Pulse Wave Velocity Estimation

via Magnetic Resonance Advection Imaging

Simon Hubmer

Johannes Kepler University, Linz

20.12.2016, Mülheim a.d. Ruhr

Joint work with:
A. Neubauer, R. Ramlau, H. Voss

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

JOHANNES KEPLER
UNIVERSITY LINZ

Introduction

Introduction

Two important abbreviations:

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

Problem

Estimate the PWV from dynamic MRI data!

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

Problem

Estimate the PWV from dynamic MRI data!

Three natural questions:

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

Problem

Estimate the PWV from dynamic MRI data!

Three natural questions:

- What is the PWV?

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

Problem

Estimate the PWV from dynamic MRI data!

Three natural questions:

- What is the PWV?
- Why do we want to estimate it?

Introduction

Two important abbreviations:

- PWV - Pulse Wave Velocity
- MRI - Magnetic Resonance Imaging

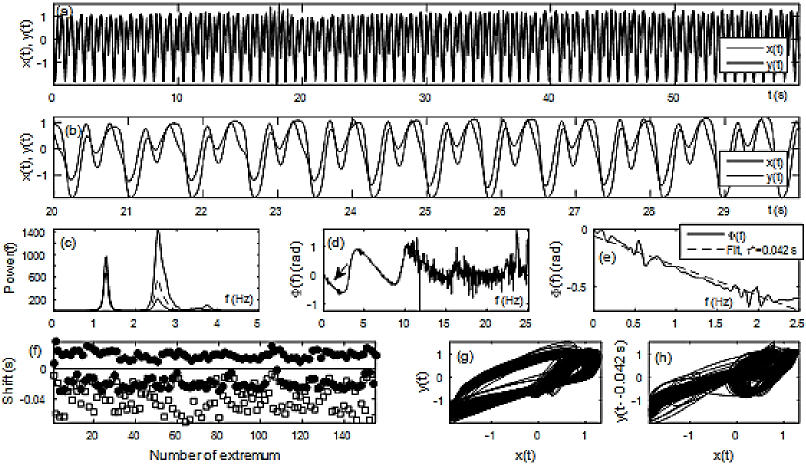
Problem

Estimate the PWV from dynamic MRI data!

Three natural questions:

- What is the PWV?
- Why do we want to estimate it?
- How can we estimate it?

The Pulse Wave



The PWV is used as a prognostic marker for:

The PWV is used as a prognostic marker for:

- cardiovascular morbidity and mortality in the elderly

The PWV is used as a prognostic marker for:

- cardiovascular morbidity and mortality in the elderly
- patients with diabetes and hypertension

The PWV is used as a prognostic marker for:

- cardiovascular morbidity and mortality in the elderly
- patients with diabetes and hypertension
- aortic stiffness $\rightarrow$ small-vessel disease and cognitive decline

The PWV is used as a prognostic marker for:

- cardiovascular morbidity and mortality in the elderly
- patients with diabetes and hypertension
- aortic stiffness $\rightarrow$ small-vessel disease and cognitive decline

Moens-Korteweg formula:

$$\boxed{\text{PWV} = \sqrt{\frac{Eh}{\rho_B d}}}$$

The PWV is used as a prognostic marker for:

- cardiovascular morbidity and mortality in the elderly
- patients with diabetes and hypertension
- aortic stiffness $\rightarrow$ small-vessel disease and cognitive decline

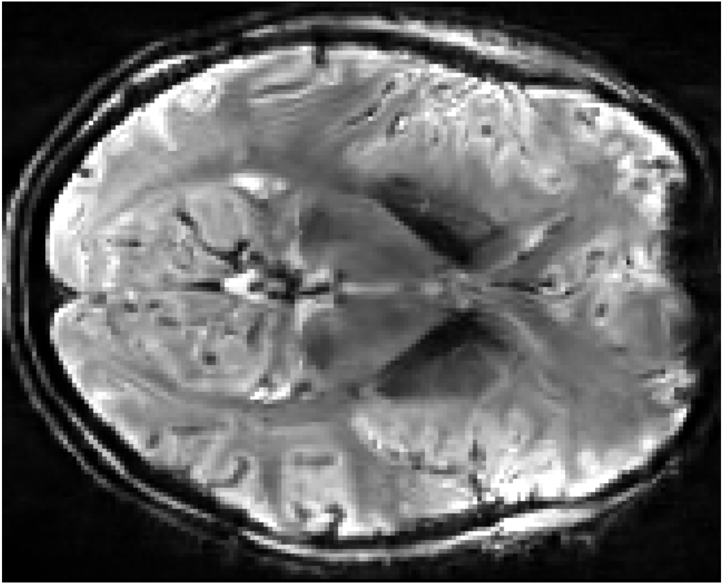
Moens-Korteweg formula:

$$\boxed{\mathrm{PWV} = \sqrt{\frac{Eh}{\rho_B d}}}$$

The parameters are

- h - vessel wall thickness,
- d - vascular diameter,
- E - vessel all's Young modulus,
- ρ_B - density of the blood.

Introduction and Motivation



How to estimate the PVW $\rightarrow$ MRAI

How to estimate the PVW $\rightarrow$ MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

How to estimate the PVW $\to$ MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

Continuity equation

$$\boxed{\frac{\partial}{\partial t}\rho(x, y, z, t) + \operatorname{div}\left(v(x, y, z)\rho(x, y, z, t)\right) = 0\,.}$$

How to estimate the PVW $\rightarrow$ MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

Continuity equation and $\nabla \cdot v = 0$

$$\boxed{\frac{\partial}{\partial t}\rho(x, y, z, t) + \operatorname{div}\left(v(x, y, z)\rho(x, y, z, t)\right) = 0\,.}$$

How to estimate the PVW → MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

Continuity equation and $\nabla \cdot v = 0$ leads to

$$\boxed{\frac{\partial}{\partial t}\rho(x, y, z, t) + v(x, y, z) \cdot \nabla\rho(x, y, z, t) = 0\,.}$$

How to estimate the PVW → MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

Continuity equation and $\nabla \cdot v = 0$ leads to

$$\boxed{\frac{\partial}{\partial t}\rho(x, y, z, t) + v(x, y, z) \cdot \nabla\rho(x, y, z, t) = 0\,.}$$

Advection (Transport, Optical Flow) equation ⇒

How to estimate the PVW $\to$ MRAI

Problem variables

- MRI signal $\rho(x, y, z, t)$,
- pulse wave velocity $v(x, y, z)$.

Continuity equation and $\nabla \cdot v = 0$ leads to

$$\boxed{\frac{\partial}{\partial t}\rho(x, y, z, t) + v(x, y, z) \cdot \nabla\rho(x, y, z, t) = 0\,.}$$

Advection (Transport, Optical Flow) equation $\Rightarrow$

MRAI = Magnetic Resonance Advection Imaging

Challenges in MRAI

Challenges in MRAI

Challenges with the advection equation:

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

- High amount of noise in the MRI data.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

- High amount of noise in the MRI data.
- Huge data sets.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

- High amount of noise in the MRI data.
- Huge data sets.
- Low spatiotemporal resolution.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

- High amount of noise in the MRI data.
- Huge data sets.
- Low spatiotemporal resolution.

Challenges with the method:

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

Challenges with the data:

- High amount of noise in the MRI data.
- Huge data sets.
- Low spatiotemporal resolution.

