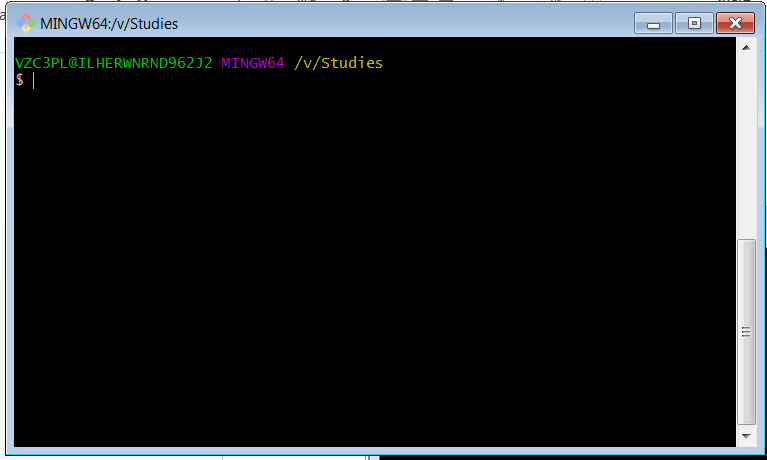
***Fresh start***

Procedure for when only a GitHub remote repository exists and no local workspace exists

1. Go to V:/Studies
2. Disconnect from GM network
3. Right click: GitBash here
4. Type: clear, to clear the screen

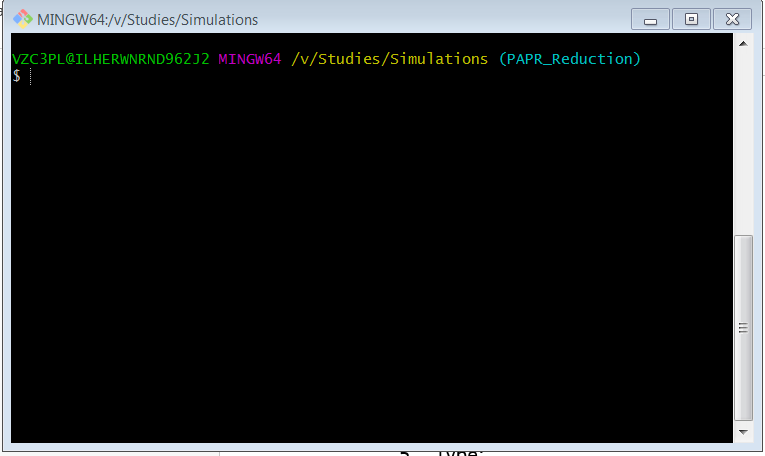


1. Type:

$ git clone -b PAPR\_Reduction https://github.com/liorkissos/VLF-Simulations.git ./Simulation

Comment: the destination (./Simlation in the example above) folder needs to not exist initially. The git command will create it

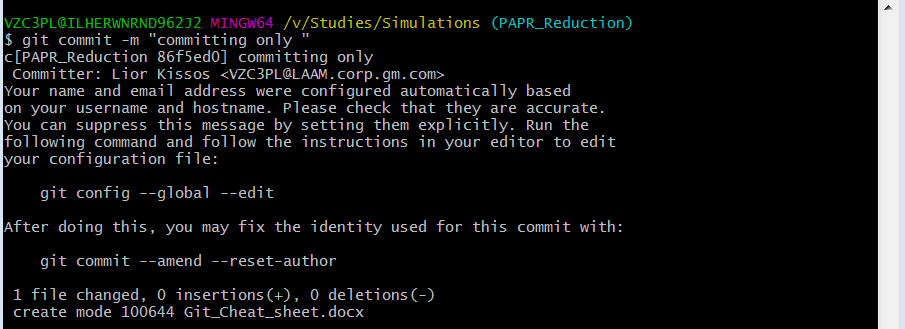
1. Enter
2. Connect to GM network and turn on Matlab
3. Work on the Matlab files: make sure the files do not have spaces in their names. When needed, add \_. E.g: readme\_Copy.txt instead of readme – Copy.xt
4. Go to V/Studies
5. Disconnect from GM network
6. Right click: GitBash here
7. Type: clear , to clear the screen



