Inverse Problems in Space Physics

Bernd Inhester, May 2002

The program

Part I: Examples
- Image deblurring
- Tomography
- Radiative transfer inversion
- Helioseismology

Part II: Mainly direct methods
- Fourier transforms
- Singular value decomposition
- Backus-Gilbert or Mollifier method

Part III: Mainly Iterative methods
- Tikhonov regularization
- Linear iteration algorithms
- Nonlinear problems
- Artificial neural network and Gencode

Tikhonov regularization: Bayesian view

We consider a measurement of $\boldsymbol{f}$ as an experiment which modifies the probablity distribution of expected model parameters $\boldsymbol{g}$. The inversion problem is equivalent to the determination of the

$$\text{conditional probability} \quad p(\boldsymbol{g}|\boldsymbol{f}) \quad \text{of } \boldsymbol{g} \text{ given } \boldsymbol{f}$$

From Bayes theorem

$$p(\boldsymbol{g}|\boldsymbol{f}) = p(\boldsymbol{f}|\boldsymbol{g})\frac{p(\boldsymbol{g})}{p(\boldsymbol{f})}$$

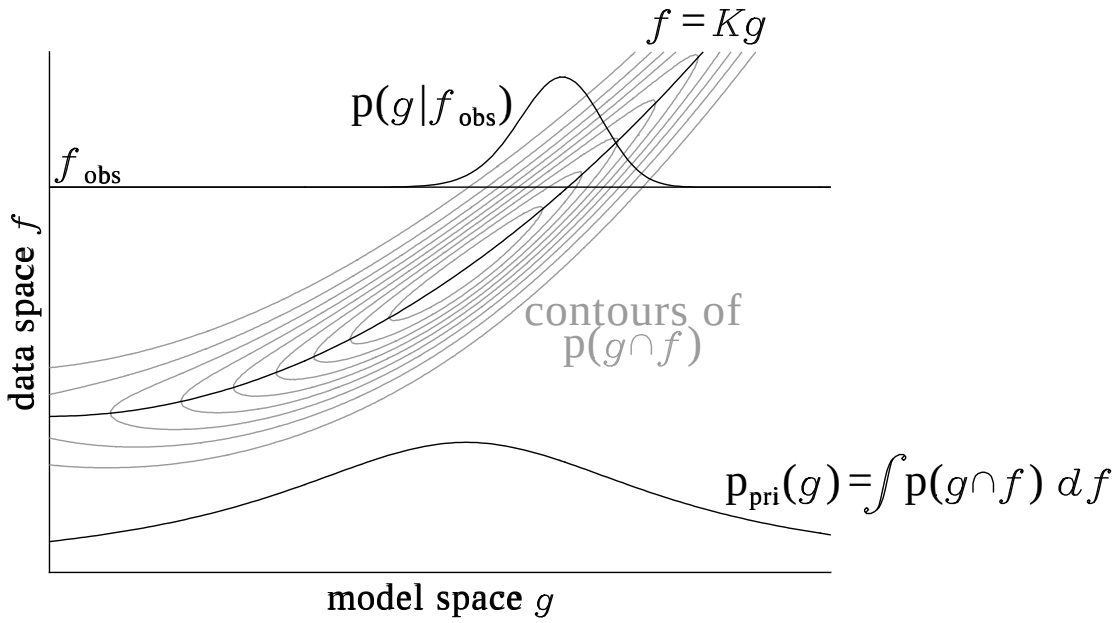


Illustration of Bayes theorem for a nonlinear problem $\boldsymbol{f} = \boldsymbol{K}(\boldsymbol{g})\boldsymbol{g}$ *and an observation* $\boldsymbol{f}_{\text{obs}}$. *The probability* $p(\boldsymbol{g})$ *in Bayes theorem is renamed* $p_{\text{pri}}(\boldsymbol{g})$.

Here $p(\boldsymbol{f}|\boldsymbol{g})$ is the probability that we measure $\boldsymbol{f}$ if the model $\boldsymbol{g}$ is given (forward problem). Typically,

$$p(\boldsymbol{f}|\boldsymbol{g}) \propto \exp - \sum_{i=1}^{n} \frac{1}{\sigma_i^2}[(\boldsymbol{K}\boldsymbol{g})_i - f_i]^2$$

for Gaussian noise with variance σ_i^2 of datum f_i. An alternative is Poisson noise often used in image processing.

Tikhonov regularization: A-priori knowledge

The probability $p(\boldsymbol{f})$ in Bayes theorem is assumed a constant and ignored if $\boldsymbol{f}$ is the observation we have just made.

The probability $p(\boldsymbol{g})$ in Bayes theorem is more interesting. Formally it is the pdf of the model $\boldsymbol{g}$ without any knowledge of $\boldsymbol{f}$ (a-priori pdf).

$$p(\boldsymbol{g}) \equiv p_{\text{pri}}(\boldsymbol{g}) = \int_{\text{data space}} p(\boldsymbol{g} \cap \boldsymbol{f})\, d\boldsymbol{f}$$

If we have no prior idea how $\boldsymbol{g}$ might look like, $p_{\text{pri}}(\boldsymbol{g})$ is uniform and another constant which we could ignore. Any other knowledge or even guess can be cast into $p_{\text{pri}}(\boldsymbol{g})$. Popular nontrivial choices are:

$$p_{\text{pri}}(\boldsymbol{g}) \propto \exp -c \sum_{i=1}^{m} (g_i - g_i^{\text{pri}})^2$$

($\boldsymbol{g}$ is probably close to a standard model $\boldsymbol{g}^{\text{pri}}$)

$$p_{\text{pri}}(\boldsymbol{g}) \propto \exp -c \sum_{i=1}^{m} (g_{i+1} - 2g_i + g_{i-1})^2$$

($\boldsymbol{g}$ is probably smooth and has minimal 2$^{\text{nd}}$ derivative)

$$p_{\text{pri}}(\boldsymbol{g}) \propto \exp -c \sum_{i=1}^{m} g_i \left[\ln \left(\frac{g_i}{g_i^{\text{pri}}} \right) - 1 \right]$$

($\boldsymbol{g}$ is positive and has max entropy deviation from standard)

The constant c expresses the certainty of the a-priori knowledge or assumption. Other forms of $\ln p_{\text{pri}}$ work as well as long as they are convex functions of $\boldsymbol{g}$.

Tikhonov regularization: Minimizing function

Inverse problem $\longrightarrow$ get the most probable model $\boldsymbol{g}$ from the maximum of conditional probability $p(\boldsymbol{g}|\boldsymbol{f})$ for given observations $\boldsymbol{f}$. Equivalently, we minimize $-\ln p(\boldsymbol{g}|\boldsymbol{f})$, or

$$\begin{aligned}\Phi(\boldsymbol{g}) &= \sum_{i=1}^{n}[(\boldsymbol{K}\boldsymbol{g})_i - f_i]^2 - \sigma^2 \ln p_{\rm pri}(\boldsymbol{g}) \\ &= \|\boldsymbol{K}\boldsymbol{g} - \boldsymbol{f}\|^2 + \mu\|\boldsymbol{g} - \boldsymbol{g}^{\rm pri}\|^2\end{aligned}$$

where we assumed the first a-priori pdf from the previous list and $\sigma_i^2 = \sigma^2$ independent of i.

• Instead of finding the root of a linear problem $\boldsymbol{K}\boldsymbol{g} - \boldsymbol{f}$ we now have to minimize a quadratic function $\Phi(\boldsymbol{g})$.

• $\Phi(\boldsymbol{g})$ has many names: merit function, cost function, misfit function, objective function. In my view it has something of a thermodynamic potential.

• The regularization parameter μ is the product of the certainty c of the a-priory model and the noise variance of the data σ^2.

The data term and the regularization term in $\Phi(\boldsymbol{g})$ can be combined formally because the a-priory information is equivalent to an observation.

$$\Phi(\boldsymbol{g}) = \|\widetilde{\boldsymbol{K}}\boldsymbol{g} - \widetilde{\boldsymbol{f}}\|^2 \quad \text{where} \quad \widetilde{\boldsymbol{K}} = \begin{pmatrix} \boldsymbol{K} \\ \sqrt{\mu}\mathbf{1} \end{pmatrix}, \quad \widetilde{\boldsymbol{f}} = \begin{pmatrix} \boldsymbol{f} \\ \sqrt{\mu}\boldsymbol{g}^{\rm pri} \end{pmatrix}$$

and $\mathbf{1}$ is the unit matrix in data space. When we discuss iterative schemes, we often mean $\widetilde{\boldsymbol{K}}$ and $\widetilde{\boldsymbol{f}}$ when we write $\boldsymbol{K}$ and $\boldsymbol{f}$.

