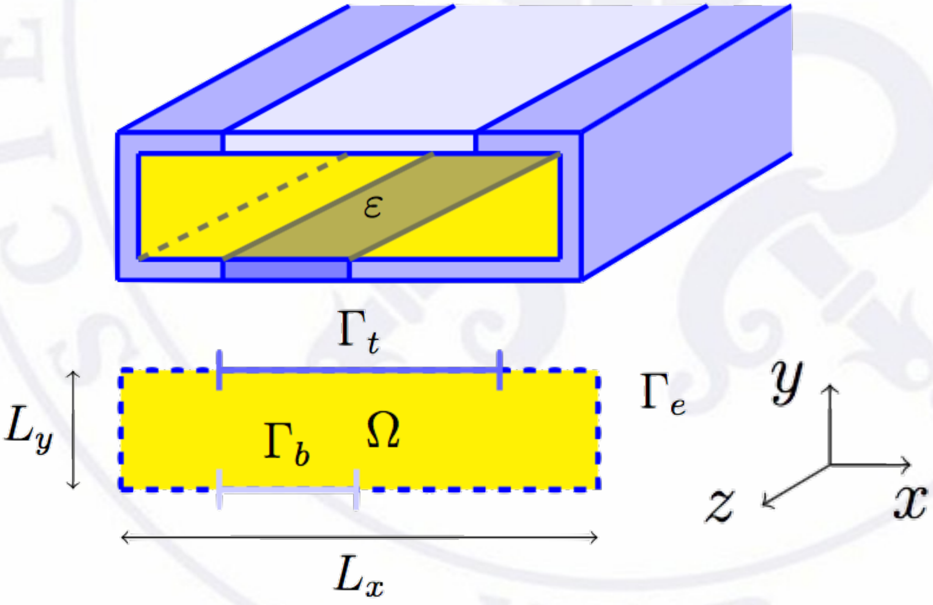
MATH-H301-Numerical Calculation

Presentation of the Project: Study of a capacitor

Alban Dietrich

1. Introduction
2. Program
    - Question 1: the prob function
    - Question 2: the residue
    - Question 3: the potential
    - Question 4: the E field
    - Question 5: capacity
    - Question 6: AGMG
    - The Makefile
3. Improvements



