Advanced Cognitive Neuroscience

Week 43: Forward modelling and dipole estimation

The course plan

Week 36:
Lesson 0: What is it all about?
Class 0: Setting up UCloud and installing MNE-Python

Week 37:
No Teaching

Week 38:
Lesson 1: Workshop paradigm: Measuring visual subjective experience + MR Recordings
Class 1: Running an MEG analysis of visual responses

Week 39:
MEG workshop: Measuring and predicting visual subjective experience

Week 40:
Lesson 2: Basic physiology and Evoked responses
Class 2: Evoked responses to different levels of subjective experience

Week 41:
Lesson 3: Multivariate statistics
Class 3: Predicting subjective experience in sensor space
Deadline for feedback: Video Explainer

Week 42:
Autumn Break

Week 43:
Lesson 4: Forward modelling and dipole estimation
Class 4: Creating a forward model and fitting dipoles

Week 44:
Lesson 5: Inverse modelling: Minimum-norm estimate
Class 5: Predicting subjective experience in source space

Week 45:
Lesson 6: Inverse modelling: Beamforming
Class 6: Predicting subjective experience in source space, continued

Week 46:
Lesson 7: What about that other cortex? - the cerebellar one
Class 7: Oral presentations (part 1)
Deadline for feedback: Lab report

Week 47:
Lesson 8: Guest lecture: Laura Bock Paulsen: Respiratory analyses
Class 8: Oral presentations (part 2)

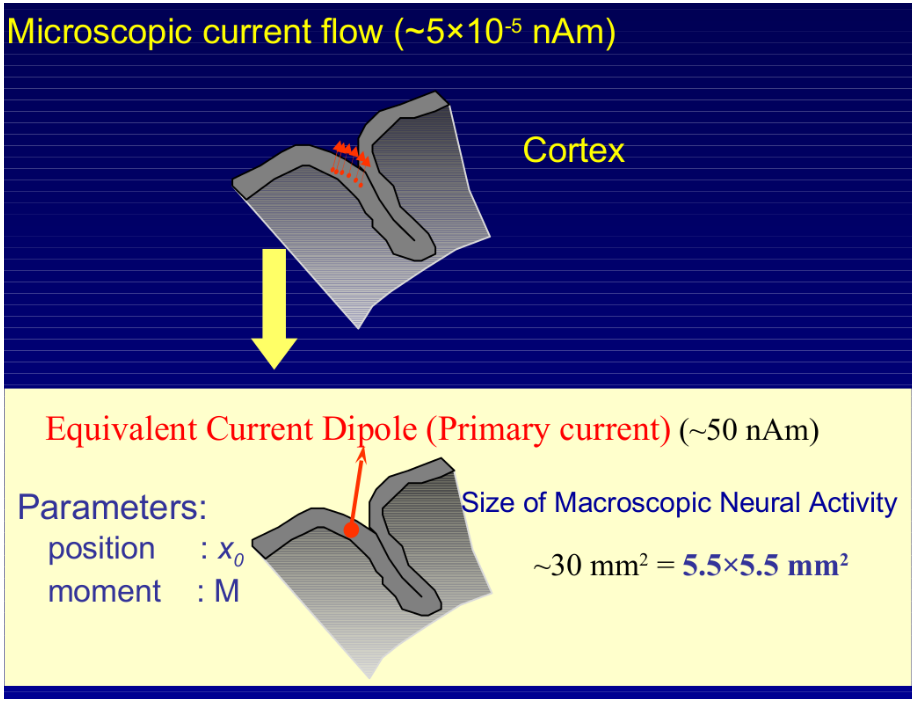
Week 48:
Lesson 9: Guest lecture: Barbara Pomiechowska: Using OPM-MEG to study brain and cognitive development in infancy
Class 9: Oral presentations (part 3)

Week 49:
Lesson 0 again: What was it all about?
Class 10: Oral presentations (part 4)

Learning goals

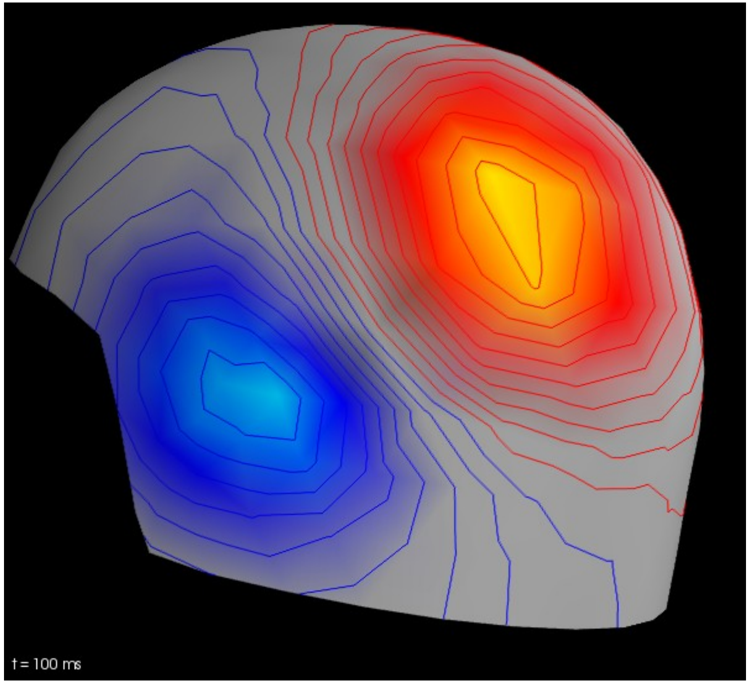
- Learning
  - what the ingredients of a forward model are
  - how the forward model links sources of the brain to sensors
  - what co-registration of MRI and MEG amounts to
  - how to fit dipoles

Equivalent Current Dipole



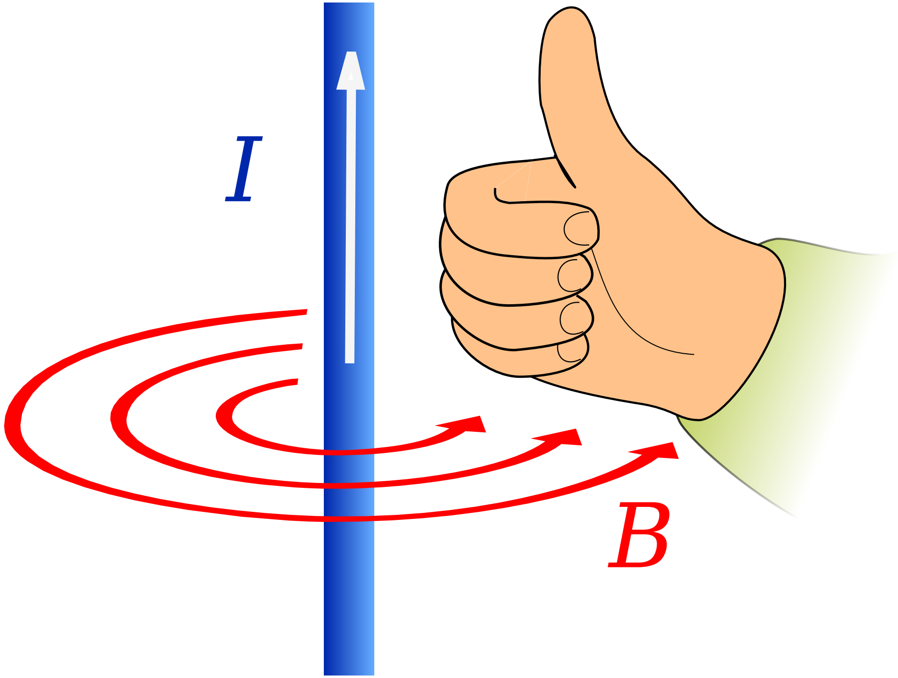
Stephanie Sillekens

Field plot

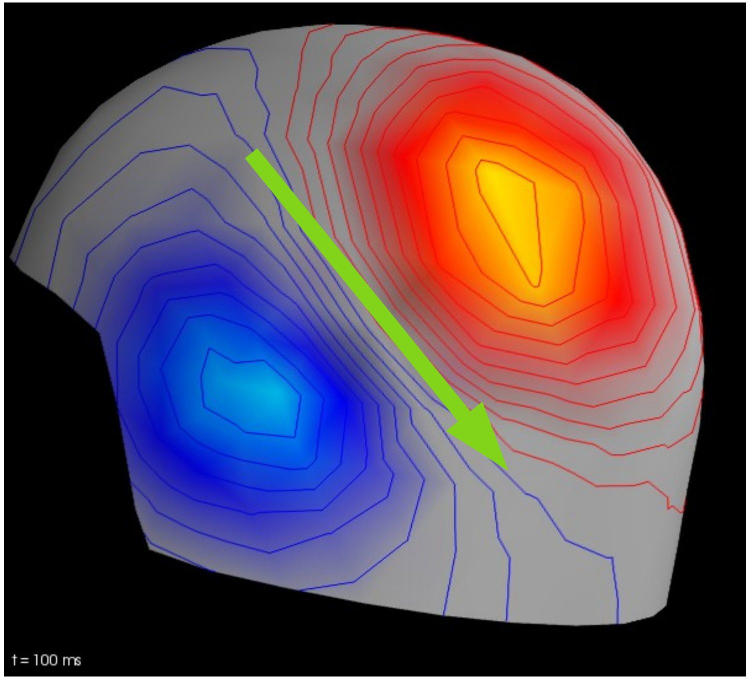


Which way does current flow in the underlying dipole?

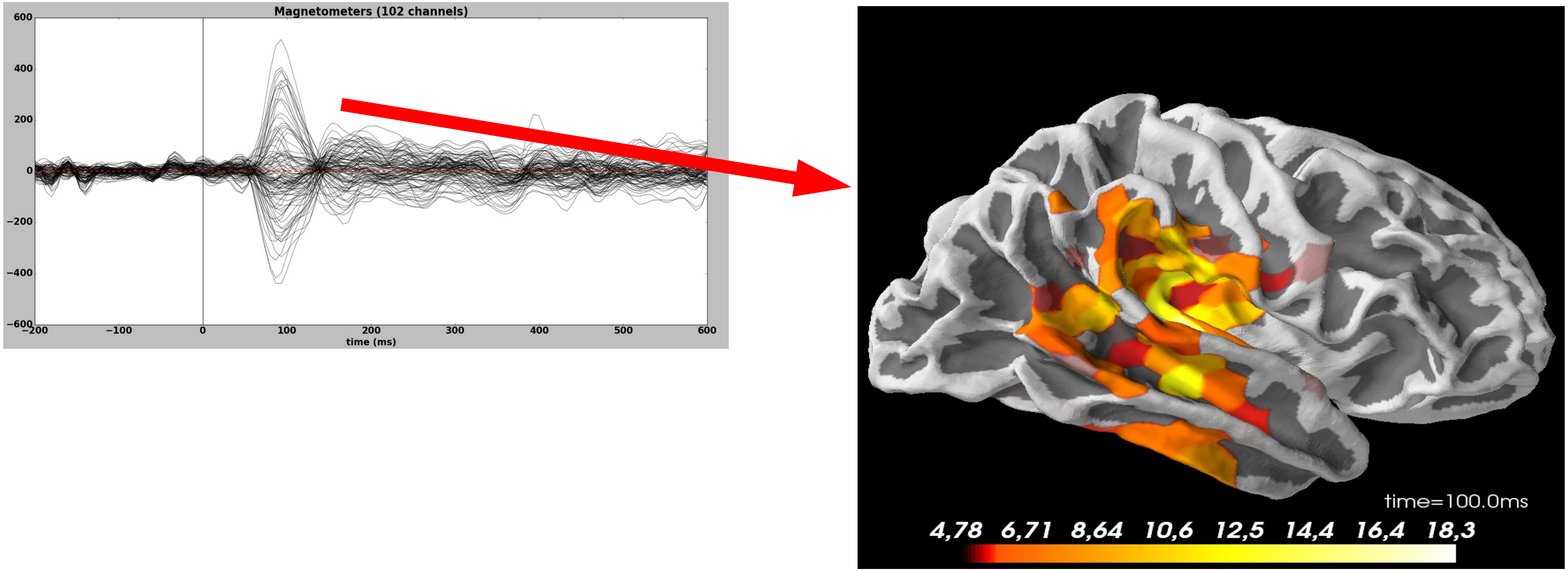
Right-hand rule



Field plot with very rough source localization



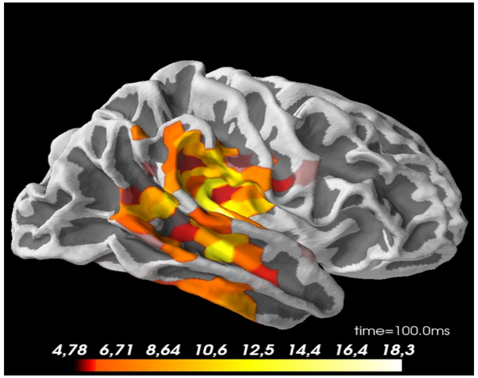
But... how do we get to here?