• Since $\mathbf{1}$ has unit eigenvalues, the smallest SVD eigenvalue of $\widetilde{\boldsymbol{K}}$ is now $\sqrt{\mu}$ - any nullspace has vanished.

• Generalizations to other regularization operators $\mu\mathbf{1} \longrightarrow \mu\boldsymbol{R}$ are obvious. The final nullspace of $\widetilde{\boldsymbol{K}}$ is $\mathcal{N}(\boldsymbol{K}) \cap \mathcal{N}(\boldsymbol{R})$.

Tikhonov regularization: SVD analysis

The consequences of the Tikhonov regularization term which stems from the a-prori knowledge about the model can best be studied by an SVD analysis of Φ.

The minimum of $\Phi(\boldsymbol{g}) = \|\boldsymbol{K}\boldsymbol{g} - \boldsymbol{f}\|^2 + \mu\|\boldsymbol{g} - \boldsymbol{g}^{\mathrm{pri}}\|^2$ is identical with the root of its first derivative

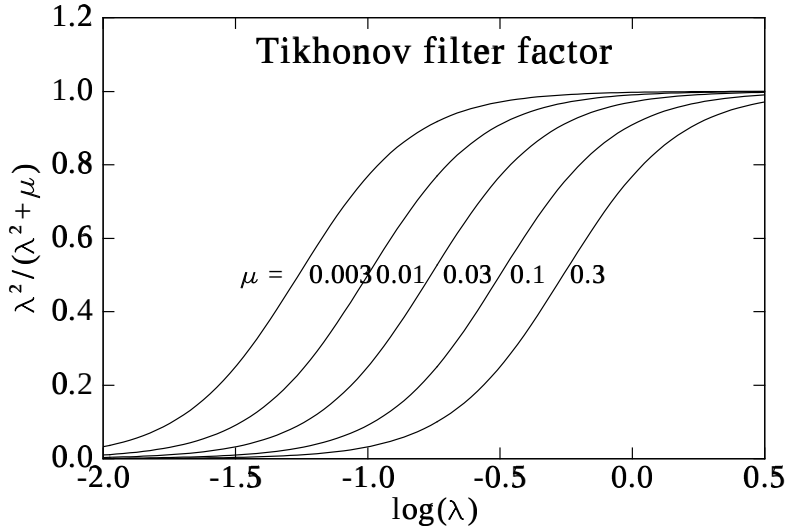
$$\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{g} - \boldsymbol{K}^T\boldsymbol{f} + \mu(\boldsymbol{g} - \boldsymbol{g}^{\mathrm{pri}}) = 0$$

Insertion of the SVD system $\boldsymbol{K} = \sum_{i=1}^{n} \boldsymbol{u}_i \lambda_i \boldsymbol{v}_i$

$$\sum_{i=1}^{n} \boldsymbol{v}_i \left[\lambda_i^2(\boldsymbol{v}_i \cdot \boldsymbol{g}) - \lambda_i(\boldsymbol{u}_i \cdot \boldsymbol{f}) + \mu(\boldsymbol{v}_i \cdot \boldsymbol{g}) - \mu(\boldsymbol{v}_i \cdot \boldsymbol{g}^{\mathrm{pri}})\right] = 0$$

$$\text{or} \quad \boldsymbol{g} = \sum_{i=1}^{n} \boldsymbol{v}_i(\boldsymbol{v}_i \cdot \boldsymbol{g}) = \sum_{i=1}^{n} \boldsymbol{v}_i \Big[\underbrace{\frac{F(\lambda_i)}{\lambda_i}(\boldsymbol{u}_i \cdot \boldsymbol{f})}_{\substack{\text{filtered} \\ \text{SVD} \\ \text{inverse}}} + \underbrace{\frac{\mu}{\lambda_i^2 + \mu}(\boldsymbol{v}_i \cdot \boldsymbol{g}^{\mathrm{pri}})}_{\substack{\sim \text{ nullspace} \\ \text{part of } \boldsymbol{g}^{\mathrm{pri}}}} \Big]$$

It differs from the SVD inverse by filter factors $F(\lambda) = \dfrac{\lambda^2}{\lambda^2 + \mu}$ and a selective contribution from $\boldsymbol{g}^{\mathrm{pri}}$ where the λ_i are small.



Filter factor $F(\lambda)$ for Tikhonov regularization

Tikhonov regularization: Regularization parameter

There is no perfect theory about how to chose the optimal parameter μ. It does obviously make no sense to minimize

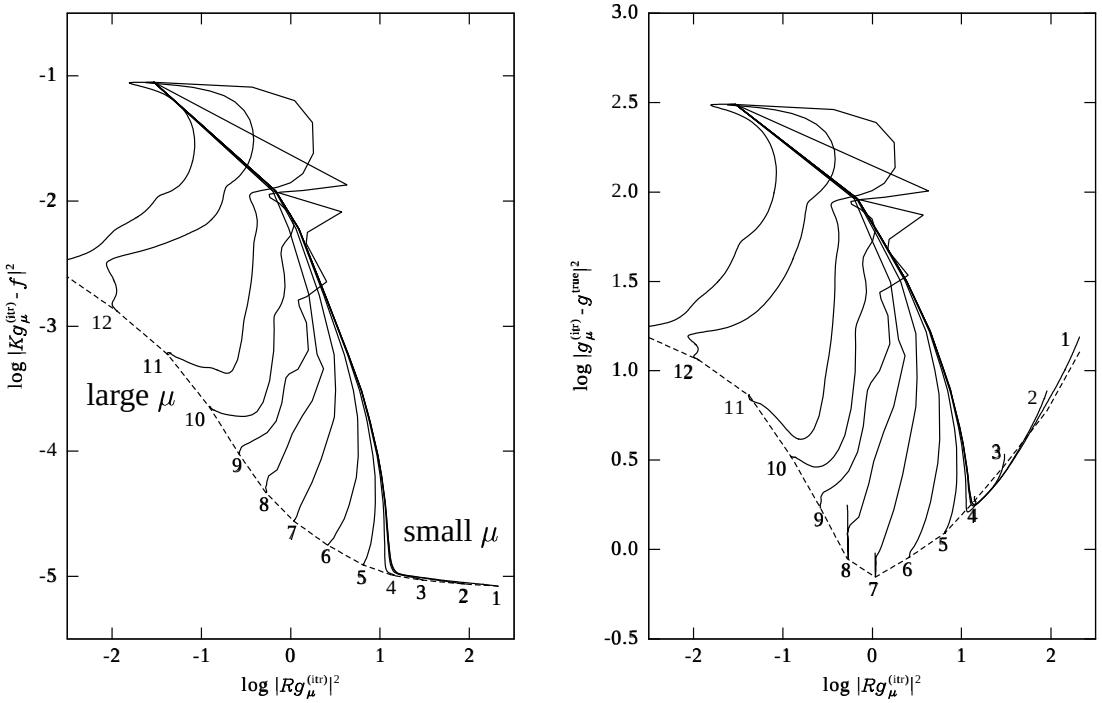
$$\Phi(\boldsymbol{g}_\mu) = \|\boldsymbol{K}\boldsymbol{g}_\mu - \boldsymbol{f}\|^2 + \mu\|\boldsymbol{g}_\mu - \boldsymbol{g}^{\mathrm{pri}}\|^2$$

any further after $\|\boldsymbol{K}\boldsymbol{g}_\mu - \boldsymbol{f}\|^2 < n\sigma^2$ has been reached.

