PGE382L – CP3 (Spring, 2024)

Solve:

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}-\frac{\partial^2 u}{\partial x^2}=-u \;\Rightarrow\; \frac{\partial u}{\partial t}=\frac{\partial u}{\partial x}+\frac{\partial^2 u}{\partial x^2}-u$$

CASE 1 - Crank Nicholson, Central differences:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{1}{2}\left[-\frac{u_{i+1}^{n+1}-u_{i-1}^{n+1}}{2\Delta x}+\frac{u_{i+1}^{n+1}-2u_i+u_{i-1}^{n+1}}{\Delta x^2}-u_i^{n+1}\right]+\frac{1}{2}\left[-\frac{u_{i+1}^{n}-u_{i-1}^{n}}{2\Delta x}+\frac{u_{i+1}^{n}-2u_i^n+u_{i-1}^{n}}{\Delta x^2}-u_i^{n}\right]$$

Factor and reorganize LHS/RHS:

$$\frac{u_i^{n+1}}{\Delta t}-\frac{1}{2}\left[-\frac{u_{i+1}^{n+1}-u_{i-1}^{n+1}}{2\Delta x}+\frac{u_{i+1}^{n+1}-2u_i+u_{i-1}^{n+1}}{\Delta x^2}-u_i^{n+1}\right]=\frac{1}{2}\left[-\frac{u_{i+1}^{n}-u_{i-1}^{n}}{2\Delta x}+\frac{u_{i+1}^{n}-2u_i^n+u_{i-1}^{n}}{\Delta x^2}-u_i^{n}\right]+\frac{u_i^n}{\Delta t}$$

Multiply by 2 and reorganize:

$$u_{i-1}^{n+1}\left[-\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]+u_i^{n+1}\left[\frac{2}{\Delta t}+\frac{2}{\Delta x^2}+1\right]+u_{i+1}^{n+1}\left[\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]$$
$$=u_{i-1}^{n}\left[\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]+u_i^{n}\left[\frac{2}{\Delta t}-\frac{2}{\Delta x^2}-1\right]+u_{i+1}^{n}\left[-\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]$$

BC @ $i=0$ defines the equation at $i=1$

$$u_0^t=e^t \quad\Rightarrow$$

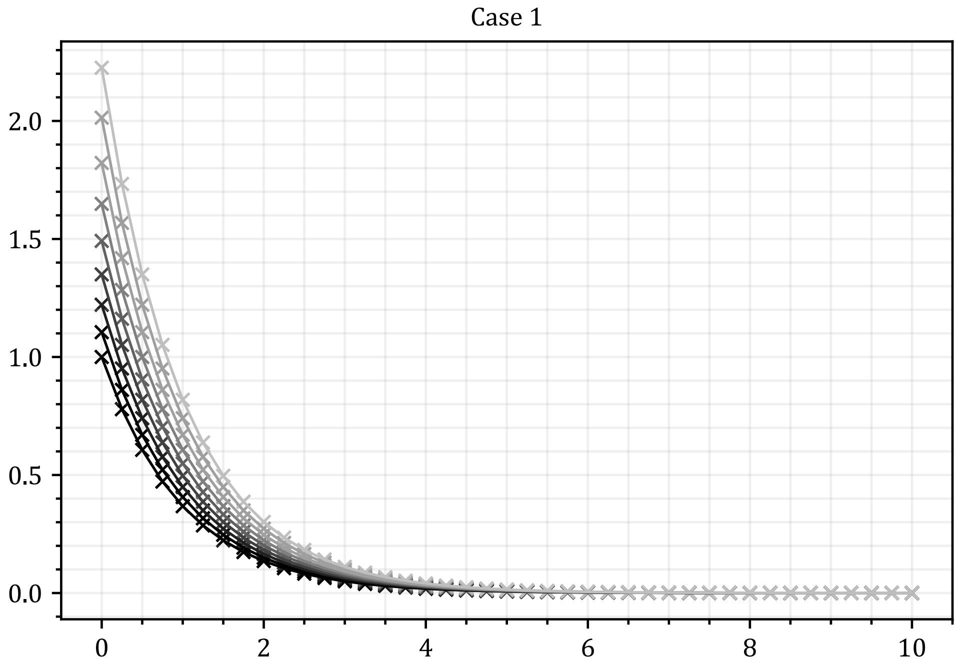
$$u_1^{n+1}\left[\frac{2}{\Delta t}+\frac{2}{\Delta x^2}+1\right]+u_2^{n+1}\left[\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]$$
$$=\; u_0^{n}\left[\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]+u_1^{n}\left[\frac{2}{\Delta t}-\frac{2}{\Delta x^2}-1\right]+u_2^{n}\left[-\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]-u_0^{n+1}\left[-\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]$$

