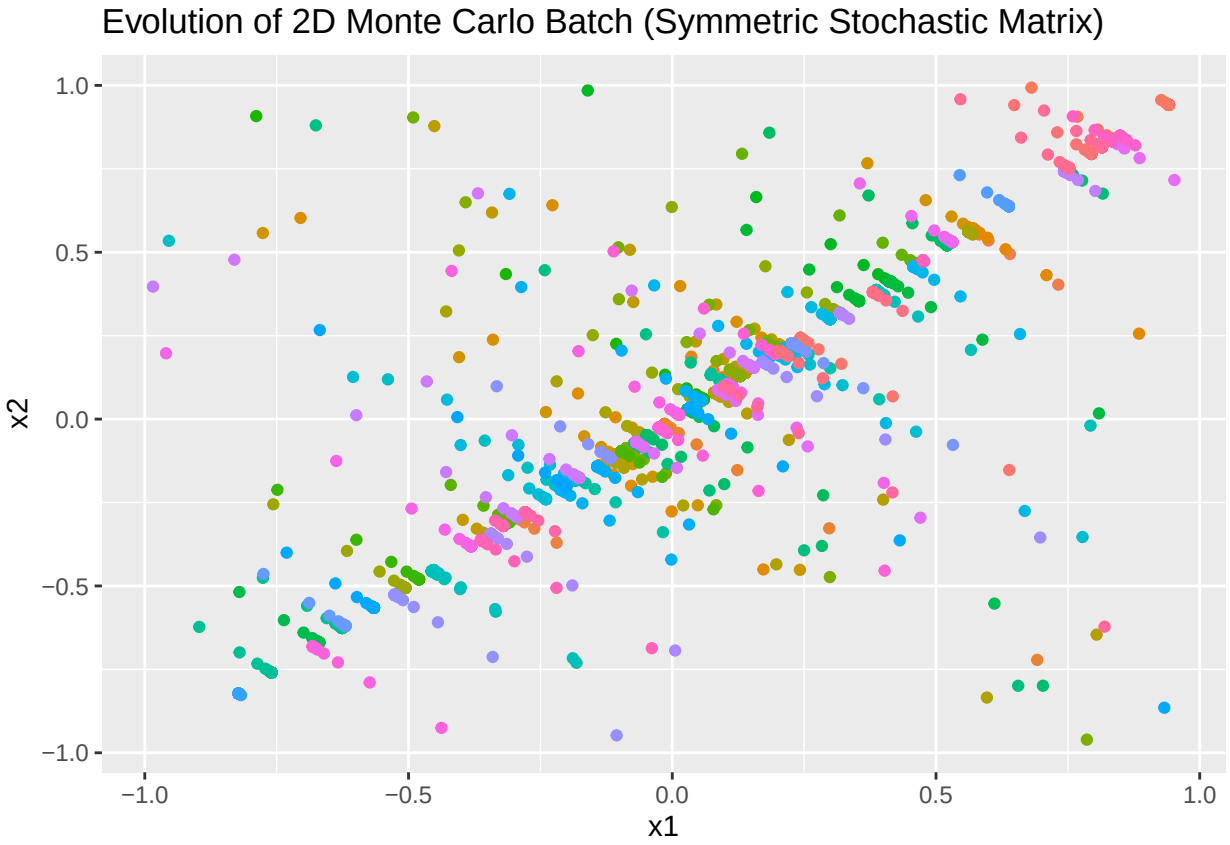
```
##          x1          x2 time element_index
## 1 0.6388979 -0.15255888    0             1
## 2 0.4181180  0.06822105    1             1
## 3 0.3205129  0.16582615    2             1
## 4 0.2773624  0.20897661    3             1
## 5 0.2582859  0.22805310    4             1
## 6 0.2498524  0.23648667    5             1
```

```
tail(evolved_batch)
```

```
##             x1        x2 time element_index
## 2095 0.7946909 0.7946923   15           100
## 2096 0.7946913 0.7946919   16           100
## 2097 0.7946915 0.7946917   17           100
## 2098 0.7946915 0.7946917   18           100
## 2099 0.7946916 0.7946916   19           100
## 2100 0.7946916 0.7946916   20           100
```