$\longrightarrow$ Mozorov's discrepancy principle:

choose $\alpha > 1$ (typical $2 \ldots 3$), initial $\mu > 0$, continue=true
do while (continue)
find $\boldsymbol{g}_\mu$ = argmin $\Phi(\boldsymbol{g}_\mu)$
if $\|\boldsymbol{K}\boldsymbol{g}_\mu - \boldsymbol{f}\|^2 \simeq \alpha n\sigma^2$ continue = false
elseif $\|\boldsymbol{K}\boldsymbol{g}_\mu - \boldsymbol{f}\|^2 > \alpha n\sigma^2$ reduce μ
elseif $\|\boldsymbol{K}\boldsymbol{g}_\mu - \boldsymbol{f}\|^2 < \alpha n\sigma^2$ enhance μ
end while

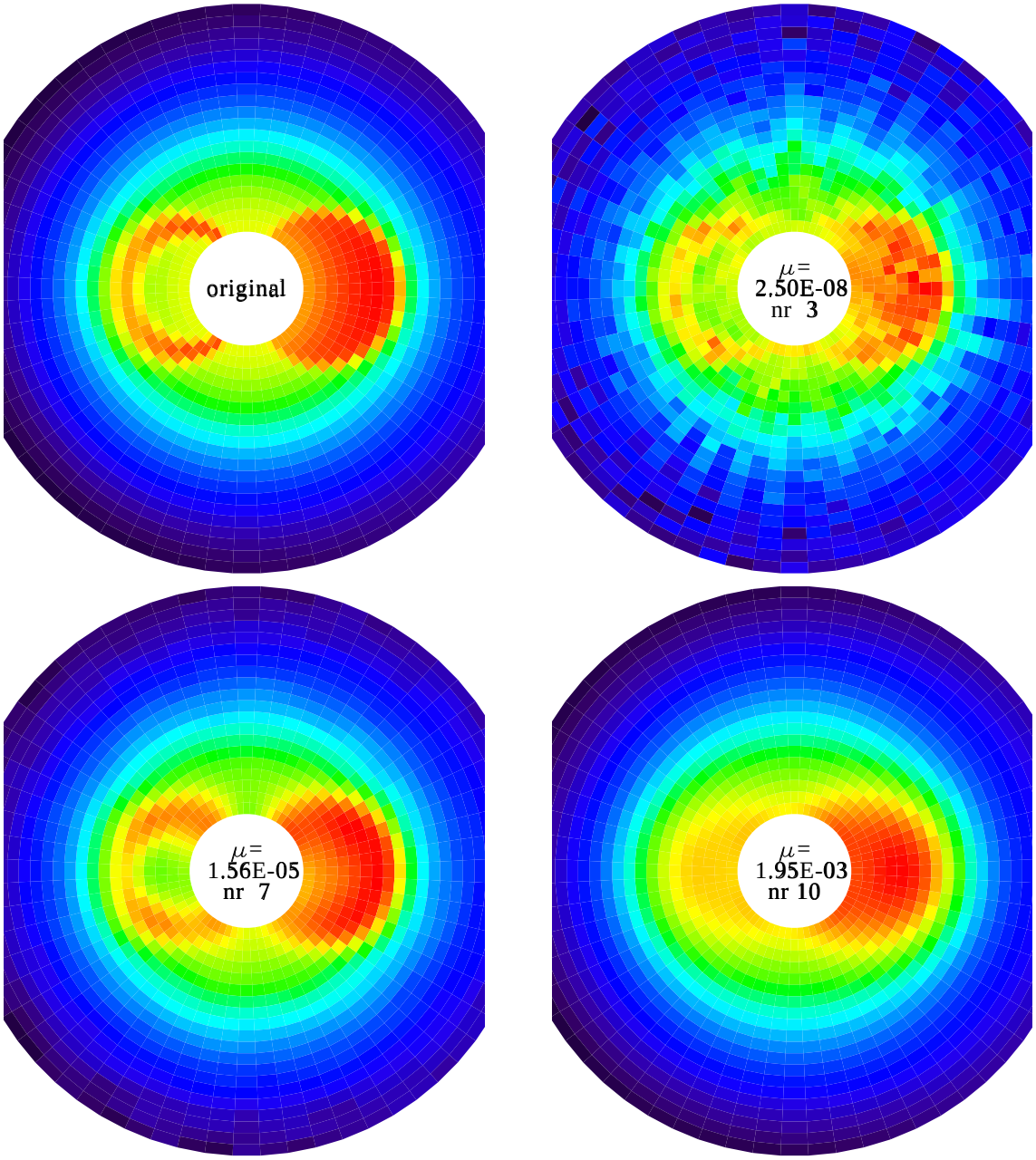
$\longrightarrow$ Another way is to plot the L-curve. The optimal μ is where the L-curve has its strongest positive curvature.



Iterative solutions for different μ. Data error (left) and model error (right) for the iteration path towards the solution vs magnitude of the regularization term. L-curve is the dashed curve in the left diagram.

Tikhonov regularization: Solution examples

The solutions $\boldsymbol{g}_\mu$ in the previous diagram are taken from a 2D tomography test problem. The regularization operator $\boldsymbol{R}$ was chosen as a 2nd order differential operator. Hence, the solutions become smoother from right to left as μ increases.



Example solutions $\boldsymbol{g}_\mu$ for different μ. The upper left shows $\boldsymbol{g}^{\text{true}}$ for comparison. The solution numbers refer to the numbers along the L-curve.

Iterative methods: Landweber iteration

The goal of an iteration is to decrease $\Phi(\boldsymbol{g}) = \|\boldsymbol{K}\boldsymbol{g} - \boldsymbol{f}\|^2$ stepwise by replacing $\boldsymbol{g}^{(\mathrm{itr})} \longrightarrow \boldsymbol{g}^{(\mathrm{itr}+1)}$. To insure convergence each step must proceed downhill, i.e., $\boldsymbol{g}^{(\mathrm{itr}+1)} - \boldsymbol{g}^{(\mathrm{itr})}$ must have a component along the descent direction

$$\underbrace{-\frac{1}{2}\nabla\Phi(\boldsymbol{g}^{(\mathrm{itr})})}_{\boldsymbol{d}^{(\mathrm{itr})}} = -\boldsymbol{K}^T(\underbrace{\boldsymbol{K}\boldsymbol{g}^{(\mathrm{itr})} - \boldsymbol{f}}_{\text{data error } \boldsymbol{r}^{(\mathrm{itr})}}) = -\boldsymbol{K}^T\boldsymbol{K}(\underbrace{\boldsymbol{g}^{(\mathrm{itr})} - \boldsymbol{g}^{(\mathrm{true})}}_{\text{model error } \boldsymbol{e}^{(\mathrm{itr})}})$$

The simplest iteration scheme is the Landweber interation:

$$\boldsymbol{g}^{(\mathrm{itr}+1)} = \boldsymbol{g}^{(\mathrm{itr})} - \alpha\boldsymbol{d}^{(\mathrm{itr})}$$

The model error then changes at each step by (simply subtract $\boldsymbol{g}^{\mathrm{true}}$ from both sides)

$$\boldsymbol{e}^{(\mathrm{itr}+1)} = \boldsymbol{e}^{(\mathrm{itr})} - \alpha\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{e}^{(\mathrm{itr})} = (\mathbf{1} - \alpha\boldsymbol{K}^T\boldsymbol{K})^{\mathrm{itr}}\boldsymbol{e}^{(0)}$$

$\longrightarrow$ error only declines if $1 > \alpha\times$(greatest eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}$).

The error at iteration step (itr+1) can be expressed as a polynom P_{itr} of order itr of the matrix $\boldsymbol{K}^T\boldsymbol{K}$ applied to the initial error:

$$\boldsymbol{e}^{(\mathrm{itr}+1)} = P_{\mathrm{itr}}(\boldsymbol{K}^T\boldsymbol{K})\ \boldsymbol{e}^{(0)}$$

For the Landweber iteration the polynom is $P_{\mathrm{itr}}(\lambda^2) = (1-\alpha\lambda^2)^{\mathrm{itr}}$. For optimum α, the polynom $P_{\mathrm{itr}}(\lambda_i^2)$ is minimal at each eigenvalue λ_i^2 of $\boldsymbol{K}^T\boldsymbol{K}$.
The error improvement $\boldsymbol{e}^{(\mathrm{itr}+1)}$ - $\boldsymbol{e}^{(0)}$ is built up as a vector of

$$\text{Krylov space} \quad \mathcal{K}^{\mathrm{itr}}(\boldsymbol{K}^T\boldsymbol{K}, \boldsymbol{e}^{(0)})$$
$$= \mathrm{span}\{\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{e}^{(0)}, (\boldsymbol{K}^T\boldsymbol{K})^2\boldsymbol{e}^{(0)}, \ldots, (\boldsymbol{K}^T\boldsymbol{K})^{\mathrm{itr}}\boldsymbol{e}^{(0)}\}$$

which expands with every iteration step.