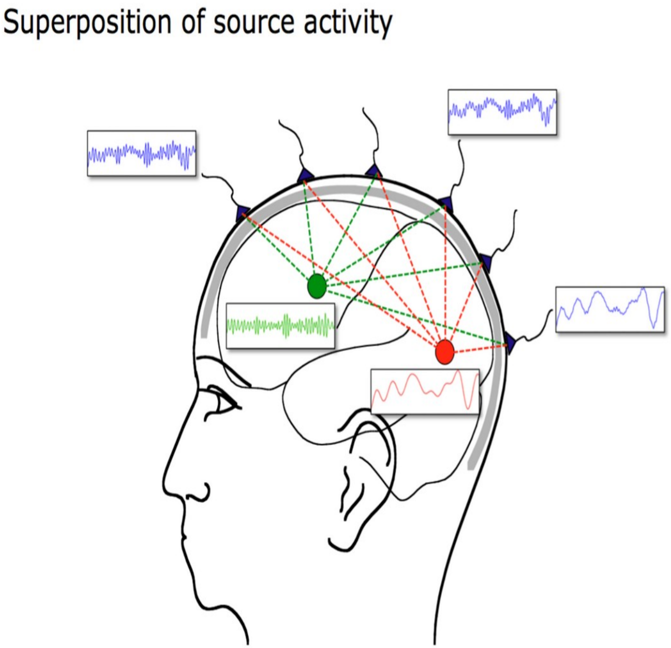
FORWARD MODELLING
(some slides from and inspired by FieldTrip)

We want to go here:

Brain activation



Problem



The problem in a nutshell

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X}$$

Y: the true signal (magnetic field)
X: the neural generators
W: a weighting matrix (the leadfield)

The problem in a nutshell

$$\boldsymbol{b}(t)=\boldsymbol{L}(r)\boldsymbol{s}(\boldsymbol{r},t)+\boldsymbol{n}(t)$$

b*(t)*: the measured magnetic field at time *t*
s*(****r****, t)*: the sources at position ***r*** at time *t*
L*(r):* leadfield of sources a position ***r*** (a weighting matrix)
n(*t*): normally distributed noise at each time *t*

Inverse modelling

$\hat{\boldsymbol{s}}(\boldsymbol{r}, t)$

physiological source
(what is the brain origin of
the potentials and fields)

$\boldsymbol{L}(\boldsymbol{r})$

volume conductor
(how potentials and fields
spread inside the head)

$\boldsymbol{b}(t)$

observed
electric potential or
magnetic field

inverse model
(source reconstruction)

Forward modelling

forward model

$$\boldsymbol{s}(\boldsymbol{r}, t) \xrightarrow{\boldsymbol{L}(\boldsymbol{r})} \hat{\boldsymbol{b}}(t)$$

physiological source (what is the brain origin of the potentials and fields)	volume conductor (how potentials and fields spread inside the head)	***expected*** electric potential or magnetic field

IMPORTANT

Without a plausible forward model restricting the solution space, an infinite number of inverse models could explain the data

Ingredients for a forward model, ***L***(***r***)

- A source model *src*
  - Telling us *the origin* of brain activity
- A volume conductor *bem*
  - Telling us how electric currents *spread* within the head *bem solution*
- Sensor positions *info*
  - Telling us *where* the sensors are relative to the sources
  - These need to be transformed into MR space

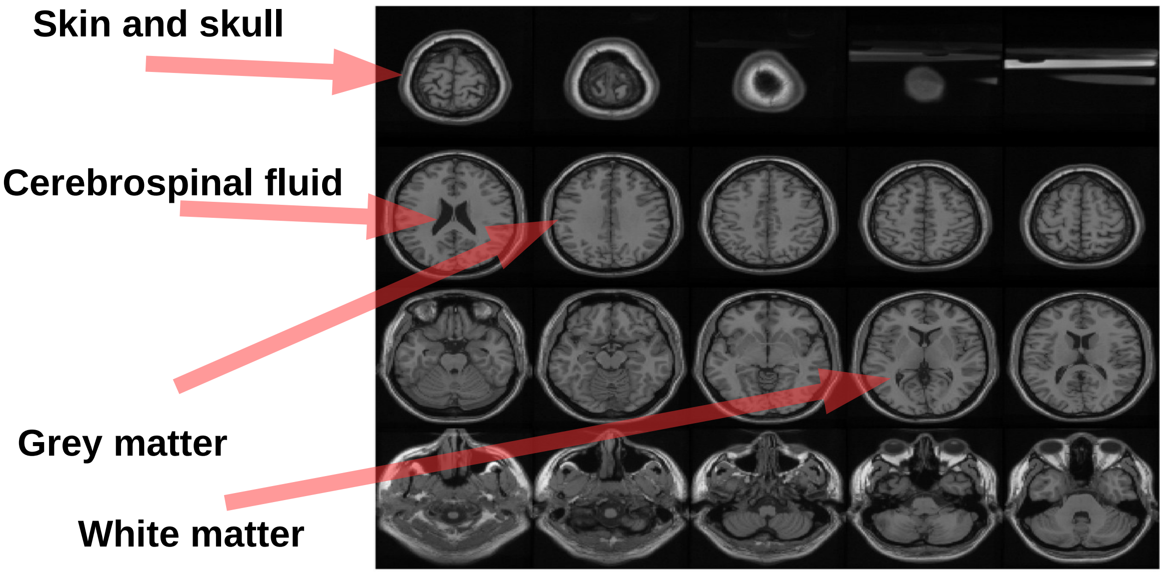
Info structure

```
In [29]: info = mne.io.read_info('workshop_2025_raw.fif')
```

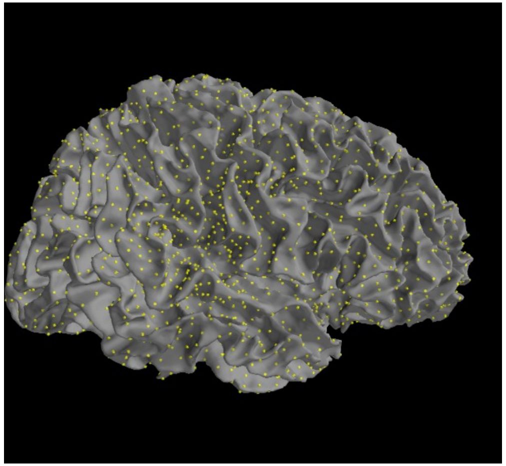
```
In [30]: info['dev_head_t']
Out[30]:
<Transform | MEG device->head>
[[ 0.99853134  0.04444526  0.03098306 -0.00409153]
 [-0.04400658  0.99892318 -0.01469979  0.00523906]
 [-0.03160303  0.01331474  0.99941182  0.05708658]
 [ 0.          0.          0.          1.        ]]
```

```
In [32]: info['chs'][3]
Out[32]:
{'scanno': 4,
 'logno': 111,
 'kind': 1 (FIFFV_MEG_CH),
 'range': 1.9073486328125e-05,
 'cal': 1.329999999022391e-10,
 'coil_type': 3022 (FIFFV_COIL_VV_MAG_T1),
 'loc': array([-0.1066    ,  0.0464    , -0.0604    , -0.0127    ,  0.0057    ,
        -0.99990302, -0.186801  , -0.98240298, -0.0033    , -0.98232698,
         0.18674099,  0.013541  ]),
 'unit': 112 (FIFF_UNIT_T),
 'unit_mul': 0 (FIFF_UNITM_NONE),
 'ch_name': 'MEG0111',
 'coord_frame': 1 (FIFFV_COORD_DEVICE)}
```

What do we see in the MR?

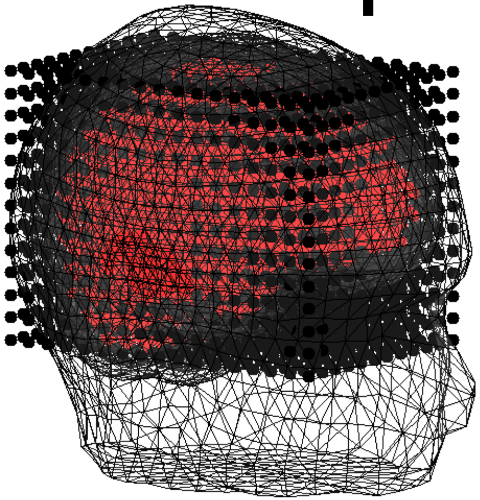


Source model examples



```
#%% CORTICAL SURFACE SOURCE SPACE

for subject in subjects:
    cort_src = mne.source_space.setup_source_space(subject,
                                                   subjects_dir=subjects_dir,
                                                   n_jobs=-1)
    bem_path = join(subjects_dir, subject, 'bem')
    write_filename = subject + '-oct-6-src.fif'
    mne.source_space.write_source_spaces(join(bem_path, write_filename),
                                         cort_src)
```

Restricted to the cortical surface



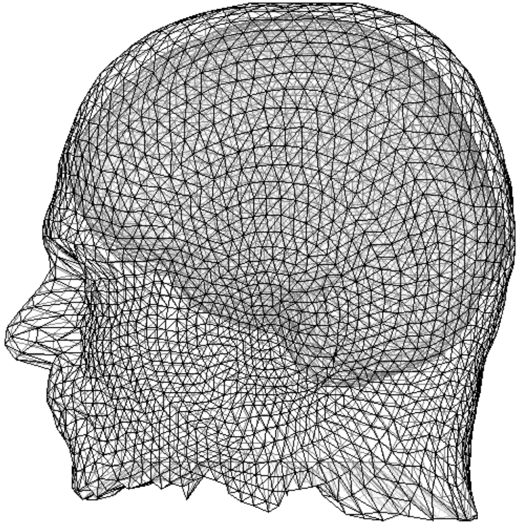
```
#%% VOLUME SOURCE SPACE

for subject in subjects:

    bem_path = join(subjects_dir, subject, 'bem')


    vol_src = mne.source_space.setup_volume_source_space(subject=subject,
                                                         bem=bem_solution,
                                                         subjects_dir=subjects_dir)
    write_filename = subject + '-volume-5-mm-src.fif'

    mne.source_space.write_source_spaces(join(bem_path, write_filename),
                                         vol_src)
```

Volumetric grid

https://github.com/ualsbombe/2025_advanced_cognitive_neuroscience/blob/main/experiment/mr_processing/python/bem_processing.py

Volume conductor (head model)

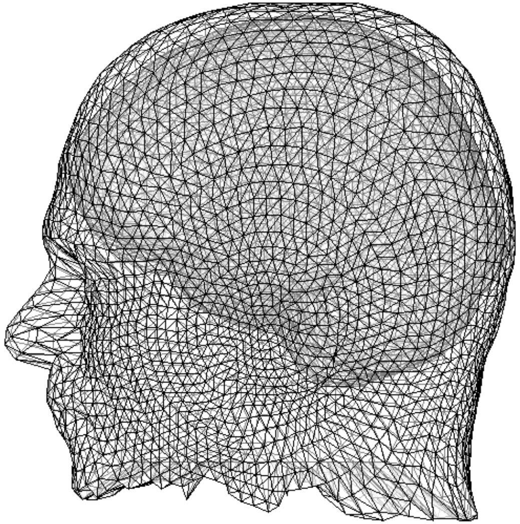
An anatomical model that models the conductivities of different tissues *bem*



Volume conductor (head model)

For *EEG*, we need to model the brain, skull and scalp with different conductivities (we are measuring volume currents)

```
## BOUNDARY ELEMENT METHOD

## describe the surfaces and their conductivities
#bem_model = mne.bem.make_bem_model(subject=subject, subjects_dir=subjects_dir,
#                            conductivity=[0.3, 0.006, 0.3]) ## three layer model

## model how electrical potentials spread to the electrodes and how the
# currents of the brain are related to the magnetic field measured at the
# sensors
#bem_solution = mne.bem.make_bem_solution(bem_model)

bem_solution = mne.bem.read_bem_solution('../../subjects/sample/bem/sample-5120-5120-5120-bem-sol.fif')

print(bem_solution['solution'])
```

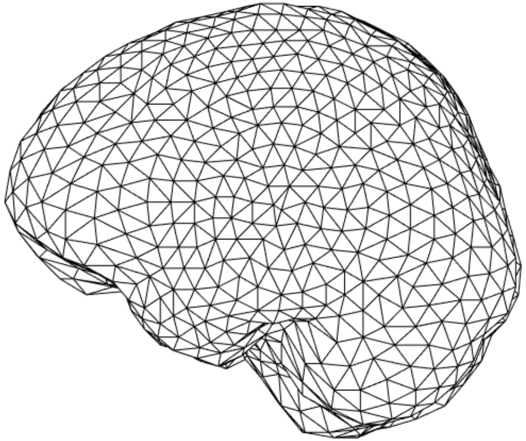


Volume conductor (head model)

For MEG, the model can be much simpler since the magnetic field is roughly proportional to the magnitude of the primary current

(This is why MEG has better spatial resolution than EEG)

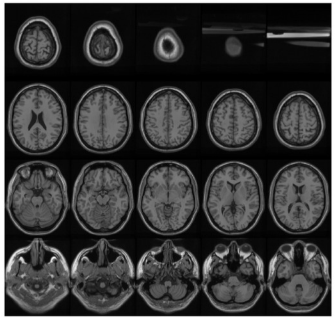
```
#%% BEM MODEL

for subject in subjects:
    bem_path = join(subjects_dir, subject, 'bem')

    ## single-layer model
    write_filename = subject + '-5120-bem.fif'
    bem_surfaces = mne.bem.make_bem_model(subject, conductivity=[0.3],
                                          subjects_dir=subjects_dir)
    mne.bem.write_bem_surfaces(join(bem_path, write_filename),
                               bem_surfaces)
```



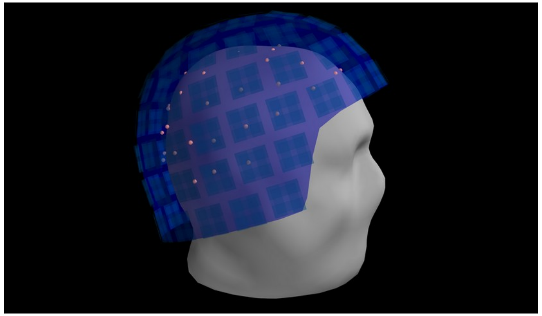
https://github.com/ualsbombe/2025_advanced_cognitive_neuroscience/blob/main/experiment/mr_processing/python/bem_processing.py

Sensor positions

- The MR-images and the helmet/electrode data will be represented in different coordinate systems
  - Thus, they have to be co-registered