Step 3: Analyze the batch

```
# Plot the evolution arrays of the batch elements
batch_data <- evolved_batch
# 2d plot
batch_2d_plot(batch_data, "(Symmetric Stochastic Matrix)")
```



Example 2: A Symmetric Normal Matrix

Step 0: Setup the matrix

```
# Set seed
set.seed(6)
# Set parameters
M <- 9
mu <- 0
sd <- 1
# Generate matrix
P <- RM_normal(M, c(mu,sd), symm = T)
```

The Matrix

```
##              [,1]       [,2]        [,3]       [,4]        [,5]        [,6]
##  [1,]  0.26960598 -0.6299854  0.86865983  1.7271955  0.02418764  0.36802518
##  [2,] -0.62998541  1.7278511 -1.17859974  0.6532067 -0.36856649 -0.59955464
##  [3,]  0.86865983  2.0148645  0.51874901 -1.4049179  2.01486448 -1.18815834
##  [4,] -1.17859974  0.4924386 -1.30920430 -0.6714594  0.49243855 -1.17939052
##  [5,]  1.72719552  0.3680252  0.05460517  0.7386219 -0.88516413 -0.43233430
##  [6,]  0.65320671 -0.5995546  0.19038081  1.7076774 -0.59677030 -0.05413369
##  [7,] -1.40491794 -1.1881583 -1.05871745 -1.1697359  0.55377583 -0.03808156
##  [8,]  0.02418764 -1.1793905 -0.42162386  1.1379026  0.04487299 -0.16026528
##  [9,] -0.36856649 -0.4323343  0.46161167 -0.1704941 -1.09437298  0.24581094
##              [,7]       [,8]        [,9]
##  [1,] -1.30920430  0.7386219  0.04487299
##  [2,]  0.05460517  1.7076774 -1.09437298
##  [3,]  0.19038081 -1.1697359 -0.03808156
##  [4,] -1.05871745  1.1379026 -0.16026528
##  [5,] -0.42162386 -0.1704941  0.24581094
##  [6,]  0.46161167 -0.5967703  1.26325824
##  [7,]  0.21413762  0.5537758 -1.05951952
##  [8,]  1.26325824 -0.3891779 -0.44724564
##  [9,] -1.05951952 -0.4472456  2.60809809
```