Iterative methods: SVD analysis of Landweber interation

What does the solution look like if we stop at iteration itr ?

$$\boldsymbol{g}^{(\mathrm{itr}+1)} = \boldsymbol{g}^{(\mathrm{itr})} - \alpha \boldsymbol{d}^{(\mathrm{itr})} = (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K}) \boldsymbol{g}^{(\mathrm{itr})} + \alpha \boldsymbol{K}^T \boldsymbol{f}$$

where we inserted the expression for $\boldsymbol{d}$. Hence,

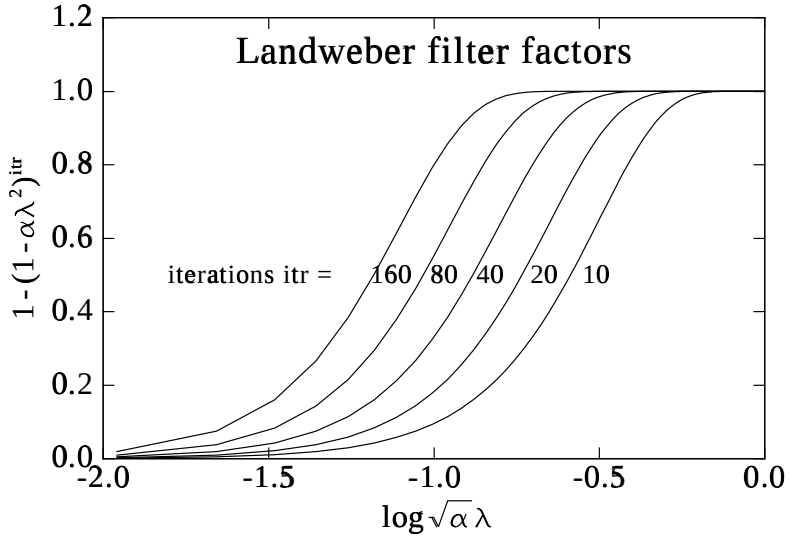
$$\begin{aligned}
\boldsymbol{g}^{(1)} &= (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K}) \boldsymbol{g}^{(0)} + \alpha \boldsymbol{K}^T \boldsymbol{f} \\
\boldsymbol{g}^{(2)} &= (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K})^2 \boldsymbol{g}^{(0)} + (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K})\, \alpha \boldsymbol{K}^T \boldsymbol{f} + \alpha \boldsymbol{K}^T \boldsymbol{f} \\
&\vdots \\
\boldsymbol{g}^{(\mathrm{itr})} &= (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K})^{\mathrm{itr}} \boldsymbol{g}^{(0)} + \sum_{j=0}^{\mathrm{itr}-1} (\mathbf{1} - \alpha \boldsymbol{K}^T \boldsymbol{K})^j\, \alpha \boldsymbol{K}^T \boldsymbol{f}
\end{aligned}$$

Replace $\boldsymbol{K}^T \boldsymbol{K}$ by its SVD system

$$\boldsymbol{g}^{(\mathrm{itr})} = \sum_{i=1}^{n} \boldsymbol{v}_i \Big[\underbrace{P_{\mathrm{itr}}(\lambda_i^2)}_{\text{decays like } \boldsymbol{e}^{(\mathrm{itr})}} (\boldsymbol{v}_i \cdot \boldsymbol{g}^{(0)}) + \frac{F_{\mathrm{itr}}(\lambda_i)}{\lambda_i} (\boldsymbol{u}_i \cdot \boldsymbol{f}) \Big]$$

where the filter factors for the Landweber iteration are

$$F_{\mathrm{itr}}(\lambda) = \alpha \lambda^2 \sum_{j=0}^{\mathrm{itr}-1} (1 - \alpha \lambda^2)^j = 1 - (1 - \alpha \lambda^2)^{\mathrm{itr}}$$



Filter factor $F(\lambda)$ of Landweber iteration

Iterative methods: Steepest descent

The Landweber iteration can be improved if the stepsize α is chosen with more care.

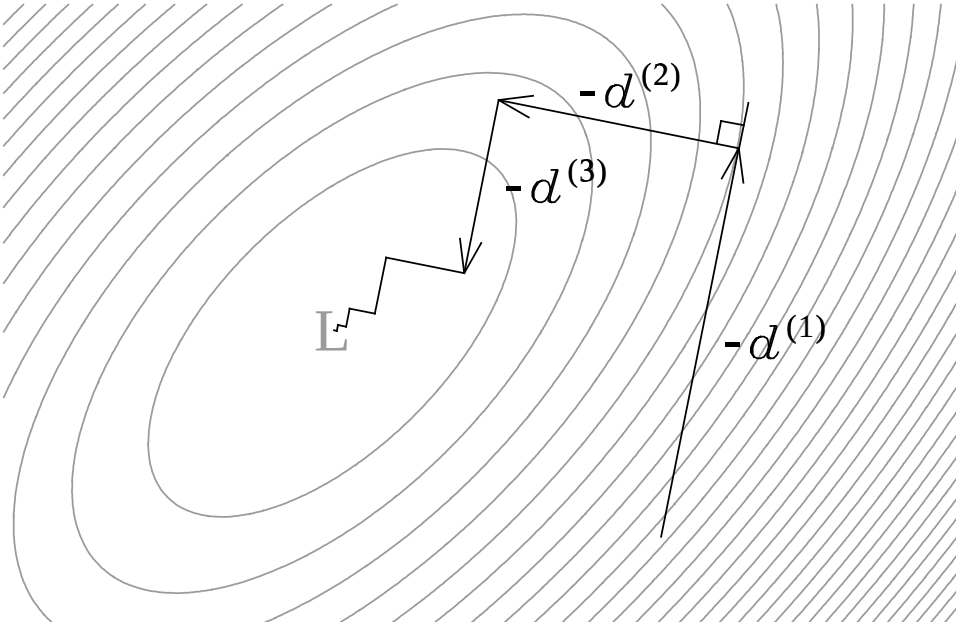
The method of steepest descent determines the stepsize α at each step so that the minimum along the descent direction is reached (line search):

$$\begin{aligned}\widehat{\Phi}(\alpha) &\equiv \Phi(\boldsymbol{g}^{(\mathrm{itr})} - \alpha \boldsymbol{d}^{(\mathrm{itr})}) \\ &= \|\boldsymbol{K}(\boldsymbol{g}^{(\mathrm{itr})} - \alpha \boldsymbol{d}^{(\mathrm{itr})}) - \boldsymbol{f}\|^2 = \|\boldsymbol{r}^{(\mathrm{itr})} - \alpha \boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}\|^2 \\ &= \|\boldsymbol{r}^{(\mathrm{itr})}\|^2 - 2\alpha(\boldsymbol{r}^{(\mathrm{itr})} \cdot \boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}) + \alpha^2 \|\boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}\|^2\end{aligned}$$

This gives a stepsize of

$$\alpha = \frac{(\boldsymbol{r}^{(\mathrm{itr})} \cdot \boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})})}{\|\boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}\|^2} = \frac{((\boldsymbol{K}^T\boldsymbol{r}^{(\mathrm{itr})}) \cdot \boldsymbol{d}^{(\mathrm{itr})})}{\|\boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}\|^2} = \frac{\|\boldsymbol{d}^{(\mathrm{itr})}\|^2}{\|\boldsymbol{K}\boldsymbol{d}^{(\mathrm{itr})}\|^2}$$

$\longrightarrow$ Consequence: The new descent direction $\boldsymbol{d}^{(\mathrm{itr}+1)}$ is perpendicular to the old $\boldsymbol{d}^{(\mathrm{itr})}$.



Sequence of iteration steps of the steepest decent algorithm

Iterative methods: Conjugate gradients

An essential further improvement is achieved if the new search direction is chosen not orthogonal to the old, but so that it is conjugate to the old.

