Inverse Problems in Imaging
Lecture 4

Simon R. Arridge[1]

[1] Department of Computer Science, University College London, UK

UCL, Term 2

Outline

1. Iterative Methods
   - Gradient Descent and Landweber Method
   - Conjugate Gradients
   - Regularisation within Iterative Methods
   - LSQR

Outline

1. Iterative Methods
    - Gradient Descent and Landweber Method
    - Conjugate Gradients
    - Regularisation within Iterative Methods
    - LSQR

- When the dimension of the problem becomes large the SVD is an impractical tool, and direct inversion of A is often impossible.

- When the dimension of the problem becomes large the SVD is an impractical tool, and direct inversion of A is often impossible.
- Even storage of a matrix representation of A becomes difficult if the dimensions is very large.

- When the dimension of the problem becomes large the SVD is an impractical tool, and direct inversion of A is often impossible.
- Even storage of a matrix representation of A becomes difficult if the dimensions is very large.
- This leads to the requirement for iterative methods.

- When the dimension of the problem becomes large the SVD is an impractical tool, and direct inversion of A is often impossible.
- Even storage of a matrix representation of A becomes difficult if the dimensions is very large.
- This leads to the requirement for iterative methods.
- We first consider linear problems.

Consider the minimisation of a quadratic form

$$\frac{1}{2}||\mathrm{A}\boldsymbol{f} - \boldsymbol{g}||^2 \rightarrow \min \quad \equiv \quad \Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2}\left\langle \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}, \boldsymbol{f}\right\rangle - \left\langle \mathrm{A}^{\mathrm{T}}\boldsymbol{g}, \boldsymbol{f}\right\rangle \rightarrow \min \qquad (4.1)$$

Consider the minimisation of a quadratic form

$$\frac{1}{2}||\mathrm{A}\boldsymbol{f} - \boldsymbol{g}||^2 \to \min \quad \equiv \quad \Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2}\left\langle \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}, \boldsymbol{f}\right\rangle - \left\langle \mathrm{A}^{\mathrm{T}}\boldsymbol{g}, \boldsymbol{f}\right\rangle \to \min \qquad (4.1)$$

The gradient is

$$\nabla_{\boldsymbol{f}}\Phi(\boldsymbol{f}, \boldsymbol{g}) = \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} - \mathrm{A}^{\mathrm{T}}\boldsymbol{g} \qquad (4.2)$$

which coincides with the negative residual associated with the least-squares criterion

$$\boldsymbol{r} = \mathrm{A}^{\mathrm{T}}(\boldsymbol{g} - \mathrm{A}\boldsymbol{f}) \qquad (4.3)$$

Consider the minimisation of a quadratic form

$$\frac{1}{2}||\mathrm{A}\boldsymbol{f} - \boldsymbol{g}||^2 \to \min \quad \equiv \quad \Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2}\left\langle \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}, \boldsymbol{f}\right\rangle - \left\langle \mathrm{A}^{\mathrm{T}}\boldsymbol{g}, \boldsymbol{f}\right\rangle \to \min \tag{4.1}$$

The gradient is

$$\nabla_{\boldsymbol{f}}\Phi(\boldsymbol{f}, \boldsymbol{g}) = \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} - \mathrm{A}^{\mathrm{T}}\boldsymbol{g} \tag{4.2}$$

which coincides with the negative residual associated with the least-squares criterion

$$\boldsymbol{r} = \mathrm{A}^{\mathrm{T}}(\boldsymbol{g} - \mathrm{A}\boldsymbol{f}) \tag{4.3}$$

Given an approximation $\boldsymbol{f}_k$ for the solutions, in a neighbourhood of the point the functional Φ decreases most rapidly in the direction of the *negative* gradient, i.e. in the direction of the residual $\boldsymbol{r}$.