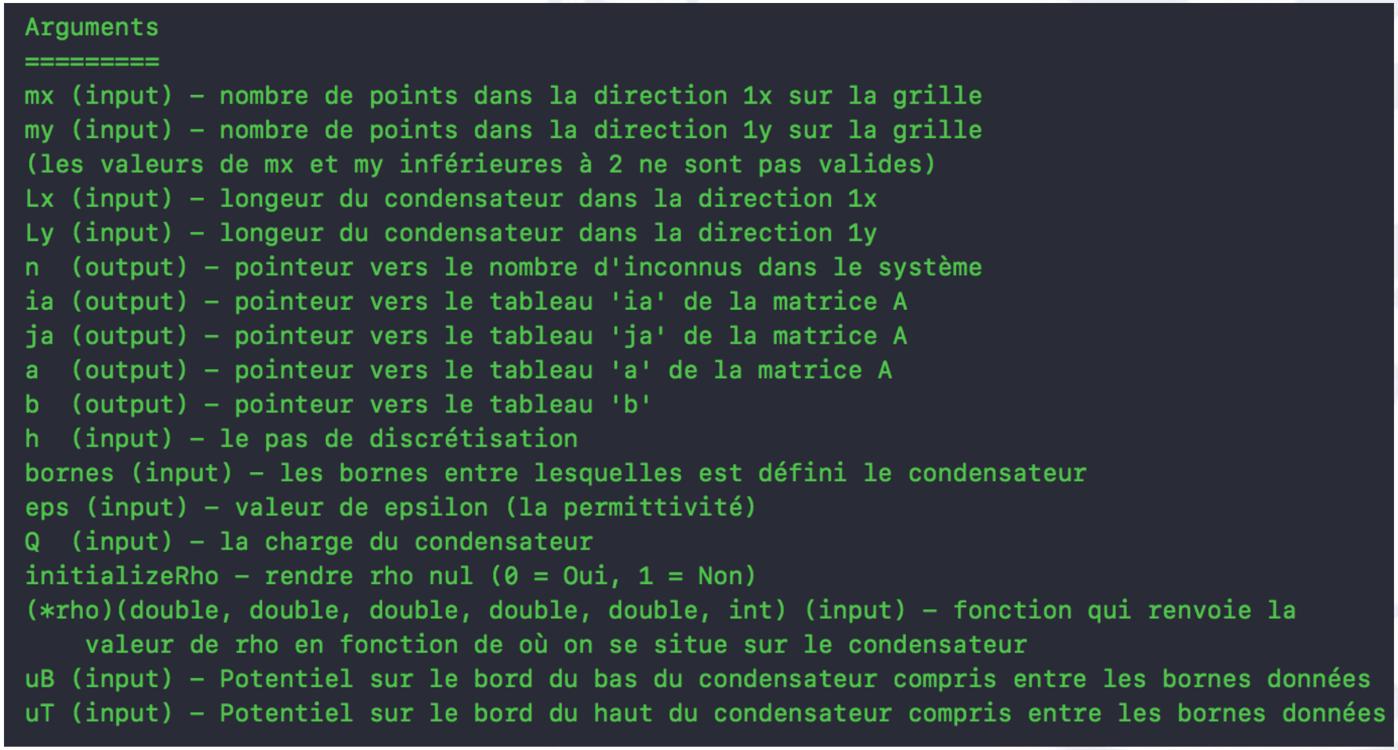
1. Introduction

- General code, adaptable to all projects
- Menu available
- Comparison of the residuals of the 2 solvers on a graph, not very relevant

