1928
K. N. Toosi University
of Technology

Faculty of Electrical Engineering

Course: Functional Brain Imaging Systems

Professor: Dr. Ali Khadem

Computer assignment 2

Inverse Problem

Spring 2025

Prepared By:

Ramin Tavakoli, Mohammadreza Shahsavari, Mahdi Molaii

Introduction to the Inverse Problem in Functional Brain Imaging

This computer assignment (CA) explores the inverse problem in MEG and EEG, which involves estimating electrical current sources within the brain based on measurements from external sensors. While the forward problem predicts signals generated by known sources, the inverse problem works backward to reconstruct brain activity patterns from recorded data. This mathematically challenging process is fundamental to source localization and functional brain mapping, enabling researchers to identify active brain regions during specific tasks.

1. MEG Inverse problem.

Visualize MEG Sensor Coordinates

Using the `MEG_Sens.py` script that you implemented in the previous computer assignment (CA), we visualized all the sensors positioned on the scalp, as shown in Figure 1. In this visualization, we establish the locations of 33 sensors on the upper hemisphere of the scalp, which is modeled as a sphere with a radius of $r_3 = 0.09$ m.

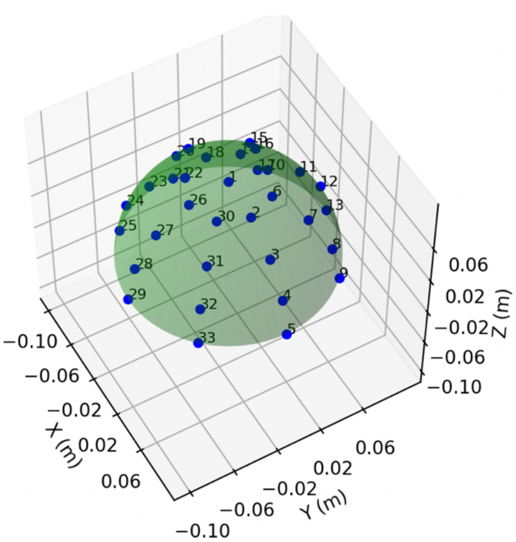

Figure 1) Visualization of all sensors placed on the scalp

Save Electric Current Source Locations

Save the 104 randomly generated electric current source locations from Task 2 of CA1 in a file named **`Diapole_coordinates_1.npz`** within the `dataset` folder.

Task 1

In this task, consider the 104 previously saved locations, along with the location of the current source q_0 defined in Task 5 of the previous CA. This brings the total number of electric current sources to 105.
Your goal is to visualize all 105 electric current sources located in the cerebral cortex.

`Diapole_Loc.py`

```
# ---------------------  Calculate Coordinates of Diapole  ------------------
#TODO: Load the diapole coordinates from the dataset file
data1 = np.load("Dataset/MEG/Diapole_coordinates_1.npz")  # Loading Daipole

#TODO: Define the radius of the hemisphere
radius = 0.07  # radius of the hemisphere

#TODO: Extract the x, y, z coordinates from the loaded data
# rq_x = ...
# rq_y = ...
# rq_z = ...

#TODO: Concatenate the coordinates into a single array
# rq = ...

#TODO: Define the angles for the new diapole in radians
# theta = ...
# phi = ...
# radius = ...

#TODO: Convert spherical coordinates to Cartesian coordinates
# x_0, y_0, z_0 = ...

#TODO: Create the position vector for the new diapole
# rq_0 = ...

#TODO: Define the orientation vector for the new diapole
# q_0 = ...

#TODO: Append the new diapole position to the existing coordinates
# rq = ...
#TODO: Save the updated diapole coordinates to a new file
np.savez("Dataset/MEG/Diapole_coordinates_2.npz", x=rq[:,0], y=rq[:,1], z=rq[:,2],
rq=rq)
```

- Explain the code and output.

