

# Numerical Simulation Methods in Geophysics, Exercise 8: 1D/2D EM

## 1. MGPY+MGIN

*thomas.guenther@geophysik.tu-freiberg.de*



**TUBAF**  
Die Ressourcenuniversität.  
Seit 1765.

# Tasks for today

- have some drinks and sweets
- finish 2D Poisson problem with singular source

$$-\nabla \cdot \sigma \nabla u = I\delta(\mathbf{r} - \mathbf{r}_s)$$

Analytical solution:  $u = -\frac{I}{2\pi\sigma} \ln r$

- 1D EM problem  $\Rightarrow$  2D EM problem

# Tasks

- write function for analytical solution and plot
- generate a model with two electrodes at the surface
- start with homogeneous conductivity
- create stiffness matrix and load vector
- solve the matrix-vector equation
- plot solution and compare with analytical solution
- use inhomogeneous conductivity and try secondary field approach

# Complex or real-valued?

The complex-valued system

$$(\mathbf{A} + \imath\omega\mathbf{M})\mathbf{u} = (\mathbf{A} + \imath\omega\mathbf{M})(\mathbf{u}_r + \imath\mathbf{u}_i) = \mathbf{b}_r + \imath\mathbf{b}_i$$

can be transferred into a doubled real-valued system

$$\mathbf{A}\mathbf{u}_r + \imath\mathbf{A}\mathbf{u}_i + \imath\omega\mathbf{M}\mathbf{u}_r - \omega\mathbf{M}\mathbf{u}_i = \mathbf{b}_r + \imath\mathbf{b}_i$$

$$\begin{pmatrix} A & -\omega M \\ \omega M & A \end{pmatrix} \begin{pmatrix} u_r \\ u_i \end{pmatrix} = \begin{pmatrix} b_r \\ b_i \end{pmatrix}$$