Program

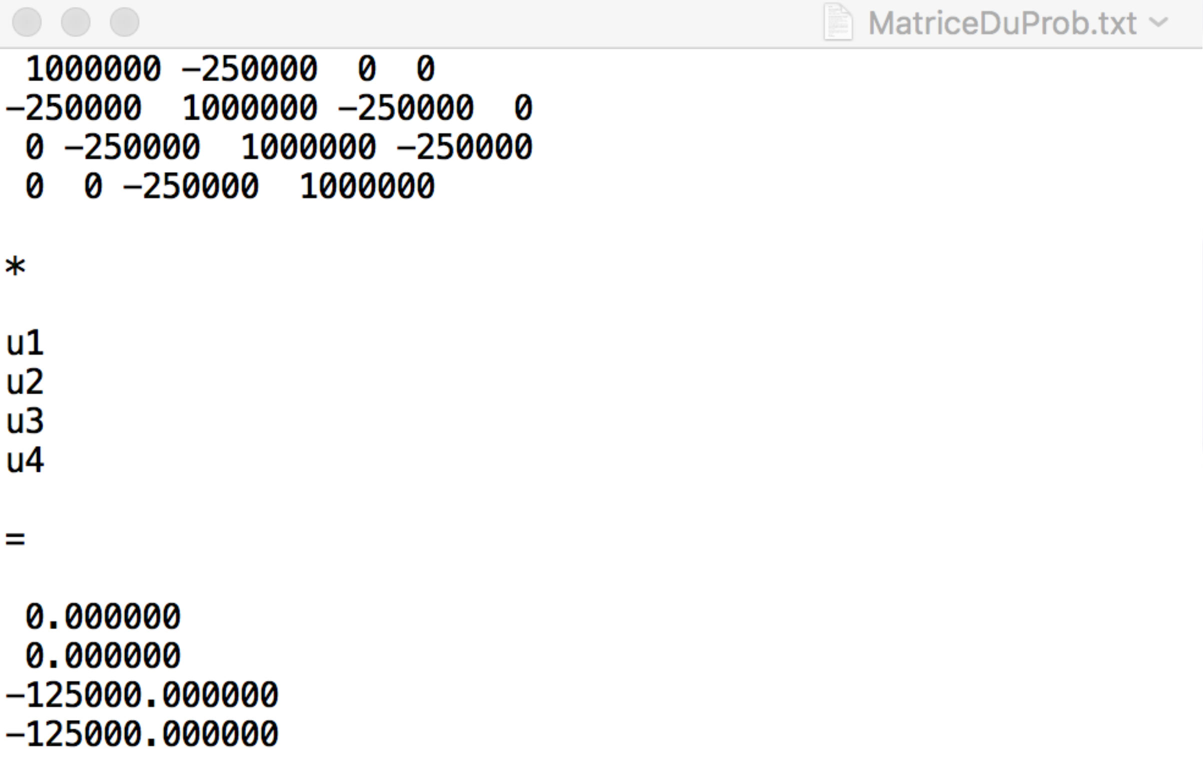
QUESTION 1: the prob function

- Modification of the prob.c file:
  - terminals
  - h
  - uT and uB



```
Arguments
=========
mx (input) - nombre de points dans la direction 1x sur la grille
my (input) - nombre de points dans la direction 1y sur la grille
(les valeurs de mx et my inférieures à 2 ne sont pas valides)
Lx (input) - longeur du condensateur dans la direction 1x
Ly (input) - longeur du condensateur dans la direction 1y
n  (output) - pointeur vers le nombre d'inconnus dans le système
ia (output) - pointeur vers le tableau 'ia' de la matrice A
ja (output) - pointeur vers le tableau 'ja' de la matrice A
a  (output) - pointeur vers le tableau 'a' de la matrice A
b  (output) - pointeur vers le tableau 'b'
h  (input) - le pas de discrétisation
bornes (input) - les bornes entre lesquelles est défini le condensateur
eps (input) - valeur de epsilon (la permittivité)
Q  (input) - la charge du condensateur
initializeRho - rendre rho nul (0 = Oui, 1 = Non)
(*rho)(double, double, double, double, double, int) (input) - fonction qui renvoie la
    valeur de rho en fonction de où on se situe sur le condensateur
uB (input) - Potentiel sur le bord du bas du condensateur compris entre les bornes données
uT (input) - Potentiel sur le bord du haut du condensateur compris entre les bornes données
```

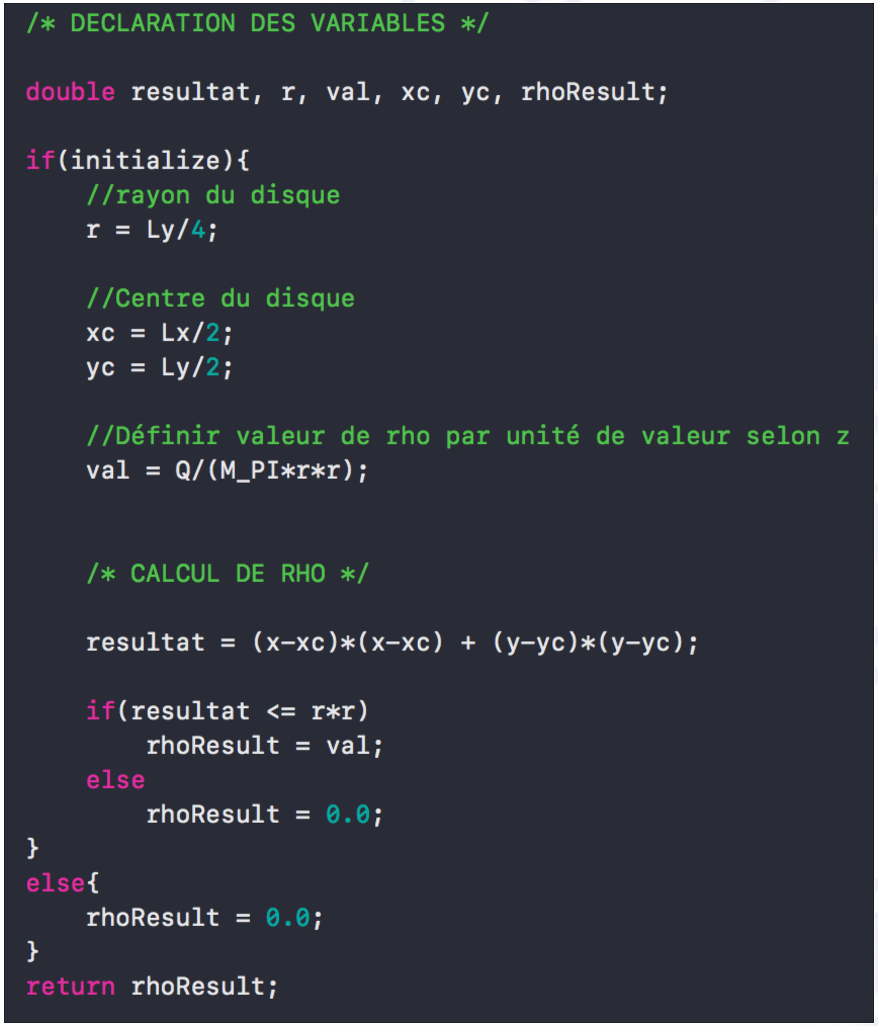
- affMatrice() function: transcribe the linear problem Ax = b into a file



```
 1000000 -250000   0   0
-250000   1000000 -250000   0
 0 -250000   1000000 -250000
 0   0 -250000   1000000

*

u1
u2
u3
u4

=

 0.000000
 0.000000
-125000.000000
-125000.000000
```