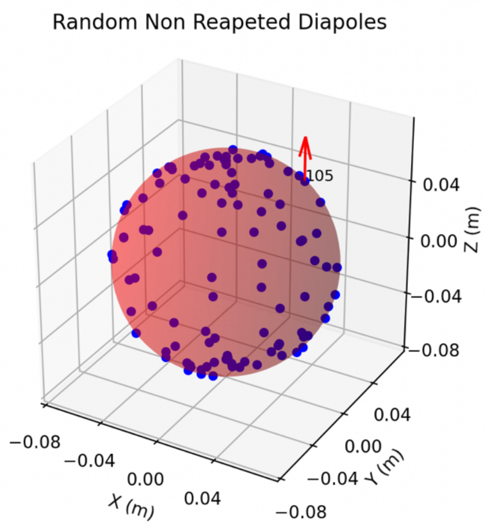

Figure 2) Visualization of all (104 + 1) current sources located on the cerebral cortex

Setup for Inverse Problem

Ensure that all required files obtained are added to the **`dataset`** folder:

- Using the locations of the 105 electric current sources **`(Diapole_coordinates_2.npz)`**, compute the lead field matrix. Save this matrix in a file named **`MEG_Lead_Field_1.npz`** and place it in the `dataset` folder.

- Save the measurements obtained from the forward MEG problem, which includes data from 33 sensors and one current source q_0 from the Task 5 of previous CA, in a file named **`MEG_Measurement_Vector.npz`**. Ensure this file is also placed in the `dataset` folder.

`utility_functions.py` contains the functions that we need for our calculations.

Task 2: Solving MEG inverse Problem Using Minimum Norm Imaging

This task focuses on solving the inverse MEG problem using measurements from **MEG_Measurement_Vector.npz** to estimate current source distributions. The process involves:

Analyze the lead field matrix rank to determine the appropriate solution method (direct minimum norm or pseudo-inverse approach).

Using the Minimum Norm Imaging method, compute the estimated current sources at all 105 locations. Calculate the magnitude of each estimated current source vector. Visualize the resulting field distribution across diapole locations on the hemispherical surface to assess the inverse problem's accuracy.

Task2.py

```
# ----------------------- Loading Diapole and MEG_Lead_Field_1 -----------------------
#TODO: Load the diapole coordinates data from file
# data1 = ...
#TODO: Load the MEG lead field matrix
# data2 = ...
#TODO: Load the MEG measurement vector
# data3 = ...

#TODO: Extract the x, y, z coordinates from the daipole data
# rq_x = ...
# rq_y = ...
# rq_z = ...
# rq = ...

#TODO: Define the orientation vector for the diapole
# q_0 = ...

#TODO: Extract the lead field matrix and measurement vector
# G = ...
# B_r = ...

#TODO: Calculate and print the rank and shape of the lead field matrix
# Rank = ...
print("Shape of G = ", G.shape)
print("Rank of G = ", Rank)

# --------------------------- calculate Current_source_vector -----------------------
#TODO: Calculate the current source vector using the minimum norm solution
# q = ...
#TODO: Initialize an array to store the magnitude of each current source
# norm_q = ...

#TODO: Calculate the magnitude of each current source
# for i in range(0, 105):
    # ...
#TODO: Reshape the current source vector for easier analysis
# q1 = ...

#TODO: Print the estimated diapole orientation and related information
print("q_0 vector = ", q1[104, :])
print("diapole number = ", np.argmax(norm_q), "\nmaximum norm_q = ",
      np.max(norm_q), "\nq_0 norm = ", norm_q[104])
```

What is the relative error in estimating the actual electric current source vector q_0 and the total vector of electric current sources $\vec{q}$? (Calculate the relative error as $\frac{||(\hat{\vec{q}}_0 - \vec{q}_0)||}{||\vec{q}_0||}$, $\frac{||(\hat{\vec{q}} - \vec{q})||}{||\vec{q}||}$.)

```
#  ---------------------------- Calculate the relative error ----------------------------
#TODO: Reshape the current source vector if not already done
# q1 = ...

#TODO: Calculate the relative error between estimated and true diapole orientation
# relative_q0_error = ...
print('The relative q0 error =', relative_q0_error)
#TODO: Calculate the relative error across all estimated sources
# relative_q_error = ...
print('The relative q error =', relative_q_error)
```

- Explain the code and output.
- Has the location of the real electric current source q_0, which is at $(r0, \theta0, \varphi0)$ = $(7cm,$ 45deg, 45deg) been correctly identified?

