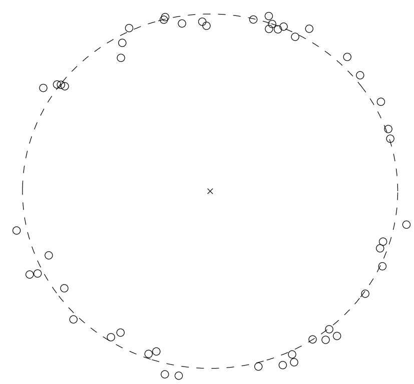
Additional problems for homework #1

1. *Least-squares fit of a circle to points.* In this problem we use least squares to fit a circle to given points (u_i, v_i) in a plane, as shown in the figure.



The variables (u_c, v_c) denote the center of the circle and R its radius. A point (u, v) is on the circle if $(u - u_c)^2 + (v - v_c)^2 = R^2$. We formulate the fitting problem as an optimization problem

$$\text{minimize} \quad \sum_{i=1}^{m} \left((u_i - u_c)^2 + (v_i - v_c)^2 - R^2\right)^2$$

with three variables u_c, v_c, R.

(a) Show that the problem can be written as a linear least-squares problem

$$\text{minimize } \|Ax - b\|_2^2 \tag{1}$$

if we make a change of variables and use as variables

$$x_1 = u_c, \qquad x_2 = v_c, \qquad x_3 = u_c^2 + v_c^2 - R^2.$$

(b) Use the normal equations $A^T Ax = A^T b$ of the least-squares problem to show that the optimal solution $\hat{x}$ of the least-squares problem satisfies

$$\hat{x}_1^2 + \hat{x}_2^2 - \hat{x}_3 \geq 0.$$

This is necessary to compute $R = \sqrt{\hat{x}_1^2 + \hat{x}_2^2 - \hat{x}_3}$ from $\hat{x}$.

(c) Test your formulation on the problem data in the file `circlefit.m` on the course website. The commands

```
circlefit;
plot(u, v, 'o');
axis equal
```

will create a plot of the $m = 50$ points (u_i, v_i) in the figure.

Use the MATLAB command `x = A \ b` to solve the least squares problem (1).

2. This problem is an introduction to the MATLAB software package CVX that will be used in the course. CVX can be downloaded from `www.cvxr.com`.

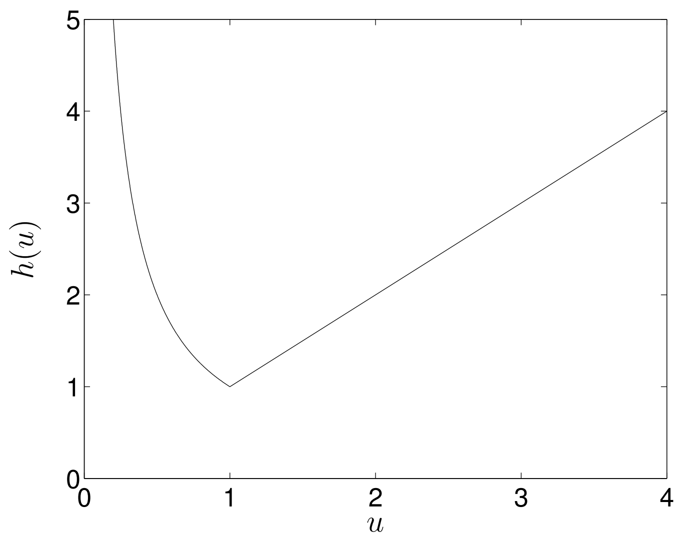
We consider the illumination problem of lecture 1. We take $I_{\mathrm{des}} = 1$ and $p_{\max} = 1$, so the problem is

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m, \end{array} \tag{2}$$

with variable $p \in \mathbf{R}^m$. As mentioned in the lecture, the problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} h(a_k^T p) \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m, \end{array} \tag{3}$$

where $h(u) = \max\{u, 1/u\}$ for $u > 0$. The function h, shown in the figure below, is nonlinear, nondifferentiable, and convex.



To see the equivalence between (2) and (3), we note that

$$\begin{aligned} f_0(p) &= \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ &= \max_{k=1,\ldots,n} \max\{\log(a_k^T p), \log(1/a_k^T p)\} \end{aligned}$$

$$
\begin{aligned}
&= \log \max_{k=1,\ldots,n} \max\{a_k^T p, 1/a_k^T p\} \\
&= \log \max_{k=1,\ldots,n} h(a_k^T p),
\end{aligned}
$$

and since the logarithm is a monotonically increasing function, minimizing f_0 is equivalent to minimizing $\max_{k=1,\ldots,n} h(a_k^T p)$.

The problem data are given in the file `illum_data.m` posted on the course website. Executing this file in MATLAB creates the $n \times m$-matrix A (which has rows a_k^T). There are 10 lamps ($m = 10$) and 20 patches ($n = 20$).

Use the following methods to compute five approximate solutions and the exact solution, and compare the answers (the vectors p and the corresponding values of $f_0(p)$).