$\neq$

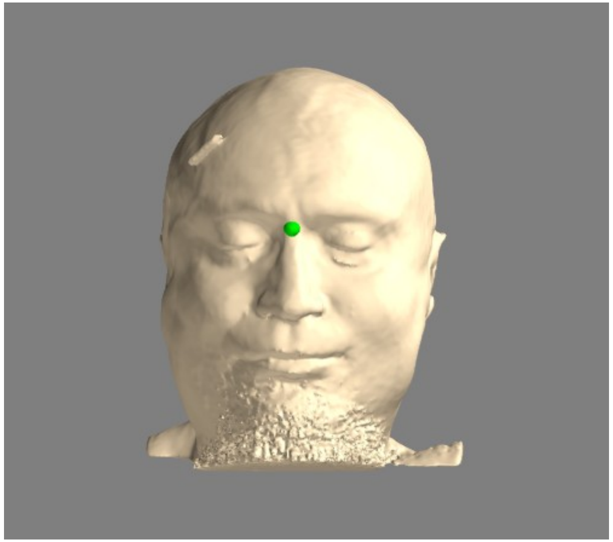


Co-registration

Construct a head model based on the MRI

Co-registration

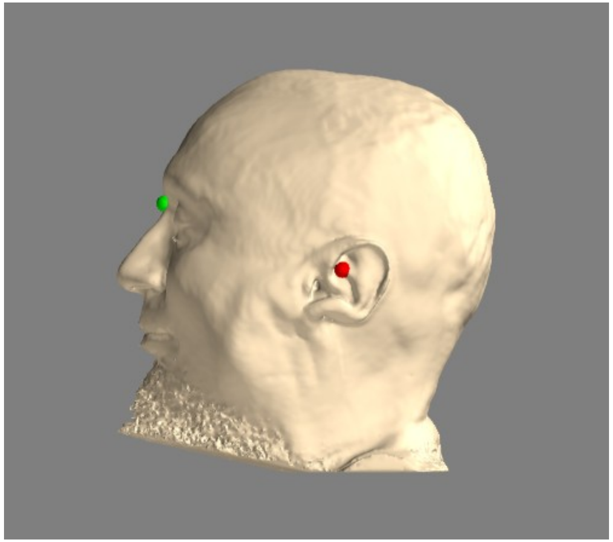
1. Nasion



Plot in fiducial points (these are acquired before recording)

Co-registration

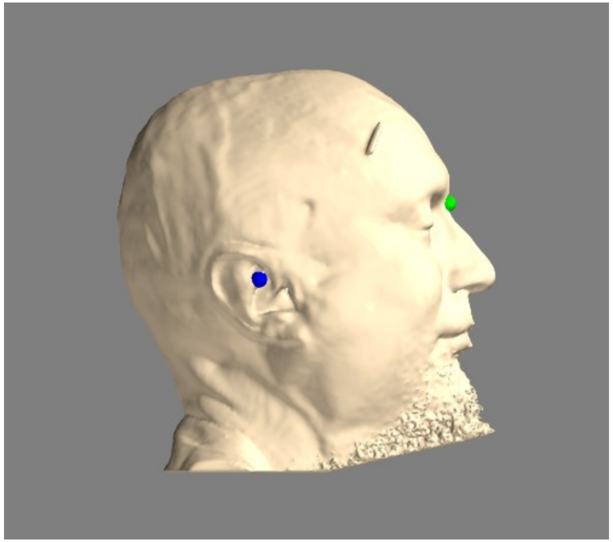
1. Nasion
2. Left pre-auricular point



Plot in fiducial points (these are acquired before recording)

Co-registration

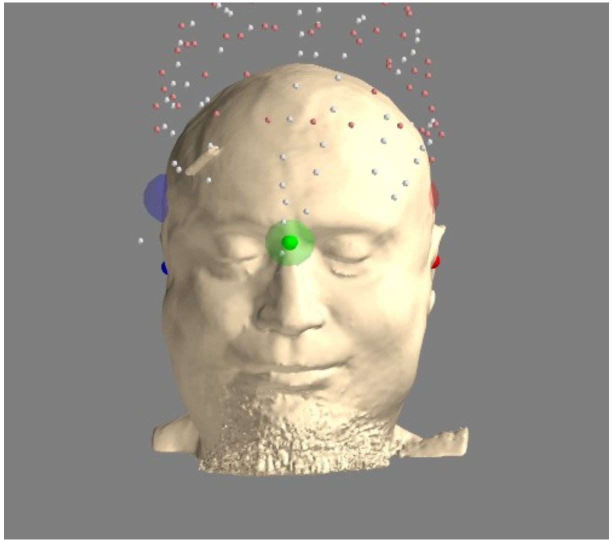
1. Nasion
2. Left pre-auricular point
3. Right pre-auricular point



Plot in fiducial points (these are acquired before recording)

Co-registration

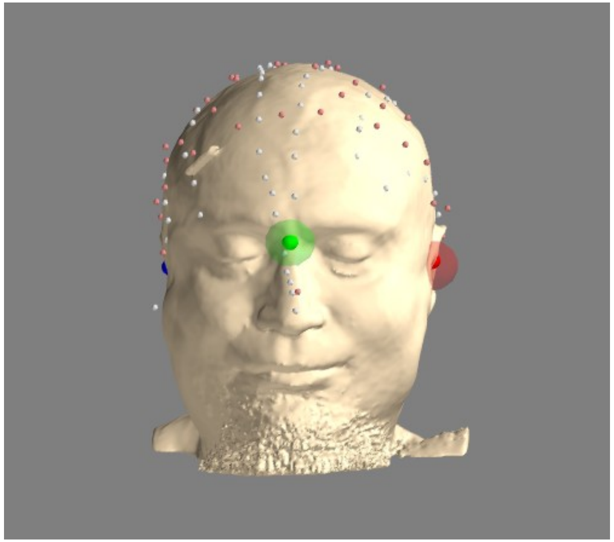
1. Nasion
2. Left pre-auricular point
3. Right pre-auricular point



Plot extra head points

Co-registration

1. Nasion
2. Left pre-auricular point
3. Right pre-auricular point

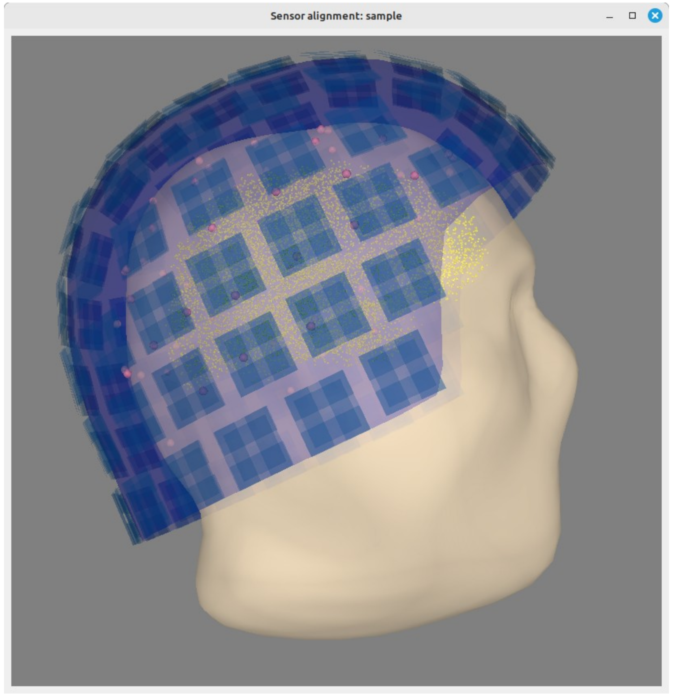


```
In [7]: trans = mne.read_trans('workshop_2025-trans.fif')

In [8]: trans
Out[8]:
<Transform | head->MRI (surface RAS)>
[[ 0.99968028 -0.02514233 -0.00269039  0.00789862]
 [ 0.02285611  0.94399983 -0.3291533  -0.00342108]
 [ 0.01081541  0.32898656  0.9442727  -0.03508078]
 [ 0.          0.          0.          1.        ]]
```

```
In [32]: info['chs'][3]
Out[32]:
{'scanno': 4,
 'logno': 111,
 'kind': 1 (FIFFV_MEG_CH),
 'range': 1.9073486328125e-05,
 'cal': 1.329999999022391e-10,
 'coil_type': 3022 (FIFFV_COIL_VV_MAG_T1),
 'loc': array([-0.1066    ,  0.0464    , -0.0604    , -0.0127    ,  0.0057    ,
       -0.99990302, -0.186801  , -0.98240298, -0.0033    , -0.98232698,
        0.18674099,  0.013541  ]),
 'unit': 112 (FIFF_UNIT_T),
 'unit_mul': 0 (FIFF_UNITM_NONE),
 'ch_name': 'MEG0111',
 'coord_frame': 1 (FIFFV_COORD_DEVICE)}
```

Use an algorithm to minimize the distance between points and head shape. The MR and the MEG are now co-registered

Check alignment




```
# extract source space
src = fwd['src']
##plot alignment does not work on UCloud
mne.viz.plot_alignment(info=epochs_sample.info, src=src,
                       subjects_dir='../../subjects/',
                       subject="sample", surfaces="outer_skin",
                       trans='sample_audvis_raw-trans.fif')
```

With all ingredients in place

WE CAN GET THE FORWARD MODEL

- Volume conduction model
  - *bem*
- Source model
  - *src*
- Transformed sensor positions
  - *info*
  - *trans*

```
## GET FORWARD MODEL

info = evoked.info
trans = join(MEG_path, ?subject? + '-trans.fif')
src = join(subjects_dir, ?subject?, 'bem', ?subject? + '-oct-6-src.fif')
bem = join(subjects_dir, ?subject?, 'bem', ?subject? + '-5120-bem-sol.fif')

fwd = mne.make_forward_solution(
                            info, trans,
                            src, bem)
```

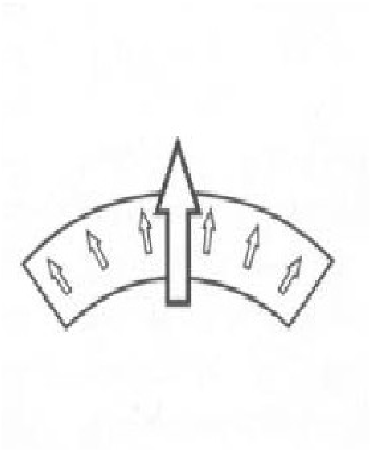
forward model

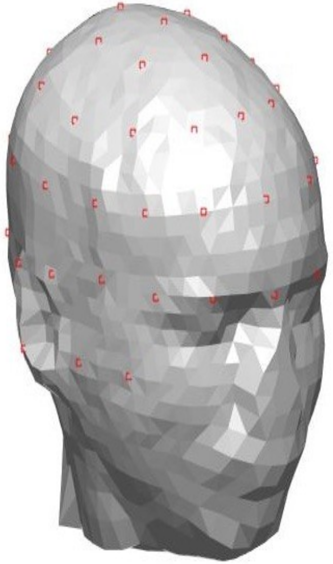
$\mathbf{s}(\mathbf{r}, t)$ | $\mathbf{L}(\mathbf{r})$ | $\hat{\mathbf{b}}(t)$

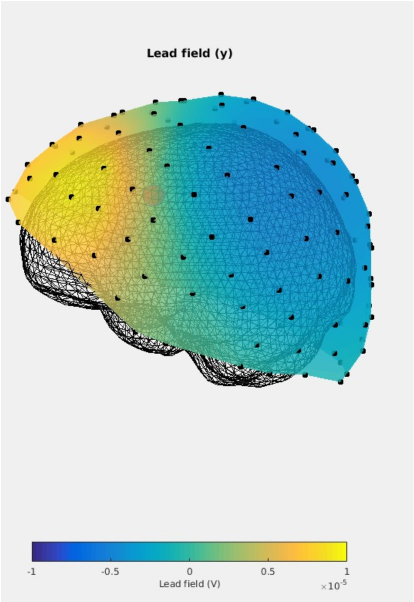
physiological source
(source model)

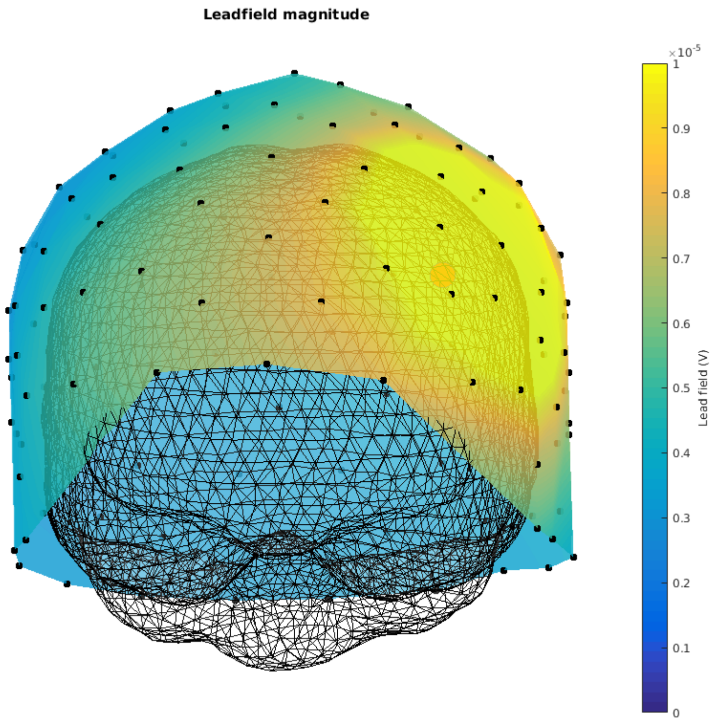
volume conductor
(head model)