The Eigenvalues

```
eigen_frame(P)
```

```
##          Re       Im row_i
## 1  -0.10399 -0.15539     1
## 2  -0.10399  0.15539     1
## 3  -0.18463  0.00000     1
## 4  -0.05121  0.00000     1
## 5  -0.35573 -0.28293     1
## 6  -0.35573  0.28293     1
## 7   0.31997  0.06061     1
## 8   0.31997 -0.06061     1
## 9  -0.01229  0.00000     1
## 10  0.60610  0.00000     2
## 11  0.60610  0.00000     2
## 12 -0.02246  0.00000     2
## 13 -0.17749  0.00000     2
```

```
## 14 -0.03028  0.05576     2
## 15 -0.03028 -0.05576     2
## 16  0.02410 -0.09115     2
## 17  0.02410  0.09115     2
## 18  0.18149  0.00000     2
## 19  0.06305 -0.51889     3
## 20  0.06305  0.51889     3
## 21  0.03536  0.00000     3
## 22 -0.56805  0.00000     3
## 23  0.59469  0.00000     3
## 24  0.59469  0.00000     3
## 25 -0.59349  0.00000     3
## 26 -0.59349  0.00000     3
## 27 -0.56756  0.00000     3
## 28  0.20238  0.07579     4
## 29  0.20238 -0.07579     4
## 30 -0.33936  0.00000     4
## 31  0.05530  0.00000     4
## 32  0.30963  0.20013     4
## 33  0.30963 -0.20013     4
## 34  0.09201  0.02117     4
## 35  0.09201 -0.02117     4
## 36  0.19351  0.00000     4
## 37 -0.00467 -0.12897     5
## 38 -0.00467  0.12897     5
## 39  0.43150  0.00000     5
## 40 -0.04679  0.00000     5
## 41 -0.07943  0.32180     5
## 42 -0.07943 -0.32180     5
## 43  0.33236 -0.01130     5
## 44  0.33236  0.01130     5
## 45  0.14426  0.00000     5
## 46 -0.07983  0.12924     6
## 47 -0.07983 -0.12924     6
## 48  0.55629  0.00000     6
## 49  0.37590  0.00000     6
## 50 -0.02645 -0.24221     6
## 51 -0.02645  0.24221     6
## 52  0.54578 -0.07201     6
## 53  0.54578  0.07201     6
## 54  0.51046  0.00000     6
## 55 -0.05409  0.30697     7
## 56 -0.05409 -0.30697     7
## 57 -0.40167  0.00000     7
## 58  0.03077  0.00000     7
## 59  0.14718  0.11009     7
## 60  0.14718 -0.11009     7
## 61 -0.06862  0.01561     7
## 62 -0.06862 -0.01561     7
## 63 -0.32781  0.00000     7
## 64 -0.02756  0.23133     8
## 65 -0.02756 -0.23133     8
## 66  0.43810  0.00000     8
## 67  0.06047  0.00000     8
```

```
## 68  0.07405 -0.28178     8
## 69  0.07405  0.28178     8
## 70  0.11579  0.10815     8
## 71  0.11579 -0.10815     8
## 72 -0.46349  0.00000     8
## 73 -0.25873  0.14019     9
## 74 -0.25873 -0.14019     9
## 75 -0.00891  0.00000     9
## 76  0.70145  0.00000     9
## 77 -0.09166  0.05922     9
## 78 -0.09166 -0.05922     9
## 79  0.27357  0.07315     9
## 80  0.27357 -0.07315     9
## 81 -0.06062  0.00000     9
```

Step 1: Get the batch

```
# Set batch parameters
B <- 100
# Create batch
batch <- make_batch(M = M, B = B)
head(batch)
```

```
##            x1         x2         x3         x4          x5         x6
## 1 -0.04070632  0.8922378  0.9740657  0.4485278 -0.01106800 -0.6037098
## 2  0.64676080  0.8610923 -0.9942109 -0.2012166  0.35944675 -0.2557878
## 3 -0.94910317 -0.7664982  0.3276610  0.9985725  0.76266919  0.5934003
## 4  0.14872891 -0.9079017 -0.8628192  0.6779868 -0.39852092 -0.5209503
## 5 -0.71352980 -0.9320286 -0.6531758  0.9893077  0.07481969 -0.2443750
## 6 -0.40847367 -0.0574896 -0.5034317 -0.1766644 -0.89289404 -0.1135333
##            x7         x8          x9
## 1 -0.87823312  0.9018124 -0.83237921
## 2 -0.19575995 -0.5325527  0.31532668
## 3  0.64135987 -0.9057326  0.07605654
## 4  0.98524870 -0.4475074  0.98107982
## 5  0.05282391 -0.6270022 -0.67814647
## 6  0.99590605  0.4076312 -0.15609459
```

Step 2: Evolve the batch

```
# Set evolution parameters
steps <- 20
# Evolve batch
evolved_batch <- evolve_batch(batch, steps, with_steps = T)
# View
head(evolved_batch)
```

```
##             x1           x2           x3            x4         x5         x6
## 1  -0.04070632    0.8922378    0.9740657  0.4485278479 -0.0110680 -0.6037098
## 2   0.89340682    4.4484221   -1.1190702 -0.0008681338  2.6887784 -2.5144866
## 3  -1.15108119    6.3487892   -8.7654245  3.5720391666 -0.1324768 -3.0677049
## 4 -13.49382004  -11.2565447  -33.6484867 22.4332115687  5.0232651 -4.3648379
## 5 -64.39240983 -106.7865663 -121.4386480  7.6282426263 35.6864319 -8.8460320
## 6 -65.32396004 -475.1807328 -234.4972195  0.2264540528 60.5696915 93.1006388
```