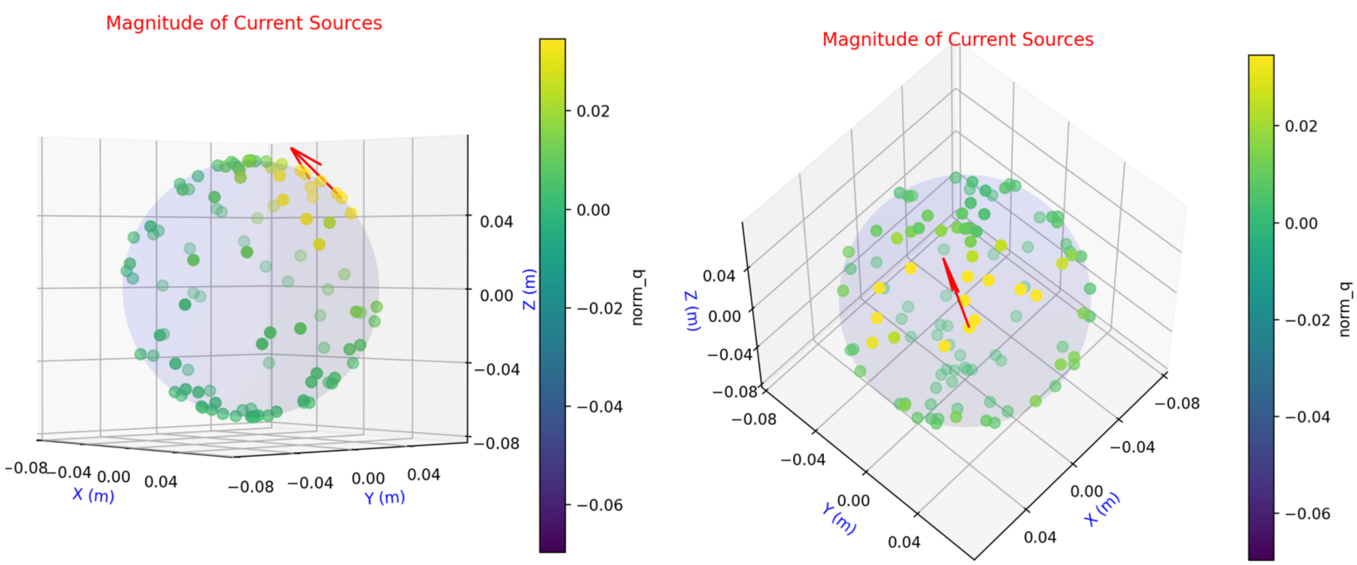

Figure 3) Visualization of current source vectors and display of their magnitude

Task 3 (optional)

plot the estimated vector size of the electric current source as a surface plot with the θ and φ axes.