Consider the minimisation of a quadratic form

$$\frac{1}{2}||\mathrm{A}\boldsymbol{f}-\boldsymbol{g}||^2 \to \min \quad \equiv \quad \Phi(\boldsymbol{f},\boldsymbol{g}) = \frac{1}{2}\left\langle \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f},\boldsymbol{f}\right\rangle - \left\langle \mathrm{A}^{\mathrm{T}}\boldsymbol{g},\boldsymbol{f}\right\rangle \to \min \tag{4.1}$$

The gradient is

$$\nabla_{\boldsymbol{f}}\Phi(\boldsymbol{f},\boldsymbol{g}) = \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} - \mathrm{A}^{\mathrm{T}}\boldsymbol{g} \tag{4.2}$$

which coincides with the negative residual associated with the least-squares criterion

$$\boldsymbol{r} = \mathrm{A}^{\mathrm{T}}(\boldsymbol{g} - \mathrm{A}\boldsymbol{f}) \tag{4.3}$$

Given an approximation $\boldsymbol{f}_k$ for the solutions, in a neighbourhood of the point the functional Φ decreases most rapidly in the direction of the *negative* gradient, i.e. in the direction of the residual $\boldsymbol{r}$. Thus for small values τ, we can guarantee that a new solution

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k + \tau\boldsymbol{r}_k \tag{4.4}$$

will give a smaller value of Φ.

How do we choose τ ?

[1] More generally the minimisation of the one-dimensional functional $\phi(\tau)$ would be non-quadratic and would require a *line-search* method.

How do we choose τ ? It makes sense to minimise the one dimensional function

$$\phi(\tau) = \Phi(\boldsymbol{f}_k + \tau \boldsymbol{r}_k, \boldsymbol{g}) = \Phi(\boldsymbol{f}_k, \boldsymbol{g}) + \frac{\tau^2}{2} ||\mathsf{A}\boldsymbol{r}_k||^2 - \tau ||\boldsymbol{r}_k||^2 \,. \tag{4.5}$$

[1] More generally the minimisation of the one-dimensional functional $\phi(\tau)$ would be non-quadratic and would require a *line-search* method.

How do we choose τ ? It makes sense to minimise the one dimensional function

$$\phi(\tau) = \Phi(\boldsymbol{f}_k + \tau \boldsymbol{r}_k, \boldsymbol{g}) = \Phi(\boldsymbol{f}_k, \boldsymbol{g}) + \frac{\tau^2}{2}||\mathsf{A}\boldsymbol{r}_k||^2 - \tau||\boldsymbol{r}_k||^2 \,. \tag{4.5}$$

Since this is a quadratic function of τ we have immediately that

$$\tau_k = \frac{||\boldsymbol{r}_k||^2}{||\mathsf{A}\boldsymbol{r}_k||^2}\,, \tag{4.6}$$

which gives rise to the *steepest descent* update rule. [1]

[1] More generally the minimisation of the one-dimensional functional $\phi(\tau)$ would be non-quadratic and would require a *line-search* method.

How do we choose τ ? It makes sense to minimise the one dimensional function

$$\phi(\tau) = \Phi(\boldsymbol{f}_k + \tau \boldsymbol{r}_k, \boldsymbol{g}) = \Phi(\boldsymbol{f}_k, \boldsymbol{g}) + \frac{\tau^2}{2} ||\mathrm{A}\boldsymbol{r}_k||^2 - \tau ||\boldsymbol{r}_k||^2 \, . \tag{4.5}$$

Since this is a quadratic function of τ we have immediately that

$$\tau_k = \frac{||\boldsymbol{r}_k||^2}{||\mathrm{A}\boldsymbol{r}_k||^2} \, , \tag{4.6}$$

which gives rise to the *steepest descent* update rule. [1] We can see immediately that the residuals have the properties

- They can be obtained iteratively

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \tau_k \mathrm{A}^{\mathrm{T}} \mathrm{A} \boldsymbol{r}_k \, , \tag{4.7}$$

[1] More generally the minimisation of the one-dimensional functional $\phi(\tau)$ would be non-quadratic and would require a *line-search* method.