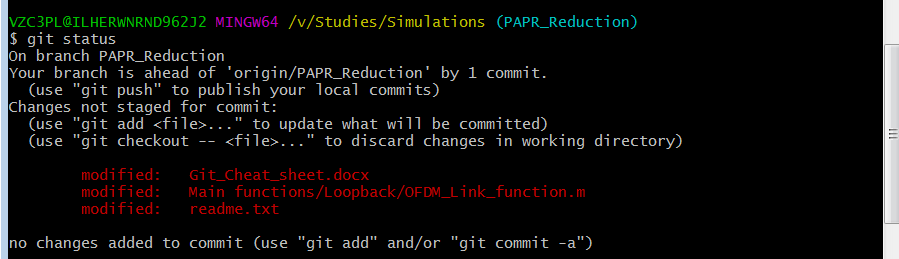
1. Type: git status

Check the changes

1. (Optional ) Type: git diff
2. (Optional) Type: git add <file name>. e.g: git add readme\_Copy.txt, to stage a file. Staging is adding the file to the list of tracked files
3. Type: git log, to see the history of commits
4. Type: git commit –m “message” , to commit the staged files that have changed from last commit (or from cloning)

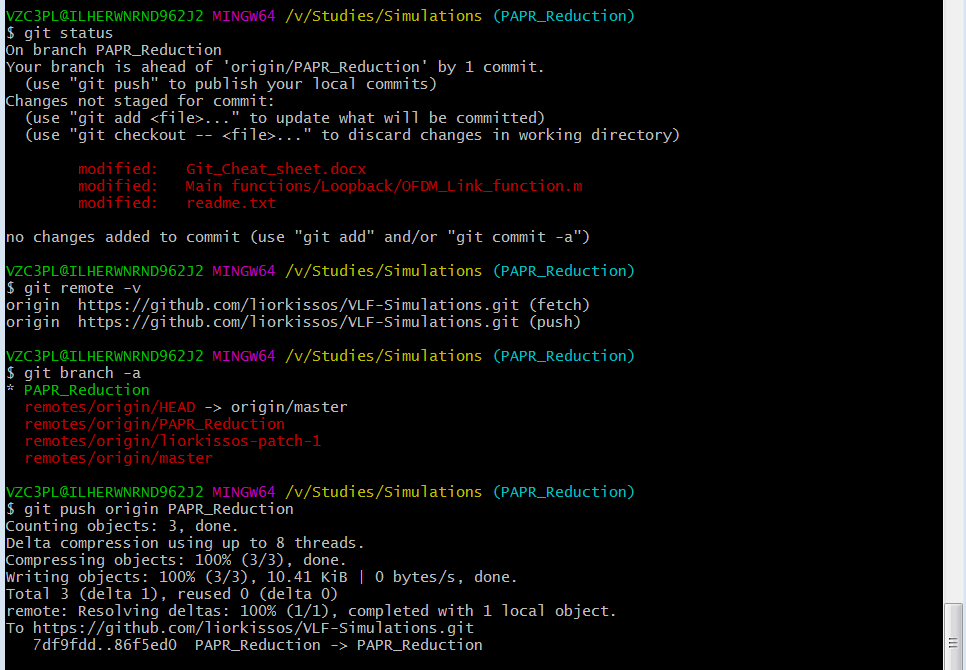


1. If the text editor opens type the message and then type ESC-> :q! (?)
2. Type: git status, to see some details about current status of local repository versus remote repository