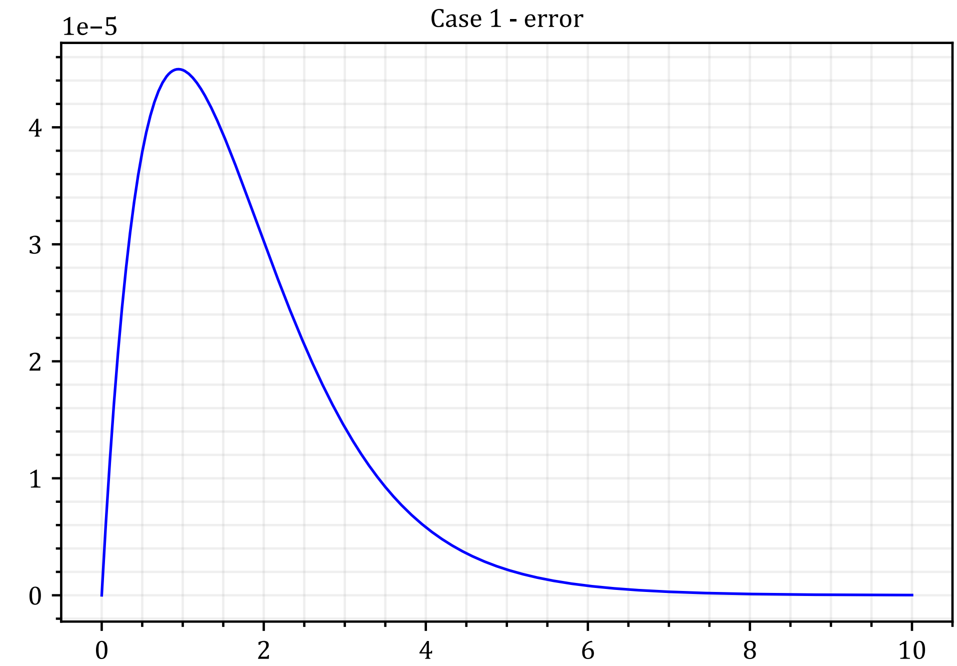
BC @ $i=N$

$$\frac{u_{N+1}-u_{N-1}}{2\Delta x}=-u_N \qquad\Rightarrow\qquad u_{N+1}=u_{N-1}-2\Delta x\, u_N$$

$$u_{N-1}^{n+1}\left[-\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]+u_N^{n+1}\left[\frac{2}{\Delta t}+\frac{2}{\Delta x^2}+1\right]+\left[u_{N-1}^{n+1}-2\Delta x\, u_N^{n+1}\right]\left[\frac{1}{2\Delta x}-\frac{1}{\Delta x^2}\right]$$
$$=u_{N-1}^{n}\left[\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]+u_N^{n}\left[\frac{2}{\Delta t}-\frac{2}{\Delta x^2}-1\right]+\left[u_{N-1}^{n}-2\Delta x\, u_N^{n}\right]\left[-\frac{1}{2\Delta x}+\frac{1}{\Delta x^2}\right]$$

SOLUTION – CASE 1





CASE 2 - Crank Nicholson, Central differences for u_{xx} and Backward difference for u_x:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{1}{2}\left[-\frac{u_i^{n+1}-u_{i-1}^{n+1}}{\Delta x}+\frac{u_{i+1}^{n+1}-2u_i+u_{i-1}^{n+1}}{\Delta x^2}-u_i^{n+1}\right]+\frac{1}{2}\left[-\frac{u_i^n-u_{i-1}^n}{\Delta x}+\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\Delta x^2}-u_i^n\right]$$