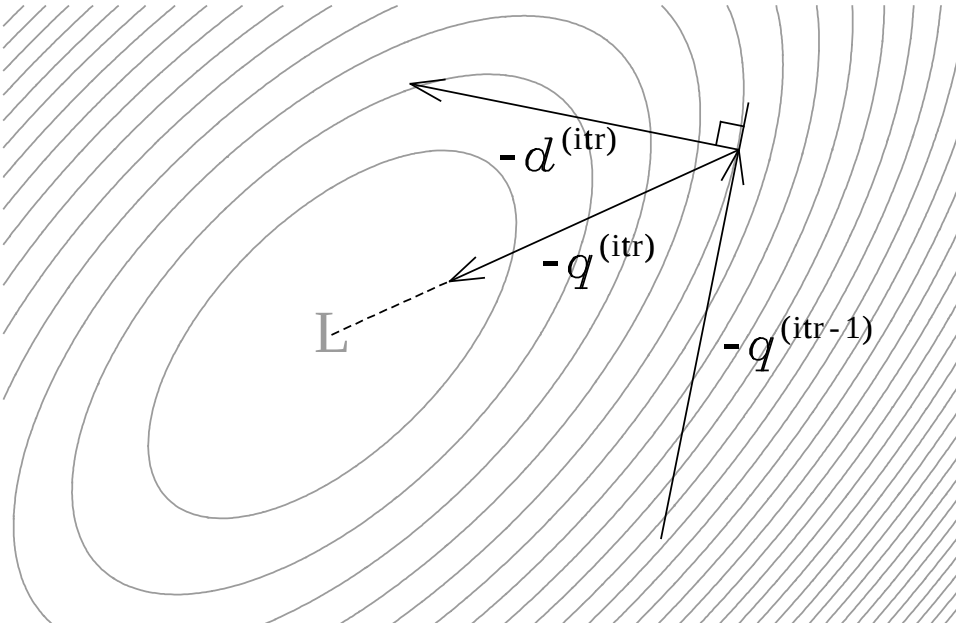
CG does not advance exactly along the descent direction but

$$\boldsymbol{g}^{(\mathrm{itr}+1)} = \boldsymbol{g}^{(\mathrm{itr})} - \alpha \boldsymbol{q}^{(\mathrm{itr})} \quad \text{where} \quad \boldsymbol{q}^{(\mathrm{itr})} = \boldsymbol{d}^{(\mathrm{itr})} + \beta \boldsymbol{q}^{(\mathrm{itr}-1)}$$

β is chosen so that $\boldsymbol{q}^{(\mathrm{itr})}$ and $\boldsymbol{q}^{(\mathrm{itr}-1)}$ are $\boldsymbol{K}^T\boldsymbol{K}$-orthogonal (conjugacy) which makes successive data error vectors orthogonal

$$\begin{aligned} 0 &= (\boldsymbol{q}^{(\mathrm{itr}-1)} \cdot \boldsymbol{K}^T\boldsymbol{K}\boldsymbol{q}^{(\mathrm{itr})}) \\ &= (\boldsymbol{K}\boldsymbol{q}^{(\mathrm{itr}-1)} \cdot \boldsymbol{K}\boldsymbol{q}^{(\mathrm{itr})}) = (\boldsymbol{r}^{(\mathrm{itr}-1)} \cdot \boldsymbol{r}^{(\mathrm{itr})}) \end{aligned}$$

Theoretically, CG needs at most n steps to find the minimum. The error polynom $\boldsymbol{e}^{(\mathrm{itr})} = P_{\mathrm{itr}}(\boldsymbol{K}^T\boldsymbol{K})\boldsymbol{e}^{(0)}$ is constructed so that for itr=n its roots are placed exactly at the eigenvalues of $\boldsymbol{K}^T\boldsymbol{K}$. For itr<$n$, $(\boldsymbol{e}^{(\mathrm{itr})} \cdot \boldsymbol{K}^T\boldsymbol{K}\boldsymbol{e}^{(\mathrm{itr})})$ has no more contributions from the current Krylov space $\mathcal{K}^{\mathrm{itr}}(\boldsymbol{K}^T\boldsymbol{K}, \boldsymbol{e}^{(0)})$



Conjugacy of $\boldsymbol{q}^{(\mathrm{itr})}$ and $\boldsymbol{q}^{(\mathrm{itr}-1)}$ causes a 2D minimum to be reached in 2 steps.

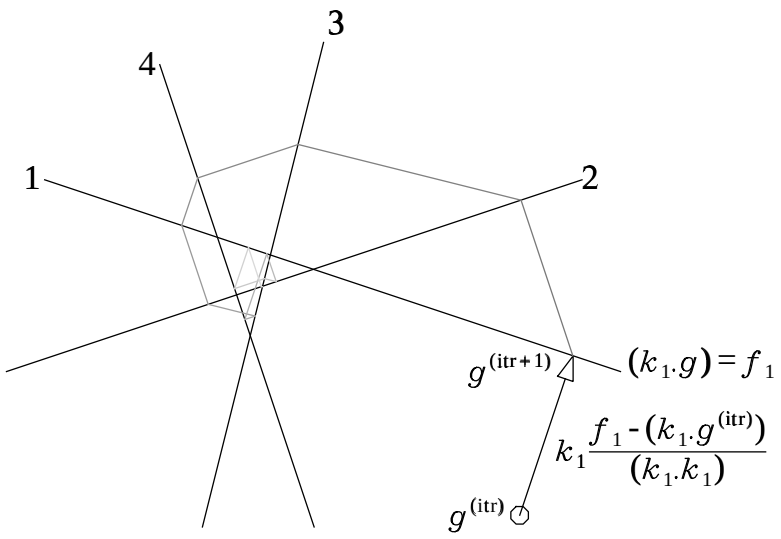
Iterative methods: Non-Krylov space schemes
Kaczmarz iteration

The idea is simple: consider each row of $\boldsymbol{Kg} - \boldsymbol{f} = 0$ as the definition of a hyperplane in model space $\mathbb{R}^n$. Each hyperplane has the i^{th} column vector of $\boldsymbol{K}^T = \{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$ as its normal. Then $\boldsymbol{g}$ has to satisfy n the hyperplane equations $f_i = (\boldsymbol{k}_i \cdot \boldsymbol{g})$

Kaczmarz's recipe is:

select initial $\boldsymbol{g}$
do while (no convergence)
  do for every datum f_i
    project $\boldsymbol{g}$ onto the plane $f_i = (\boldsymbol{k}_i \cdot \boldsymbol{g})$
  end for
end while

• Good convergence if the problem is consistent, i.e., if the planes all intersect in al least one point.
• No convergence if the problem is inconsistent $\longrightarrow$ underrelax



Kaczmarz' scheme is a sucession of projections onto planes. If the planes do not intersect in one point, $\boldsymbol{g}$ will jump around in a limited region around the point which has a minimum distance to all planes.

Iterative methods: Non-Krylov space schemes
Multiplicative reconstruction

Rather than adding a correction to $\boldsymbol{g}^{(\mathrm{itr})}$, why not multiply a correction factor? Numerous variants of such a scheme exist. The basics can be coded in a few lines:

define $\overline{\boldsymbol{g}}^{\infty} = \boldsymbol{K}\boldsymbol{f}$
select initial $\boldsymbol{g}^{(0)}$
do while (no convergence)
  set $\overline{\boldsymbol{g}}^{(\mathrm{itr})} = \boldsymbol{K}^T\boldsymbol{K}\boldsymbol{g}^{((\mathrm{itr})}$
  correct $g_i^{(\mathrm{itr}+1)} = g_i^{(\mathrm{itr})} \dfrac{\overline{g}_i^{(\infty)}}{\overline{g}_i^{(\mathrm{itr})}}$ for $i = 1, n$
end while