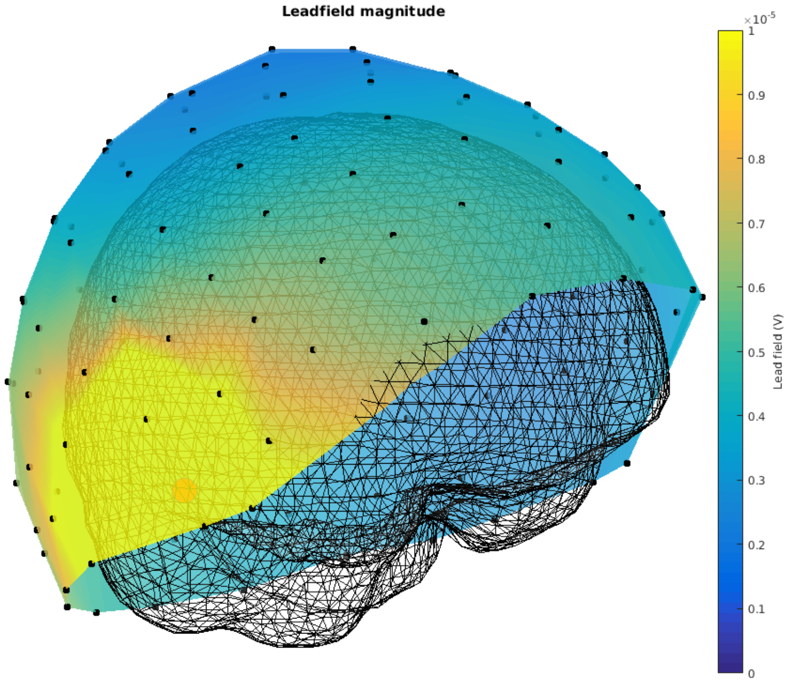
expected
potential or field
(lead field)









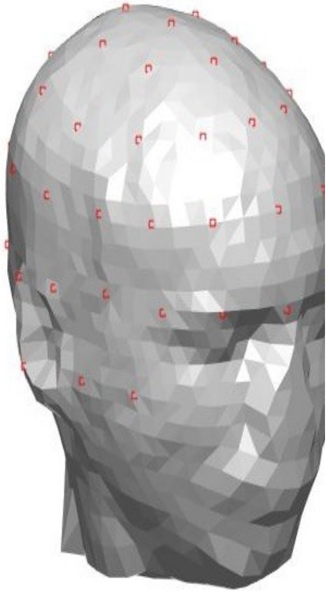


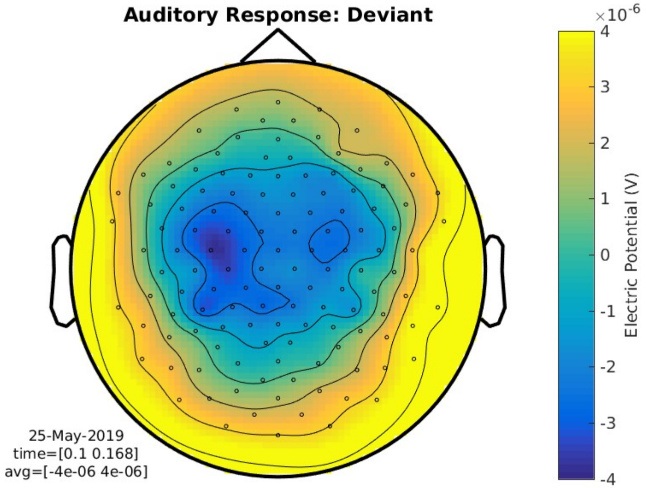
and so on for each source in the source model ...

inverse model

$\hat{\boldsymbol{s}}(\boldsymbol{r}, t) \longleftarrow \boldsymbol{L}(\boldsymbol{r}) \longleftarrow \boldsymbol{b}(t)$

physiological source (source model)	volume conductor (head model)	**observed** potential or field **(actual data)**





Now we can create the forward model (leadfield)

The forward model models how each source in the source space *would* be seen by the sensors (in *T*) in the helmet, *given* that that source is active with 1 *Am*

What’s the physical unit for the leadfield?

Now we can create the forward model (leadfield)

The forward model models how each source in the source space *would* be seen by the sensors (in T) in the helmet, *given* that that source is active with 1 Am

What’s the physical unit for the leadfield? $$\frac{T}{Am}$$

Units

$$\boldsymbol{b}(t)=\boldsymbol{L}(r)\boldsymbol{s}(\boldsymbol{r},\,t)+\boldsymbol{n}(t)$$

$$[T] \;=\; \left[\frac{T}{Am}\right]\;[Am]+\;[T]$$

Contents of the *fwd*

WHAT IS THE STORY I WANT TO TELL HERE?

```
In [44]: fwd['sol']['data'].shape
Out[44]: (366, 22494)
```

```
In [50]: fwd['src']
Out[50]: <SourceSpaces: [<surface (lh), n_vertices=155407, n_used=3732>,
<surface (rh), n_vertices=156866, n_used=3766>] head coords, subject
'sample', ~31.0 MiB>
```

```
In [49]: len(evoked_sample.ch_names)
Out[49]: 366
```

```
In [51]: len(fwd['src'])
Out[51]: 2
```

```
In [56]: fwd['sol']['data'].shape[1] // 3
Out[56]: 7498
```

```
In [52]: fwd['src'][0]['nuse'] + fwd['src'][1]['nuse']
Out[52]: 7498
```

```
In [58]: evoked_sample.ch_names[2]
Out[58]: 'MEG 0111'
```

```
In [57]: fwd['sol']['data'][2, :]
Out[57]:
array([-7.1391241e-07,  9.5892676e-07,  7.8583281e-07, ...,
        9.4759559e-07, -2.4599572e-07, -6.7779990e-07],
      shape=(22494,), dtype=float32)
```

What order will the magnetic field (in *T*) roughly be, when we have sources with current densities of 10-100 *nAm?*

```
In [57]: fwd['sol']['data'][2, :]
Out[57]:
array([-7.1391241e-07,  9.5892676e-07,  7.8583281e-07, ...,
        9.4759559e-07, -2.4599572e-07, -6.7779990e-07],
      shape=(22494,), dtype=float32)
```

Interim summary

- The forward model consists of:
  - a source space (*src*)
  - a solution of the spread of the currents (*bem*)
  - transformed (*trans*) positions of the sensor positions (*info*)
- The forward model links the source space (*src)* to the transformed sensor space (*trans**info*) through the model (*bem*)
  - Its unit is: $[\frac{T}{Am}]$

Learning goals

- For the dipole modelling
  - understanding when it is appropriate to use dipole fitting, minimum norm estimate and beamformer strategies respectively
  - understanding how the forward model is a crucial ingredient in all source modelling strategies
  - get a first understanding of the relevant equations

Inverse model types

- Dipole fitting
  - Only a single or a few dipolar sources are active at given time sample
  - The solution is *overdetermined:* there are more data channels (hundreds) than sources (a few)

$$[\boldsymbol{b}(t) - \boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})^{+}\boldsymbol{b}(t)]^2$$

- Minimum Norm Estimate
  - All (cortical) sources are active all the time
  - The solution is *underdetermined*: there are more sources (thousands) than channels (hundreds)

$$\hat{\boldsymbol{v}}_{\boldsymbol{vox}}(t) = \boldsymbol{L}_{\boldsymbol{V}}^{\boldsymbol{T}}(\boldsymbol{G} + \epsilon\boldsymbol{I})^{(-1)}\boldsymbol{b}(t)$$

- Beamformer
  - All sources are estimated independently from one another

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})}{\boldsymbol{L}^{\boldsymbol{T}}(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})}$$

Inverse model types

- Dipole fitting
  - Only a single or a few dipolar sources are active at given time sample
  - The solution is *overdetermined:* there are more data channels (hundreds) than sources (a few)

$$[\boldsymbol{b}(t) - \underline{\boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})^{+}}\boldsymbol{b}(t)]^{2}$$

- Minimum Norm Estimate
  - All (cortical) sources are active all the time
  - The solution is *underdetermined*: there are more sources (thousands) than channels (hundreds)

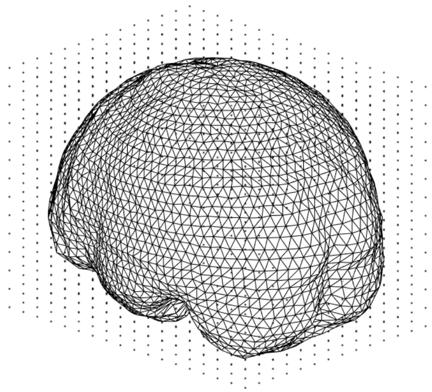
$$\hat{\boldsymbol{v}}_{\boldsymbol{vox}}(t) = \underline{\boldsymbol{L}_{\boldsymbol{V}}^{\boldsymbol{T}}(}\boldsymbol{G} + \epsilon\boldsymbol{I})^{(-1)}\boldsymbol{b}(t)$$

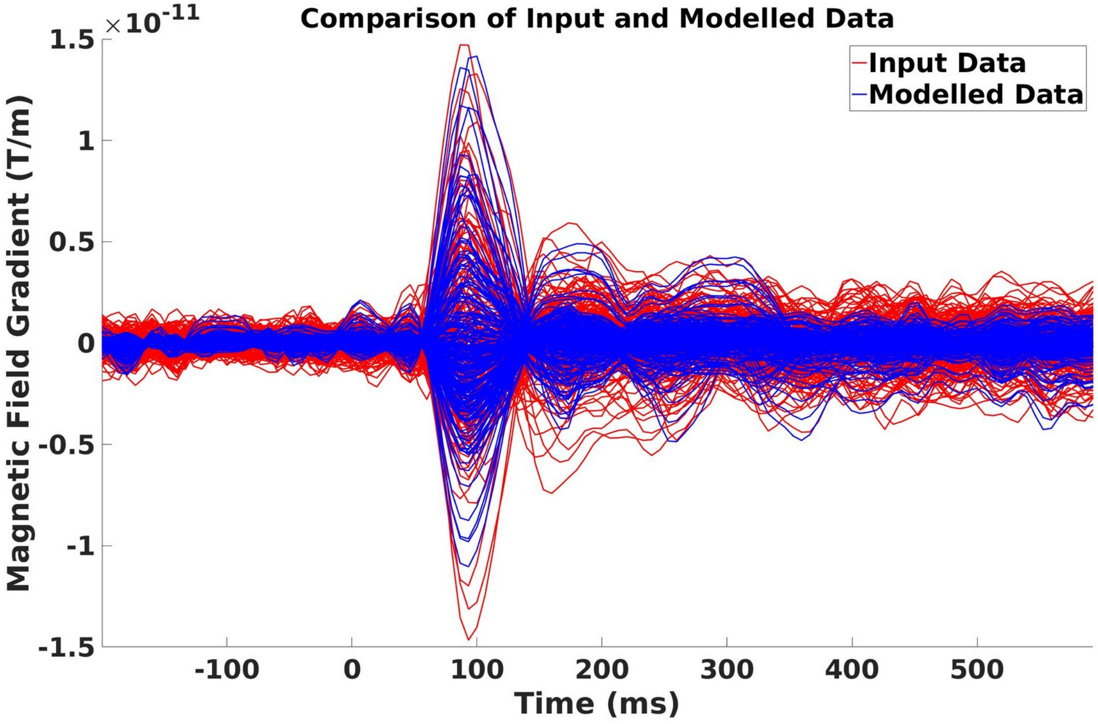
- Beamformer
  - All sources are estimated independently from one another

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{R}^{-1}\underline{\boldsymbol{L}}(\boldsymbol{r})}{\underline{\boldsymbol{L}}^{\boldsymbol{T}}(\boldsymbol{r})\boldsymbol{R}^{-1}\underline{\boldsymbol{L}}(\boldsymbol{r})}$$

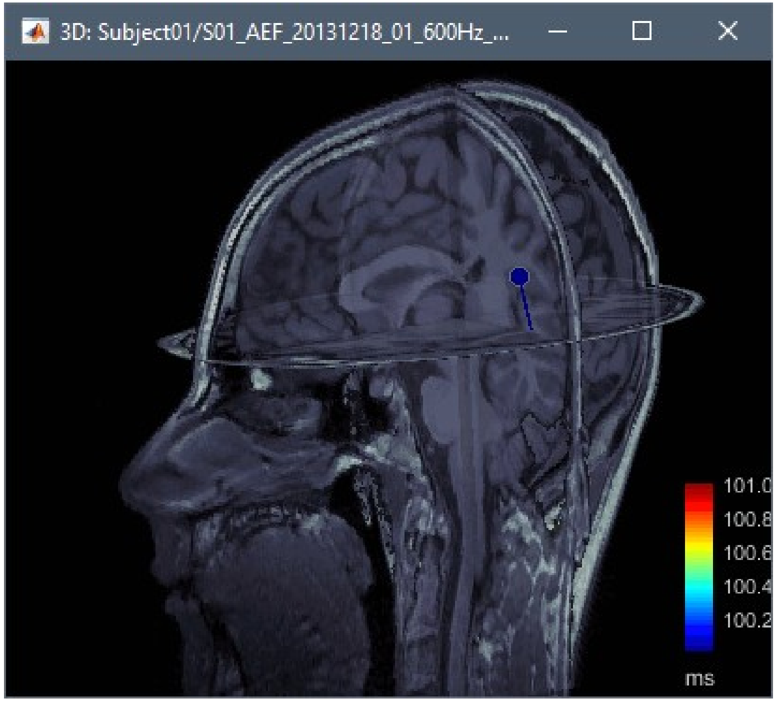
Dipole fits

- Two-step procedure for minimizing the error, where the error is the mismatch between MEG data and *inverse* model
  1. Scan all the sources within your source space to find the optimal starting position
  2. From the optimal starting position, do gradient descent until the error cannot be minimized further.