Challenges with the method:

- Treatment of boundary conditions.

Challenges in MRAI

Challenges with the advection equation:

- Difficult solution concept for non-Lipschitz velocities.
- Forward problem is already hard to solve.

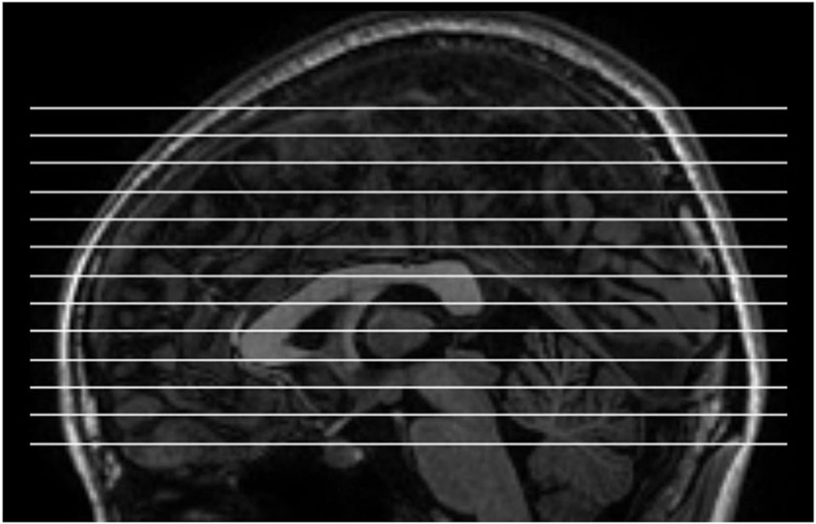
Challenges with the data:

- High amount of noise in the MRI data.
- Huge data sets.
- Low spatiotemporal resolution.

Challenges with the method:

- Treatment of boundary conditions.
- Partial data $\longleftrightarrow$ slice-time-acquisition problem.

Slice-Time Acquisition



Solution Strategy - Discretization

Solution Strategy - Discretization

The MRI data ρ is only available at points

$$(x_i, y_j, z_k, t_{k,l})$$

Solution Strategy - Discretization

The MRI data ρ is only available at points

$$(x_i, y_j, z_k, t_{k,l})$$

where

$$x_i = x_0 + i\Delta x\,, \quad y_j = y_0 + j\Delta y\,, \quad z_k = z_0 + k\Delta z\,,$$
$$t_{k,l} = (k + (K+1)l)\Delta t\,,$$

Solution Strategy - Discretization

The MRI data ρ is only available at points

$$(x_i, y_j, z_k, t_{k,l})$$

where

$$x_i = x_0 + i\Delta x\,, \quad y_j = y_0 + j\Delta y\,, \quad z_k = z_0 + k\Delta z\,,$$
$$t_{k,l} = (k + (K+1)l)\Delta t\,,$$

$$0 \leq i \leq I\,, \quad 0 \leq j \leq J\,, \quad 0 \leq k \leq K\,, \quad 0 \leq l \leq L\,.$$

Solution Strategy - Discretization

The MRI data ρ is only available at points

$$(x_i, y_j, z_k, t_{k,l})$$

where

$$x_i = x_0 + i\Delta x\,, \quad y_j = y_0 + j\Delta y\,, \quad z_k = z_0 + k\Delta z\,,$$
$$t_{k,l} = (k + (K+1)l)\Delta t\,,$$

$$0 \leq i \leq I\,, \quad 0 \leq j \leq J\,, \quad 0 \leq k \leq K\,, \quad 0 \leq l \leq L\,.$$

Idea: Discretize the advection equation according to the data!!

Solution Strategy - Discretization

The continuous advection equation

$$\frac{\partial}{\partial t}\rho(x, y, z, t) + v(x, y, z) \cdot \nabla \rho(x, y, z, t) = 0\,,$$

Solution Strategy - Discretization

The continuous advection equation

$$\frac{\partial}{\partial t}\rho(x,y,z,t) + v(x,y,z)\cdot\nabla\rho(x,y,z,t) = 0\,,$$

then becomes a discrete linear system of equations:

$$\frac{\rho_{i,j,k,l}-\rho_{i,j,k,l-1}}{(K+1)\Delta t} + D_{x_i}\rho_{i,j,k,l}\,v_{1,i,j,k} + D_{y_j}\rho_{i,j,k,l}\,v_{2,i,j,k} + D_{z_k}\rho_{i,j,k,l}\,v_{3,i,j,k} = 0\,,$$

Solution Strategy - Discretization

The continuous advection equation

$$\frac{\partial}{\partial t}\rho(x,y,z,t) + v(x,y,z)\cdot\nabla\rho(x,y,z,t) = 0\,,$$

then becomes a discrete linear system of equations:

$$\frac{\rho_{i,j,k,l} - \rho_{i,j,k,l-1}}{(K+1)\Delta t} + D_{x_i}\rho_{i,j,k,l}\, v_{1,i,j,k} + D_{y_j}\rho_{i,j,k,l}\, v_{2,i,j,k} + D_{z_k}\rho_{i,j,k,l}\, v_{3,i,j,k} = 0\,,$$

where

$$\rho_{i,j,k,l} = \rho(x_i, y_j, z_k, t_{k,l})\,, \quad v_{m,i,j,k} = v_m(x_i, y_j, z_k)\,, \quad m = 1,2,3\,.$$

Discretization - Finite Differences

$$D_{x_i}\rho_{i,j,k,l} := \begin{cases} \dfrac{\rho_{i+1,j,k,l} - \rho_{i-1,j,k,l}}{2\Delta x}, & 1 \leq i \leq I-1 \\ \dfrac{\rho_{1,j,k,l} - \rho_{0,j,k,l}}{\Delta x}, & i = 0 \\ \dfrac{\rho_{I,j,k,l} - \rho_{I-1,j,k,l}}{\Delta x}, & i = I \end{cases}$$

$$D_{y_j}\rho_{i,j,k,l} := \begin{cases} \dfrac{\rho_{i,j+1,k,l} - \rho_{i,j-1,k,l}}{2\Delta y}, & 1 \leq j \leq J-1 \\ \dfrac{\rho_{i,1,k,l} - \rho_{i,0,k,l}}{\Delta y}, & j = 0 \\ \dfrac{\rho_{i,J,k,l} - \rho_{i,J-1,k,l}}{\Delta y}, & j = J \end{cases}$$

Discretization - Finite Differences

$$D_{z_k}\rho_{i,j,k,l} := \begin{cases} \dfrac{(1-r)(\rho_{i,j,k+1,l}-\rho_{i,j,k-1,l+1}) + r(\rho_{i,j,k+1,l-1}-\rho_{i,j,k-1,l})}{2\Delta z}, & 1 \leq k \leq K-1, 1 \leq l < L \\ \dfrac{(1-r)\rho_{i,j,k+1,L} - (1+r)\rho_{i,j,k-1,L} + r(\rho_{i,j,k+1,L-1}+\rho_{i,j,k-1,L-1})}{2\Delta z}, & 1 \leq k \leq K-1, l = L \\ \dfrac{(1-r)\rho_{i,j,1,l} + r\rho_{i,j,1,l-1} - \rho_{i,j,0,l}}{\Delta z}, & k = 0, 1 \leq l \leq L \\ \dfrac{\rho_{i,j,K,l} - (1-r)\rho_{i,j,K-1,l+1} - r\rho_{i,j,K-1,l}}{\Delta z}, & k = K, 1 \leq l < L \\ \dfrac{\rho_{i,j,K,L} - (1+r)\rho_{i,j,K-1,L} + r\rho_{i,j,K-1,L-1}}{\Delta z}, & k = K, l = L \end{cases}$$

$$r := \frac{1}{K+1}$$

Solution Strategy - Discretization

Define the vectors:

- $\vec{\rho}_0$ - consists of all ρ $(l = 0)$ values.
- $\vec{\rho}$ - consists of all ρ $(l > 0)$ values,

Solution Strategy - Discretization

Define the vectors:

- $\vec{\rho}_0$ - consists of all ρ $(l = 0)$ values.
- $\vec{\rho}$ - consists of all ρ $(l > 0)$ values,

Then,

$$\frac{\rho_{i,j,k,l} - \rho_{i,j,k,l-1}}{(K+1)\Delta t} + D_{x_i}\rho_{i,j,k,l}\, v_{1,i,j,k} + D_{y_j}\rho_{i,j,k,l}\, v_{2,i,j,k} + D_{z_k}\rho_{i,j,k,l}\, v_{3,i,j,k} = 0\,,$$

can be written in the form

$$A(\vec{v})\vec{\rho} = b(\vec{v}, \vec{\rho}_0)\,.$$

Solution Strategy - Discretization

Define the vectors:

- $\vec{\rho}_0$ - consists of all ρ $(l = 0)$ values.
- $\vec{\rho}$ - consists of all ρ $(l > 0)$ values,

Then,

$$\frac{\rho_{i,j,k,l} - \rho_{i,j,k,l-1}}{(K+1)\Delta t} + D_{x_i}\rho_{i,j,k,l}\, v_{1,i,j,k} + D_{y_j}\rho_{i,j,k,l}\, v_{2,i,j,k} + D_{z_k}\rho_{i,j,k,l}\, v_{3,i,j,k} = 0\,,$$

can be written in the form

$$A(\vec{v})\vec{\rho} = b(\vec{v}, \vec{\rho}_0)\,.$$

We denote the solution $\vec{\rho}$ of this equation with $\rho(\vec{v}, \vec{\rho}_0)$.

The Inverse Problem

The Inverse Problem

We define the following operator

$$F : \mathcal{X} \to \mathcal{Y}\,, \quad (\vec{v}, \vec{\rho}_0) \mapsto (\rho(\vec{v}, \vec{\rho}_0), \vec{\rho}_0)\,,$$

The Inverse Problem

We define the following operator

$$F : \mathcal{X} \to \mathcal{Y}\,, \quad (\vec{v}, \vec{\rho}_0) \mapsto (\rho(\vec{v}, \vec{\rho}_0), \vec{\rho}_0)\,,$$

where the inner products on $\mathcal{X}$ and $\mathcal{Y}$ are given by

$$\langle\, (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0)\,\rangle_{\mathcal{X}} := \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0\,,$$
$$\langle\, (\vec{\rho}, \vec{\rho}_0), (\vec{w}, \vec{w}_0)\,\rangle_{\mathcal{Y}} := \vec{\rho}^{\,T} \vec{w} + \vec{\rho}_0^{\,T} \vec{w}_0\,.$$

The Inverse Problem

We define the following operator

$$F : \mathcal{X} \to \mathcal{Y}\,, \quad (\vec{v}, \vec{\rho}_0) \mapsto (\rho(\vec{v}, \vec{\rho}_0), \vec{\rho}_0)\,,$$

where the inner products on $\mathcal{X}$ and $\mathcal{Y}$ are given by

$$\langle\, (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0)\,\rangle_{\mathcal{X}} := \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0\,,$$
$$\langle\, (\vec{\rho}, \vec{\rho}_0), (\vec{w}, \vec{w}_0)\,\rangle_{\mathcal{Y}} := \vec{\rho}^{\,T} \vec{w} + \vec{\rho}_0^{\,T} \vec{w}_0\,.$$

We can now write our problem in standard form, i.e.,

$$\text{"}\; F\,(\vec{v}, \vec{\rho}_0) = \left(\vec{\rho}^{\,\delta}, \vec{\rho}_0^{\,\delta}\right) \;\text{"}$$

Derivative and Adjoint

Derivative and Adjoint

The Frechet derivative is given by

$$F'(\vec{v}, \vec{\rho}_0)(\Delta\vec{v}, \Delta\vec{\rho}_0) = (\rho'(\vec{v}, \vec{\rho}_0)(\Delta\vec{v}, \Delta\vec{\rho}_0), \Delta\vec{\rho}_0)\,,$$

Derivative and Adjoint

The Frechet derivative is given by

$$F'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0) = (\rho'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0),\Delta\vec{\rho}_0)\,,$$

where

$$A(\vec{v})[\rho'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0)] = -(A'(\vec{v})\Delta\vec{v})\rho(\vec{v},\vec{\rho}_0)+b'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0)\,.$$

Derivative and Adjoint

The Frechet derivative is given by

$$F'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0) = (\rho'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0),\Delta\vec{\rho}_0)\,,$$

where

$$A(\vec{v})[\rho'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0)] = -(A'(\vec{v})\Delta\vec{v})\rho(\vec{v},\vec{\rho}_0)+b'(\vec{v},\vec{\rho}_0)(\Delta\vec{v},\Delta\vec{\rho}_0)\,.$$

It's adjoint is given by

$$F'(\vec{v},\vec{\rho}_0)^*(\vec{w},\vec{w}_0) = \begin{pmatrix} H^{-1}\left(-D_A(\vec{v},\rho(\vec{v},\vec{\rho}_0))^T + b'_{\Delta\vec{\rho}_0}(\vec{v},\vec{\rho}_0)^T\right)A(\vec{v})^{-T}\vec{w} \\ b'_{\Delta\vec{v}}(\vec{v},\vec{\rho}_0)^T A(\vec{v})^{-T}\vec{w} + \vec{w}_0 \end{pmatrix}.$$

Divergence Free Condition

In the derivation of the advection equation we used

$$\mathrm{div}\,[v(x,y,z)] = 0\,.$$

The reconstruction method should take that into account.

Divergence Free Condition

In the derivation of the advection equation we used

$$\mathrm{div}\,[v(x,y,z)] = 0\,.$$

The reconstruction method should take that into account.

- **Idea:** Choose space $\mathcal{X}$ as a divergence free space.

Divergence Free Condition

In the derivation of the advection equation we used

$$\operatorname{div}\left[v(x,y,z)\right] = 0\,.$$

The reconstruction method should take that into account.

- **Idea:** Choose space $\mathcal{X}$ as a divergence free space.
- **Problem:** Frechet derivative becomes unhandy.

Divergence Free Condition

In the derivation of the advection equation we used

$$\text{div}\,[v(x,y,z)] = 0\,.$$

The reconstruction method should take that into account.

- **Idea:** Choose space $\mathcal{X}$ as a divergence free space.
- **Problem:** Frechet derivative becomes unhandy.
- **Solution:** Enforce *weak* divergence free condition.