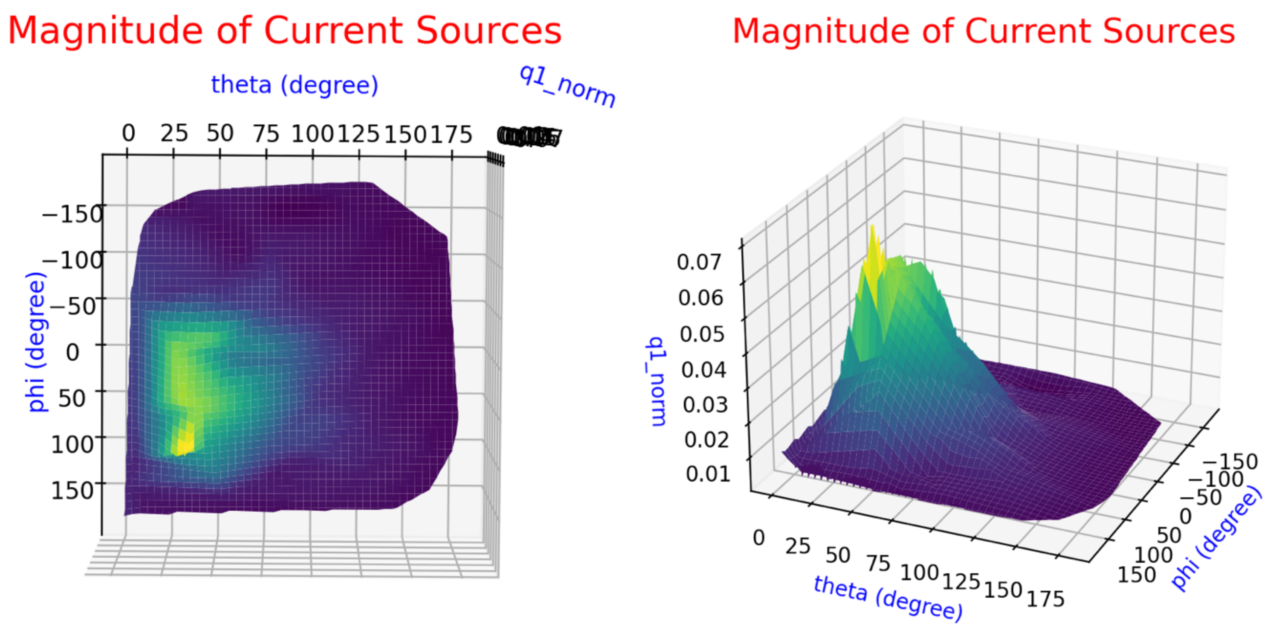

Figure 4) Visualization of current source vectors and their magnitude in the form of a surface

- Where exactly is the maximum of the above surface plot located? Is it at the location of the actual current source? And what is its size with respect to the color bar?

Task 4: Solving MEG inverse Problem Using the parametric least squares method

Solve the inverse MEG problem using parametric least squares, focusing solely on the known source location q_0 at coordinates ($r0, \theta0, \varphi0$) = (7cm, 45°, 45°). The goal is to estimate the current source components $\vec{q}0$ = [qx, qy, qz].
Generate and save the lead field matrix for this single source location as **`MEG_Lead_Field_2.npz`** in the dataset folder.
Before applying the least squares method, check the rank of the lead field matrix to determine whether to use the standard least squares method or the pseudo-inverse approach.

`Task4.py`

```
# -------------------------- Loading Diapole and MEG_Lead_Field_2 --------------------
-
#TODO: Load the diapole coordinates data from file
# data1 = ...

#TODO: Load the MEG lead field matrix
# data2 = ...

#TODO: Load the MEG measurement vector
# data3 = ...

#TODO: Extract the x, y, z coordinates from the diapole data
# rq_x = ...
# rq_y =  ...
# rq_z = ...
# rq = ...

#TODO: Define the orientation vector for the diapole
# q_0 = ...

#TODO: Extract the lead field matrix and measurement vector
# G = ...
# B_r = ...

#TODO: Calculate and print the rank of the lead field matrix
# Rank = ...
# print("Rank of G = ", Rank)

# --------------------------- calculate Current_source_vector -----------------------

#TODO: Calculate the current source vector using the pseudoinverse
# q = ...
# print('shap of q = ', q.shape)
# print('q = ', q)

#TODO: Calculate the magnitude of the current source
```

```
# norm_q = ...
# print('norm of q = ', norm_q)



#  -------------------------- Calculate the relative error ---------------------------

#TODO: Calculate the relative error between estimated and true diapole orientation
# relative_q0_error = ...
# print('The relative q0 error =', ...)

#TODO: Calculate the relative error (same calculation as above in this case)
# relative_q_error = ...
# print('The relative q error =', ...)
```

- Explain the code and output.
- What is the relative error of the estimate of q_0 and the total vector of electric current sources $\vec{q}$ in percent?