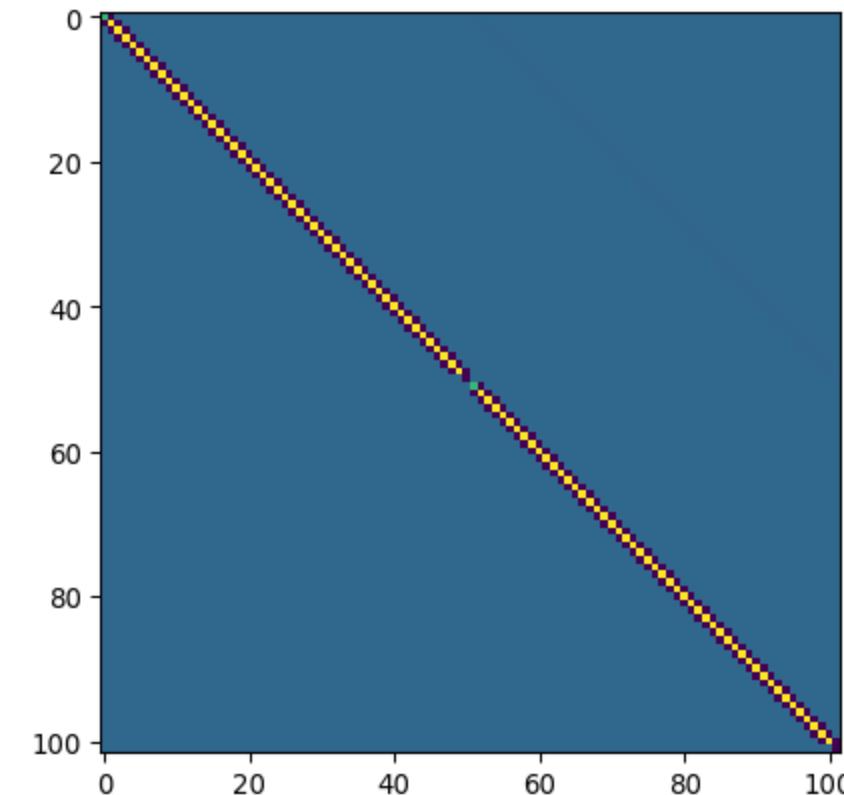
# The problem in 1D

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from poisson1d import stiffnessMatrix1DFE, massMatrix1DFE
4 T = 0.1 # 0.1
5 w = 2 * np.pi / T
6 sigma0 = 1/100 # 1/100
7 mu = np.pi * 4e-7
8 z = np.arange(-10000, 0.1, 200)
9 A, b = stiffnessMatrix1DFE(x=z, uR=1)
10 M = massMatrix1DFE(x=z, a=mu*sigma0)
11 AM = A + M * 1j * w
12 u = np.linalg.solve(AM, b)
```

# Complex-to-real conversion

$$\mathbf{B} = \begin{pmatrix} \mathbf{A} & -\omega \mathbf{M} \\ \omega \mathbf{M} & \mathbf{A} \end{pmatrix}$$

```
1 D = np.vstack([np.hstack([A, -M*w]), np.hstack([M, A])])
2 plt.imshow(D)
3 d = np.hstack([b, b*0])
4 uri = np.linalg.solve(D, d)
5 u = uri[:len(z)] + uri[len(z):] * 1j
```



# Complex-to-real conversion

$$\mathbf{B} = \begin{pmatrix} \mathbf{A} & -\omega\mathbf{M} \\ \omega\mathbf{M} & \mathbf{A} \end{pmatrix}$$

# Secondary field approach

Consider the field to consist of a primary (background) and an secondary (anomalous) field  $F = F_0 + F_a$

solution for  $F_0$  known, e.g. analytically or 1D (semi-analytically)

⇒ form equations for  $F_a$ , because

- $F_a$  is weaker or smoother (e.g.  $F_0 \propto 1/r$  at sources)
- boundary conditions easier to set (e.g. homogeneous Dirichlet)

# Secondary field Helmholtz equation

The equation  $-\nabla^2 F - k^2 F = 0$  is solved by the primary field for  $k_0$ :

$-\nabla^2 F_0 - k_0^2 F_0 = 0$  and the total field for  $k_0 + \delta k$ :

$$-\nabla^2(F_0 + F_a) - (k_0^2 + \delta k^2)(F_0 + F_a) = 0$$

$$-\nabla^2 F_a - k^2 F_a = \delta k^2 F_0$$

 **Note**

Source terms only arise at anomalous terms, weighted by the primary field.

# Secondary field for EM

Maxwells equations  $k^2 = -\omega\mu\sigma$

$$-\nabla^2 \mathbf{E}_0 + \omega\mu\sigma \mathbf{E}_0 = 0$$

leads to

$$-\nabla^2 \mathbf{E}_a + \omega\mu\sigma \mathbf{E}_a = -\omega\mu\delta\sigma \mathbf{E}_0$$

 Note

Source terms only arise at anomalous conductivities and increase with primary field

# Secondary field for EM

Maxwells equations  $k^2 = -\omega\mu\sigma$

$$-\nabla^2 \mathbf{E}_0 + \omega\mu\sigma \mathbf{E}_0 = 0$$

leads to

$$-\nabla^2 \mathbf{E}_a + \omega\mu\sigma \mathbf{E}_a = -\omega\mu\delta\sigma \mathbf{E}_0$$

 Note

Source terms only arise at anomalous conductivities and increase with primary field

# Secondary field for EM

$$-\nabla^2 \mathbf{E}_a + \omega\mu\sigma \mathbf{E}_a = -\omega\mu\delta\sigma \mathbf{E}_0$$

leads to the discretized form (**A**-stiffness, **M**-mass)

$$\mathbf{A}\mathbf{E}_a + \omega\mathbf{M}_\sigma E_a = -\omega\mathbf{M}_{\delta\sigma} \mathbf{E}_0$$

```
1 A = stiffnessMatrix1DFE(x=z)
2 M = massMatrix1DFE(x=z, a=w*mu*sigma)
3 dM = massMatrix1DFE(x=z, a=w*mu*(sigma-sigma0))
4 u = uAna + solve(A+M*w*1j, dM@uAna * w*1j)
```