Divergence Free Condition

In the derivation of the advection equation we used

$$\operatorname{div}\,[v(x,y,z)] = 0\,.$$

The reconstruction method should take that into account.

- **Idea:** Choose space $\mathcal{X}$ as a divergence free space.
- **Problem:** Frechet derivative becomes unhandy.
- **Solution:** Enforce *weak* divergence free condition.

$$\Longrightarrow F(\vec{v},\vec{\rho}_0) := (\rho(\vec{v},\vec{\rho}_0), \vec{\rho}_0, D\vec{v})\,.$$

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

The matrix H should approximate the H^1 inner product.

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

The matrix H should approximate the H^1 inner product.

- **Idea:** Derive H from FEM basis functions.

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

The matrix H should approximate the H^1 inner product.

- **Idea:** Derive H from FEM basis functions.
- **Problem:** Matrix H becomes hard to invert.

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

The matrix H should approximate the H^1 inner product.

- **Idea:** Derive H from FEM basis functions.
- **Problem:** Matrix H becomes hard to invert.
- **Solution 1:** Use only the diagonal entries.

Choosing the matrix H

Remember the inner product:

$$\langle (\vec{v}, \vec{\rho}_0), (\vec{x}, \vec{w}_0) \rangle_{\mathcal{X}} = \vec{v}^{\,T} H \vec{x} + \vec{\rho}_0^{\,T} \vec{w}_0 \,.$$

The matrix H should approximate the H^1 inner product.

- **Idea:** Derive H from FEM basis functions.
- **Problem:** Matrix H becomes hard to invert.
- **Solution 1:** Use only the diagonal entries.
- **Solution 2:** Use Wavelets instead of H.

Nonlinear Inverse Problems

Nonlinear Inverse Problems

Problem

$$F(x) = y^{\delta}$$

Nonlinear Inverse Problems

Problem

$$F(x) = y^\delta$$

Important questions:

Nonlinear Inverse Problems

Problem

$$F(x) = y^{\delta}$$

Important questions:

- Existence and uniqueness of solutions.

Nonlinear Inverse Problems

Problem

$$F(x) = y^\delta$$

Important questions:

- Existence and uniqueness of solutions.
- How to approximate/compute particular solutions.

Nonlinear Inverse Problems

Problem

$$F(x) = y^\delta$$

Important questions:

- Existence and uniqueness of solutions.
- How to approximate/compute particular solutions.
- *How to do it in an efficient way!!*

Nonlinear Inverse Problems and TPG methods

- Tikhonov Regularization (suitable α)

$$\min_x \left\{ \frac{1}{2} \left\| F(x) - y^\delta \right\|^2 + \frac{\alpha}{2} \left\| x - x_0 \right\|^2 \right\} .$$

- Tikhonov Regularization (suitable α)

$$\min_x \left\{ \frac{1}{2} \left\| F(x) - y^\delta \right\|^2 + \frac{\alpha}{2} \left\| x - x_0 \right\|^2 \right\} .$$

- Landweber Iteration (suitable stopping rule)

$$x_{k+1}^\delta = x_k^\delta + F'(x_k^\delta)^* (y^\delta - F(x_k^\delta)) .$$

- Tikhonov Regularization (suitable α)

$$\min_x \left\{ \frac{1}{2} \left\| F(x) - y^\delta \right\|^2 + \frac{\alpha}{2} \|x - x_0\|^2 \right\} .$$

- Landweber Iteration (suitable stopping rule)

$$x_{k+1}^\delta = x_k^\delta + F'(x_k^\delta)^*(y^\delta - F(x_k^\delta)) .$$

- Levenberg Marquardt method

$$x_{k+1}^\delta = x_k^\delta + (F'(x_k^\delta)^* F'(x_k^\delta) + \alpha I)^{-1} F'(x_k^\delta)^*(y^\delta - F(x_k^\delta)) .$$

- Tikhonov Regularization (suitable α)

$$\min_x \left\{ \frac{1}{2} \left\| F(x) - y^\delta \right\|^2 + \frac{\alpha}{2} \left\| x - x_0 \right\|^2 \right\} .$$

- Landweber Iteration (suitable stopping rule)

$$x_{k+1}^\delta = x_k^\delta + F'(x_k^\delta)^*(y^\delta - F(x_k^\delta)) .$$

- Levenberg Marquardt method

$$x_{k+1}^\delta = x_k^\delta + (F'(x_k^\delta)^* F'(x_k^\delta) + \alpha I)^{-1} F'(x_k^\delta)^*(y^\delta - F(x_k^\delta)) .$$

- Iteratively regularized Gauss-Newton method

$$x_{k+1}^\delta = x_k^\delta + (F'(x_k^\delta)^* F'(x_k^\delta) + \alpha I)^{-1} (F'(x_k^\delta)^*(y^\delta - F(x_k^\delta)) + \alpha_k (x_0 - x_k^\delta)) .$$

Connection: Residual Functional

$$\Phi(x) = \frac{1}{2}\left\|F(x) - y^\delta\right\|^2$$

Connection: Residual Functional

$$\Phi(x) = \frac{1}{2}\left\|F(x) - y^\delta\right\|^2$$

- Tikhonov = Minimize{ $\Phi(x)$ + Regularization(x) }.

Connection: Residual Functional

$$\Phi(x) = \frac{1}{2} \left\| F(x) - y^\delta \right\|^2$$

- Tikhonov = Minimize{ $\Phi(x)$ + Regularization(x) }.
- Landweber = Gradient Descent for $\Phi(x)$.

Connection: Residual Functional

$$\Phi(x) = \frac{1}{2} \left\| F(x) - y^\delta \right\|^2$$

- Tikhonov = Minimize{ $\Phi(x)$ + Regularization(x) }.
- Landweber = Gradient Descent for $\Phi(x)$.
- Levenberg Marquardt = 2nd order descent for $\Phi(x)$.

Connection: Residual Functional

$$\Phi(x) = \frac{1}{2} \left\| F(x) - y^\delta \right\|^2$$

- Tikhonov = Minimize{ $\Phi(x)$ + Regularization(x) }.
- Landweber = Gradient Descent for $\Phi(x)$.
- Levenberg Marquardt = 2nd order descent for $\Phi(x)$.
- Iteratively regularized Gauss-Newton
  = 2nd order descent for $\Phi(x)$ + Tikhonov Type Stabilization

Pros and Cons

Pros and Cons

- Tikhonov Regularization
  - Pros: Weak conditions for analysis, very versatile.
  - Cons: Computation of the minimum.

Pros and Cons

- Tikhonov Regularization
  - Pros: Weak conditions for analysis, very versatile.
  - Cons: Computation of the minimum.
- Landweber Iteration
  - Pros: Easily implementable, often produces good results.
  - Cons: Strong conditions for analysis, slow convergence.

Pros and Cons

- Tikhonov Regularization
  - Pros: Weak conditions for analysis, very versatile.
  - Cons: Computation of the minimum.
- Landweber Iteration
  - Pros: Easily implementable, often produces good results.
  - Cons: Strong conditions for analysis, slow convergence.
- Second Order Methods
  - Pros: Fast convergence.
  - Cons: Even stronger conditions for analysis.