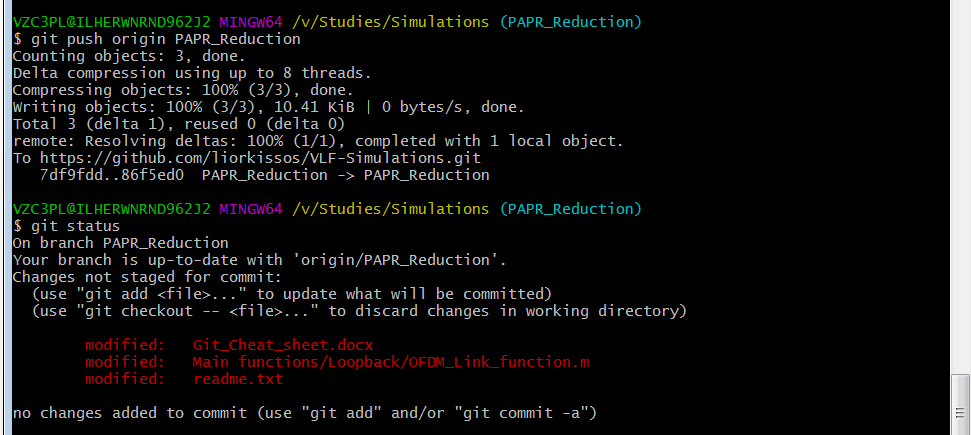


1. Type: git remote –v , to see what is the name of the remote repository (usually :”origin”)
2. Type: git branch –a, to see what branches are available (e.g; master or PAPR\_reduction)
3. Type: git push origin PAPR\_Reduction, (git push <remote repository name> <branch name>) to push into GitHub to the correct branch

In the example 3 files were changed, committed and pushed. You can see the in the test after the push command



After the push action, all is aligned



***Ongoing workflow***

Procedure for when 2 or more computers are working on the same workspace. In principle, if you do not work on the same workspace on the same time from the 2 PC’s, it is better not to follow the procedure below and simply push and clone every time you switch PC

1. git status

the local repository should be aligned with the remote tracking local repository

1. git branch –r

to see the names of the remote tracking local branches. They should be aligned with the last cloning fro the remote repository

1. git fetch

this command imports from remote repository to the remote tracking repository, as a median step before merging

1. git status

we should now notice that there are conflicts between the local repository and the remote tracking repository

1. git merge origin/master . do not do unless you are sure !!!!!!!!!

***Useful tricks***

1. **stage a file**

git add -A

stages all files

git add .

stages new and modified, without deleted

1. If we want to **change the default name** of “origin” given to the remote repository , and Allocate a new name to the GitHub URL, to save the need to type in the URL every time we need it;

git remote –v

we will see the current remote repository name (say “origin”)

git remote remove origin

then we will add the new nickname we want to the remote repoisitiory

git remote add Hello\_World\_Repo https://github.com/liorkissos/hello-world.git

to check

go remote –v

1. **If we want to unstage a file:**

git reset <file path>

e.g; git reset Lior3.txt (if we are already inside the directory that contains Lior3.txt)

1. **Text editor:**

To exit it:

Vim text editor: ESC->:wq

VI text editor: ESC-> :q!

Handling branches

**Watch branches**

git branch , local branches only

git branch –r, remote branches only

git branch –a, all branches

**Add a branch**

git branch My\_Branch

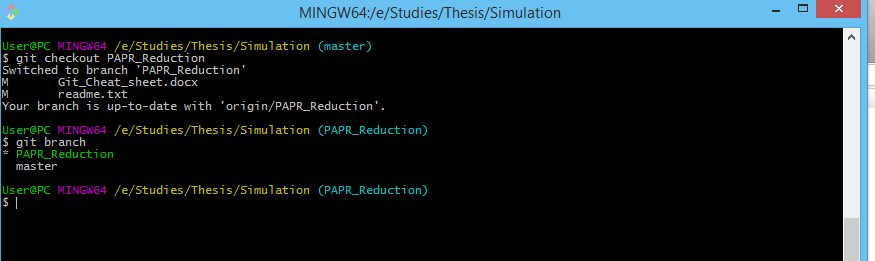
git branch –a

**switch to branch**

git checkout My\_Branch

git branch –a

until pushing the branching will exist only in the local repository



**delete branch**

git branch –d My\_Branch

git branch –a

**push the local branch to remote**

git push –u origin My\_Branch

git branch –a

this will create a new branch in the remote repository

**delete remote branch**

$ git push origin --delete My\_Branch

Structure

Local Repository

Remote

Tracking

Branches

(origin/master, origin/my\_branch…)

Local

Branches

(master, my\_branch\_…)

merge

pull

fetch

push

Remote Repository

Git pull is not recommended!!!!