How do we choose τ ? It makes sense to minimise the one dimensional function

$$\phi(\tau) = \Phi(\boldsymbol{f}_k + \tau \boldsymbol{r}_k, \boldsymbol{g}) = \Phi(\boldsymbol{f}_k, \boldsymbol{g}) + \frac{\tau^2}{2}||\mathrm{A}\boldsymbol{r}_k||^2 - \tau||\boldsymbol{r}_k||^2 \,. \tag{4.5}$$

Since this is a quadratic function of τ we have immediately that

$$\tau_k = \frac{||\boldsymbol{r}_k||^2}{||\mathrm{A}\boldsymbol{r}_k||^2} \,, \tag{4.6}$$

which gives rise to the *steepest descent* update rule. [1] We can see immediately that the residuals have the properties

- They can be obtained iteratively

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \tau_k \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{r}_k \,, \tag{4.7}$$

- They satisfy an orthogonality condition

$$\langle \boldsymbol{r}_{k+1}, \boldsymbol{r}_k \rangle = 0 \,. \tag{4.8}$$

[1] More generally the minimisation of the one-dimensional functional $\phi(\tau)$ would be non-quadratic and would require a *line-search* method.

Since $A^{\mathrm{T}}A$ is positive-definite the method has a simple geometric interpretation.

Since $A^{\mathrm{T}}A$ is positive-definite the method has a simple geometric interpretation.

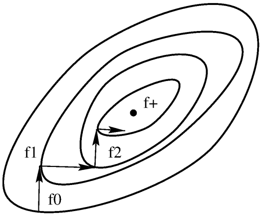
- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.

Since $\mathrm{A}^{\mathrm{T}}\mathrm{A}$ is positive-definite the method has a simple geometric interpretation.

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.

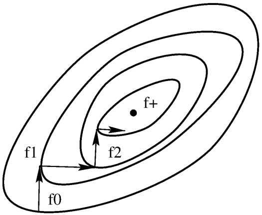
Since $A^{\mathrm{T}}A$ is positive-definite the method has a simple geometric interpretation.

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.



Since $A^T A$ is positive-definite the method has a simple geometric interpretation.

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.



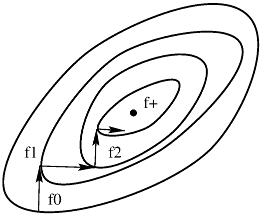
- The steepest-descent method works equally well for any optimisation problem sufficiently close to a local minimum.

Iterative Methods

Steepest Descent

Since $A^T A$ is positive-definite the method has a simple geometric interpretation.

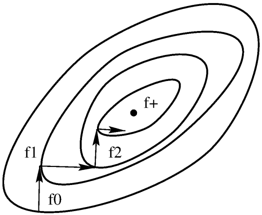
- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.



- The steepest-descent method works equally well for any optimisation problem sufficiently close to a local minimum.
- Stopping the iterations before minimisation has a regularising effect.

Since $A^{\mathrm{T}}A$ is positive-definite the method has a simple geometric interpretation.

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.



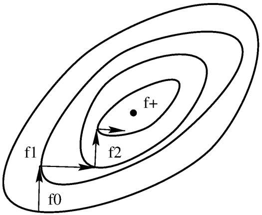
- The steepest-descent method works equally well for any optimisation problem sufficiently close to a local minimum.
- Stopping the iterations before minimisation has a regularising effect.
- It can be unreasonably slow : In the case of an anisotropic optimisation surface, the iterations can only descend very gradually to the minimum.

Iterative Methods

Steepest Descent

Since $A^{\mathrm{T}}A$ is positive-definite the method has a simple geometric interpretation.

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_2$ the line $\boldsymbol{f}_1 - \boldsymbol{f}_2$ is tangent to $\Phi(\boldsymbol{f}_2, \boldsymbol{g})$.



- The steepest-descent method works equally well for any optimisation problem sufficiently close to a local minimum.
- Stopping the iterations before minimisation has a regularising effect.
- It can be unreasonably slow : In the case of an anisotropic optimisation surface, the iterations can only descend very gradually to the minimum.
- It is almost always better to use a conjugate gradient method!