Pros and Cons

- Tikhonov Regularization
  - Pros: Weak conditions for analysis, very versatile.
  - Cons: Computation of the minimum.
- Landweber Iteration
  - Pros: Easily implementable, often produces good results.
  - Cons: Strong conditions for analysis, slow convergence.
- Second Order Methods
  - Pros: Fast convergence.
  - Cons: Even stronger conditions for analysis. Inversion of

$$(F'(x)^* F'(x) + \alpha_k I)$$

in every iteration step $\rightarrow$ difficult and takes time.

Acceleration Techniques

Acceleration Techniques

- Landweber Iteration with operator approximation:

$$x_{k+1}^\delta = x_k^\delta + \tilde{F}'(x_k^\delta)^*(y^\delta - \tilde{F}(x_k^\delta)) \,.$$

Acceleration Techniques

- Landweber Iteration with operator approximation:

$$x_{k+1}^{\delta} = x_k^{\delta} + \tilde{F}'(x_k^{\delta})^*(y^{\delta} - \tilde{F}(x_k^{\delta}))\,.$$

- Landweber Iteration in Hilbert Scales:

$$x_{k+1}^{\delta} = x_k^{\delta} + L^{-2s} F'(x_k^{\delta})^*(y^{\delta} - F(x_k^{\delta}))\,.$$

Acceleration Techniques

- Landweber Iteration with operator approximation:

$$x_{k+1}^{\delta} = x_k^{\delta} + \tilde{F}'(x_k^{\delta})^*(y^{\delta} - \tilde{F}(x_k^{\delta}))\,.$$

- Landweber Iteration in Hilbert Scales:

$$x_{k+1}^{\delta} = x_k^{\delta} + L^{-2s} F'(x_k^{\delta})^*(y^{\delta} - F(x_k^{\delta}))\,.$$

- Landweber Iteration with intelligent stepsizes:

$$x_{k+1}^{\delta} = x_k^{\delta} + \alpha_k^{\delta} F'(x_k^{\delta})^*(y^{\delta} - F(x_k^{\delta}))\,.$$

Acceleration Techniques

- Landweber Iteration with operator approximation:

$$x_{k+1}^\delta = x_k^\delta + \tilde{F}'(x_k^\delta)^*(y^\delta - \tilde{F}(x_k^\delta))\,.$$

- Landweber Iteration in Hilbert Scales:

$$x_{k+1}^\delta = x_k^\delta + L^{-2s} F'(x_k^\delta)^*(y^\delta - F(x_k^\delta))\,.$$

- Landweber Iteration with intelligent stepsizes:

$$x_{k+1}^\delta = x_k^\delta + \alpha_k^\delta F'(x_k^\delta)^*(y^\delta - F(x_k^\delta))\,.$$

Examples: Steepest Descent, Barzilai-Borwein, Neubauer.

Nesterov Acceleration

Nesterov Acceleration

General minimization problem

$$\min_x \{\Phi(x)\} \ .$$

Nesterov Acceleration

General minimization problem

$$\min_x \{\Phi(x)\} \, .$$

Yurii Nesterov: Instead of using gradient descent:

$$x_{k+1} = x_k - \omega \nabla \Phi(x_k) \, ,$$

Nesterov Acceleration

General minimization problem

$$\min_x \{\Phi(x)\} .$$

Yurii Nesterov: Instead of using gradient descent:

$$x_{k+1} = x_k - \omega \nabla \Phi(x_k) ,$$

use the following iteration:

$$\begin{aligned} z_k &= x_k + \tfrac{k-1}{k+\alpha-1}(x_k - x_{k-1}) \\ x_{k+1} &= z_k - \omega \nabla \Phi(z_k) . \end{aligned}$$

What's so good about that?

What's so good about that?

- **Assume:** Φ is convex.

What's so good about that?

- **Assume:** Φ is convex.
- **Gradient Descent:**

$$\left\| \Phi(x_k) - \Phi(x^\dagger) \right\| = \mathcal{O}(k^{-1})$$

What's so good about that?

- **Assume:** Φ is convex.
- **Gradient Descent:**

$$\left\|\Phi(x_k) - \Phi(x^\dagger)\right\| = \mathcal{O}(k^{-1})$$

- **Nesterov Acceleration:**

$$\left\|\Phi(x_k) - \Phi(x^\dagger)\right\| = \mathcal{O}(k^{-2})$$

Application to Nonlinear Ill-Posed Problems

Application to Nonlinear Ill-Posed Problems

For our problem, the method reads as

$$
\begin{aligned}
z_k^\delta &= x_k^\delta + \tfrac{k-1}{k+\alpha-1}(x_k^\delta - x_{k-1}^\delta) \\
x_{k+1}^\delta &= z_k^\delta + \alpha_k^\delta \, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.
\end{aligned}
$$

Application to Nonlinear Ill-Posed Problems

For our problem, the method reads as

$$\begin{aligned} z_k^\delta &= x_k^\delta + \tfrac{k-1}{k+\alpha-1}(x_k^\delta - x_{k-1}^\delta) \\ x_{k+1}^\delta &= z_k^\delta + \alpha_k^\delta \, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta)) \,. \end{aligned}$$

There is a generalization to deal with

$$\min\{\Phi(x) + \Psi(x)\} \,.$$

Application to Nonlinear Ill-Posed Problems

For our problem, the method reads as

$$z_k^\delta = x_k^\delta + \tfrac{k-1}{k+\alpha-1}(x_k^\delta - x_{k-1}^\delta)$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta \, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta)) \, .$$

There is a generalization to deal with

$$\min\{\Phi(x) + \Psi(x)\} \, .$$

This can be used to incorporate sparsity constraints via

$$z_k^\delta = x_k^\delta + \tfrac{k-1}{k+\alpha-1}(x_k^\delta - x_{k-1}^\delta) \, ,$$
$$x_{k+1}^\delta = S_{\alpha_k^\delta \alpha, p}\left(z_k^\delta + \alpha_k^\delta \, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\right) \, .$$

Neubauer strikes again

Neubauer strikes again

- **Assumptions:** Linear operator $F(x) = Tx$, source condition $x^\dagger \in \mathcal{R}((T^*T)^\mu)$, a priori stopping rule.

Neubauer strikes again

- **Assumptions:** Linear operator $F(x) = Tx$, source condition $x^\dagger \in \mathcal{R}((T^*T)^\mu)$, a priori stopping rule.

- If $0 \leq \mu \leq \frac{1}{2}$, then

$$k(\delta) = \mathcal{O}(\delta^{-\frac{1}{2\mu+1}}) , \qquad \left\| x_{k(\delta)}^\delta - x^\dagger \right\| = o(\delta^{\frac{2\mu}{2\mu+1}}) .$$

Neubauer strikes again

- **Assumptions:** Linear operator $F(x) = Tx$, source condition $x^\dagger \in \mathcal{R}((T^*T)^\mu)$, a priori stopping rule.