Factor and reorganize LHS/RHS:

$$u_{i-1}^{n+1}\left[-\frac{1}{\Delta x}-\frac{1}{\Delta x^2}\right]+u_i^{n+1}\left[\frac{1}{\Delta x}+\frac{2}{\Delta \mathrm{t}}+\frac{2}{\Delta x^2}+1\right]+u_{i+1}^{n+1}\left[-\frac{1}{\Delta x^2}\right]$$
$$=u_{i-1}^n\left[\frac{1}{\Delta x}+\frac{1}{\Delta x^2}\right]+u_i^n\left[\frac{2}{\Delta \mathrm{t}}-\frac{2}{\Delta x^2}-1-\frac{1}{\Delta x}\right]+u_{i+1}^n\left[\frac{1}{\Delta x^2}\right]$$

BC @ $i=0$ defines the equation at $i=1$

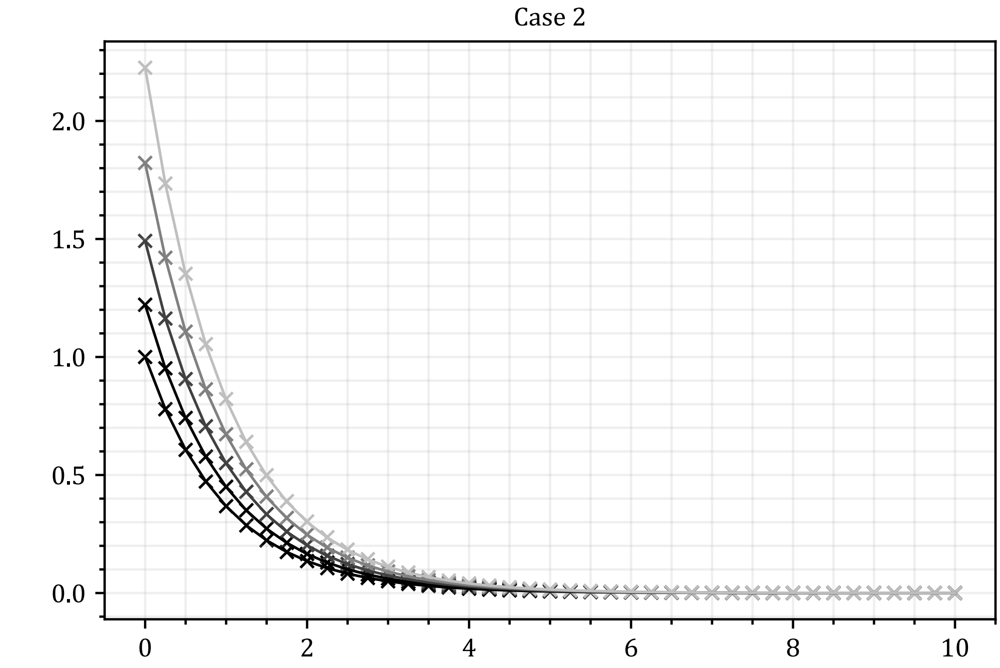
$$u_0^t=e^t: \qquad u_1^{n+1}\left[\frac{1}{\Delta x}+\frac{2}{\Delta \mathrm{t}}+\frac{2}{\Delta x^2}+1\right]+u_2^{n+1}\left[-\frac{1}{\Delta x^2}\right]$$
$$=u_0^n\left[\frac{1}{\Delta x}+\frac{1}{\Delta x^2}\right]+u_1^n\left[\frac{2}{\Delta \mathrm{t}}-\frac{2}{\Delta x^2}-1\right]+u_2^n\left[-\frac{1}{\Delta x}-\frac{1}{\Delta x}+\frac{1}{\Delta x^2}\right]+u_0^{n+1}\left[\frac{1}{\Delta x^2}+\frac{1}{\Delta x}\right]$$