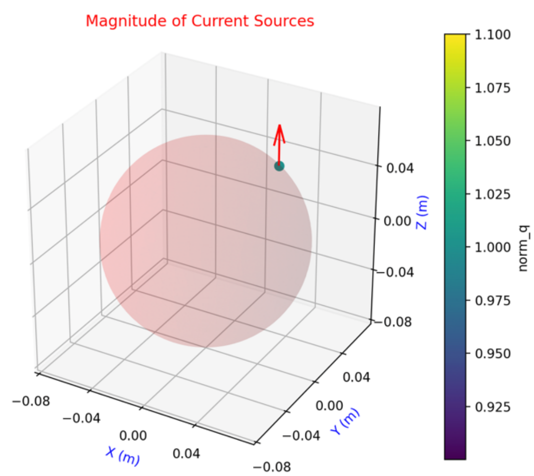

Figure 5) Visualization of the current source vector

2. EEG Inverse problem.

In this part you used the same sensor that you have in the MEG inverse problem part, we visualized all the sensors positioned on the scalp, as shown in Figure 1.

Setup for Inverse Problem

Ensure that all required files like MEG part are added to the **`dataset`** folder:

- **`Diapole_coordinates_1.npz`**: for 104 electric current sources.
- **`Diapole_coordinates_2.npz`**: for 104 electric current sources plus q_0, which is at ($r0$, $\theta 0$, $\varphi 0$) = ($7cm$, 45deg, 45deg).
- **`EEG_Lead_Field_1.npz`**: for 105 electric current sources.
- **`EEG_Lead_Field_2.npz`**: for q_0.
- **`MEG_Measurement_Vector.npz`** : the measurements obtained from the forward EEG problem from the Task 10 of previous CA.

Task 5: Solving EEG iverse Problem Using Minimum Norm Imaging

This task focuses on solving the inverse EEG problem using measurement data obtained from the previous computer assignment (Task 10) involving 33 sensors and a single current source.

Using the measurements stored in **EEG_Measurement_Vector.npz**, we will estimate the current source distribution and assess its accuracy.

Finally, create a visualization showing the field distribution on the diapole locations and plot the hemisphere surface to evaluate the accuracy of your inverse solution.

Task5.py

```
# ----------------------- Loading Diapole and EEG_Lead_Field_1 -----------------------
#TODO: Load the diapole coordinates data from file
# data1 = ...
#TODO: Load the EEG lead field matrix
# data2 = ...
#TODO: Load the EEG measurement vector
# data3 = ...

#TODO: Extract coordinates from the diapole data
# rq_x = ...
# rq_y = ...
```

```python
# rq_z = ...
# rq = ...

#TODO: Define the orientation vector for the diapole
# q_0 = ...

#TODO: Extract the lead field matrix and measurement vector
# L = ...
# V = ...

#TODO: Calculate and print the rank and shape of the lead field matrix
# Rank = ...
print("Rank of L = ", Rank)
print("Shape of L = ", L.shape)

# --------------------------- Calculate Current Source Vector -------------------------
#TODO: Calculate the current source vector using the minimum norm solution
# q = ...

#TODO: Initialize an array to store the magnitude of each current source
# norm_q = ...

#TODO: Calculate the magnitude of each current source
# for i in range(0, 105):
#     ...

#TODO: Reshape the current source vector for easier analysis
# q1 = ...

#TODO: Print the estimated diapole orientation and related information
print("q_0 vector = ", q1[104, :])
print("diapole number = ", np.argmax(norm_q), "\nmaximum norm_q = ",
      np.max(norm_q), "\nq_0 norm = ", norm_q[104])
# --------------------------- Calculate the Relative Error --------------------------
#TODO: Reshape the current source vector (if not already done)
# q1 = ...

#TODO: Calculate relative q0 error
# relative_q0_error = ...
print('The relative q0 error =', relative_q0_error)
#TODO: Calculate relative q error
# relative_q_error = ...
print('The relative q error =', relative_q_error)
```

- Explain the code and output.
- Has the location of the real electric current source q_0, which is at $(r0, \theta 0, \varphi 0) = (7cm,$ 45deg, 45deg) been correctly identified?
- What is the relative error in estimating the actual electric current source vector q_0 and the total vector of electric current sources $\vec{q}$?