- If $0 \leq \mu \leq \frac{1}{2}$, then

$$k(\delta) = \mathcal{O}(\delta^{-\frac{1}{2\mu+1}}), \qquad \left\| x_{k(\delta)}^\delta - x^\dagger \right\| = o(\delta^{\frac{2\mu}{2\mu+1}}).$$

- If $\mu > \frac{1}{2}$, then

$$k(\delta) = \mathcal{O}(\delta^{-\frac{2}{2\mu+3}}), \qquad \left\| x_{k(\delta)}^\delta - x^\dagger \right\| = o(\delta^{\frac{2\mu+1}{2\mu+3}}).$$

Neubauer strikes again

- **Assumptions:** Linear operator $F(x) = Tx$, source condition $x^\dagger \in \mathcal{R}((T^*T)^\mu)$, a priori stopping rule.
- If $0 \leq \mu \leq \frac{1}{2}$, then

$$k(\delta) = \mathcal{O}(\delta^{-\frac{1}{2\mu+1}}) \,, \qquad \left\| x_{k(\delta)}^\delta - x^\dagger \right\| = o(\delta^{\frac{2\mu}{2\mu+1}}) \,.$$

- If $\mu > \frac{1}{2}$, then

$$k(\delta) = \mathcal{O}(\delta^{-\frac{2}{2\mu+3}}) \,, \qquad \left\| x_{k(\delta)}^\delta - x^\dagger \right\| = o(\delta^{\frac{2\mu+1}{2\mu+3}}) \,.$$

- Similar results also when using the discrepancy principle.

Andreas Neubauer, *On Nesterov Acceleration for Landweber Iteration of Linear Ill-Posed Problems*, in press

Two-Point Gradient (TPG) Methods

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Question: Do they converge under standard assumptions?

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Question: Do they converge under standard assumptions?

- Yes for linear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$ $\leftarrow$ Neubauer

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Question: Do they converge under standard assumptions?

- Yes for linear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$ $\leftarrow$ Neubauer
- Yes for $\lambda_k^\delta \to 0$ fast enough.

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta) \,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^* (y^\delta - F(z_k^\delta)) \,.$$

Question: Do they converge under standard assumptions?

- Yes for linear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$ $\leftarrow$ Neubauer
- Yes for $\lambda_k^\delta \to 0$ fast enough.
- Yes for some explicit choices of λ_k^δ.

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta) \,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^* (y^\delta - F(z_k^\delta)) \,.$$

Question: Do they converge under standard assumptions?

- Yes for linear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$ $\leftarrow$ Neubauer
- Yes for $\lambda_k^\delta \to 0$ fast enough.
- Yes for some explicit choices of λ_k^δ.
- Yes for λ_k^δ defined via a backtracking search.

Two-Point Gradient (TPG) Methods

How about general methods of the form

$$z_k^\delta = x_k^\delta + \lambda_k^\delta (x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Question: Do they converge under standard assumptions?

- Yes for linear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$ $\leftarrow$ Neubauer
- Yes for $\lambda_k^\delta \to 0$ fast enough.
- Yes for some explicit choices of λ_k^δ.
- Yes for λ_k^δ defined via a backtracking search.

Open: Convergence for nonlinear problems and $\lambda_k^\delta = \frac{k-1}{k+\alpha-1}$

Convergence Conditions

Convergence Conditions

- Nonlinearity Condition

$$\left\| F(x) - F(\tilde{x}) - F'(x)(x - \tilde{x}) \right\| \leq \eta \left\| F(x) - F(\tilde{x}) \right\| ,$$

$$x, \tilde{x} \in \mathcal{B}_{4\rho}(x_0) \subset \mathcal{D}(F) , \qquad \eta < \frac{1}{2} .$$

Convergence Conditions

- Nonlinearity Condition

$$\left\| F(x) - F(\tilde{x}) - F'(x)(x - \tilde{x}) \right\| \leq \eta \left\| F(x) - F(\tilde{x}) \right\| ,$$
$$x, \tilde{x} \in \mathcal{B}_{4\rho}(x_0) \subset \mathcal{D}(F) , \qquad \eta < \frac{1}{2} .$$

- Parameters $0 \leq \lambda_k^\delta \leq 1$ and stepsizes $\alpha_k^\delta > 0$ satisfy

Convergence Conditions

- Nonlinearity Condition

$$\left\| F(x) - F(\tilde{x}) - F'(x)(x - \tilde{x}) \right\| \leq \eta \left\| F(x) - F(\tilde{x}) \right\| \,,$$
$$x, \tilde{x} \in \mathcal{B}_{4\rho}(x_0) \subset \mathcal{D}(F) \,, \qquad \eta < \frac{1}{2} \,.$$

- Parameters $0 \leq \lambda_k^\delta \leq 1$ and stepsizes $\alpha_k^\delta > 0$ satisfy

$$\lambda_k^\delta(\lambda_k^\delta + 1) \left\| x_k^\delta - x_{k+1}^\delta \right\|^2 - \left(1 + \frac{\Psi}{\mu}\right) \alpha_k^\delta \left\| F(z_k^\delta) - y^\delta \right\|^2$$
$$+ (\alpha_k^\delta)^2 \left\| F'(z_k^\delta)^*(F(z_k^\delta) - y^\delta) \right\|^2 \leq 0 \,.$$

Convergence Conditions

- Nonlinearity Condition

$$\left\| F(x) - F(\tilde{x}) - F'(x)(x - \tilde{x}) \right\| \leq \eta \left\| F(x) - F(\tilde{x}) \right\| \,, \qquad x, \tilde{x} \in \mathcal{B}_{4\rho}(x_0) \subset \mathcal{D}(F) \,, \qquad \eta < \frac{1}{2} \,.$$

- Parameters $0 \leq \lambda_k^\delta \leq 1$ and stepsizes $\alpha_k^\delta > 0$ satisfy

$$\lambda_k^\delta(\lambda_k^\delta + 1) \left\| x_k^\delta - x_{k+1}^\delta \right\|^2 - \left(1 + \frac{\Psi}{\mu}\right) \alpha_k^\delta \left\| F(z_k^\delta) - y^\delta \right\|^2 + (\alpha_k^\delta)^2 \left\| F'(z_k^\delta)^*(F(z_k^\delta) - y^\delta) \right\|^2 \leq 0 \,.$$

- Parameters λ_k^δ satisfy

$$\sum_{k=0}^{\infty} \lambda_k^0 \left\| x_k^0 - x_{k-1}^0 \right\| < \infty \,.$$

Some Possible Choices

Some Possible Choices

For the stepsizes α_k^δ, one can use

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,
- the steepest descent stepsize or the minimal error stepsize.

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,
- the steepest descent stepsize or the minimal error stepsize.

The parameters λ_k^δ can be chosen

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,
- the steepest descent stepsize or the minimal error stepsize.

The parameters λ_k^δ can be chosen

- as any sequence decaying sufficiently fast,

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,
- the steepest descent stepsize or the minimal error stepsize.

The parameters λ_k^δ can be chosen

- as any sequence decaying sufficiently fast,
- explicitly via

$$\lambda_k^\delta = \min\left\{ -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{\Psi(\tau\delta)^2}{\mu\bar{\omega}^2 \left\| x_k^\delta - x_{k-1}^\delta \right\|^2}}, 1 \right\},$$

Some Possible Choices

For the stepsizes α_k^δ, one can use

- a constant stepsize $\alpha_k^\delta = \omega$,
- the steepest descent stepsize or the minimal error stepsize.

The parameters λ_k^δ can be chosen

- as any sequence decaying sufficiently fast,
- explicitly via

$$\lambda_k^\delta = \min\left\{ -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{\Psi(\tau\delta)^2}{\mu\bar{\omega}^2 \left\| x_k^\delta - x_{k-1}^\delta \right\|^2}}, 1 \right\},$$

- via a backtracking algorithm.

