Simulation and Optimization
Lab 1

Chen Jiang Hang, Tomas Robenek, Marija Nikolic

Transport and Mobility Laboratory

February 25, 2014

Project Groups

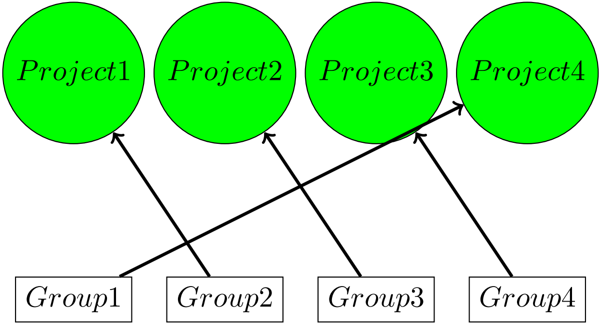
Projects

Project	Optimization algorithm Matlab coding
1	Preconditioned Projected Gradient Method
2	Interior Point Method
3	Augmented Lagrangian Method
4	Sequential Quadratic Programming Method

Groups

Group	Members
1	Milosavljevic Predrag,martand singhal,Almeida Samora Irene,Chiappino Pepe Anush
2	Stepanova Lidia,Markov Iliya Dimitrov,Perez Saez Francisco Javier,Fernandez Antolin Anna
3	Saeedmanesh Mohammadreze,kyriaki goulouti,Pisaroni Michele,Cuendet Gabriel Louis
4	stefano moret,Wallerand Anna Sophia,stephane bungener,Rujeerapaiboon Napat

Assignment


$Project1$
$Project2$
$Project3$
$Project4$
$Group1$
$Group2$
$Group3$
$Group4$

Grading

- 80% will be based on your Matlab coding and presentation on your assigned project;
- 20% will be based on your peer review (questions and challenges) on other groups' presentations.

Some tips for Matlab coding

Some tips

- Matrix multiplication: $y = (AB)x = A(Bx)$, which way is better?
- To solve the linear equation system $Ax = b$ where A is square, in Matlab, $x = A\backslash b$ is more efficient than $x = inv(A) * b$
- Matlab discriminates in favor of upper-right triangular matrices for inversion! If your matrix is lower-left triangular, first transpose it, invert the result, and transport back.
- Use built-in functions of Matlab as frequently as possible!

Exercises today

Line search

- Objective: find a step α^* such that both Wolfe's conditions are verified.
- Input:
  1. Function $f : \mathcal{R}^n \to \mathcal{R}$, continuously differentiable
  2. Gradient $\nabla f : \mathcal{R}^n \to \mathcal{R}^n$
  3. Vector $x \in \mathcal{R}^n$
  4. Descent direction d such that $\nabla f(x)^T d < 0$
  5. First approximation of the solution $\alpha_0 > 0$
  6. Parameters β_1 and β_2 such that $0 < \beta_1 < \beta_2 < 1$ (e.g., $\beta_1 = 10^{-4}$ and $\beta_2 = 0.99$)
  7. Parameter $\lambda > 1$

Line search

- Initialization: $i = 0$,$\alpha_l = 0$,$\alpha_r = +\infty$
- Iterations:
  1. If α_i verify both conditions, then $\alpha^* = \alpha_i$. STOP.
  2. If α_i violates Wolfe 1, then the step is too long and
     $$\alpha_r = \alpha_i$$
     $$\alpha_{i+1} = \frac{\alpha_l + \alpha_r}{2}$$
  3. If α_i verifies Wolfe 1 and violate Wolfe 2, then the step is too short and
     $$\alpha_l = \alpha_i$$
     $$\alpha_{i+1} = \begin{cases} \frac{\alpha_l+\alpha_r}{2}, & \text{if } \alpha_r < +\infty; \\ \lambda\alpha_i, & \text{otherwise.} \end{cases}$$
  4. $i = i + 1$

Line search

Try your code for the following example:

$$f(x_1, x_2) = x_1^2 + x_2^2$$

- Starting point: (1,1)
- $d = (-0.5, -1)$

Modified Cholesky Factorization

- Objective: Modify a matrix in order to make it positive-definite
- Input: Symmetric matrix $A \in \mathcal{R}^{n \times n}$
- Output: A lower triangular matrix L and $\tau >= 0$ such that $A + \tau I = LL^T$ is positive-definite

Modified Cholesky Factorization

Frobenius Norm of a matrix:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

- Initialization: $k = 0$; if $\min_i a_{ii} > 0$, then $\tau_k = 0$, otherwise $\tau_k = \frac{1}{2}\|A\|_F$
- Iterations:
  1. Compute the Cholesky factorization LL^T of $A + \tau_k I$
  2. If the factorization is successful, STOP.
  3. Else $\tau_{k+1} = \max\{2\tau_k, \frac{1}{2}\|A\|_F\}$
  4. $k = k + 1$

Matlab Hint:

- norm(A,'fro')
- chol
- try...catch...end

Modified Cholesky Factorization

Try your Matlab code on the following matrix:

$$A = \begin{bmatrix} 6 & 3 & 4 & 8 \\ 3 & 6 & 5 & 1 \\ 4 & 5 & 10 & 7 \\ 8 & 1 & 7 & 5 \end{bmatrix}$$