Two vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ are said to be conjugate with respect to a matrix B if

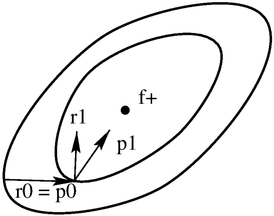
$$\langle \boldsymbol{p}, \mathrm{B}\boldsymbol{q} \rangle = 0$$

Then, given any level surface (ellipsoid) of the discrepency functional and a point on this surface, the direction of the conjugate gradient at this point is the direction orthogonal to the tangent plane w.r.t the matrix $\mathrm{A}^{\mathrm{T}}\mathrm{A}$.
We observe that a descent method such as described in the previous section builds the solution at the j^{th} iteration, from a vector space spanned by the set of vectors

$$\mathcal{K}^{(j)}(\mathrm{A}, \boldsymbol{g}) = \left\{ \mathrm{A}^{\mathrm{T}}\boldsymbol{g}, \mathrm{A}^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\boldsymbol{g}, \ldots \left(\mathrm{A}^{\mathrm{T}}\mathrm{A}\right)^{(j)} \mathrm{A}^{\mathrm{T}}\boldsymbol{g} \right\} \tag{4.9}$$

where it is assumed that all of the vectors in the set are mutually independent. This space is called the *Krylov subspace* of A.

- An optimal solution may be found by choosing at each iteration a direction that is the least-squares solution in this Krylov subspace.
- At the first iteration the subspace is of dimension one, and our only choice is the same as steepest descents.
- At subsequent directions we choose a direction that points towards the centre of an ellipsoid given by the intersection of the functional level surface with the Krylov subspace.



If P_j is the projection operator onto $\mathcal{K}^{(j)}(\mathrm{A}, \boldsymbol{g})$ then we need to solve

$$\begin{aligned} ||\mathrm{A}\mathrm{P}_j \boldsymbol{f} - \boldsymbol{g}||^2 \quad &\rightarrow \quad \min \\ \text{subject to} \quad & \quad \mathrm{P}_j \boldsymbol{f} = \boldsymbol{f} \end{aligned} \tag{4.10}$$

which is given by

$$\mathrm{P}_j^{\mathrm{T}} \mathrm{A}^{\mathrm{T}} \mathrm{A} \mathrm{P}_j \boldsymbol{f} = \mathrm{P}_j \mathrm{A}^{\mathrm{T}} \boldsymbol{g} \tag{4.11}$$

The CG method is based on the construction of two bases $\{\boldsymbol{r}_k\}$ and $\{\boldsymbol{p}_k\}$ as follows

-
$$\boldsymbol{r}_0 = \boldsymbol{p}_0 = \mathrm{A}^{\mathrm{T}}\boldsymbol{g} \tag{4.12}$$

- compute
$$\alpha_k = \frac{||\boldsymbol{r}_k||^2}{\langle \boldsymbol{r}_k, \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \rangle} \tag{4.13}$$

- compute
$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \tag{4.14}$$

- compute
$$\beta_k = \frac{\langle \boldsymbol{r}_{k+1}, \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \rangle}{\langle \boldsymbol{p}_k, \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \rangle} \tag{4.15}$$

- compute
$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k \tag{4.16}$$

Then the iterative scheme for the solutions is

$$\begin{aligned} \boldsymbol{f}_0 &= 0 \\ \boldsymbol{f}_{k+1} &= \boldsymbol{f}_k + \alpha_k \boldsymbol{p}_k \end{aligned} \tag{4.17}$$

The coefficients α_k, β_k are chosen so that the following orthogonality conditions hold $\langle \boldsymbol{r}_{k+1}, \boldsymbol{r}_k \rangle = 0\,, \quad \langle \boldsymbol{p}_{k+1}, \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \rangle = 0\,,$

i.e. the sequence of $\{\boldsymbol{r}_k\}$ are such that $\boldsymbol{r}_1 \perp \boldsymbol{r}_0$, $\boldsymbol{r}_2 \perp \boldsymbol{r}_1$, etc. and $\boldsymbol{p}_1$ is conjugate to $\boldsymbol{p}_0$, $\boldsymbol{p}_2$ is conjugate to $\boldsymbol{p}_1$ etc. Furthermore we can show that