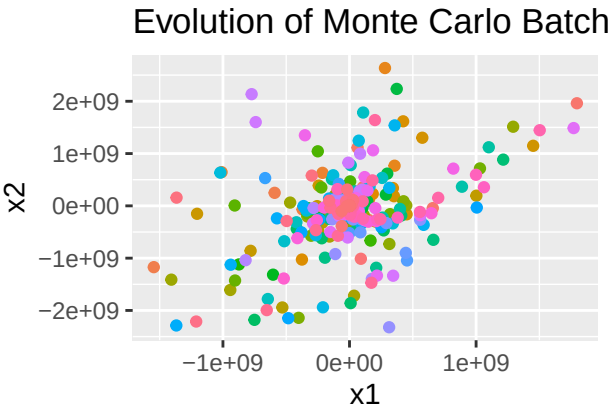
```
##            x7         x8           x9 time element_index
## 1  -0.8782331  0.9018124   -0.8323792    0             1
## 2   1.3716611  0.7617005   -3.4963601    1             1
## 3   1.5271654 12.6334262  -18.2137218    2             1
## 4  30.6269924 30.2377006  -65.9176159    3             1
## 5  97.3596707 72.1196516 -212.7728560    4             1
## 6 365.5374035 41.0172012 -575.3679760    5             1
```

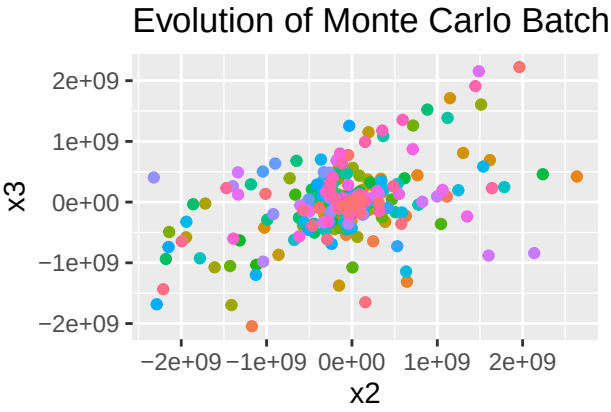
```
tail(evolved_batch)
```

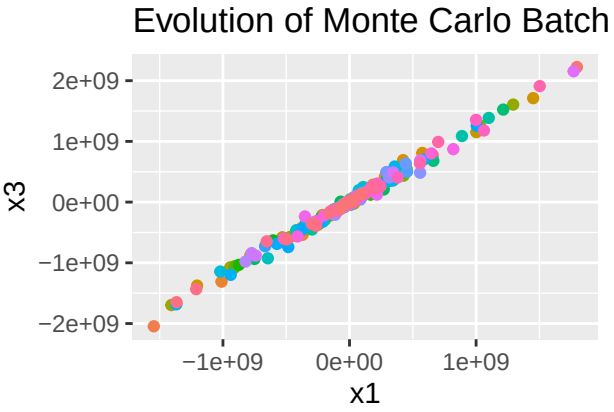
```
##               x1          x2        x3         x4          x5          x6
## 2095   -728700.1     4513747  -1182398    1538295  -1158882.5  -1590139.1
## 2096  -7542272.2     5293629  -8523404    2734045  -2524878.5  -2993915.7
## 2097 -21795794.0   -11448084 -29582611    3963894  -6493426.4   -794137.8
## 2098 -62249787.5  -101996034 -73886776  -14048050    549063.2  20623805.3
## 2099 -61222516.4  -386197571 -95907369  -74473824  29625674.8 108126909.1
## 2100  89843542.8 -1011872549 135806059 -306344254 209117638.9 338027813.9
##             x7         x8           x9 time element_index
## 2095  -1104567    2463490     632684.1   15           100
## 2096   1306505    9596460   -5748924.5   16           100
## 2097  23822360   18327250  -31317969.2   17           100
## 2098  81887931   25113401 -105675021.5   18           100
## 2099 247247907  -79277300 -233522876.5   19           100
## 2100 357293387 -474641783 -256198784.0   20           100
```

