Assignment 3
Stochastic optimization
v1.1

5DA001 — Non-linear optimization

The deadline for this assignment can be found at:
http://www8.cs.umu.se/kurser/5DA001/HT16/timetable.html
(Link Planning and Readings on the course homepage.)

- The submission should consist of:
  - The complete report, including a front page with the following information:
    1. Your name (and of your colleague if you work in pairs).
    2. The course name.
    3. The assignment number.
    4. Your username(s) at the Department of Computing Science.
    5. The version of the submission (in case of re-submissions).
  - An appendix with the source code.

- To simplify feedback, the report, except appendices, must have page numbers and each section should be numbered.

- If you submit a report with linked references (e.g. written in \LaTeX), please verify that references are ok and not “figure ??”.

- It should be possible to understand your report without knowing the specification in detail. Thus, it is recommended you start your report with a short summary of the specification.

- Your report should be submitted as a pdf file uploaded via the https://www8.cs.umu.se/~labres/py/handin.cgi page, also available via the results link at the bottom left of the course home page.

- Furthermore, the source code should be available in a folder called edu/5da001/assN in your home folder, where N is the assignment number. You will probably have to create the folder yourself.

- The submitted code should be Matlab-compatible. If you develop your code in Octave, test your code in Matlab before submitting it!

- Auxiliary code and data needed for this assignment will be placed at http://www8.cs.umu.se/kurser/5DA001/HT16/assignment3/.
1 Introduction

The goal of this assignment is to give you experience with the RANSAC algorithm.

2 Task

You may work by yourself or in pairs. The assignment consists of a first task that everyone should do and an second task where you can choose between two options.

2.1 First task

2.1.1 Implementation

Implement the Adaptive RANSAC algorithm as described in the lecture notes, applied to the line fitting problem. The data set contains of a point set $\{p_i\}$ with outliers. You may use the function `linedata` to generate suitable test data. An example of running the minimum working example in `linedata.m` is shown in Figure 1.

2.1.2 Experimentation

Evaluate your algorithm on data sets with a varying degree of outliers $\epsilon = 0.1, 0.2, \ldots, 0.9$. Is there a cutoff where the algorithm fails? What if your data
set contains two lines + outliers? Can you extend your algorithm to handle two or more lines? Illustrate your evaluations in the report.

2.2 Second task

Choose among the following two tasks:

2.2.1 RANSAC, TNG

Implement an “improvement” to the basic RANSAC algorithm from the first task. Suggest the improvement yourself or google the internet for one (there are plenty). Apply it to the line fitting problem from the first task. Show how the results differ from the “vanilla” RANSAC.

2.2.2 Rigid body fitting

Modify the RANSAC algorithm from the first task to handle the 2D rigid body fitting problem. In this case, the data set is a set of (potentially incorrectly) matched points \( \{p_i, q_i\} \). Evaluate your algorithm on data sets with a varying degree of outliers \( \epsilon = 0.1, 0.2, \ldots, 0.9 \). Is there a cutoff where the algorithm fails? You may use the function `rigiddata` to generate suitable test data. An example of running the minimum working example in `rigiddata.m` is shown in Figure 2.