- the set $\{\boldsymbol{r}_k, k = 0 \ldots j\}$ form an orthogonal basis of $\mathcal{K}^{(j)}(\mathrm{A}, \boldsymbol{g})$,
- the set $\{\boldsymbol{p}_k, k = 0 \ldots j\}$ form an $\mathrm{A}^{\mathrm{T}}\mathrm{A}$-orthogonal basis of $\mathcal{K}^{(j)}(\mathrm{A}, \boldsymbol{g})$,
- the set $\{\boldsymbol{r}_k, k = 0 \ldots j\}$ are the residuals

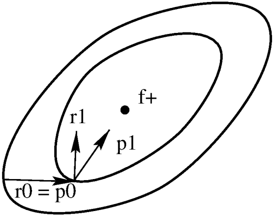
$$\boldsymbol{r}_k = \mathrm{A}^{\mathrm{T}}\boldsymbol{g} - \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}_k$$

Comment: the following alternative expression for α_k, β_k holds :

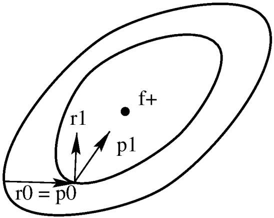
$$\alpha_k = \frac{||\boldsymbol{r}_k||^2}{\langle \boldsymbol{p}_k, \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{p}_k \rangle} \qquad \beta_k = \frac{||\boldsymbol{r}_{k+1}||^2}{||\boldsymbol{r}_k||^2}$$

- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.

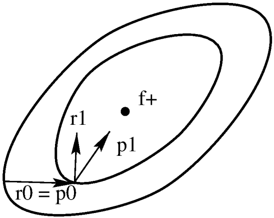
- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves to the centre of the ellipse given by the intersection of the level surface $\Phi(\boldsymbol{f}_1, \boldsymbol{g}) =$ constant with the plane spanned by $\boldsymbol{r}_0$ and $\boldsymbol{r}_1$. The direction of movement is $\boldsymbol{p}_1$.



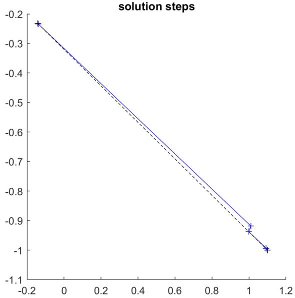
- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves to the centre of the ellipse given by the intersection of the level surface $\Phi(\boldsymbol{f}_1, \boldsymbol{g}) =$ constant with the plane spanned by $\boldsymbol{r}_0$ and $\boldsymbol{r}_1$. The direction of movement is $\boldsymbol{p}_1$.
- Then we move to centre of the ellipse given by the intersection of the level surface $\Phi(\boldsymbol{f}_2, \boldsymbol{g}) =$ constant with the three-dimensional space spanned by $\boldsymbol{r}_0$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$. The direction of movement is $\boldsymbol{p}_2$. And so on.

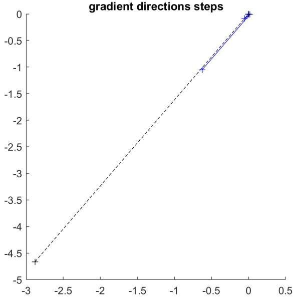


- Starting at $\boldsymbol{f}_0$ one moves perpendicular to the level set of $\Phi(\boldsymbol{f}_0, \boldsymbol{g})$ for a distance so that at the next point $\boldsymbol{f}_1$ the line $\boldsymbol{f}_0 - \boldsymbol{f}_1$ is tangent to $\Phi(\boldsymbol{f}_1, \boldsymbol{g})$.
- Then one moves to the centre of the ellipse given by the intersection of the level surface $\Phi(\boldsymbol{f}_1, \boldsymbol{g}) =$ constant with the plane spanned by $\boldsymbol{r}_0$ and $\boldsymbol{r}_1$. The direction of movement is $\boldsymbol{p}_1$.
- Then we move to centre of the ellipse given by the intersection of the level surface $\Phi(\boldsymbol{f}_2, \boldsymbol{g}) =$ constant with the three-dimensional space spanned by $\boldsymbol{r}_0$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$. The direction of movement is $\boldsymbol{p}_2$. And so on.
- If the problem is $n-$dimensional we reach the minimum in n steps.



