Graded Exercise 2 – Advanced Numerical Analysis

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Prof. D. Kressner
M. Steinlechner

Deadline for submission:

29.5.2015, 10.15 AM at the start of the exercise session.

Work on your own. Do not copy or distribute your answers and codes to other students in the class. Do not reuse any other code related to this homework. Cheating or attempts thereof will be handled according to the discplinary actions defined in the EPFL rules, see http://polylex.epfl.ch/discipline.

Guidelines for the submission:

- Answers to the theoretical questions can be handwritten or, preferably, in LaTeX.
- Print out your MATLAB code and hand it in, together with your other answers. Please make sure that is clear to which parts of the exercise you are referring to.
- Put all your MATLAB files into a Zip-file `FirstnameLastname.zip`
- Send the Zip-file to `michael.steinlechner@epfl.ch`, with the subject line "AdvancedNA: Exercise 2 submission"
- Make sure your MATLAB code runs without producing error messages. If your code does not run, you get zero points. For each plot and result that you state in your report, you have to include a MATLAB script that reproduces these results exactly.

1 ▶ Active set algorithm

a) Implement Algorithm 5.1, *Active set method for convex QP*, to solve a general quadratic program in the form of equation (5.20) in the script, with G being a symmetric positive semidefinite matrix.

```
function xstar = activeset( G, h, A, b, C, d, x0 )
```

Complete the template `activeset.m` provided on the lecture homepage.

Note that due to finite precision arithmetic, it is advisable to replace the equality check $\mathbf{p}_k = 0$ in lines 3 and 5 of the algorithm by $\|\mathbf{p}_k\| \leq 10^{-14}$. Furthermore, to safeguard against exceedingly long run times, add an upper bound `maxiter = 500` on the iteration count k.

b) Test your implementation on the following toy example:

$$\begin{aligned}
\min_{\mathbf{x}:=(x_1,x_2)^\mathsf{T}\in\mathbb{R}^2} \quad & (x_1-1)^2 + \left(x_2 - \frac{5}{2}\right)^2 \\
\text{subject to:} \quad & x_1 - 2x_2 + 2 \geq 0, \\
& -x_1 - 2x_2 + 6 \geq 0, \\
& -x_1 + 2x_2 + 2 \geq 0, \\
& x_1 \geq 0, \\
& x_2 \geq 0.
\end{aligned}$$

Transform the problem into the standard form given by equation (5.21) in the script. Start your algorithm from point $\mathbf{x}_0 = (2,0)^\mathsf{T}$ and corresponding active set $\mathcal{W}_0 = \{3,5\}$. You should obtain $\mathbf{x}_1 = \mathbf{x}_0$, $\mathbf{x}_2 = (1,0)^\mathsf{T}$, $\mathbf{x}_3 = \mathbf{x}_2$, $\mathbf{x}_4 = (1,1.5)^\mathsf{T}$. What is $\mathbf{x}_5$?

2 ▶ Finding the optimal shape of a tent

In this exercise, we want to optimize the shape of a pitched tent such that the surface energy is minimized. The energy consists of the elastic energy of the fabric and gravity acting on it.

Let the tent span a rectangular area $\Omega :=]0,1[\times]0,1[$. Let $u : \mathbb{R}^2 \to \mathbb{R}, (x,y) \mapsto u(x,y)$ describe the height of the tent at a point $(x,y) \in \Omega$. As the sides of the tent are fixed to ground by pegs, we have the boundary condition $u(x) = 0,\ \forall x \in \partial\Omega$.

To measure the energy of the tent fabric as a functional of the height, we will use the following simplified energy functional:

$$E[u] = \underbrace{\frac{1}{2}\int_0^1 \int_0^1 \nabla u(x,y) \cdot \nabla u(x,y)\, dx\, dy}_{\text{elastic energy}} + \underbrace{\mu \int_0^1 \int_0^1 u(x,y)\, dx\, dy}_{\text{gravitational energy}},$$

where $\mu > 0$ is a constant describing the graviational acceleration. We have assumed that the total mass is equal to one and is uniformly distributed over the fabric.

The shape of the tent is constrained by the position of 5 poles centered around the positions (x_k, y_k), $k = 1, 2, \ldots, 5$, with height p_i and thickness r which yield lower bounds for u at their positions. Furthermore, for obvious reasons, the height of the tent cannot be negative, $u(x,y) \geq 0$ for all points $(x,y) \in \Omega$. Hence, we have the following constraint function

$$g(x,y) = \begin{cases} p_k & \text{if } x_k - r \leq x \leq x_k + r, \quad y_k - r \leq y \leq y_k + r, \\ 0 & \text{otherwise.} \end{cases}$$

In total, we obtain the following constrained minimization problem:

$$\begin{aligned} \min_{u \in \Omega} \quad & E[u] \\ \text{subject to:} \quad & u(x,y) \geq g(x,y), \qquad (x,y) \in \Omega, \\ & u(x,y) = 0, \qquad\quad\;\; (x,y) \in \partial\Omega. \end{aligned} \tag{1}$$

Let us discretize the problem using a uniform grid with $(n+2) \times (n+2)$ points given by $(x_i, y_i) = (ih, ih)$, $h = 1/(n+1)$, $i = 0, \ldots, (n+1)$. As the values on the boundary $\partial\Omega$ are zero by the Dirichlet condition, we only have n^2 degrees of freedom corresponding to the value of u on the n^2 inner grid points, described by the vector $\mathbf{u} \in \mathbb{R}^{n^2}$.