Note: This function allowed a better understanding of the problem at the beginning of the project.

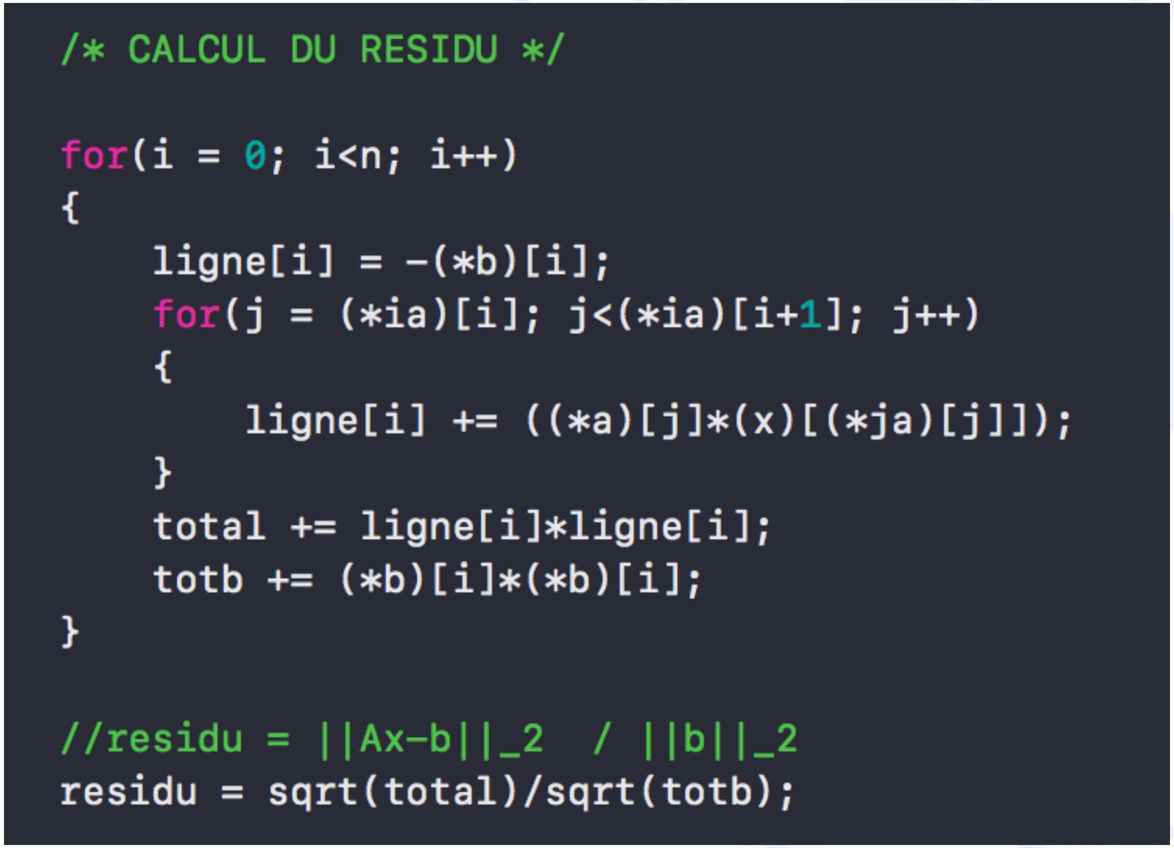
- The function that calculates rho:



```
/* DECLARATION DES VARIABLES */

double resultat, r, val, xc, yc, rhoResult;

if(initialize){
    //rayon du disque
    r = Ly/4;

    //Centre du disque
    xc = Lx/2;
    yc = Ly/2;

    //Définir valeur de rho par unité de valeur selon z
    val = Q/(M_PI*r*r);


    /* CALCUL DE RHO */

    resultat = (x-xc)*(x-xc) + (y-yc)*(y-yc);

    if(resultat <= r*r)
        rhoResult = val;
    else
        rhoResult = 0.0;
}
else{
    rhoResult = 0.0;
}
return rhoResult;
```