Results from code : `RunGradToy`.



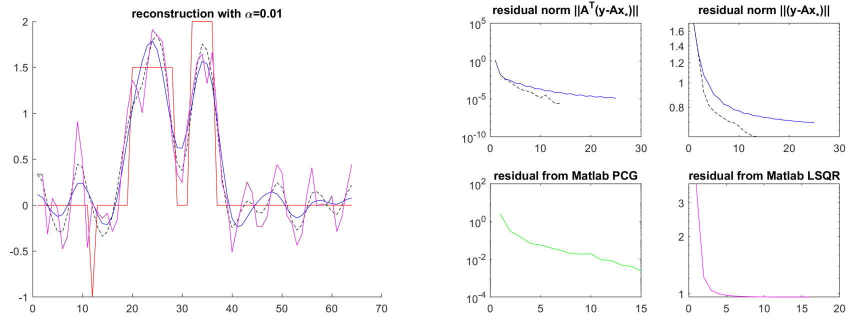


SD iterations

$$\begin{pmatrix} -0.1394 & 1.0103 & 0.9989 & 1.0927 & 1.0918 & 1.0994 & 1.0993 & 1.10 \\ -0.2341 & -0.9184 & -0.9375 & -0.9933 & -0.9949 & -0.9995 & -0.9996 & -1.00 \end{pmatrix}$$

CG iterations $\begin{pmatrix} -0.1432 & 1.1000 \\ -0.2317 & -1.0000 \end{pmatrix}$

Results from code : `RunGradLinBlur`.



Left: Results from solving normal equations $(A^TA + \alpha I)\boldsymbol{f} = A^T\boldsymbol{g}$ with SteepestDescent (blue) and CG (dashed black), and LSQR equations (magenta). Right : residual norms using Steepest Descent, CG, and Matlab functions `pcg` and `lsqr`.

- **Key point:** Early stopping of iterative methods has a regularising effect.

- **Key point:** Early stopping of iterative methods has a regularising effect.
- However, it is difficult to interpret this in terms of a penalty function and/or a Bayesian prior.

- **Key point:** Early stopping of iterative methods has a regularising effect.
- However, it is difficult to interpret this in terms of a penalty function and/or a Bayesian prior.
- The conjugate gradient method presented in eq. 4.11 is generating iterative approximations to the solution of the *Normal Equations*

$$(\mathrm{A}^{\mathrm{T}}\mathrm{A})\boldsymbol{f} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$$

- **Key point:** Early stopping of iterative methods has a regularising effect.
- However, it is difficult to interpret this in terms of a penalty function and/or a Bayesian prior.
- The conjugate gradient method presented in eq. 4.11 is generating iterative approximations to the solution of the *Normal Equations*

$$(\mathrm{A}^{\mathrm{T}}\mathrm{A})\boldsymbol{f} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$$

- We can also enforce a regularisation explicitly by choosing to solve

$$(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha\Gamma)\boldsymbol{f} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$$

- **Key point:** Early stopping of iterative methods has a regularising effect.
- However, it is difficult to interpret this in terms of a penalty function and/or a Bayesian prior.
- The conjugate gradient method presented in eq. 4.11 is generating iterative approximations to the solution of the *Normal Equations*

$$(\mathsf{A}^{\mathrm{T}}\mathsf{A})\boldsymbol{f} = \mathsf{A}^{\mathrm{T}}\boldsymbol{g}$$

- We can also enforce a regularisation explicitly by choosing to solve

$$(\mathsf{A}^{\mathrm{T}}\mathsf{A} + \alpha\Gamma)\boldsymbol{f} = \mathsf{A}^{\mathrm{T}}\boldsymbol{g}$$

- if $\Gamma = \mathsf{I}$ then the generated Krylov space is just an affine transformation of eq. 4.9 and therefore the same space of solutions are reached **but a different norm is minimised**.

