# Homework sheet 1 2023-10-20

**Due date: 2023-05-23 08:59**

The goal of this homework, is to create your first iterative CT reconstruction. Here, we will focus on strictly convex smooth problems (such as least squares).

### Delivery format

- Create a folder called "hw03" under the "homework" folder in your git repository.
- Answer all questions to the related to the homework in a file called "README.md" in the above created folder. **The readme must be a complete and standalone solution**. You are highly encouraged to provide instructions on how to recreate your results (e.g. run this script with this dataset, or run through this Jupyter notebook).
- Put all scripts and images you've created to reproduce and comprehend your answers in the same folder. Examples for this, are scripts to create plots, visualization, certain reconstructions to compare different strengths and weaknesses and similar things.
- The concrete implementations of the homework, must be put into the "aomip" folder. You are free to choose the structure, layout and form of your implementation.
- Once you've finished your homework, commit your changes and create a git tag with `git tag -a "hw03" -m "Tag for homework 3"` and push your changes. This needs to be tagged and pushed to your "master" branch before the deadline ends. Please use this exact command.
- Finally, once you push the changes with the tag to your repository. Create a release in GitLab. You find it in your project, under the menu 'Deployments'.

### Note

Generally, we expect and reward both explorativ work and good software engineering. Hence, we highly encourage you to look at the provided resources, do your own research and try different things and play around with the tools provided to you during the course.

Without explicitly permissions you are not allowed to pull any other dependencies. If you need dependencies to load or import a dataset, please get in touch with us and let us know the limitations. If you need to open `TIFF` files, use the already included dependency `tifffile`. Use it with `tifffile.imread`. See some examples at `https://pypi.org/project/tifffile/#examples`. This can also handle 3D TIFF-files, which makes this quite handy.

## Homework 1: More gradient based methods

Now we can start looking into advanced gradient based methods. They mostly differ in the exact problem formulation, and their convergence rate.

### i) Nesterov's Methods

In the last homework, we implemented the basic gradient descent algorithm. Over the years there have been many different algorithm, which improve on the convergence of gradient descent.

One famous example is gradient descent with Nesterov's momentum. Another widely known algorithms is the Optimized Gradient Method (OGM) as proposed by Kim and Fessler (see references below).

Implement the OGM1 method as presented in the paper by Kim and Fessler. And apply the algorithm to X-ray CT. How does it compare to the vanilla gradient descent version?

### (Optional:) Fast Gradient Method

You can also implement the *FGM1* from the Kim and Fessler paper and compare it to the other two.

### ii) Landweber iteration

Landweber iteration is a special case of gradient descent for the least squares problem. The update step is given as

$$x_{k+1} = x_k - \lambda A'(Ax_k - b)$$

The update step is quite clear, you forward project your current estimate, compute the error residual with the measurement and then backproject that residual to receive the new estimate.

The basic landweber iteration converges for $0 < \lambda < \frac{2}{\sigma_1^2}$, where $\sigma_1$ is the largest singular value of $A$. The largest singular value of $A$ can be found using Power iterations. Using:

$$||C||_2 = \sigma_1(C)$$

$$||C'C||_2 = ||C||_2^2$$

we can use the the power iterations to compute the largest singular value of $A'A$. The update step for the power iterations are given by:

$$b_{k+1} = \frac{Cb_k}{||Cb_k||_2}$$

$\sigma_1$ is the given by $\sigma_1 = \frac{\langle b_n, Cb_n \rangle}{||b_n||_2}$. Some side notes: the initial vector for the power iterations should be non-zero and of unit length.

Implement the Landweber iterations and the power iterations to compute the largest singular value of $A$. Can you empirically validate the mathematical bound of the step length? Can you find a decent enough step length choice that gives reasonable reconstructions in many cases?

### iii) (Optional): SIRT

Landweber iteration can be further generalized to the following update step:

$$x_{k+1} = x_k - \lambda DA'M(Ax_k - b)$$

The above landweber iteration uses $D = M = I$. One interesting and often used case in the CT community is the *Simultaneous Iterative Reconstruction Technique* (SIRT). There,

$$D = \frac{1}{A'\mathbf{1}} \qquad M = \frac{1}{A\mathbf{1}}$$

where $\mathbf{1}$ is the one vector. $D$ and $M$ are diagonal matrices, where the diagonal entries correspond to the inverse of the column and row sums of $A$ respectively.