Example Problem: SPECT

Example Problem: SPECT

$$A(f,\mu)(s,\omega) := \int_{\mathbb{R}} f(s\omega^{\perp} + t\omega) \exp\left(-\int_{t}^{\infty} \mu(s\omega^{\perp} + r\omega)\, dr \right) dt \,.$$

Example Problem: SPECT

$$A(f,\mu)(s,\omega) := \int_{\mathbb{R}} f(s\omega^{\perp} + t\omega) \exp\left(- \int_{t}^{\infty} \mu(s\omega^{\perp} + r\omega)\, dr \right) dt \,.$$

$\lambda_k^\delta = 0$	Backtracking λ_k^δ	Explicit λ_k^δ	Nesterov λ_k^δ	k_*	Time
x				3433	489 s
	x			631	90 s
		x		345	77 s
			x	205	30 s

Example Problem: SPECT

$$A(f,\mu)(s,\omega) := \int_{\mathbb{R}} f(s\omega^{\perp} + t\omega) \exp\left(- \int_{t}^{\infty} \mu(s\omega^{\perp} + r\omega)\, dr \right) dt \,.$$

$\lambda_k^\delta = 0$	Backtracking λ_k^δ	Explicit λ_k^δ	Nesterov λ_k^δ	k_*	Time
x				3433	489 s
	x			631	90 s
		x		345	77 s
			x	205	30 s

Reading suggestion: *Convergence Analysis of a Two-Point Gradient Method for Nonlinear Ill-Posed Problems*, Hubmer, Ramlau, submitted.

Application to MRAI

Application to MRAI

Iterative procedure

$$z_k^\delta = x_k^\delta + \tfrac{k-1}{k+2}(x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = S_{\alpha_k^\delta \alpha, p}\left(z_k^\delta + \alpha_k^\delta \, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\right)\,.$$

Application to MRAI

Iterative procedure

$$\begin{aligned} z_k^\delta &= x_k^\delta + \tfrac{k-1}{k+2}(x_k^\delta - x_{k-1}^\delta)\,, \\ x_{k+1}^\delta &= S_{\alpha_k^\delta \alpha, p}\left(z_k^\delta + \alpha_k^\delta F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\right)\,. \end{aligned}$$

Steepest descent stepsize:

$$\alpha_k^\delta = \frac{\left\| s_k \right\|^2}{\left\| F'(z_k^\delta) s_k^\delta \right\|^2}\,, \qquad s_k^\delta = F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Application to MRAI

Iterative procedure

$$z_k^\delta = x_k^\delta + \tfrac{k-1}{k+2}(x_k^\delta - x_{k-1}^\delta)\,,$$
$$x_{k+1}^\delta = S_{\alpha_k^\delta \alpha, p}\left(z_k^\delta + \alpha_k^\delta\, F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\right)\,.$$

Steepest descent stepsize:

$$\alpha_k^\delta = \frac{\|s_k\|^2}{\left\|F'(z_k^\delta)s_k^\delta\right\|^2}\,, \qquad s_k^\delta = F'(z_k^\delta)^*(y^\delta - F(z_k^\delta))\,.$$

Discrepancy principle:

$$\left\|y^\delta - F(z_{k_*}^\delta)\right\| \leq \tau\delta < \left\|y^\delta - F(z_k^\delta)\right\|\,, \quad 0 \leq k \leq k_*\,.$$

Implementation Details

- **Software:** MATLAB R2015b.
- **Solver:** biCGstab with iLU preconditioner.
- **Parallelization:** As far as possible.
- **Essential:** Stefan Engblom's *fsparse.m* file.

Implementation Details

- **Software:** MATLAB R2015b.
- **Solver:** biCGstab with iLU preconditioner.
- **Parallelization:** As far as possible.
- **Essential:** Stefan Engblom's *fsparse.m* file.

$\Longrightarrow$ *Runs on a standard home computer in acceptable time!!!*

(Real-world data set has 3 million unknowns)

Simulation Outline

Simulation Outline

Steps of the data creation:

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
    - Choose a constant velocity $\bar{v}$ pointing in vessel direction.

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.
   - Notice that then $\rho(x, y, z, t) = \rho_0(x - \bar{v}_1 t, y - \bar{v}_2 t, z - \bar{v}_3 t)$.

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.
   - Notice that then $\rho(x, y, z, t) = \rho_0(x - \bar{v}_1 t, y - \bar{v}_2 t, z - \bar{v}_3 t)$.
   - Sample at the right space-time points to get $\rho_{i,j,k,l}$

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.
   - Notice that then $\rho(x, y, z, t) = \rho_0(x - \bar{v}_1 t, y - \bar{v}_2 t, z - \bar{v}_3 t)$.
   - Sample at the right space-time points to get $\rho_{i,j,k,l}$
3. Combine the vessel contributions.

Simulation Outline

Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.
   - Notice that then $\rho(x, y, z, t) = \rho_0(x - \bar{v}_1 t, y - \bar{v}_2 t, z - \bar{v}_3 t)$.
   - Sample at the right space-time points to get $\rho_{i,j,k,l}$
3. Combine the vessel contributions.
4. Add a random data error of magnitude δ.

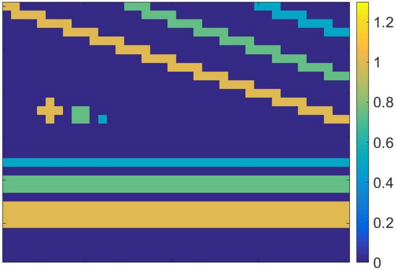
Simulation Outline

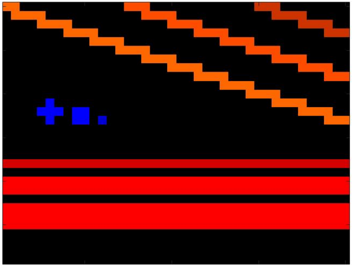
Steps of the data creation:

1. Prepare a phantom of size $40 \times 30 \times 30$ featuring several vessels of different thickness and orientation.
2. For every vessel:
   - Choose a constant velocity $\bar{v}$ pointing in vessel direction.
   - Choose an initial signal ρ_0 of sinusoidal form.
   - Notice that then $\rho(x, y, z, t) = \rho_0(x - \bar{v}_1 t, y - \bar{v}_2 t, z - \bar{v}_3 t)$.
   - Sample at the right space-time points to get $\rho_{i,j,k,l}$
3. Combine the vessel contributions.
4. Add a random data error of magnitude δ.

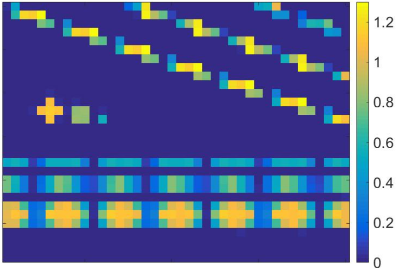
$\Longrightarrow$ Run the algorithm using the discrepancy principle ($\tau = 1.1$).