- **Key point:** As far as the “black box” Krylov solver is concerned we are solving an equation

$$\mathrm{H}\boldsymbol{x} = \boldsymbol{y}$$

- **Key point:** As far as the "black box" Krylov solver is concerned we are solving an equation

$$\mathsf{H}\boldsymbol{x} = \boldsymbol{y}$$

- What does the "tolerance" monitor ? It is simply the "error"

$$\|\boldsymbol{y} - \mathsf{H}\boldsymbol{x}\|^2$$

- **Key point:** As far as the “black box” Krylov solver is concerned we are solving an equation
$$\mathrm{H}\boldsymbol{x} = \boldsymbol{y}$$
- What does the “tolerance” monitor ? It is simply the “error”
$$\|\boldsymbol{y} - \mathrm{H}\boldsymbol{x}\|^2$$
- But when using $\mathrm{H} = \mathrm{A}^{\mathrm{T}}\mathrm{A}$, $\boldsymbol{y} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$, $\boldsymbol{x} = \boldsymbol{f}$ this corresponds to an “image space error”
$$\left\|\mathrm{A}^{\mathrm{T}}\boldsymbol{g} - \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}\right\|^2$$

- **Key point:** As far as the "black box" Krylov solver is concerned we are solving an equation
$$\mathrm{H}\boldsymbol{x} = \boldsymbol{y}$$
- What does the "tolerance" monitor ? It is simply the "error"
$$\|\boldsymbol{y} - \mathrm{H}\boldsymbol{x}\|^2$$
- But when using $\mathrm{H} = \mathrm{A}^{\mathrm{T}}\mathrm{A}$, $\boldsymbol{y} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$, $\boldsymbol{x} = \boldsymbol{f}$ this corresponds to an "image space error"
$$\left\|\mathrm{A}^{\mathrm{T}}\boldsymbol{g} - \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}\right\|^2$$
- The function being minimised is
$$\Phi = \boldsymbol{f}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} - 2\boldsymbol{g}^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}$$

  Not what was wanted !

- **Key point:** As far as the "black box" Krylov solver is concerned we are solving an equation
$$\mathrm{H}\boldsymbol{x} = \boldsymbol{y}$$
- What does the "tolerance" monitor ? It is simply the "error"
$$\|\boldsymbol{y} - \mathrm{H}\boldsymbol{x}\|^2$$
- But when using $\mathrm{H} = \mathrm{A}^{\mathrm{T}}\mathrm{A}$, $\boldsymbol{y} = \mathrm{A}^{\mathrm{T}}\boldsymbol{g}$, $\boldsymbol{x} = \boldsymbol{f}$ this corresponds to an "image space error"
$$\left\|\mathrm{A}^{\mathrm{T}}\boldsymbol{g} - \mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}\right\|^2$$
- The function being minimised is
$$\Phi = \boldsymbol{f}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} - 2\boldsymbol{g}^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\mathrm{A}\boldsymbol{f}$$
  Not what was wanted !
- (One of the) differences between different Krylov solvers corresponds to the definition of what norm is being minimised

- For Inverse Problems, the most appropriate solver is **LSQR**.

- For Inverse Problems, the most appropriate solver is **LSQR**.
- We'll write this as

$$\boldsymbol{f}_{\alpha,\mathrm{L}} = \arg\min_{\boldsymbol{f}} \left[\Phi = \left\| \begin{pmatrix} \mathrm{A} \\ \sqrt{\alpha}\mathrm{L} \end{pmatrix} \boldsymbol{f} - \begin{pmatrix} \boldsymbol{g} \\ \mathbf{0} \end{pmatrix} \right\|^2 \right]$$

- For Inverse Problems, the most appropriate solver is **LSQR**.
- We'll write this as

$$\boldsymbol{f}_{\alpha,\mathrm{L}} = \arg\min_{\boldsymbol{f}} \left[\Phi = \left\| \begin{pmatrix} \mathrm{A} \\ \sqrt{\alpha}\mathrm{L} \end{pmatrix} \boldsymbol{f} - \begin{pmatrix} \boldsymbol{g} \\ \mathbf{0} \end{pmatrix} \right\|^2 \right]$$