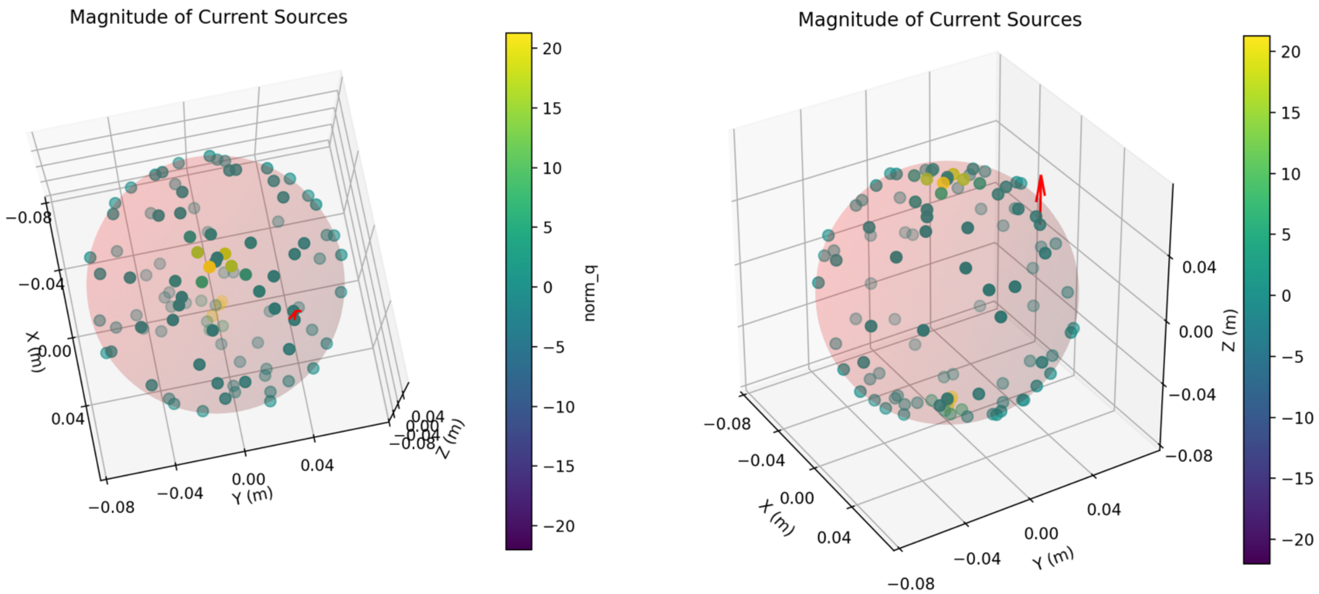

Figure 6) Visualization of current source vectors and display of their magnitude

Task 6 (optional)

Create a surface plot displaying the estimated current source vector magnitudes across theta (θ) and phi (ϕ) coordinates. This visualization will help analyze the spatial distribution of the estimated source strengths and allow for comparison with the actual source location.