Intuitively, this adds a correction factor to the residual error and the back projection.

Compare SIRT to landweber iterations, how do they compare? Explore choosing the step length? Can you find anything on how to choose the step length for SIRT?

### iv) Conjugate Gradient

As you have implemented a couple if methods now, you will notice that they usually converge to good looking images only after a couple of 100 iterations, which can take some time. Conjugate Gradient (CG) is a popular method that converges relatively quickly (usually after 10s of iterations). However, CG solves the quadratic functions of the form:

$$f(x) = \frac{1}{2}x^T Ax + b \cdot x + x$$

which is equivalent to solve

$$Ax = b$$

You get here by setting the gradient of $f$ to zero and rearrange a little. However, $A$ needs to be symmetric positive definite (SPD), which for the use case of CT it rarely the case. But one can use the normal equation to rely on CG:

$$A'Ax = A'b$$

which is just a reformulation of setting the gradient of the least squares problem to zero. Now we can apply CG to the normal equation, as $A'A$ is symmetric positive definite.

In the references, you'll find a great resources by Jonathan Richard Shewchuk, which your are highly encourage to read.

Implement the algorithm given in `https://en.wikipedia.org/wiki/Conjugate_gradient_method#The_resulting_algorithm`. But note, that the $A$ in the link is $A'A$ for CT, and $b$ is $A'b$ CT. Just be careful with nomenclature.

Evaluate the performance of CG compared to the previous methods, both in accuracy and runtime. Does the promise of fewer iterations hold?

### v) (Optional): Tikhonov Reguarlization with CG

As stated above the Conjugate Gradient method can't solve the Tikhonov problem yet. However, with a little trick it is possible. We stack the operator on top.

Recall the Tikhonov problem:

$$\min_x \frac{1}{2}||Ax - b||_2^2 + \frac{\lambda}{2}||Lx||_2^2$$

then this is equivalent to solve

$$\min_x \frac{1}{2}|| \begin{bmatrix} A \\ \sqrt{\lambda}L \end{bmatrix} x - \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} ||_2^2$$

For $A \in \mathbb{F}^{m\times n}$ and $L \in \mathbb{F}^{l\times n}$, then $\begin{bmatrix} A \\ \sqrt{\lambda}L \end{bmatrix} \in \mathbb{F}^{(m+l)\times n}$.

This can now be derived similar as to how we have proceeded with CG. Implement or extend your previous routine to be able to solve generalized Tikhonov problems with CG. Also investigate and compare this to the previous algorithms.

References:

- Optimized first-order methods for smooth convex minimization, Donghwan Kim, Jeffrey A. Fessler (2015)
- An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Jonathan Richard Shewchuk (1994)

# Homework 2: Solving problems other than X-ray CT

So far we have mostly looked at X-ray CT. However, to make the course more interesting we will try to look at different modalities and test algorithms on these.

Today we will look at two different modalities: Image denoising and deblurring. So far all of the theory can be applied to any inverse problem, all we need is find the appropriate forward model.

For both of these methods, it is advisable to adapt (if you haven't already) the norms and functionals from last weeks exercise to be able to handle operators other than the X-ray operator.

## Denoising

For image denoising, the forward model is simply the identity operator. First implement 3 methods that add/apply 3 kinds of noise

- Gaussian noise
- Poisson Noise
- Salt and Pepper noise (randomly set values to the min or maximum)

Then find some suitable test images (they should be grayscale to get started), add noise, setup some problem formulation (e.g. Least Squares, Tikhonov) and solve it using any of the above implemented algorithms.

What kind of problem formulations are suitable and why? Which do you think yields best results and how well do they perform on different images with different noise levels?

## Deblurring

The next modality is deblurring. The blurring process can be described as a convolution of the input signal with the point spread function (PSF). As with the FBP, the convolution is usually implemented via the Fourier Transform. Implement the convolution and deconvolution. Where convolution is a multiplication of the input singal with the filter in the Fourier Space, and the deconvolution (i.e. the adjoint), is the division of the input signal with the filter in the Fourier Space. Convolve and deconvolve a test image with a Gaussian filter. Can you recover the original filter? Try again when you add noise to the blurred image. What is the result now?

Finally, also test the problem formulations and algorithms for deconvolution. Again, setup a problem solve it using any of the above algorithms and document how they perform.