QUESTION 2: the residue

- The calculation of the residual is done using the following formula: $$\frac{\|A\mathbf{u}-\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$$

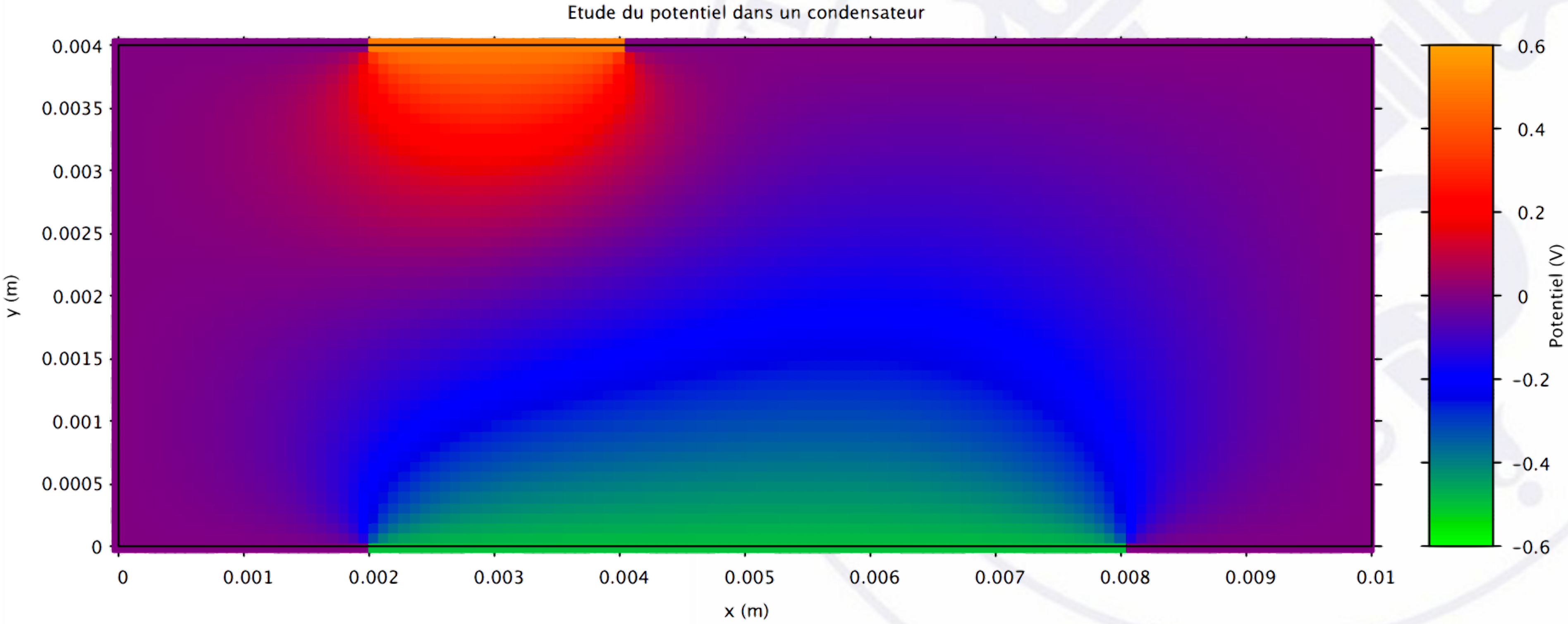


```
/* CALCUL DU RESIDU */

for(i = 0; i<n; i++)
{
    ligne[i] = -(*b)[i];
    for(j = (*ia)[i]; j<(*ia)[i+1]; j++)
    {
        ligne[i] += ((*a)[j]*(x)[(*ja)[j]]);
    }
    total += ligne[i]*ligne[i];
    totb += (*b)[i]*(*b)[i];
}

//residu = ||Ax-b||_2  / ||b||_2
residu = sqrt(total)/sqrt(totb);
```

