Rapport de projet - phase 2

MTH8408

Oihan Cordelier, Oussama Mouhtal

Lien GitHub

Ce projet est accessible sur le dépot GitHub au lien suivant : https://github.com/oihanc/mth8408_projet.

```
using Pkg
Pkg.activate("projet_env_phase2")
Pkg.add("ADNLPModels")
Pkg.add("NLPModels")
Pkg.add("Krylov")
Pkg.add("LinearOperators")
Pkg.add("JSOSolvers")
Pkg.add("Plots")

Pkg.add("OptimizationProblems")  # collection + outils pour sélectionner les problèmes
include("subsolvers.jl")
# TODO: add CUTest
Pkg.add("SuiteSparseMatrixCollection")
Pkg.add("MatrixMarket")
using LinearAlgebra, NLPModels , ADNLPModels, Printf, Krylov, LinearOperators, SuiteSparseMatrixCollecti

using OptimizationProblems, OptimizationProblems.ADNLPProblems, JSOSolvers

using Plots
# gr(fmt = :png)
```

```
function get_mm(matrix_name)
  ssmc = ssmc_db()
  pb = ssmc_matrices(ssmc, "", matrix_name)
  fetch_ssmc(pb, format="MM")
  pb_path = fetch_ssmc(pb, format="MM")
  path_mtx = pb_path[1]
  A = MatrixMarket.mmread(joinpath(path_mtx, matrix_name * ".mtx"))
  #b = MatrixMarket.mmread(joinpath(path_mtx, matrix_name * "_b.mtx"))
  return A
end


function memory(n, p)
    @assert 1   p   n "p doit être entre 1 et n"
    # indices = [floor(Int, (i-1)*n/p) + 1 for i in 1:p]
    indices = round.(Int, range(1, n, length=p))    # the last memory = n (to perform bfgs)
    indices = unique(sort(indices))                 # au cas où n/p n'est pas exact
    return indices
```

```
end

function plot_save_graphes(A, b, name, listmem)


  (xcg,statscg) = cg(A, b; atol=1e-8, rtol=1e-8,history=true)
  gr()
  plot(statscg.residuals, label="||r|| cg", lw=2, yaxis=:log, xlabel="Itération")
  for mem in listmem
    println("mem= ", mem)
    (xlbfgs,statslbfgs) = lbfgs(A, b; atol=1e-18, rtol=1e-18, mem = mem)
    p = round(Int64, 100 * mem / n)
    if p > 0
      plot!(statslbfgs.residuals, label="||r|| lbfgs $(m = p)%", lw=2)
    end
  end

  savefig("CG_versus_lbfgs_$(name).pdf")
end
```

```
plot_save_graphes (generic function with 1 method)
```

```
A = get_mm("494_bus")
n,n = size(A)
println("n= ", n)

b = randn(eltype(A), n)
p=11
listmem = memory(n, p)

println("listmem= ", listmem)


plot_save_graphes(A, b, "494_bus_2", listmem)
```

```
n= 494
listmem= [1, 50, 100, 149, 198, 248, 297, 346, 395, 445, 494]
mem= 1
mem= 50
mem= 100
mem= 149
mem= 198
mem= 248
mem= 297
mem= 346
mem= 395
mem= 445
mem= 494
```

```
"/home/corde/mth8408/projet/phase_2/CG_versus_lbfgs_494_bus_2.pdf"
```

```
problems = ["fletchcr", "nondquar", "woods", "broydn7d", "sparsine"]

# OptimizationProblems.meta.name
```

```
intersect(OptimizationProblems.meta.name, problems)
```

```
5-element Vector{String}:
 "broydn7d"
 "fletchcr"
 "nondquar"
 "sparsine"
 "woods"
```