1 Implementation

1.1 How did we get our data?

To get the data for our simulation, we used the so-called Facebook Graph API. (reference) What does Graph API mean? A Facebook Grpah API is a programming tool designed to support better access to conventions on the Facebook social media platform.[1] It allows you to extract information that is published on facebook, which is exactly what we needed.

Unfortunately, there is no implementation for Matlab, so we programmed the data-fechting in python.

1.2 Coding

1.2.1 Get access

For the coding in python we used the facebook sdk for python[2].

First, one needs to get access to the social graph. Therefore, you need to enter an access token[3].

```
token = raw_input('Enter your Access Token: ')
anonym = input('Do you want to anonymize your Data? Yes (1) or No (0)? ')
graph = facebook.GraphAPI(token)
```

1.2.2 Get graph

Having enterd a valid token, we can now access all information this token provide us. For simplicty, we'll only discuss the implementation of the non-anonymized case in details. [For more details on the anonymization, look at subsection anonymization]

```
profile = graph.get_object("me")
friends = graph.get_connections("me", "friends")
```

In our case, we're interested in data about our own profile, as well as all available information we get from our friends. The object friends is nothing else but an array that contains all id's of our friends, and for each id, there is another array data, that contains all information about this specific friend.

```
friend_list = [friend['id'] for friend in friends['data']]
```

[1] http://www.techopedia.com/definition/28984/facebook-graph-api
[2] https://github.com/pythonforfacebook/facebook-sdk
[3] https://developers.facebook.com/docs/facebook-login/access-tokens/

With this simple for-loop, one can now get a list of all his friends. Since we're not primarly interesseted in our friends, but the connection in between them, we have to access at least the mutualfriends.

```
mutualfriends = graph.get_connections(tfriend, "mutualfriends")
```

The object mutualfriends provides us a list with all mutualfriends of ”me” and friend ”tfriend” (which is nothing else but one of the id's we have in the saved in "friend_list"

Repeat this proceedure for all friends, and we'll get exatly what we're looking for - a graph of all connection of our friends in between of them.

1.2.3 Get activity

1.3 Anonymization