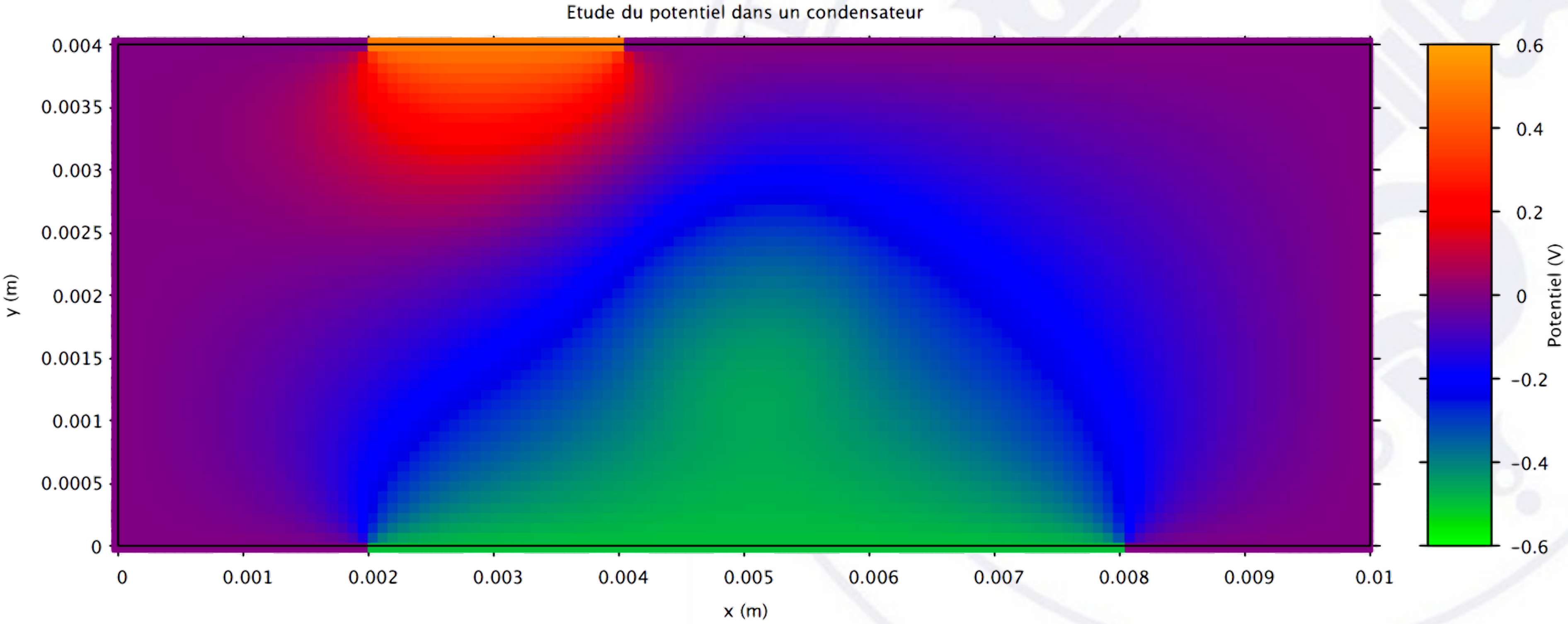
QUESTION 3: the potential

- Creation of a table containing the ≠ potential values
- Display of a graph representing it thanks to gnuplot, generated with files