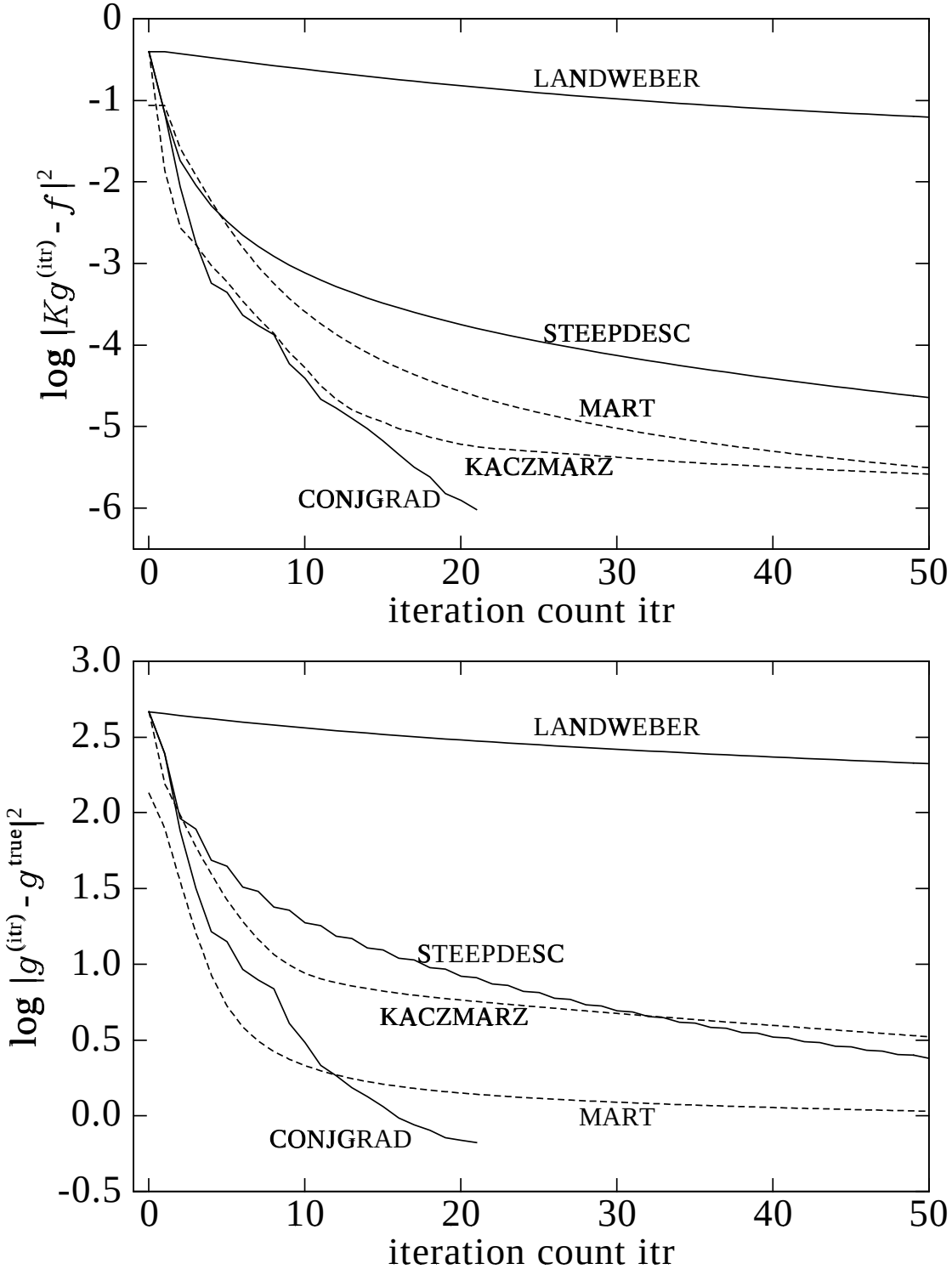
The scheme converges because the correction added to each $g_i^{(\mathrm{itr})}$

$$g_i^{(\mathrm{itr}+1)} - g_i^{(\mathrm{itr})} = g_i^{(\mathrm{itr})}\Big(\frac{\overline{g}_i^{(\infty)}}{\overline{g}_i^{(\mathrm{itr})}} - 1\Big) = \frac{g_i^{(\mathrm{itr})}}{\overline{g}_i^{(\mathrm{itr})}}\big(\overline{g}_i^{(\infty)} - \overline{g}_i^{(\mathrm{itr})}\big)$$
$$= \frac{g_i^{(\mathrm{itr})}}{\overline{g}_i^{(\mathrm{itr})}}(\boldsymbol{K}^T\boldsymbol{f} - \boldsymbol{K}^T\boldsymbol{K}\boldsymbol{g}^{(\mathrm{itr})})_i = -\frac{g_i^{(\mathrm{itr})}}{\overline{g}_i^{(\mathrm{itr})}} d_i^{(\mathrm{itr})}$$

is again along the descent direction, however with a different step size for each component i. Note $\boldsymbol{K}^T\boldsymbol{K}$ is positive and therefore the factor $g_i^{(\mathrm{itr})}/\overline{g}_i^{(\mathrm{itr})}$ is always positive or zero if the initial $\boldsymbol{g}$ is.

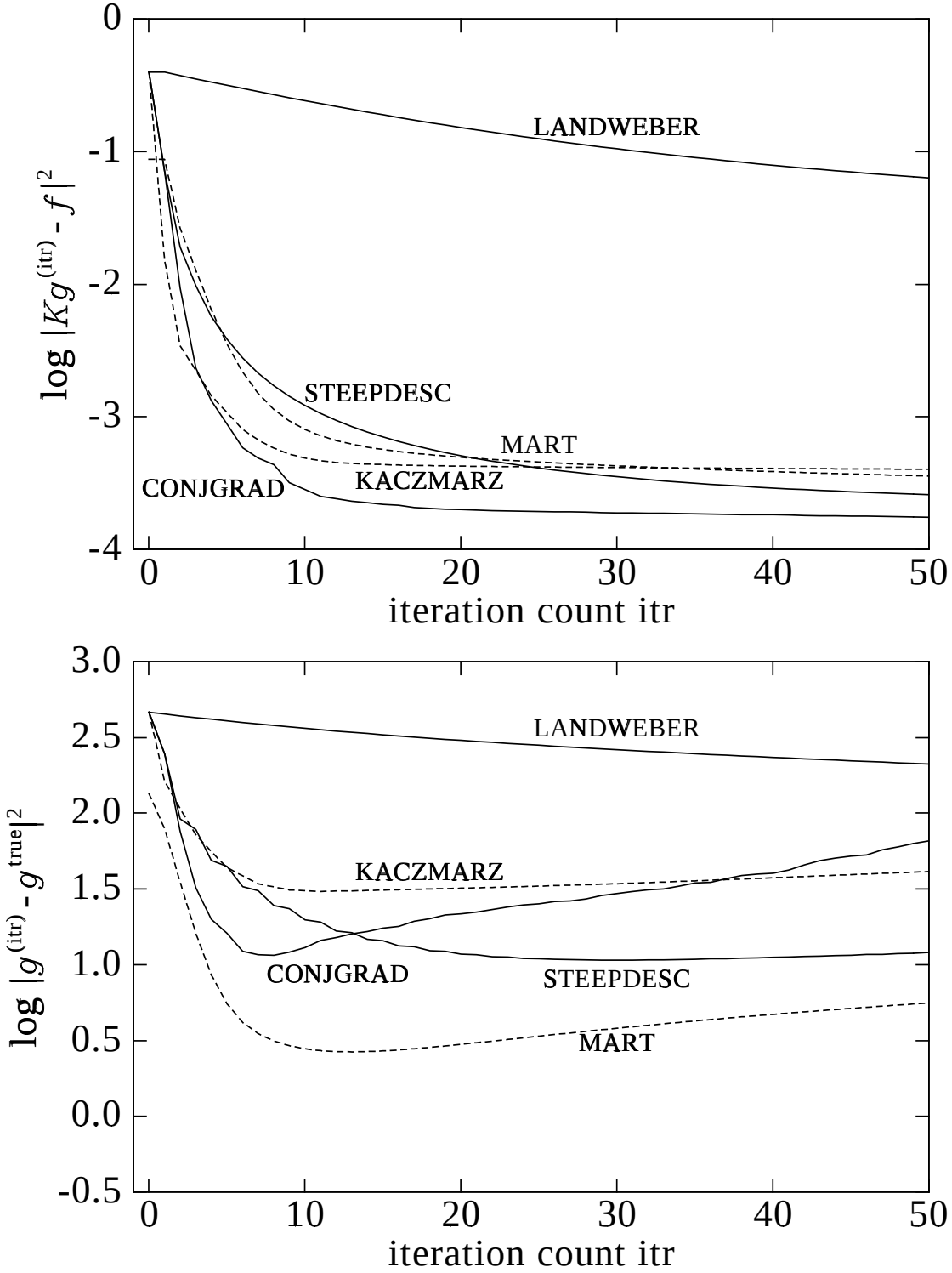
Advantages:
• $\boldsymbol{g}$ never changes sign during the iteration, i.e., negative densities or intensities are automatically avoided.
• If your model domain has a complicated boundary, use a simple square grid box and set initially all g_i outside of your domain to zero. The respective grid points will not contribute.

Iterative methods: Performance test without noise



Evolution of data error (top) and model error(bottom) for different iteration schemes. The problem to be solved was consistent.

Iterative methods: Performance test with noise - semiconvergence



Evolution of data error (top) and model error(bottom) for different iteration schemes. The problem to be solved was inconsistent, noise with standard deviatiuon of 5% of maximum signal was added to $\boldsymbol{f}$. *Note that the model error eventually increases while the data error is still iterated down.*

Nonlinear methods: Weak nonlinearity

The treatment of nonlinear problems is still at a somewhat experimental stage.

The nonlinearity is weak if the minimizing function can be expanded around an approximate solution

$$\Phi(\boldsymbol{g}) = \|\boldsymbol{h}(\boldsymbol{g}) - \boldsymbol{f}\|^2$$
$$= \|\boldsymbol{h}(\boldsymbol{g}^{\mathrm{apprx}}) + (\boldsymbol{g} - \boldsymbol{g}^{\mathrm{apprx}})\cdot\nabla\boldsymbol{h}(\boldsymbol{g}^{\mathrm{apprx}}) - \boldsymbol{f}\|^2$$

and higher derivatives can be neglected. This is the case if either
• the structure of $\boldsymbol{h}$ allows for such an expansion or
• the guess $\boldsymbol{g}^{\mathrm{apprx}}$ is sufficiently close to the expected solution.