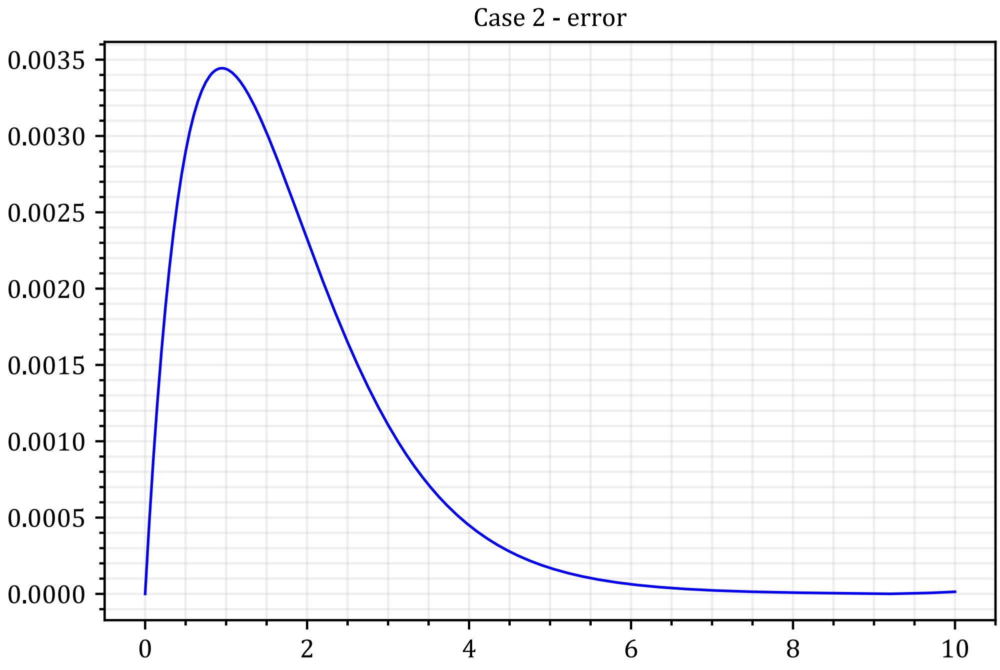
BC @ $i=N$

$$\frac{u_{N+1}-u_N}{\Delta x}=-u_N \qquad \Rightarrow \qquad u_{N+1}=u_N(1-\Delta x)$$

$$u_{N-1}^{n+1}\left[-\frac{1}{\Delta x}-\frac{1}{\Delta x^2}\right]+u_N^{n+1}\left[\frac{1}{\Delta x}+\frac{2}{\Delta \mathrm{t}}+\frac{2}{\Delta x^2}+1\right]+[u_N^{n+1}(1-\Delta x)]\left[-\frac{1}{\Delta x^2}\right]$$
$$=u_{N-1}^n\left[\frac{1}{\Delta x}+\frac{1}{\Delta x^2}\right]+u_N^n\left[-\frac{1}{\Delta x}+\frac{2}{\Delta \mathrm{t}}-\frac{2}{\Delta x^2}-1\right]+[u_N^n(1-\Delta x)]\left[\frac{1}{\Delta x^2}\right]$$