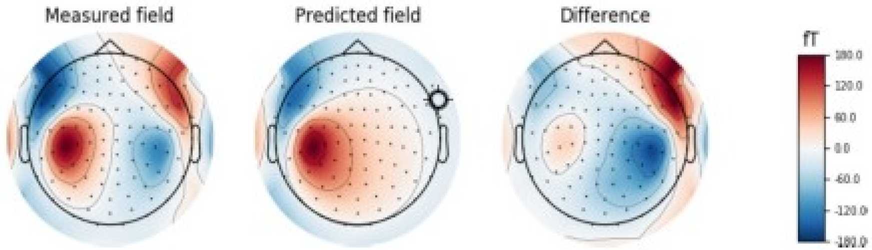




Comparison of Input and Modelled Data
×10^-11
1.5
1
0.5
0
-0.5
-1
-1.5
Input Data
Modelled Data
Magnetic Field Gradient (T/m)
-100
0
100
200
300
400
500
Time (ms)



3D: Subject01/S01_AEF_20131218_01_600Hz_...
101.0
100.8
100.6
100.4
100.2
ms



Measured field
Predicted field
Difference
fT

Dipole fitting

1. Scan all the sources within your source space to find the optimal starting position

Minimize the following expression, i.e. the discrepancy between model and observed data (“+” is the pseudoinverse)

$$\left[\boldsymbol{b}(t) - \boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})^{+}\boldsymbol{b}(t)\right]^2$$

***L*(*r*)** = the lead field for any given source *(r)*
***b*(*t*)** = the observed magnetic field or the electric potential at a given time point *(t)*
***L*(*r*)**$^{+}$ = the pseudoinverse of ***L*(*r*)**

Derivation

$\boldsymbol{b}(t)$: measured data

$\hat{\boldsymbol{s}}(\boldsymbol{r}, t)$: fitted dipole

$\boldsymbol{L}(\boldsymbol{r})$: forward solution

$\hat{\boldsymbol{b}}(t) = \boldsymbol{L}(r)\boldsymbol{s}(\boldsymbol{r}, t)$: according to forward model

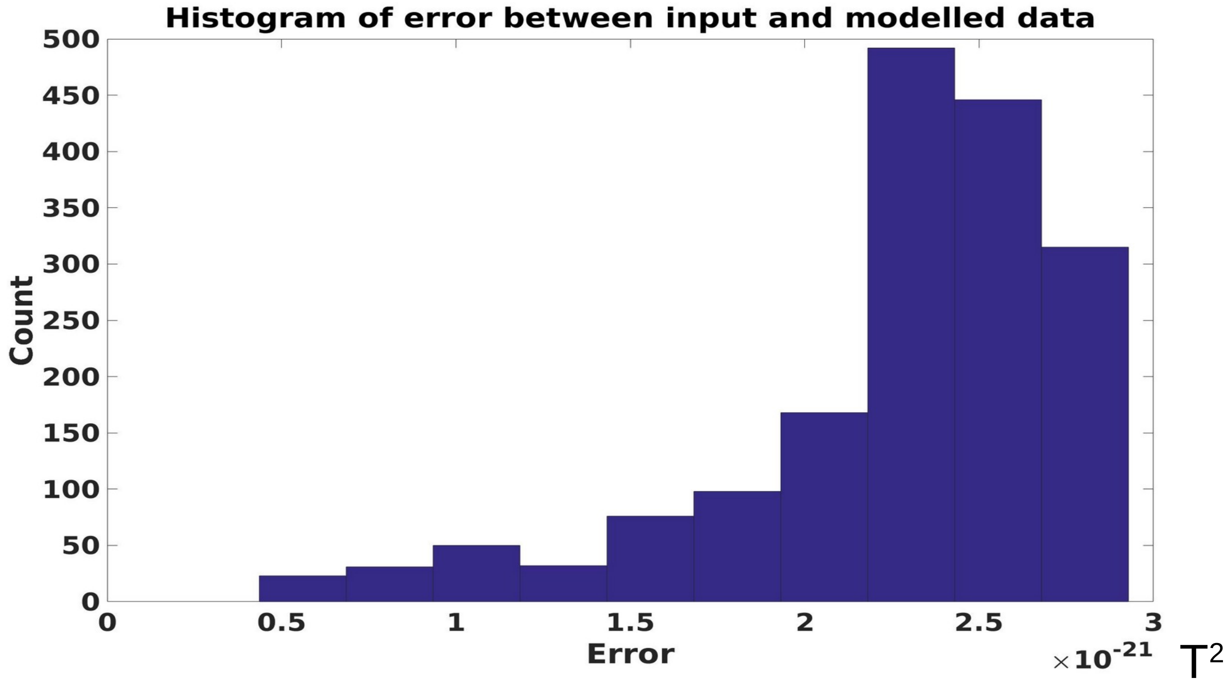
$[\boldsymbol{b}(t) - \hat{\boldsymbol{b}}(t)]^2$: to be minimised

$$\hat{\boldsymbol{s}}(\boldsymbol{r}, t) = \boldsymbol{L}(\boldsymbol{r})^+\boldsymbol{b}(t)$$

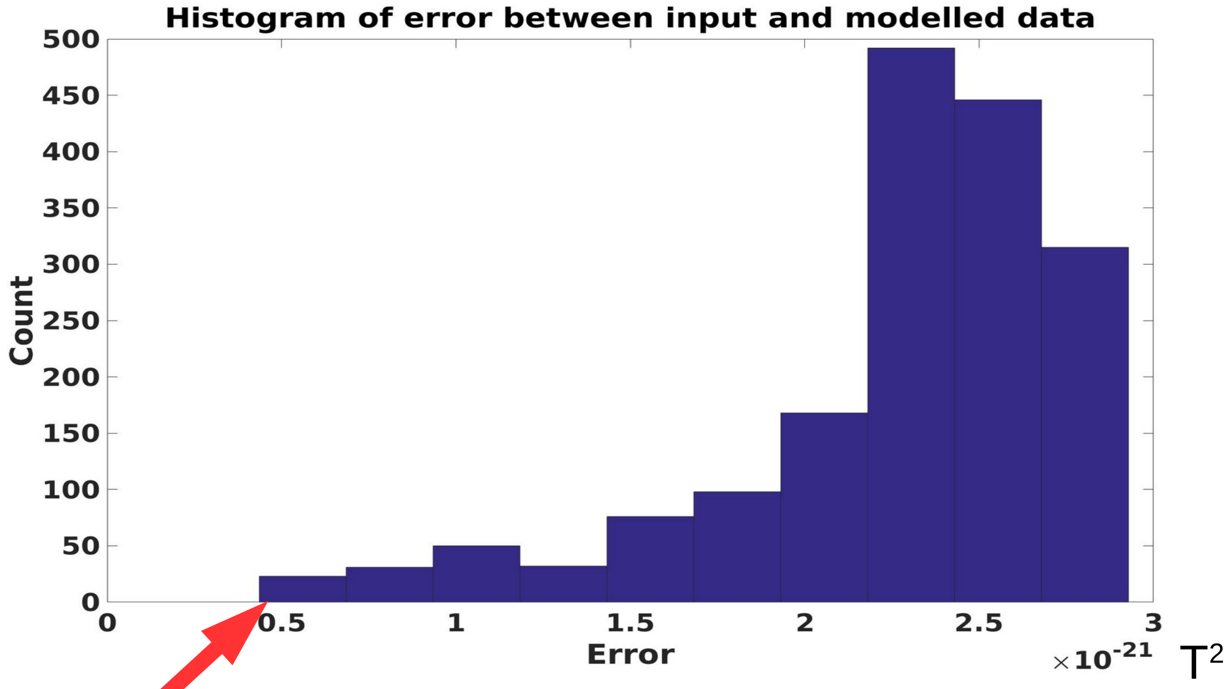
$\hat{\boldsymbol{b}}(t) = \boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})^+\boldsymbol{b}(t)$: using $\hat{\boldsymbol{s}}(\boldsymbol{r}, t)$ for $\boldsymbol{s}(\boldsymbol{r}, t)$

$[\boldsymbol{b}(t) - \boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})^+\boldsymbol{b}(t)]^2$: cost function

1. Scan all the sources within your source space



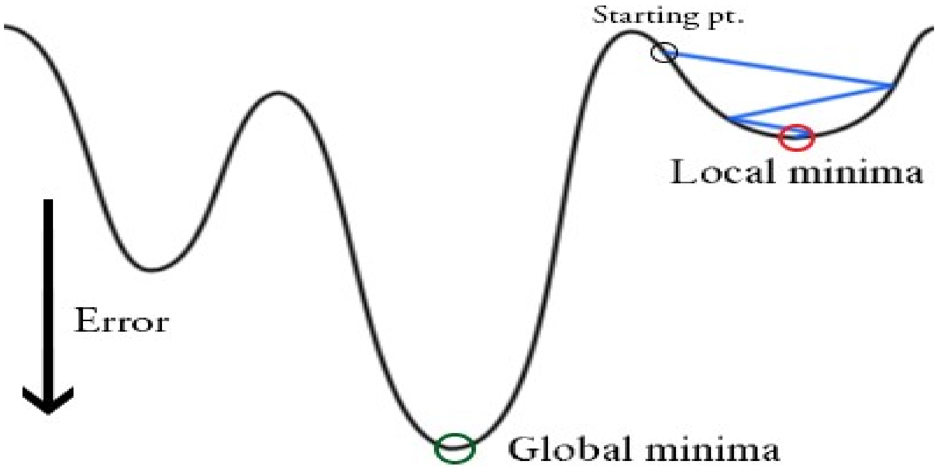
The source position corresponding to the smallest error is used as the starting position



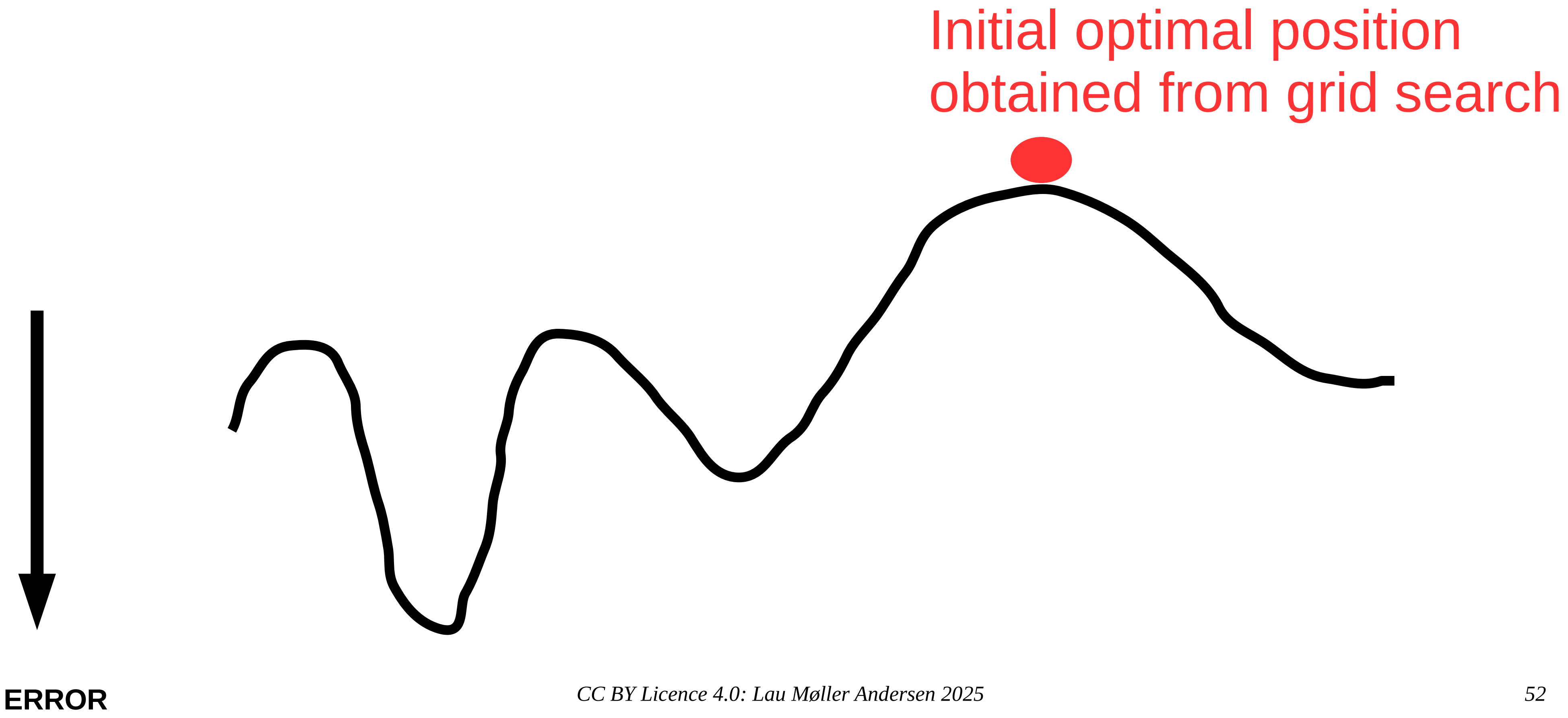
Start from corresponding position

Gradient descent

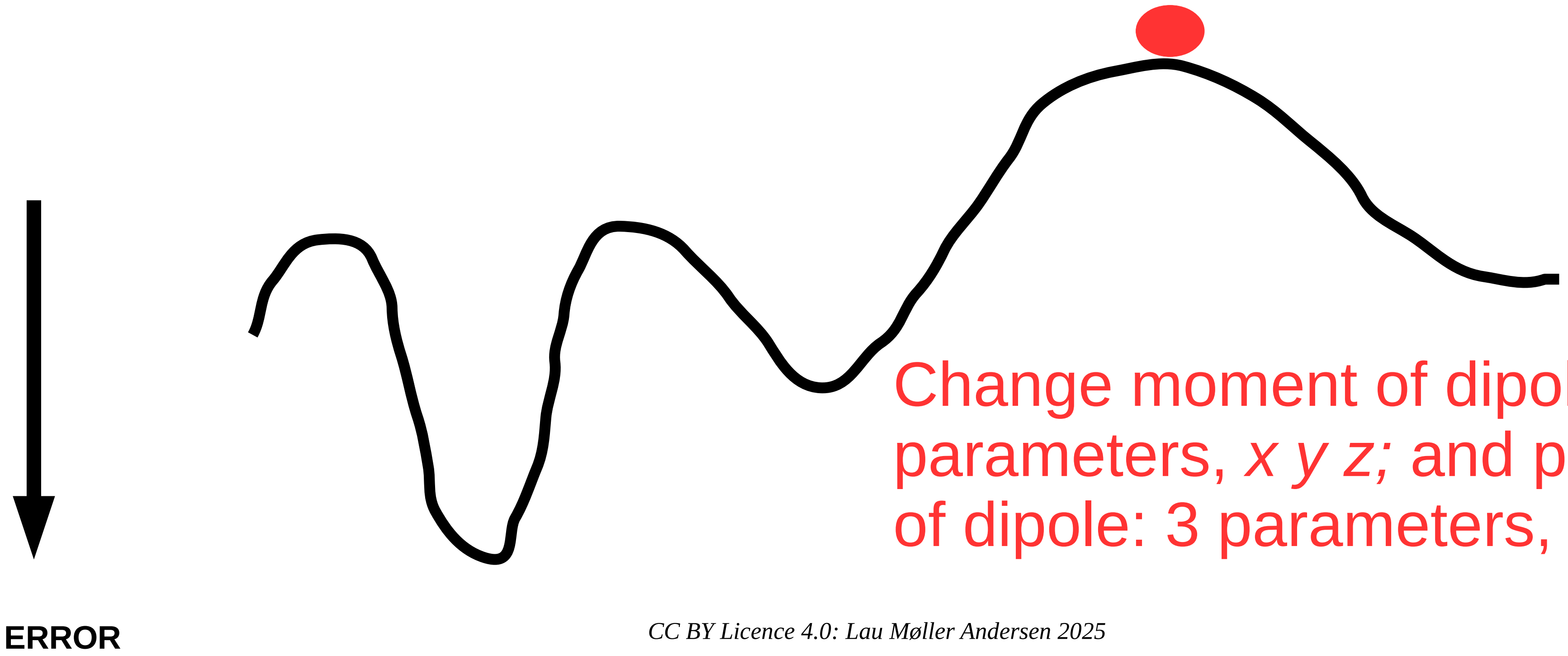
- From the optimal position, do gradient descent until the error cannot be minimized further.
  - The dangers of local minima exist