In case of a weak linearity, a safe approach is to embedd a linear problem solver in an iterative correction of the approximate solution:

select $\boldsymbol{g}^{\mathrm{apprx}}$
do while ($\|\delta\boldsymbol{g}\|^2 > 0$)
  set $\delta\boldsymbol{f} = \boldsymbol{f} - \boldsymbol{h}(\boldsymbol{g}^{\mathrm{apprx}})$, $\boldsymbol{K}^T = \nabla\boldsymbol{h}(\boldsymbol{g}^{\mathrm{apprx}})$
  minimize $\|\boldsymbol{K}\delta\boldsymbol{g} - \delta\boldsymbol{f}\|^2$
  correct $\boldsymbol{g}^{\mathrm{apprx}} = \boldsymbol{g}^{\mathrm{apprx}} + \delta\boldsymbol{g}$
end while

• If the linear solver is iterative, the external nonlinear correction and the inner linear iteration can be combined, e.g., by an improved line search.

• Generaly speaking, for nonlinear methods the emphasis is more on the fact that the right solution (global minimum rather than a local minimum) is found or that a solution is found at all , while for linear schemes the convergence speed is an issue $\longrightarrow$ switch to a linear scheme if $\boldsymbol{g}^{\mathrm{apprx}}$ is sufficiently close to the solution.

Nonlinear methods: Levenberg-Marquardt

Even with nonlinearity the Landweber iteration into descent direction should still work provided the stepsizes are small enough. However, if the nonlinearity is weak, a faster convergence would be desirable.

Expand $\Phi(\boldsymbol{g}) = \|\boldsymbol{h}(\boldsymbol{g}) - \boldsymbol{f}\|^2$ as before

$$\Phi(\boldsymbol{g}) \simeq \Phi(\boldsymbol{g}^{\mathrm{apprx}}) + 2(\delta\boldsymbol{g}\cdot\boldsymbol{d}) + (\delta\boldsymbol{g}\cdot\boldsymbol{K}^T\boldsymbol{K}\delta\boldsymbol{g}) \quad \text{where}$$

$-\boldsymbol{d} = -\boldsymbol{K}^T(\boldsymbol{h} - \boldsymbol{f})$ at $\boldsymbol{g}^{\mathrm{apprx}}$ is the descent direction

$\boldsymbol{K}^T = \nabla\boldsymbol{h}$ at $\boldsymbol{g}^{\mathrm{apprx}}$ is the Jacobian of $\boldsymbol{h}$

To approach the minimum, we have to correct $\boldsymbol{g}^{\mathrm{apprx}}$ by

$$\delta\boldsymbol{g} \quad \text{with} \quad \boldsymbol{K}^T\boldsymbol{K}\delta\boldsymbol{g} = -\boldsymbol{d}$$

while a Landweber step would be $\delta\boldsymbol{g} = -\alpha\boldsymbol{d}$
with $\frac{1}{\alpha} >$ largest eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}$

Both steps are combined in the Levenberg-Marquardt scheme:

select $\boldsymbol{g}^{(0)}$ and α, calculate $\boldsymbol{h}^{(0)} = \boldsymbol{h}(\boldsymbol{g}^{(0)})$, set itr=0
do while ($\|\delta\boldsymbol{g}\|^2 > 0$)
calculate $\boldsymbol{K}^T = \nabla\boldsymbol{h}(\boldsymbol{g}^{(\mathrm{itr})})$
set $\boldsymbol{d} = \boldsymbol{K}^T(\boldsymbol{h}^{(\mathrm{itr})} - \boldsymbol{f})$
solve $(\boldsymbol{K}^T\boldsymbol{K} + \frac{1}{\alpha}\mathbf{1})\delta\boldsymbol{g} = -\boldsymbol{d}$
correct $\boldsymbol{g}^{(\mathrm{itr}+1)} = \boldsymbol{g}^{(\mathrm{itr})} + \delta\boldsymbol{g}$
calculate $\boldsymbol{h}^{(\mathrm{itr}+1)} = \boldsymbol{h}(\boldsymbol{g}^{(\mathrm{itr}+1)})$
if $\|\boldsymbol{h}^{(\mathrm{itr}+1)} - \boldsymbol{f}\|^2 < \|\boldsymbol{h}^{(\mathrm{itr})} - \boldsymbol{f}\|^2$ enhance α
elseif $\|\boldsymbol{h}^{(\mathrm{itr}+1)} - \boldsymbol{f}\|^2 > \|\boldsymbol{h}^{(\mathrm{itr})} - \boldsymbol{f}\|^2$ reduce α
itr=itr+1
end while

Artificial neural network (ANN)

For a **linear** problem: Construct inverse matrix $\boldsymbol{W}$ which solves

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{g} \quad \text{by} \quad \boldsymbol{g} \simeq \boldsymbol{W}\boldsymbol{f}$$

$\boldsymbol{W}$ has $n + m$ unknown elements which are to be determined by a trainings phase: produce pairs $\{\boldsymbol{f}^i, \boldsymbol{g}^i\}$, $i = 1, l$ and solve the linear problem

$$\{\boldsymbol{g}^1, \ldots, \boldsymbol{g}^l\}^T = \{\boldsymbol{f}^1, \ldots, \boldsymbol{f}^l\}^T \boldsymbol{W}^T$$

Use conventional linear methods (e.g., Kaczmarz iteration) extended from vectors to matrices. l must be $\geq m$ to avoid the problem to be underdetermined. Mind linear independence of $\boldsymbol{f}^i$ and $\boldsymbol{g}^i$.

For a **nonlinear** problem: Vectors $\boldsymbol{g}$ and $\boldsymbol{f}$ are considered as layers, elementwise connected by a matrix. The matrix elements define the strength of this connection. The simplest extension of the linear case is the three layer perceptron with two matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ in between

On every layer, the "signal" is modified by the sigmoid function $\mathrm{sgm}(x) = 1/(1 + \mathrm{e}^{-x})$. Hence the inversion is done in two steps

$$\boldsymbol{h} = \mathrm{sgm}(\boldsymbol{V}\boldsymbol{f}) \ , \quad \boldsymbol{g} = \mathrm{sgm}(\boldsymbol{W}\boldsymbol{h}) \ ,$$

Since the intermediate layer has to pass all relevant information, its dimension must be at least the dimension of $\min(n, m)$.

- Training is rather expensive
- Once $\boldsymbol{W}$ and/or $\boldsymbol{V}$ are found, inversion is fast
- Adaptation to different SNR levels is a problem

Simplex Optimization

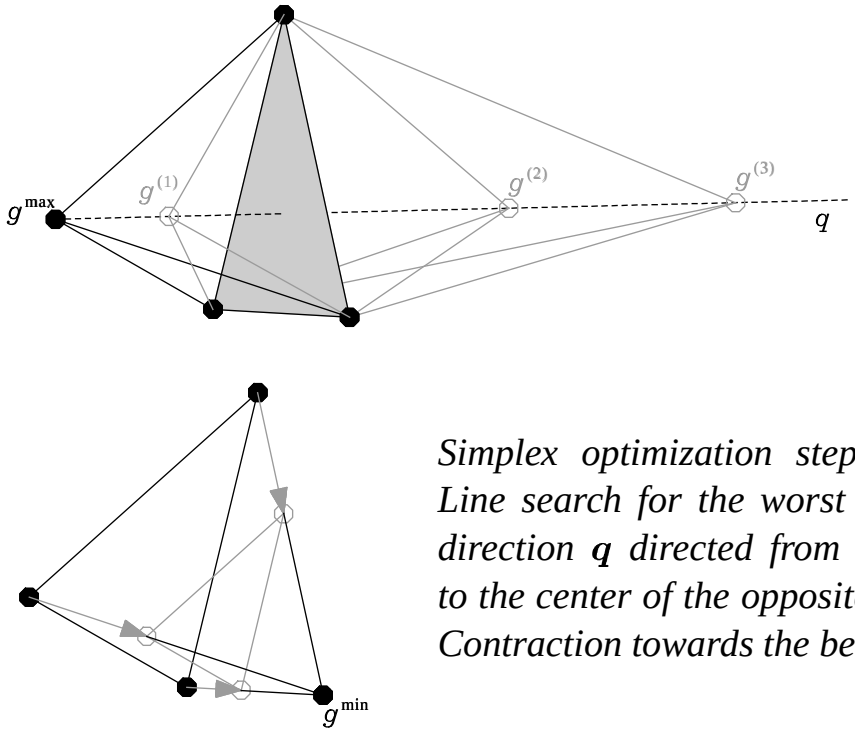
Rather than push a single $\boldsymbol{g}$ through model space on search of the minimum with nonlinear problems it may be effective to let a whole bunch of $\boldsymbol{g}$ do the search in parallel.