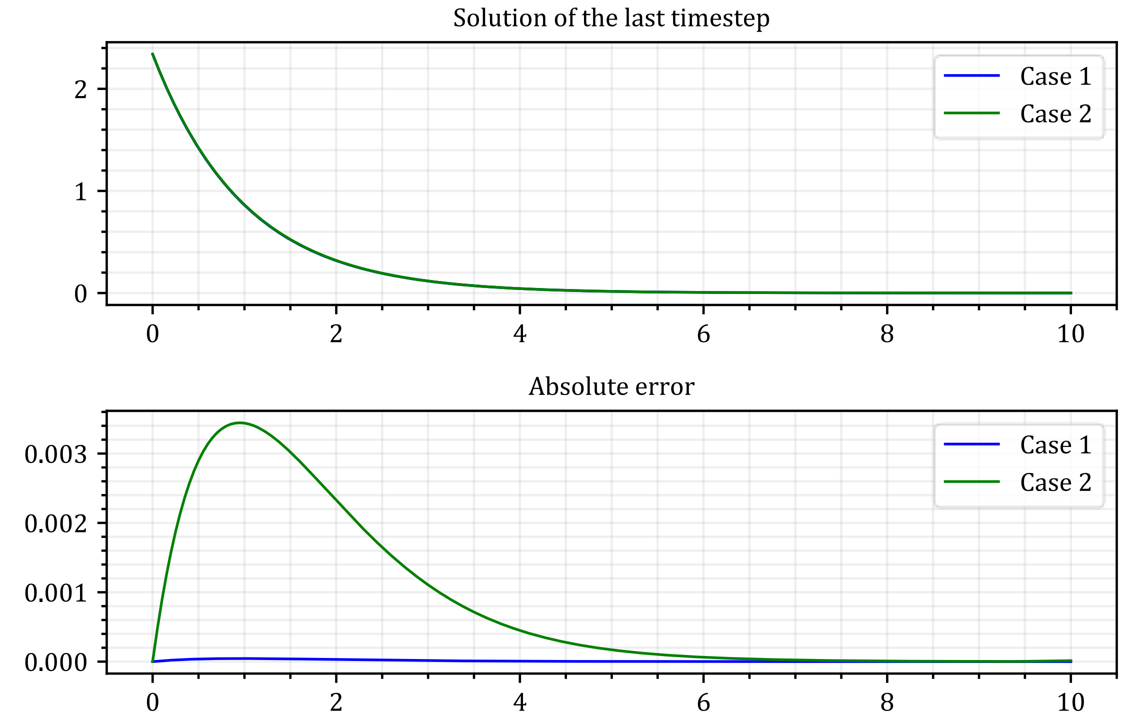
SOLUTION – CASE 2





MAXIMUM ABSOLUTE ERRORS (CASE 1 vs CASE 2)