Let $\mathbf{d}$ be the vector representing the constraint function on the grid points, see Figure 1 for a plot. It is possible to show that the minimization problem (1) can then be cast into the following quadratic program:

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{R}^{n^2}} \quad & \frac{1}{2}\mathbf{u}^\mathsf{T} G \mathbf{u} + \mathbf{u}^\mathsf{T}\mathbf{h} \\ \text{subject to:} \quad & \mathbf{u} \geq \mathbf{d}, \quad \textit{(elementwise)} \end{aligned} \tag{2}$$

where G is the $n^2 \times n^2$ 2D-Laplace matrix,

$$G = I_n \otimes L + L \otimes I_n, \quad L = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \text{ with } I_n, L \in \mathbb{R}^{n \times n},$$

and $\mathbf{h} = (\mu h^2, \ldots, \mu h^2)^\mathsf{T} \in \mathbb{R}^{n^2}$. We refer to the course on partial differential equations for more details on the discretization of PDEs.

a) Is the quadratic program (2) convex?

Hint: *Check Wikipedia for the properties of the Kronecker product and use that the eigenvalues of L are given by*

$$\lambda_k = 4\left(\sin\left(\frac{\pi k}{2(n+1)}\right)\right)^2, \quad k = 1, \ldots, n$$

b) Apply your implementation of the active set algorithm to solve the quadratic program (2). The Matlab file `tent.m` from the hompage provides a template which handles the setup of the problem and the plotting of the solution. Fill in the missing part.

   If you have implemented everything correctly, you should obtain a plot as depicted in Figure 2.

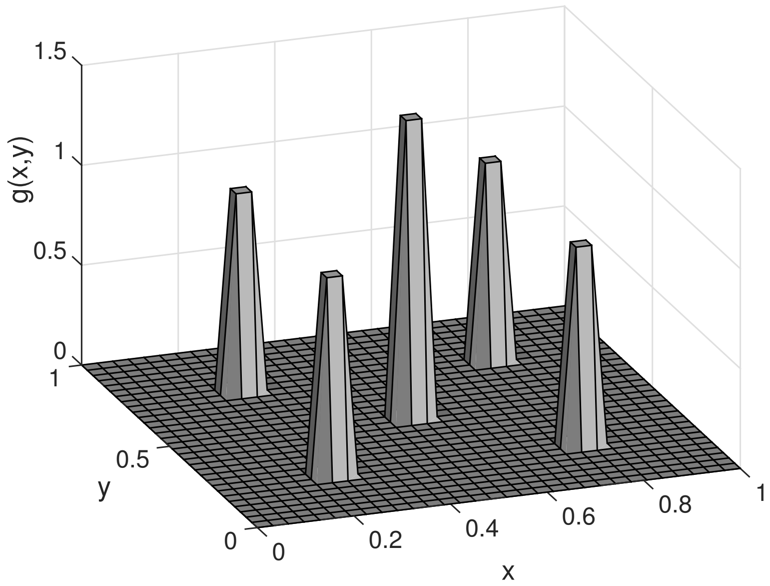

Figure 1: The constraint function $g(x, y)$, which forms a lower bound for the cost function, corresponds to the ground plane and the 5 tent poles of height p_k. Here, we have discretized the domain Ω using 32×32 grid points ($n = 30$).

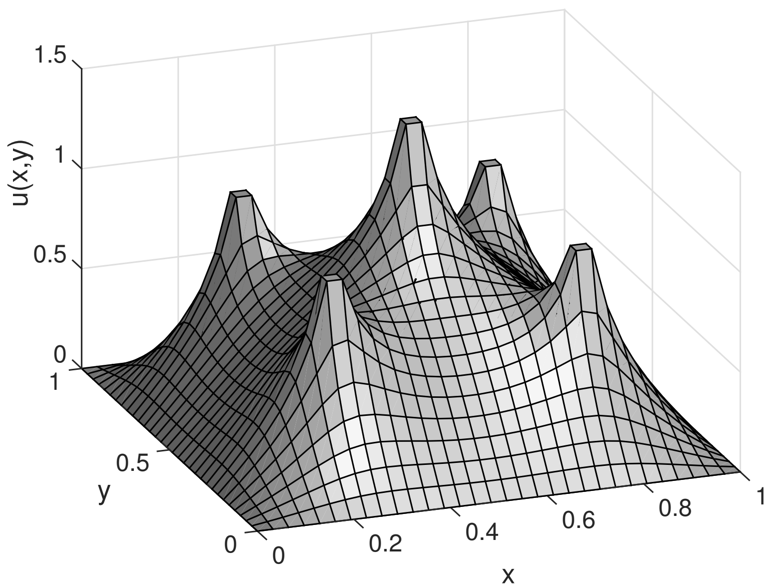

Figure 2: The resulting optimal tent surface $u(x, y)$, discretized on the domain Ω using 32×32 grid points ($n = 30$).