(a) *Equal lamp powers.* Take $p_j = \gamma$ for $j = 1, \ldots, m$. Plot $f_0(p)$ versus γ over the interval $[0, 1]$. Graphically determine the optimal value of γ, and the associated objective value. The objective function $f_0(p)$ can be evaluated in MATLAB as `max(abs(log(A*p)))`.

(b) *Least-squares with saturation.* Solve the least-squares problem

$$
\text{minimize} \quad \sum_{k=1}^{n} (a_k^T p - 1)^2 = \|Ap - \mathbf{1}\|_2^2.
$$

If the solution has negative coefficients, set them to zero; if some coefficients are greater than 1, set them to 1. Use the MATLAB command `x = A \ b` to solve a least-squares problem (minimize $\|Ax - b\|_2^2$).

(c) *Regularized least-squares.* Solve the regularized least-squares problem

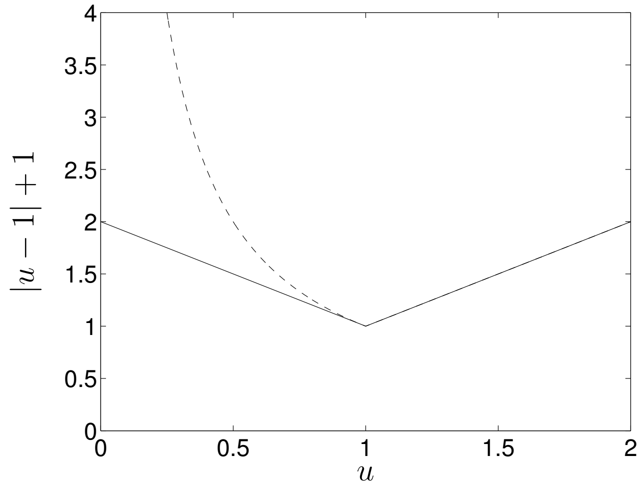
$$
\text{minimize} \quad \sum_{k=1}^{n} (a_k^T p - 1)^2 + \rho \sum_{j=1}^{m} (p_j - 0.5)^2 = \|Ap - \mathbf{1}\|_2^2 + \rho \, \|p - (1/2)\mathbf{1}\|_2^2 \,,
$$

where $\rho > 0$ is a parameter. Increase ρ until all coefficients of p are in the interval $[0, 1]$.

(d) *Chebyshev approximation.* Solve the problem

$$
\begin{array}{ll}
\text{minimize} & \max_{k=1,\ldots,n} |a_k^T p - 1| = \|Ap - \mathbf{1}\|_\infty \\
\text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m.
\end{array}
$$

We can think of this problem as obtained by approximating the nonlinear function $h(u)$ by a piecewise-linear function $|u - 1| + 1$. As shown in the figure below, this is a good approximation around $u = 1$.



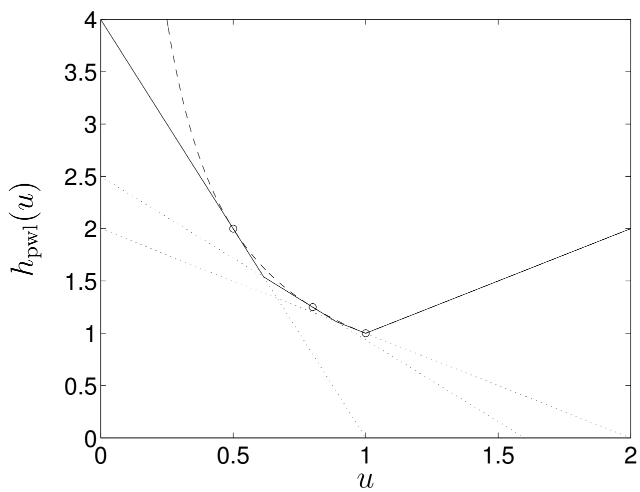
This problem can be converted to a linear program and solved using the MATLAB function `linprog`. It can also be solved directly in CVX, using the expression `norm(A*p - 1, inf)` to specify the cost function.

(e) *Piecewise-linear approximation.* We can improve the accuracy of the previous method by using a piecewise-linear approximation of h with more than two segments. To construct a piecewise-linear approximation of $1/u$, we take the pointwise maximum of the first-order approximations

$$h(u) \approx 1/\hat{u} - (1/\hat{u}^2)(u - \hat{u}) = 2/\hat{u} - u/\hat{u}^2,$$

at a number of different points $\hat{u}$. This is shown below, for $\hat{u} = 0.5,\ 0.8,\ 1$. In other words,

$$h_{\mathrm{pwl}}(u) = \max\{u,\ \frac{2}{0.5} - \frac{1}{0.5^2}u,\ \frac{2}{0.8} - \frac{1}{0.8^2}u,\ 2 - u\}.$$



Solve the problem

$$\begin{array}{ll}\text{minimize} & \max_{k=1,\ldots,n} h_{\rm pwl}(a_k^T p) \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m\end{array}$$

using `linprog` or CVX.

(f) *Exact solution.* Finally, use CVX to solve

$$\begin{array}{ll}\text{minimize} & \max_{k=1,\ldots,n} \max(a_k^T p, 1/a_k^T p) \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m.\end{array}$$

Use the CVX function `inv_pos()` to expresss the function $f(x) = 1/x$ with domain $\mathbf{R}_{++}$.