C1 Max err	C2 Max err
4.49684e-05	3.44407e-03



Code (Python)

```
MAXX = 10
MAXT = 0.85
DX = 1 / 40
DT = 1 / 100

# CASE 1

from math import factorial, pi, sin, ceil
import numpy as np
from numpy import exp, linspace, vectorize
import matplotlib.pyplot as plt

plt.style.use('paper.mplstyle')

X = np.arange(0, 10 + DX, DX)
T = np.arange(0, 0.85 + DT, DT)

nx = len(X)
nt = len(T)

# Set initial condition and BC@X=0
Uni = np.zeros( (nt, nx) )

Uni[0,:] = np.exp( - X )
Uni[:,0] = np.exp( T )

EXACT_Uni = np.zeros( (nt, nx) )
for n in np.arange( 0, nt ) :
    EXACT_Uni[n,:] = np.exp( T[n] - X )

for n in np.arange( 0, nt-1 ) :
    K = np.zeros( (nx, nx) )
    F = np.zeros( nx )
    for i in np.arange( 1, nx ) :
        K[i, i-1] += - 1/2/DX - 1/DX/DX
        K[i, i]   +=   2/DT + 2/DX/DX + 1

        F[i] += Uni[n,i-1] * ( 1/2/DX + 1/DX/DX )
        F[i] += Uni[n,i  ] * ( 2/DT - 2/DX/DX - 1  )

        # BC @ i=0
        if i == 1 :
            F[i] -=  (- 1/2/DX - 1/DX/DX) * Uni[n+1,i-1]

        if i < nx-1 :
           K[i, i+1] += 1/2/DX - 1/DX/DX
           F[i]      += Uni[n,i+1] * ( -1/2/DX + 1/DX/DX )
        # BC @ i=N
        else :
            K[i, i-1] += 1/2/DX - 1/DX/DX
            K[i, i]   += (-2*DX) * ( 1/2/DX - 1/DX/DX )
            F[i] += ( Uni[n,i-1] - 2*DX*Uni[n,i] ) * ( -
1/2/DX + 1/DX/DX )

    # Remove i=0
    K=K[1:nx,1:nx]
    F=F[1:nx]
    U = np.linalg.solve(K,F)
    Uni[n+1,1:nx] = U

# PLOT
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

range = np.arange(nt-1)[::10]
# range = np.arange(2)
colors = cm.get_cmap('gray', len(range))
print(colors)
i=0
for n in range :
    evr = 10
    c=colors(n/nt*.8)
    plt.plot( X[::evr], EXACT_Uni[n,::evr], color=c )
    plt.scatter( X[::evr], Uni[n,::evr], color=c, s=25,
marker='x' )
plt.title("Case 1")
plt.savefig("Case_1.svg")

C1_LAST = Uni[-1,:]
C1_ERR = abs(EXACT_Uni[-1,:] - Uni[-1,:])
plt.plot(X,C1_ERR)
plt.title("Case 1 - error")
plt.savefig("Case_1-Err.svg")

# CASE 2

from math import factorial, pi, sin, ceil
import numpy as np
from numpy import exp, linspace, vectorize
import matplotlib.pyplot as plt

plt.style.use('paper.mplstyle')

X = np.arange(0, 10 + DX, DX)
T = np.arange(0, 0.85 + DT, DT)

nx = len(X)
nt = len(T)

# Set initial condition and BC@X=0
Uni = np.zeros( (nt, nx) )

Uni[0,:] = np.exp( - X )
Uni[:,0] = np.exp( T )

EXACT_Uni = np.zeros( (nt, nx) )
for n in np.arange( 0, nt ) :
    EXACT_Uni[n,:] = np.exp( T[n] - X )

for n in np.arange( 0, nt-1 ) :
    K = np.zeros( (nx, nx) )
    F = np.zeros( nx )
    for i in np.arange( 1, nx ) :
        K[i, i-1] += - 1/DX - 1/DX/DX
        K[i, i]   +=   1/DX + 2/DT + 2/DX/DX + 1

        F[i] += Uni[n,i-1] * ( 1/DX + 1/DX/DX )
        F[i] += Uni[n,i  ] * ( -1/DX + 2/DT - 2/DX/DX - 1  )

        # BC @ i=0
        if i == 1 :
            F[i] +=  (1/DX/DX + 1/DX) * Uni[n+1,i-1]

        if i < nx-1 :
           K[i, i+1] += - 1/DX/DX
           F[i]      += Uni[n,i+1] * ( 1/DX/DX  )
        # BC @ i=N
        else :
            K[i, i]   += (1-DX) * ( - 1/DX/DX )
            F[i] += Uni[n,i] * ( 1 - DX ) * ( - 1/DX +
1/DX/DX )

    # Remove i=0
    K=K[1:nx,1:nx]
    F=F[1:nx]
    U = np.linalg.solve(K,F)
    Uni[n+1,1:nx] = U

# PLOT

import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

range = np.arange(nt-1)[::20]
colors = cm.get_cmap('gray', len(range))
print(colors)
i=0
for n in range :
    evr = 10
    c=colors(n/nt*.8)
    plt.plot( X[::evr], EXACT_Uni[n,::evr], color=c )
    plt.scatter( X[::evr], Uni[n,::evr], color=c, s=25,
marker='x' )
plt.title("Case 2")
plt.savefig("case2.svg")

C2_LAST = Uni[-1,:]
C2_ERR = abs(EXACT_Uni[-1,:] - Uni[-1,:])
plt.plot(X,C2_ERR)
plt.title("Case 2 - error")
plt.savefig("Case_2-Err.svg")

# COMPARE 1 VS 2

fig, [ax1,ax2] = plt.subplots(2,1)

ax1.set_title("Solution of the last timestep")
ax1.plot(X,C1_LAST, label='Case 1')
ax1.plot(X,C2_LAST, label='Case 2')
ax1.legend()

ax2.set_title("Absolute error")
ax2.plot(X,C1_ERR, label='Case 1')
ax2.plot(X,C2_ERR, label='Case 2')
ax2.legend()
fig.tight_layout()

C1_MAX = max(C1_ERR)
C2_MAX = max(C2_ERR)

print(f"{'C1 Max err':20s}{'C2 Max err':20s}")
print(50*"=")
print(f"{C1_MAX:10.5e}{C2_MAX:-20.5e}")
print(50*"=")

fig.savefig("err.svg")
```