The simplex optimization algorithm employs $n+1$ vectors $\boldsymbol{g}_i$. They form the vertices of a simplex in model space $\mathbb{R}^n$ if the n edge vectors $\boldsymbol{g}_i - \boldsymbol{g}_{n+1}$, i=1,n are linearly independent.

Search strategy:

- The new $\boldsymbol{g}_i$ are a linear combination of the old
- Maintain the linear independence of the edges
- Do not replace $\boldsymbol{g}_i$=$\boldsymbol{g}_{\min}$ which gives the lowest $\Phi(\boldsymbol{g})$
  but try to improve $\boldsymbol{g}_i$=$\boldsymbol{g}_{\max}$ which yields the largest $\Phi(\boldsymbol{g})$

Typical steps are:



Simplex optimization steps. Above: Line search for the worst with search direction $\boldsymbol{q}$ directed from the worst $\boldsymbol{g}$ to the center of the opposite face. Left: Contraction towards the best $\boldsymbol{g}$.

Advantages:
- No need to calculate derivatives of Φ
- Finds a minimum almost certainly, even if Φ is not analytic.

Disadvantage:
- Very slow convergence, many evaluations of Φ

Gencodes

Usually we want to find the global minimum of Φ rather than a local one. Algorithms which satisfy this request have to be driven by two opposing principles:

• Tendency to approach a minimum once $\boldsymbol{g}$ is close

• Diversity to explore unknown regions of model space

Gencodes take evolution as a model (which assumes that mankind as the ultimate result of evolution is the global minimum):

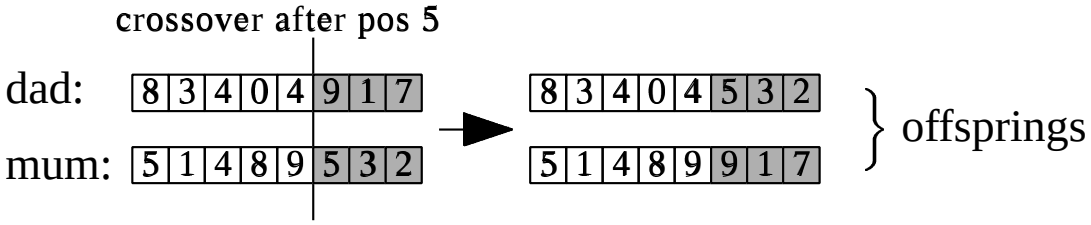
Start with a set of $\boldsymbol{g}_i$, $i = 1, k$ (population)

Calculate $\Phi_i = \Phi(\boldsymbol{g}_i)$ (fitness)

Select the fittest $\boldsymbol{g}_i$ in pairs for marriage

Encode $\boldsymbol{g}$ of each couple to a string of bits, decimals, or . . . (chromosomes)

Let each couple with some probability exchange a random part of their chromosomes (recombination):



Randomly manipulate few chromosomes of the offsprings (mutation)

Decode chromosomes back to $\boldsymbol{g}$

Replace old generation by the offsprings and goto start

Eventually, the fitness pressure (selection) should take the majority of the population to or near the global minimum of Φ.

Numerous variants exist for the different steps (Selection, recombination, mutation), all controlled by probability distributions.

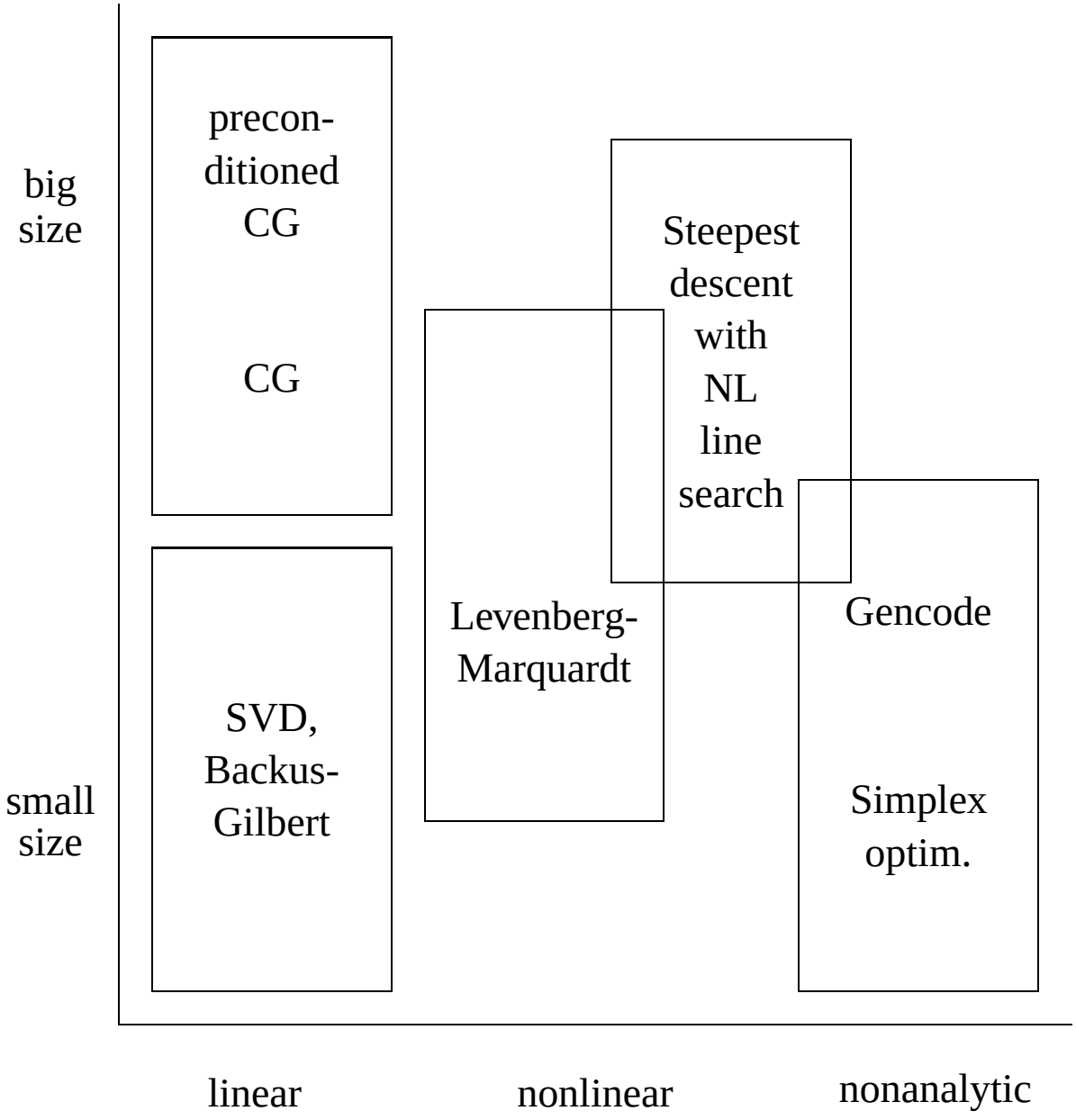
Advantages: • No need to calculate derivatives of Φ

• Finds a global minimum

Disadvantage: • Very slow convergence, many evaluations of Φ

• Many parameters (pdfs) for tuning.

Summary: Which method for which problem?



- Most methods can be combined with Tikhonov-regularization, replace $\boldsymbol{K} \longrightarrow \widetilde{\boldsymbol{K}}$ and $\boldsymbol{f} \longrightarrow \widetilde{\boldsymbol{f}}$

- If $\boldsymbol{g}$ is supposed to be ≥ 0 and has large areas with $\boldsymbol{g} \simeq 0$ try MART.
- For some nonlinear problems the repeated calculation of the Hessian, i.e., $\nabla\nabla\Phi \simeq \boldsymbol{K}^T\boldsymbol{K}$ can be very expensive. Simplified approximations sometimes work just as well.