- Expanding the norm shows that this is minimising

$$\begin{aligned} \Phi &= \boldsymbol{f}^{\mathrm{T}} \left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha \mathrm{L}^{\mathrm{T}}\mathrm{L} \right) \boldsymbol{f} - 2\boldsymbol{g}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} + \|\boldsymbol{g}\|^2 \\ &= \|\boldsymbol{g} - \mathrm{A}\boldsymbol{f}\|^2 + \alpha \|\mathrm{L}\boldsymbol{f}\|^2 \end{aligned}$$

- For Inverse Problems, the most appropriate solver is **LSQR**.
- We'll write this as

$$\boldsymbol{f}_{\alpha,\mathrm{L}} = \arg\min_{\boldsymbol{f}} \left[\Phi = \left\| \begin{pmatrix} \mathrm{A} \\ \sqrt{\alpha}\mathrm{L} \end{pmatrix} \boldsymbol{f} - \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{0} \end{pmatrix} \right\|^2 \right]$$

- Expanding the norm shows that this is minimising

$$\begin{aligned} \Phi &= \boldsymbol{f}^{\mathrm{T}} \left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha \mathrm{L}^{\mathrm{T}}\mathrm{L} \right) \boldsymbol{f} - 2\boldsymbol{g}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} + \|\boldsymbol{g}\|^2 \\ &= \|\boldsymbol{g} - \mathrm{A}\boldsymbol{f}\|^2 + \alpha \|\mathrm{L}\boldsymbol{f}\|^2 \end{aligned}$$

- This is what we want, provided that $\Gamma = \mathrm{L}^{\mathrm{T}}\mathrm{L} = \mathrm{C}^{-1}$. This can usually be constructed specifically e.g. $\mathrm{L} = \nabla \Rightarrow \mathrm{C}^{-1} = -\nabla^2$, or we find L by e.g. Cholesky Decomposition.

- For Inverse Problems, the most appropriate solver is **LSQR**.
- We'll write this as

$$\boldsymbol{f}_{\alpha,\mathrm{L}} = \arg\min_{\boldsymbol{f}} \left[\Phi = \left\| \begin{pmatrix} \mathrm{A} \\ \sqrt{\alpha}\mathrm{L} \end{pmatrix} \boldsymbol{f} - \begin{pmatrix} \boldsymbol{g} \\ \mathbf{0} \end{pmatrix} \right\|^2 \right]$$

- Expanding the norm shows that this is minimising

$$\begin{aligned} \Phi &= \boldsymbol{f}^{\mathrm{T}} \left(\mathrm{A}^{\mathrm{T}}\mathrm{A} + \alpha \mathrm{L}^{\mathrm{T}}\mathrm{L} \right) \boldsymbol{f} - 2\boldsymbol{g}^{\mathrm{T}}\mathrm{A}\boldsymbol{f} + \|\boldsymbol{g}\|^2 \\ &= \|\boldsymbol{g} - \mathrm{A}\boldsymbol{f}\|^2 + \alpha \|\mathrm{L}\boldsymbol{f}\|^2 \end{aligned}$$

- This is what we want, provided that $\Gamma = \mathrm{L}^{\mathrm{T}}\mathrm{L} = \mathrm{C}^{-1}$. This can usually be constructed specifically e.g. $\mathrm{L} = \nabla \Rightarrow \mathrm{C}^{-1} = -\nabla^2$, or we find L by e.g. Cholesky Decomposition.
- We have to supply the Matlab function `lsqr` with a "stacked" matrix and source vector e.g

$$\boldsymbol{f} = \mathtt{lsqr}\left(\begin{pmatrix} \mathrm{A} \\ \sqrt{\alpha}\mathrm{L} \end{pmatrix}, \begin{pmatrix} \boldsymbol{g} \\ \mathbf{0} \end{pmatrix}, \ldots \right)$$

Or the equivalent function handle. See documentation!