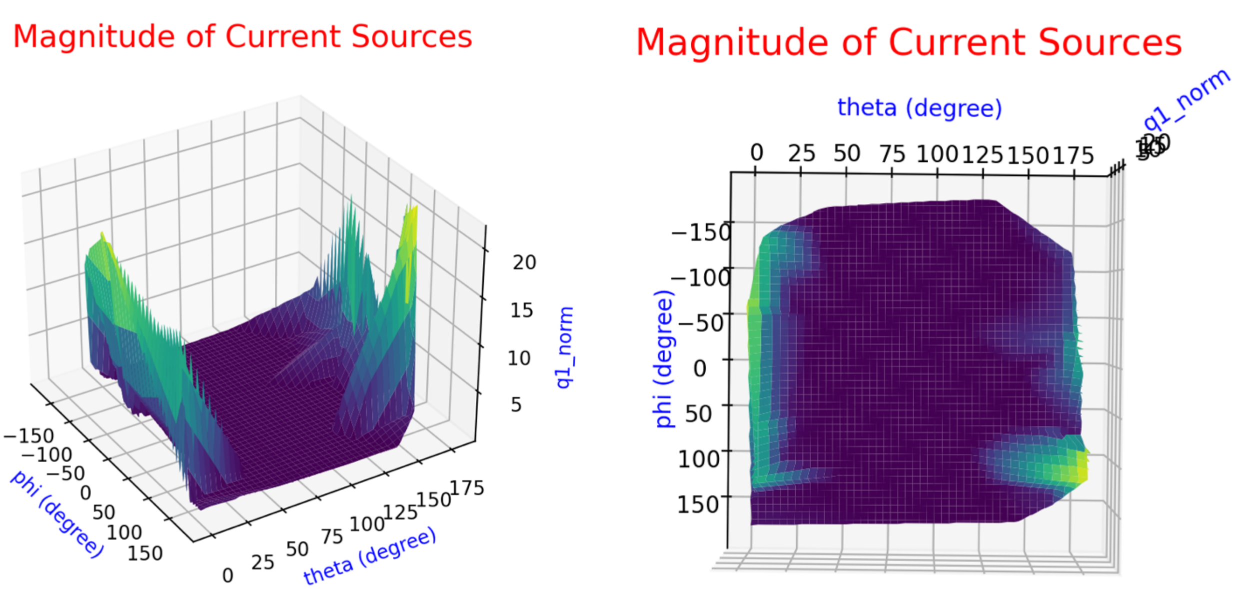

Figure 4) Visualization of current source vectors and their magnitude in the form of a surface

- Where exactly is the maximum of the above surface plot located? Is it at the location of the actual current source? And what is its size with respect to the color bar?

Task 7: Solving EEG inverse Problem Using the parametric least squares method

Now solve the inverse EEG problem using the parametric least squares method, by considering the location of the electric current source only at the actual location q_0, i.e. ($r0$, $\theta 0$, $\varphi 0$) = ($7cm$, 45deg, 45deg) and estimate q_0 = [qx, qy, qz].
Before applying the least squares method, check the rank of the lead field matrix.

Task7.py

```
# ------------------------ Loading Diapole and EEG_Lead_Field_1 ------------------------
#TODO: Load the daipole coordinates data from file
# data1 = ...
#TODO: Load the EEG lead field matrix
# data2 = ...
#TODO: Load the EEG measurement vector
# data3 = ...

#TODO: Extract coordinates from the diapole data
# rq_x = ...
# rq_y = ...
# rq_z = ...
# rq = ...

#TODO: Define the orientation vector for the diapole
# q_0 = ...

#TODO: Extract the lead field matrix and measurement vector
# L = ...
# V = dat...

#TODO: Calculate and print the rank of the lead field matrix
# Rank = ...
print("Rank of L = ", Rank)

# ---------------------------- Calculate Current Source Vector ------------------------
#TODO: Calculate the current source vector using pseudoinverse
# q = ...

#TODO: Print shape and value of current source vector
print('shape of q = ', q.shape)
print('q_0 = q =', q)

#TODO: Calculate magnitude of the current source
```

```
# norm_q = ...
print('norm of q = ', norm_q)

# -------------------------- Calculate the Relative Error ------------------------
#TODO: Calculate relative q0 error
# relative_q0_error = ...
print('The relative q0 error =', relative_q0_error)
#TODO: Calculate relative q error
# relative_q_error = ...
print('The relative q error =', relative_q_error)
```

- Explain the code and output.
- What is the relative error of the estimate of q_0 and the total vector of electric current sources $\vec{q}$ in percent?