2. Gradient descent

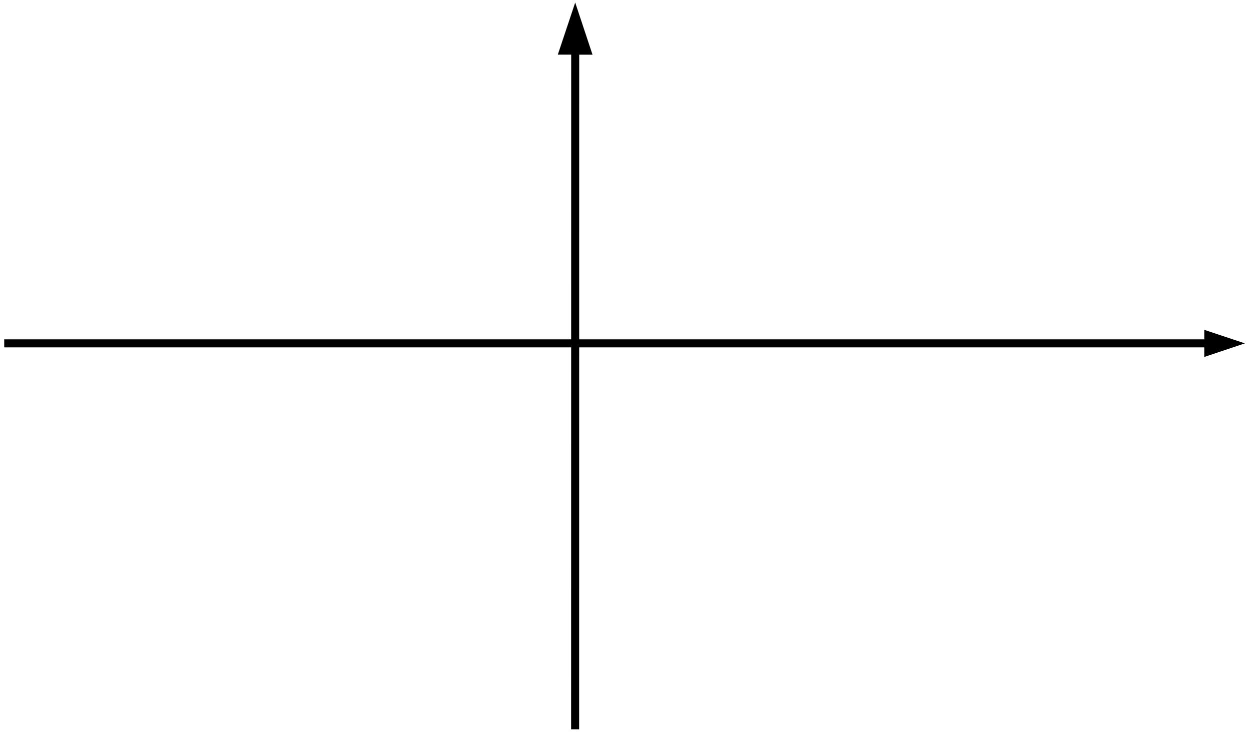


2. Gradient descent



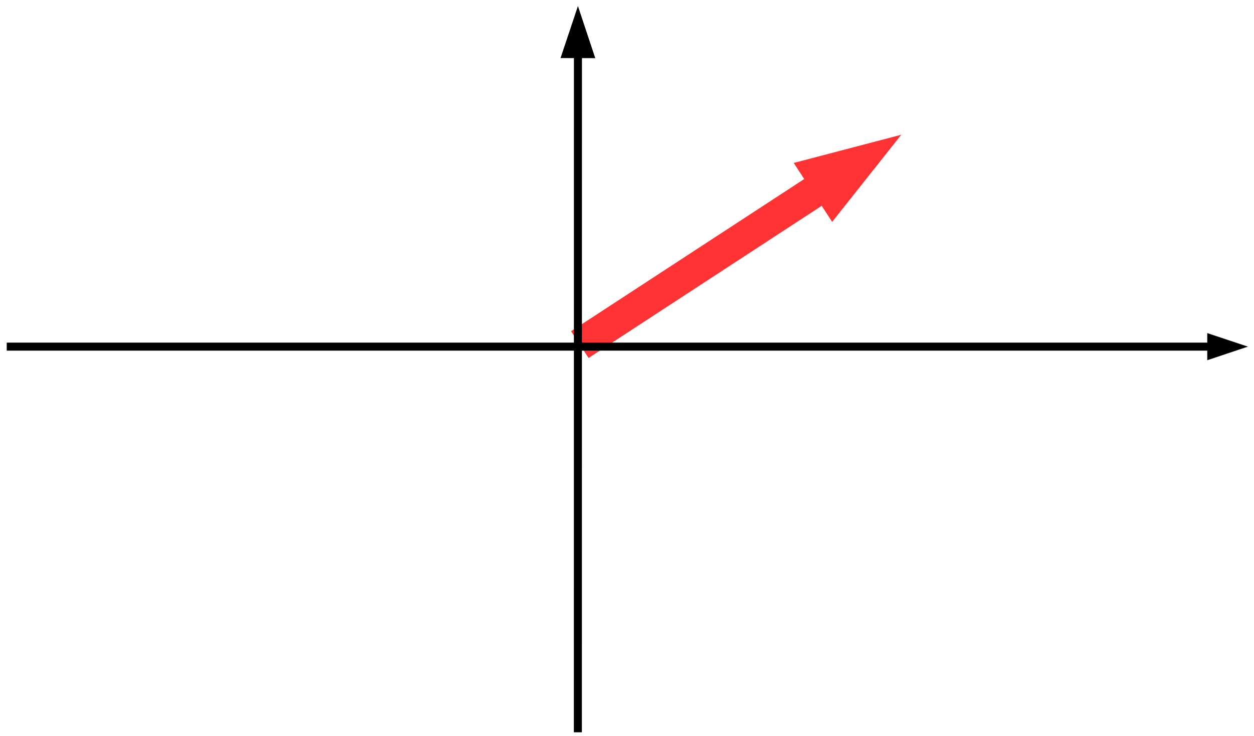
Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

Moment – vector space (example in 2d)



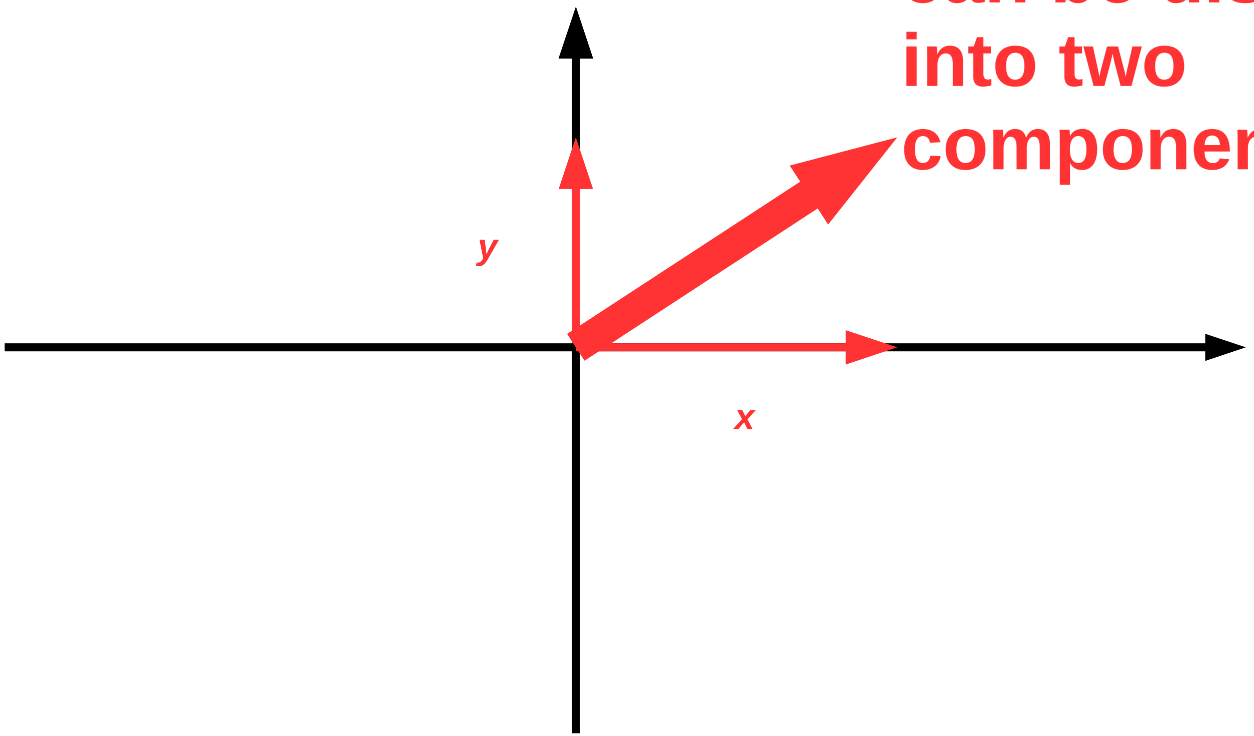
Moment – vector space (example in 2d)

DIPOLE MOMENT



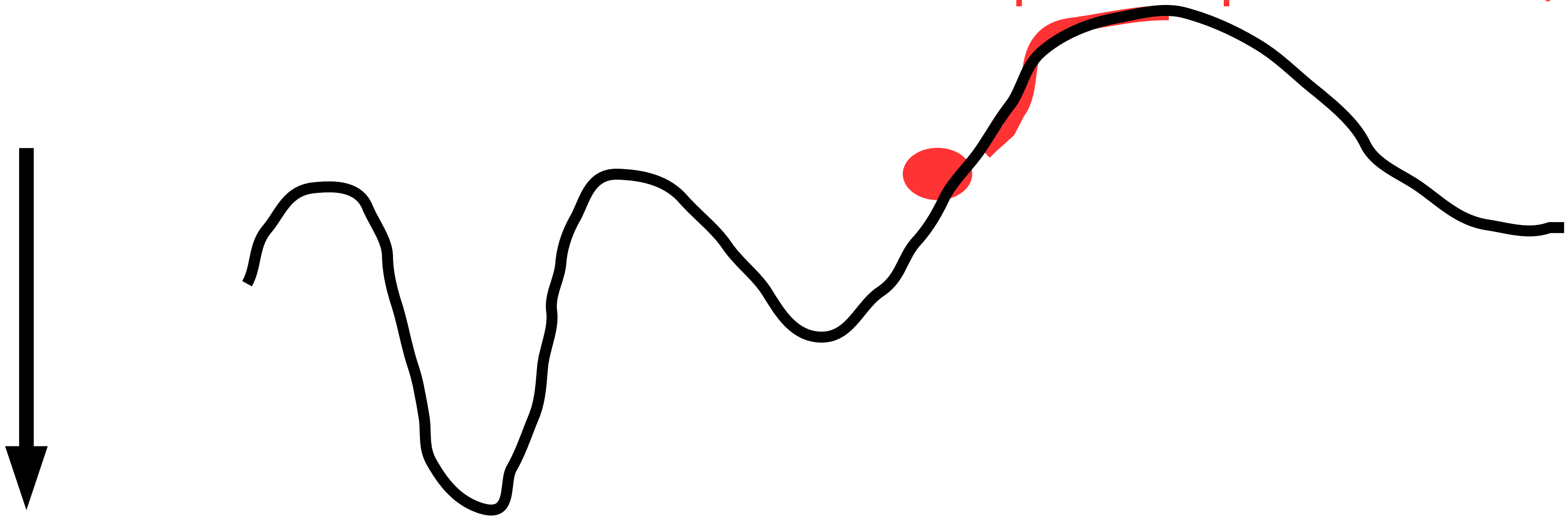
Moment – vector space (example in 2d)