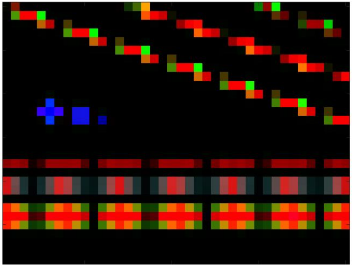
Simulation Phantom - MIP and direction MIP



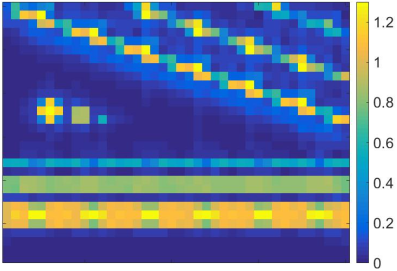


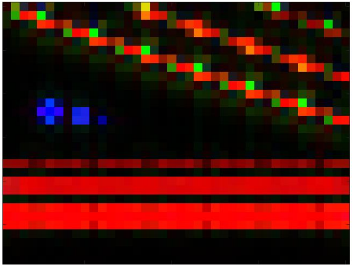
Results - Pure Method



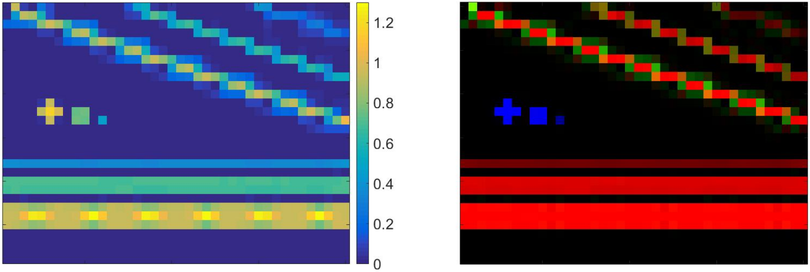


Results - Divergence-Free





Results - Divergence-Free + Wavelets + Sparsity



Natural Stimulation Data Set

Natural Stimulation Data Set

Specifications:

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.
- Pulse repetition time (TR) of 2 seconds.

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.
- Pulse repetition time (TR) of 2 seconds.
- Eight 15 minutes long segments for each subject.

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.
- Pulse repetition time (TR) of 2 seconds.
- Eight 15 minutes long segments for each subject.

Algorithm specifics:

Natural Stimulation Data Set

Specifications:

- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.
- Pulse repetition time (TR) of 2 seconds.
- Eight 15 minutes long segments for each subject.

Algorithm specifics:

- First 20 seconds of second segment were used.

Natural Stimulation Data Set

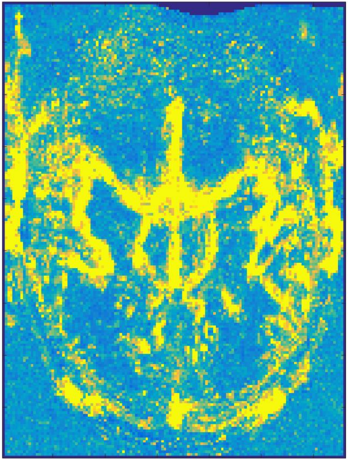
Specifications:

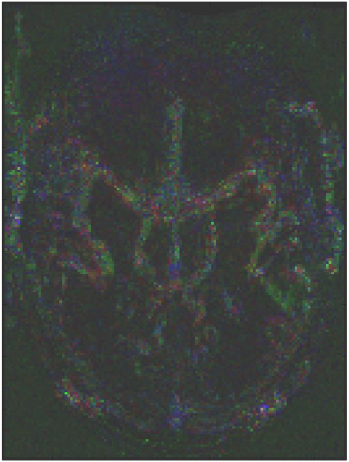
- Publicly available natural stimulation dynamic EPI data.
- Data has dimension $132 \times 175 \times 48$.
- 7.0 T MRI scanner, 1.4 mm isotropic spatial resolution.
- Pulse repetition time (TR) of 2 seconds.
- Eight 15 minutes long segments for each subject.

Algorithm specifics:

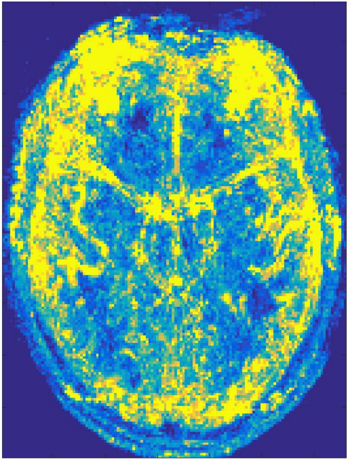
- First 20 seconds of second segment were used.
- **Stopping rule:** Residual decrease check.

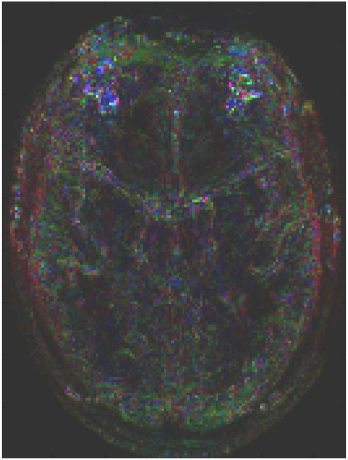
Regression Approach - Results



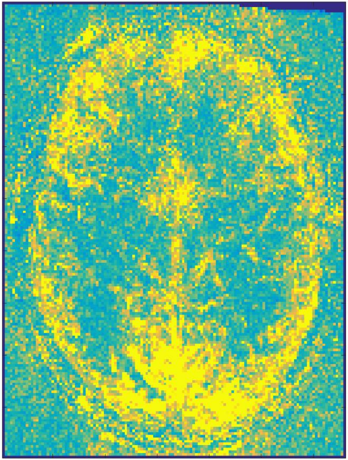


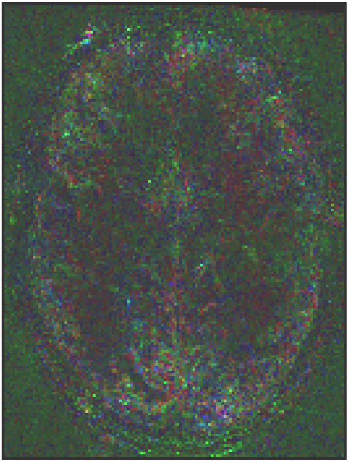
New Approach - Results



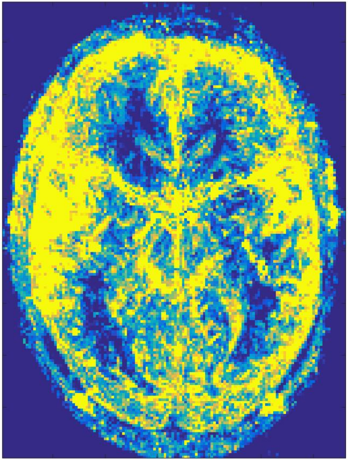


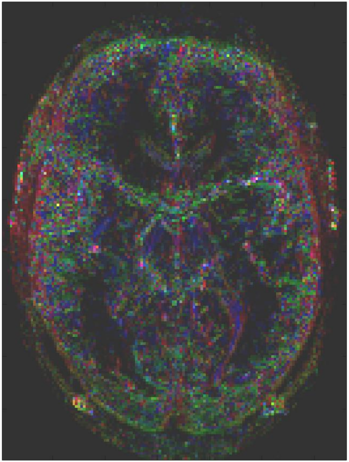
Regression Approach - Results





New Approach - Results





End

Thank you for your attention!