Step 3: Analyze the batch

```
# Plot the evolution arrays of the batch elements
batch_data <- evolved_batch
# 3d plot
batch_3d_plot(batch_data, "(Symmetric Normal Matrix)")
```

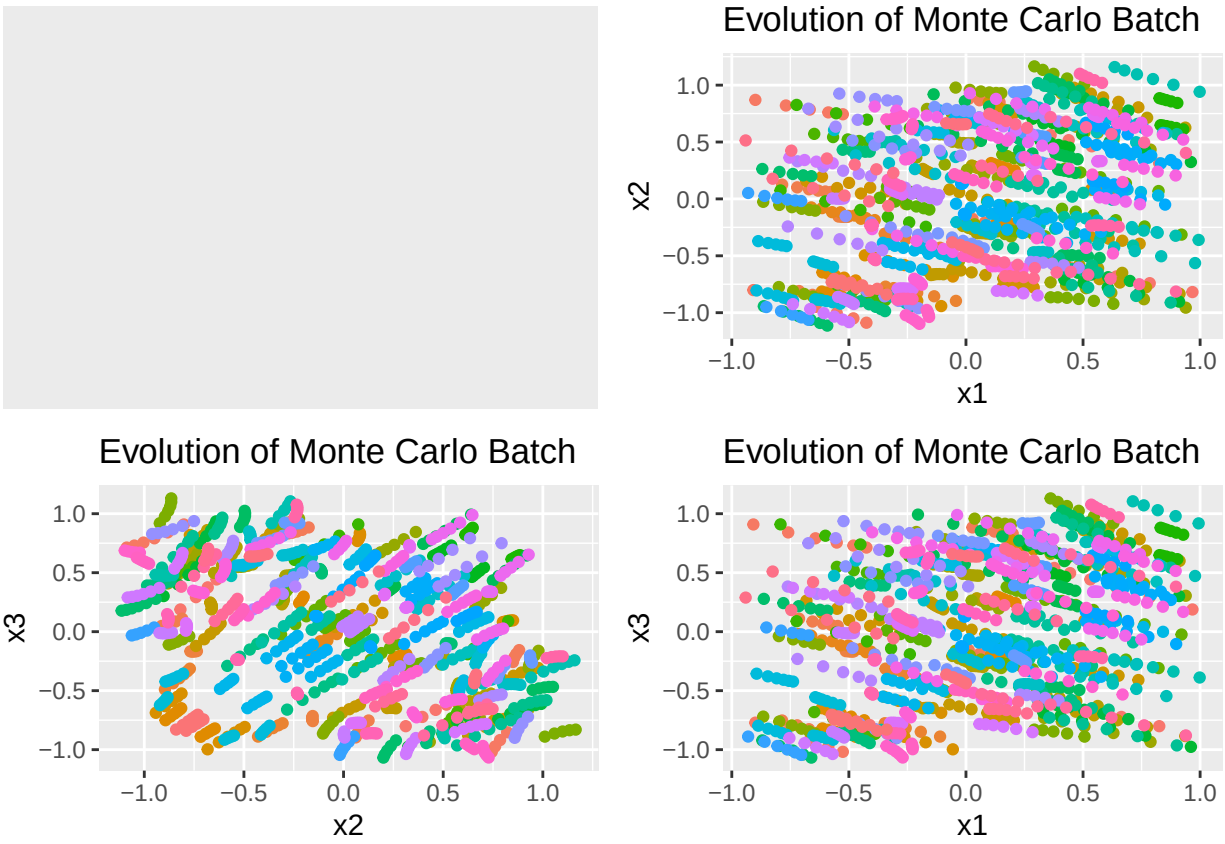








```
set.seed(27)
M <- 3
P <- RM_stoch(M, symm = T, sparsity = T)
# Set batch parameters
B <- 100
# Create batch
batch <- make_batch(M = M, B = B)
# Set evolution parameters
steps <- 10
# Evolve batch
evolved_batch <- evolve_batch(batch, steps, with_steps = T)
```

```
batch_3d_plot(evolved_batch)
```




```
#batch_2d_customplot(evolved_batch, n1 = 1, n2 = 2) + transition_time(V4)
```