DIPOLE MOMENT can be dissolved into two components



2. Gradient descent

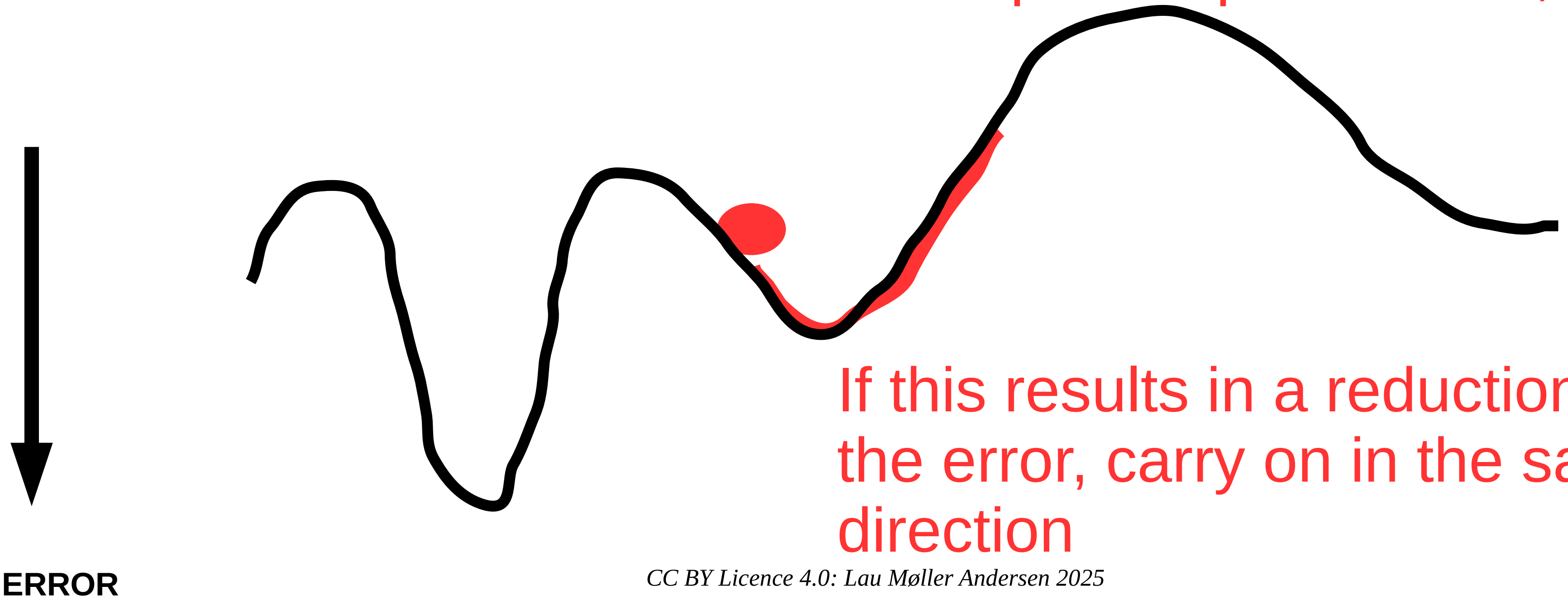
Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.



ERROR

2. Gradient descent

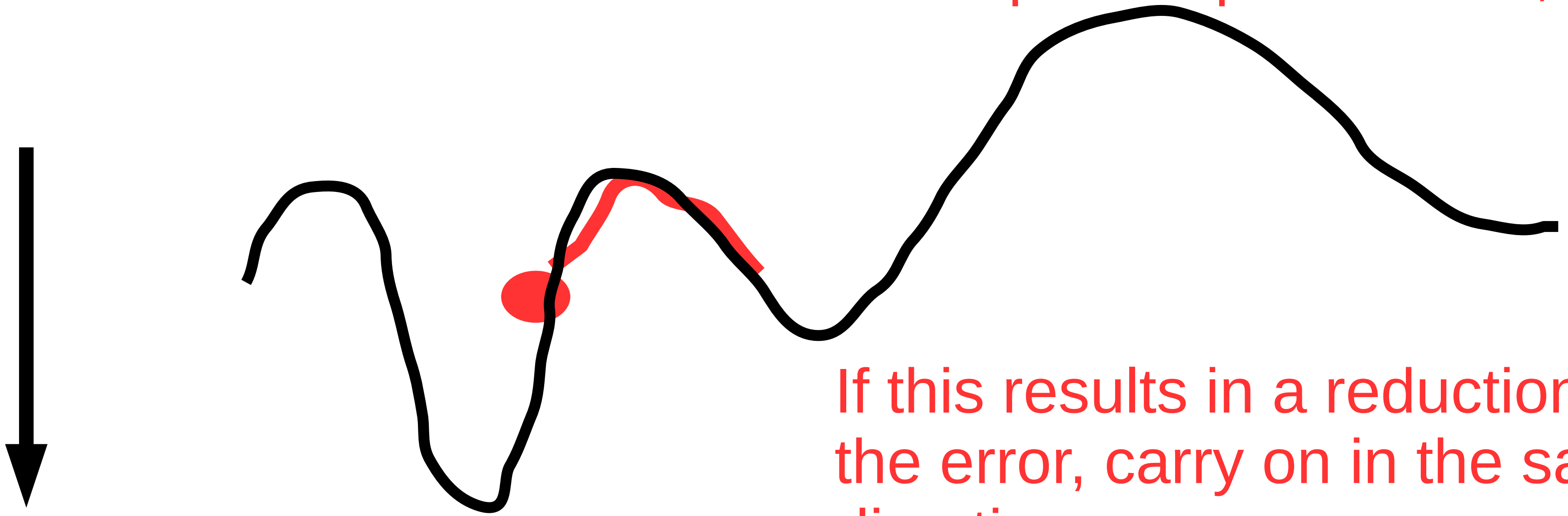
Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.



If this results in a reduction of the error, carry on in the same direction

2. Gradient descent

Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

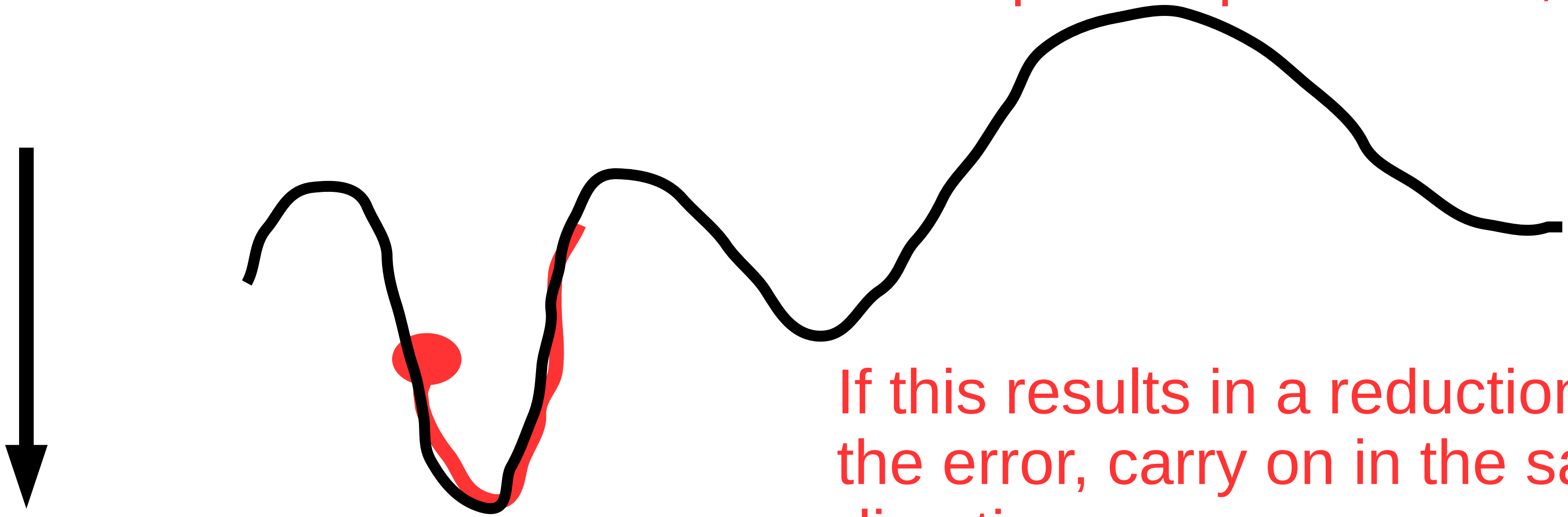


If this results in a reduction of the error, carry on in the same direction

ERROR

2. Gradient descent

Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

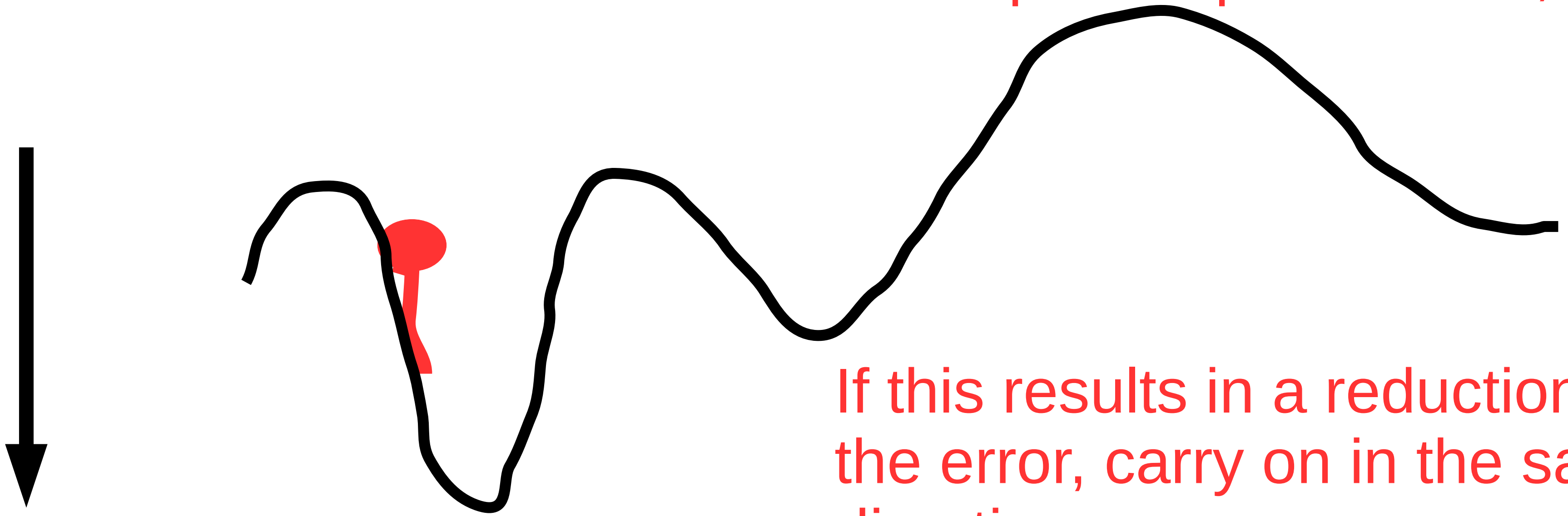


If this results in a reduction of the error, carry on in the same direction

ERROR

2. Gradient descent

Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

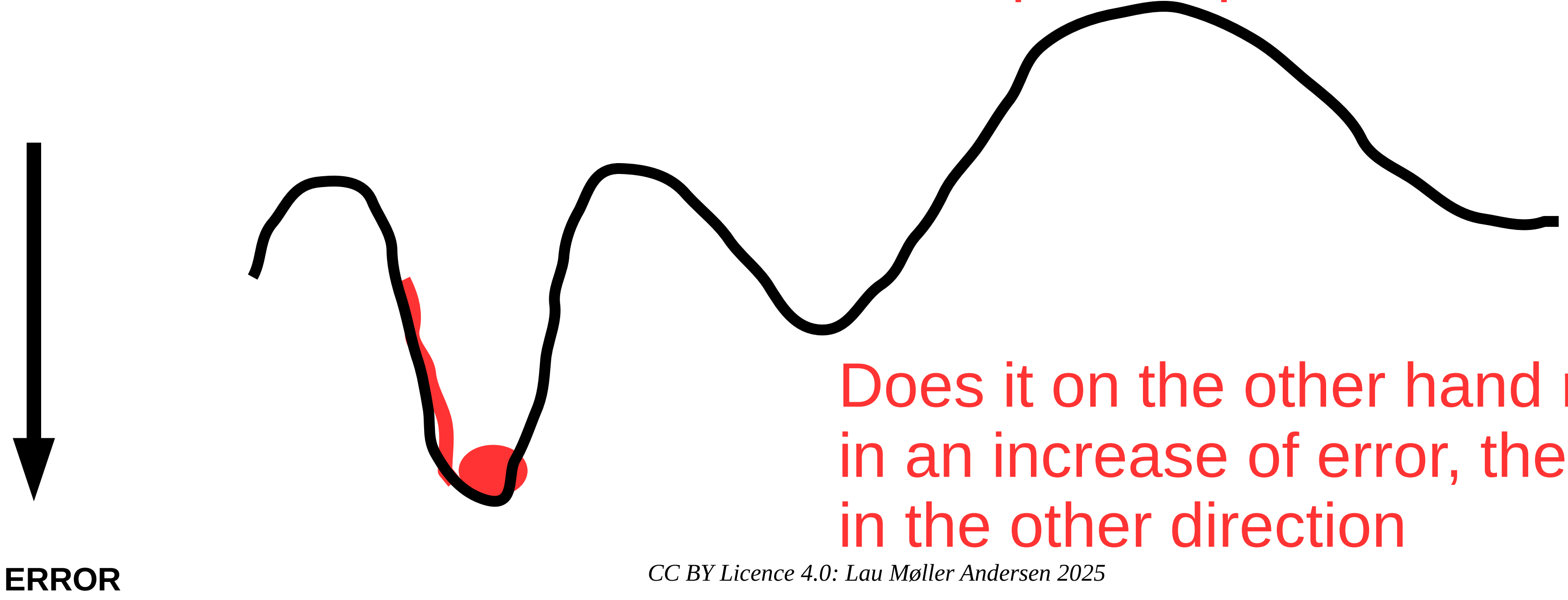


If this results in a reduction of the error, carry on in the same direction

ERROR

2. Gradient descent

Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.



Does it on the other hand result in an increase of error, then go in the other direction

2. Gradient descent

Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

And if both directions increase the error, then STOP!

