Downloading and installing Julia

https://julialang.org/downloads/

Use the latest version 1.5.2. Run the following commands to download and extract the julia package:

```
wget https://julialang-s3.julialang.org/bin/linux/x64/1.5/julia-1.5.2-linux-
x86_64.tar.gz

tar -xzvf julia-1.5.2-linux-x86_64.tar.gz
```

Add julia to PATH (replace "pathtojulia" in the following command with the actual path on your machine):

```
export PATH=/pathtojulia/julia-1.5.2/bin:$PATH
```

Installing the LsqFit package

In your terminal, type

```
julia
```

which will open the julia REPL, and and in the PEPL, type

```
using Pkg
Pkg.add("LsqFit")
```

The installation may take more than 1 minute. After installing the LsqFit package, run the following command to test it is working:

```
using LsqFit  # this may take >1 minute to load

xdata=[ 1.0
        0.9410359787896052
        0.892747272010167
        0.8440028046803102
        0.7944607563872212
        0.768315877119944
        0.7444147421008809]

ydata = [ 0.0
        0.002892932000008841
        0.007156206999979986
```

```
         0.012909014999991086
         0.020529311000018424
         0.025222962999976062
         0.030964621000000963]

p0 = [0.0, 5.0, 0.0]

@. model(x, p) = (p[1]/p[2])*x^(-p[2]) + (p[1]-p[3])*x

fit = curve_fit(model, xdata, ydata, p0)

print(fit.param)
```

The last command should print an array of 3 floating numbers

```
[0.04396688697055344, 4.864867834353412, 0.0533354593802239]
```

You may exit the julia REPL using

```
exit()
```

Running xppcm calculations

Copy `xppcm-test.jl` and `input.jl` to a working folder. Modify `input.jl` according to the instructions in the file.

Make sure Gaussian09 or Gaussian16 is properly installed, and `g09` or `g16` will actually call the program. The script will use `g16` over `g09` if both are installed.

Then run the xppcm calculations (the following command uses 24 cores/threads for the job):

```
julia --threads 24 xppcm-test.jl
```

Below is an example PBS script for running the calculation on a cluster. Modify it according your cluster specifications. Important thing is to let the computing node know the paths to `g09`/`g16` and `julia`.

```
#!/bin/bash
#PBS -q parallel
#PBS -l nodes=1:ppn=24
#PBS -l mem=48gb
#PBS -l cput=24:00:00
#PBS -N xppcm

cd $PBS_O_WORKDIR

module load Gaussian/16

/scratch/user/julia-1.5.1/bin/julia --threads 24 xppcm-test.jl
```