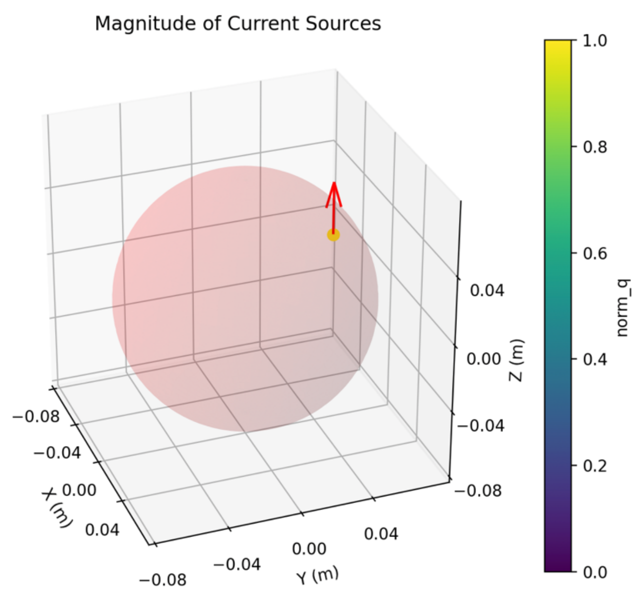

Figure 8) Visualization of estimated current source vectors

3. Probative issues

1. Consider the Imaging Method for solving the inverse problem (Weighted Minimum Norm) as follows, where $G_{m\times n}$ is the lead field matrix (n > m).

$$\begin{cases} \vec{q}_{WMN} = arg_{\vec{q}}\min\{\vec{q}^{T} W \vec{q}\} \\ \quad Subject\ to{:}\ \vec{b} = G\vec{q} \end{cases}$$

If the matrix $GW^{-1}G^T$ is invertible, prove that the answer to the inverse problem using this method is as follows:

$$\vec{q}_{WMN} = W^{-1}G^T(GW^{-1}G^T)^{-1}\vec{b}$$

(In general, the answer to this problem is $\vec{q}_{WMN} = W^{-1}G^T(GW^{-1}G^T)^{+}\vec{b}$)

2. Consider the cost function of the Imaging method for solving the inverse Regulated Minimum Norm problem as follows, where $Gm\times n$ is the lead field matrix (m < n and λ is a fixed chosen scalar).

$$C_\lambda(\vec{q}) = \left\|\vec{b} - G\vec{q}\right\|^2 + \lambda\vec{q}^T\vec{q} = \left(\vec{b} - G\vec{q}\right)^T(\vec{b} - G\vec{q}) + \lambda\vec{q}^T\vec{q}$$

Prove that the answer to the inverse problem using this method is as follows:

$$\vec{q}_\lambda = arg_{\vec{q}}\min\{C_\lambda(\vec{q})\} =$$

$$(G^TG + \lambda I_n)^{-1}G^T\vec{b} = G^T(GG^T + \lambda I_m)^{-1}\vec{b}$$

4. Research issues

1. As mentioned in class, LORETA is a widely used imaging method for solving inverse problems based on the Weighted Minimum Norm method. Examine the form of the weight matrix W in the LORETA method? What are the advantages and disadvantages of this method?

2. Explain the inverse problem solving method using beamforming and list its advantages.

توجه:

1. پاسخ ارسالی شما باید شامل یک گزارش برای پاسخ به سوالات داخل متن کارگاه در یک فایل ورد، به همراه کدهای مربوط به تسک‌ها باشد و هر دو در یک فایل rar ارسال گردند.
2. حتماً در موعد مقرر به بارگذاری پاسخ تمرین در سامانه VC اقدام کنید (تاخیر قابل قبول نیست)
3. در صورت بروز هرگونه ابهام یا سوال در مورد تمرین حتما با دستیاران آموزشی در ارتباط باشید.
4. لطفاً نام خانوادگی خود را به صورت لاتین به همراه شماره تمرین به عنوان نام فایل بارگذاری شده قرار دهید.

   به عنوان مثال :

FIS_Ramintavakoli_CA1

5. زمان کافی جهت تحقیق جهت یافتن پاسخ تمارین به دانشجویان گرامی داده شده. لازم به ذکر است که هر شخص باید برداشت و نتایج تحقیقات خود را ارائه دهد پس به همین دلیل از کپی برداری پاسخنامه یکدیگر شدیدا خودداری فرمایید و در صورت مشاهده مسئولیت عواقب منفی متوجه تمامی افراد خاطی میگردد. همچنین ممکن است جلسه های مجازی تحت عنوان پرسش و پاسخ در خصوص گزارش دوستان گرامی برگزار شود.

موفق باشید.