ERROR

2. Gradient descent

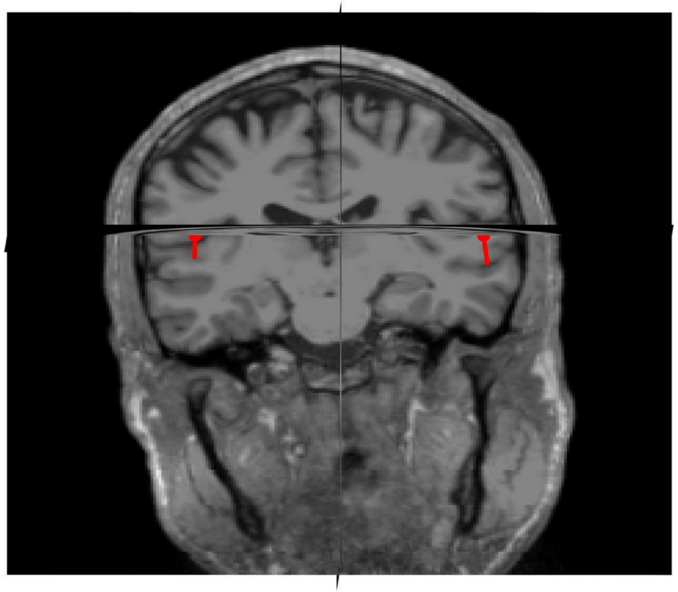
Change moment of dipole: 3 parameters, *x y z;* and position of dipole: 3 parameters, *x y z*.

You might end up in local minima though

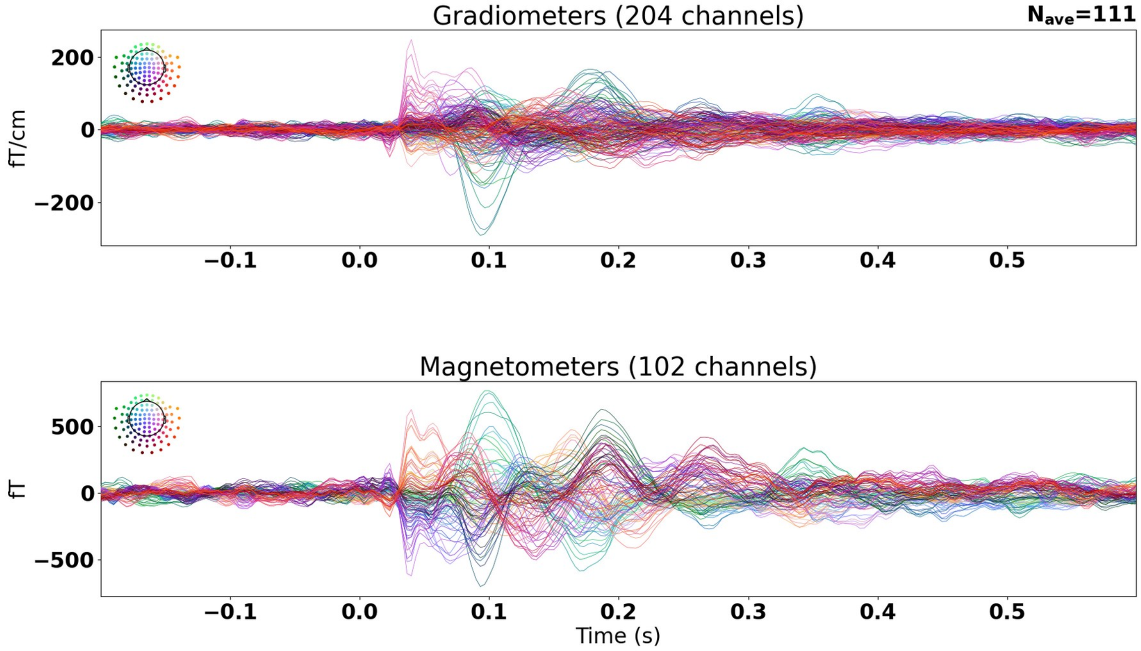
ERROR

Dipole fit results

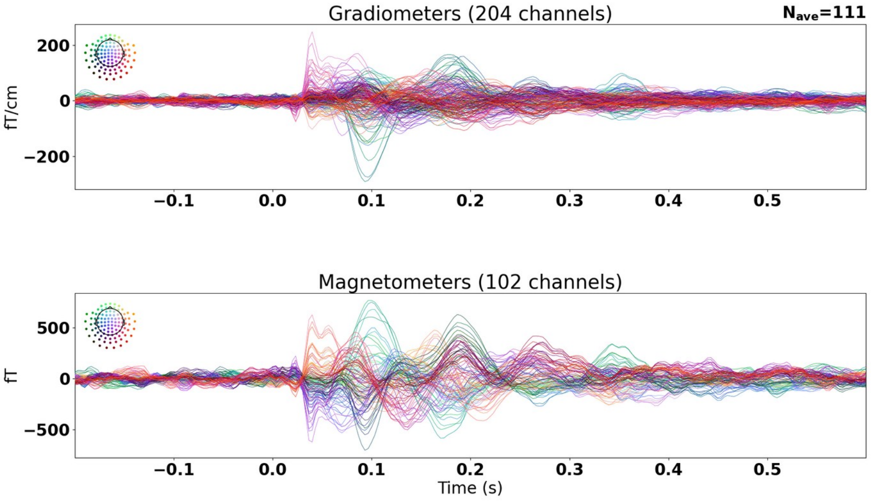
100 ms

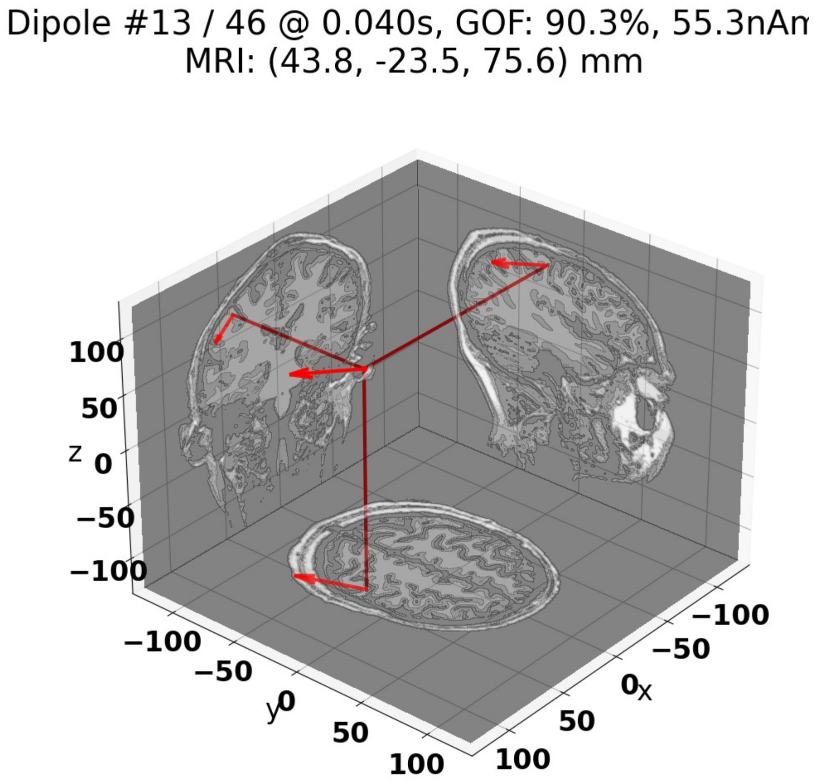


Somatosensory stimulation

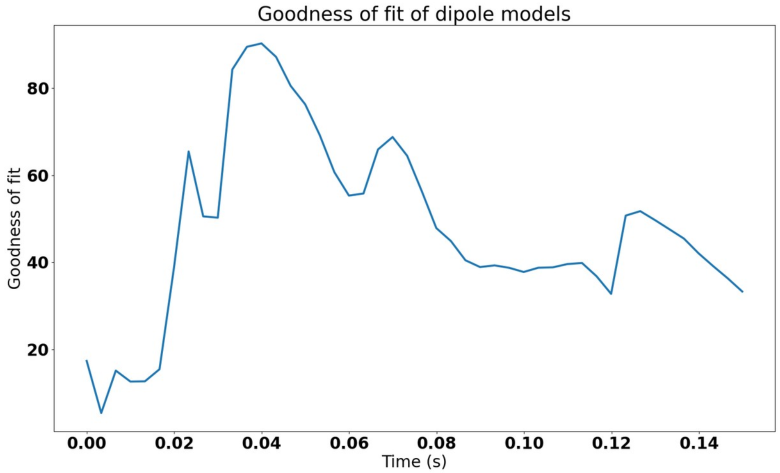


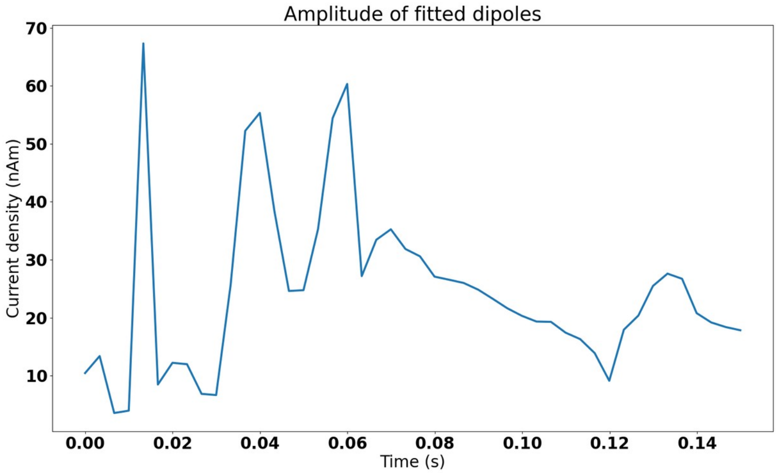
Somatosensory stimulation





Dipole fit results – goodness of fit

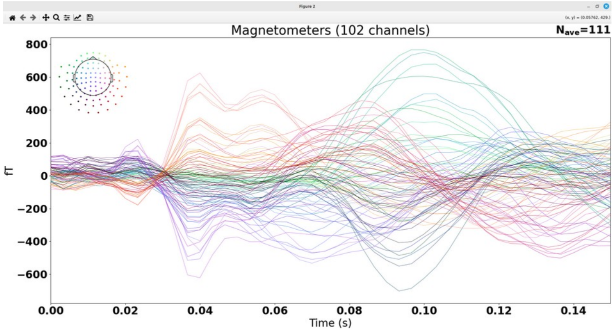


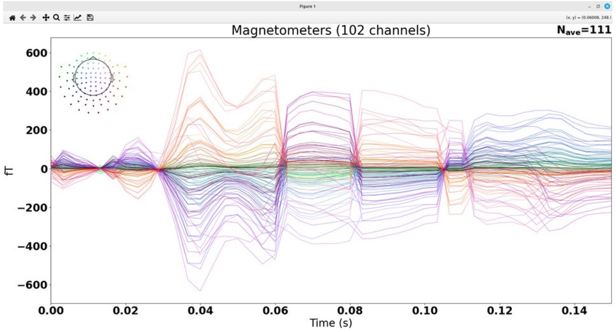


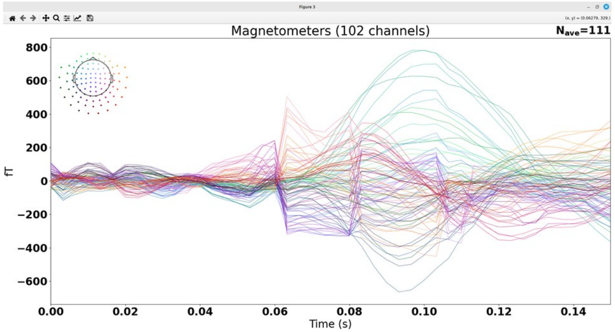
0 = dipolar model explains nothing in the data
100 = dipolar model explains everything in the data

Measured

Model







Residual (measured - model)

CODING EXAMPLE
dipole fitting

may be done in class

Summary

- Dipole fits assume that a single (or few) source(s) can explain (most of) the observed pattern ***b*** at a given *t*
- They are especially good for fitting early, sensory components.
- Several dipoles can be fitted by sequentially fitting the residuals

The course plan

Week 36:
Lesson 0: What is it all about?
Class 0: Setting up UCloud and installing MNE-Python

Week 37:
No Teaching

Week 38:
Lesson 1: Workshop paradigm: Measuring visual subjective experience + MR Recordings
Class 1: Running an MEG analysis of visual responses

Week 39:
MEG workshop: Measuring and predicting visual subjective experience

Week 40:
Lesson 2: Basic physiology and Evoked responses
Class 2: Evoked responses to different levels of subjective experience

Week 41:
Lesson 3: Multivariate statistics
Class 3: Predicting subjective experience in sensor space
Deadline for feedback: Video Explainer

Week 42:
Autumn Break

Week 43:
Lesson 4: Forward modelling and dipole estimation
Class 4: Creating a forward model and fitting dipoles

Week 44:
Lesson 5: Inverse modelling: Minimum-norm estimate
Class 5: Predicting subjective experience in source space

Week 45:
Lesson 6: Inverse modelling: Beamforming
Class 6: Predicting subjective experience in source space, continued

Week 46:
Lesson 7: What about that other cortex? - the cerebellar one
Class 7: Oral presentations (part 1)
Deadline for feedback: Lab report

Week 47:
Lesson 8: Guest lecture: Laura Bock Paulsen: Respiratory analyses
Class 8: Oral presentations (part 2)

Week 48:
Lesson 9: Guest lecture: Barbara Pomiechowska: Using OPM-MEG to study brain and cognitive development in infancy
Class 9: Oral presentations (part 3)

Week 49:
Lesson 0 again: What was it all about?
Class 10: Oral presentations (part 4)

Reading questions

- What does it mean that the minimum-norm is a non-adaptive filter?
- Why is the generalised inverse needed to estimate $\hat{\boldsymbol{\nu}}_{vox}(t)$?
- How is the spatial matched filter similar to dipole fitting?
- What happens with the MNE if data that has a non-brain origin is part of the ***b***(*t*)?

Next class – fitting dipoles to your own data