- Potential in the capacitor when rho is 0



- Potential in the capacitor when rho is Q/S where Q = -7.446149e-11 C/m (calculated using question 5 with my = 50)



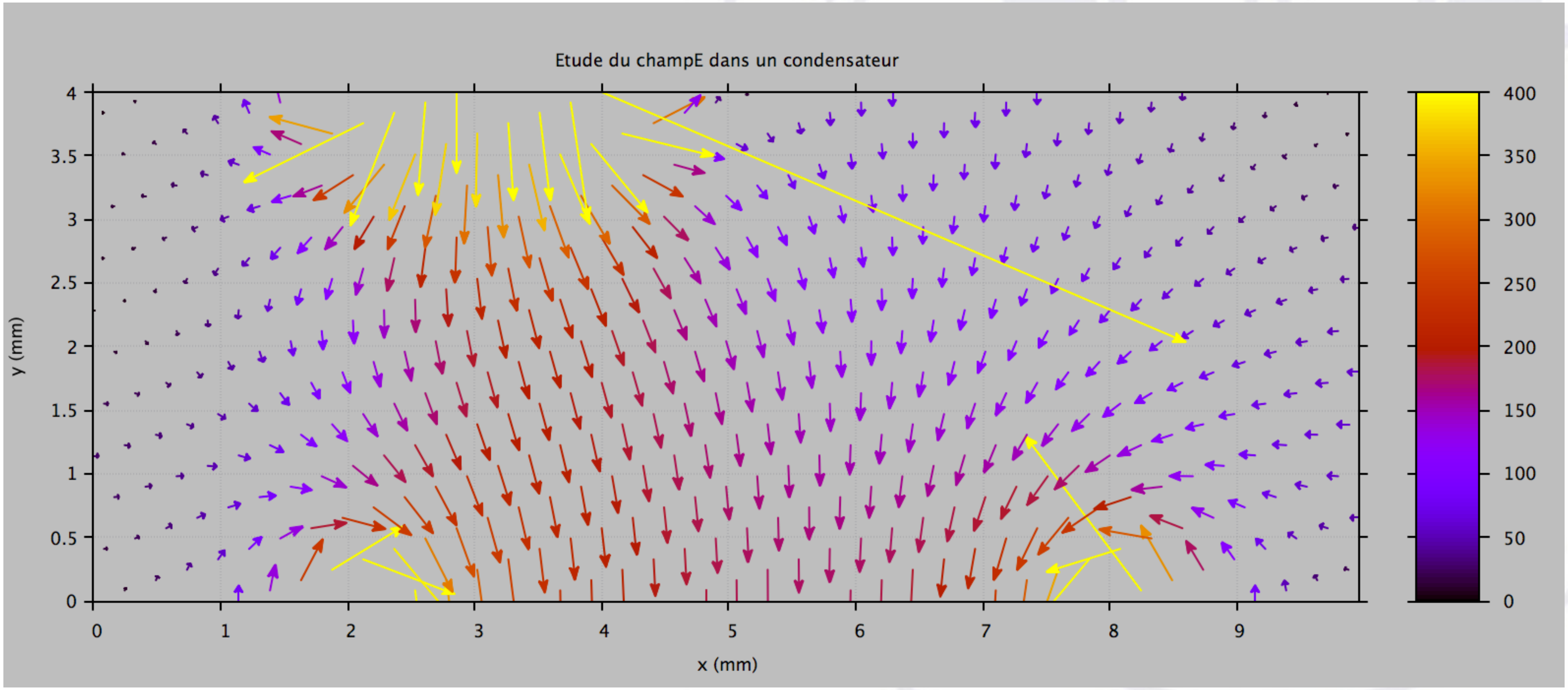
QUESTION 4: field E

- Creation of a double array containing [Ex, Ey]: arrayChE[mx*my][2]
- Ex and Ey are calculated using 3 different formulas:

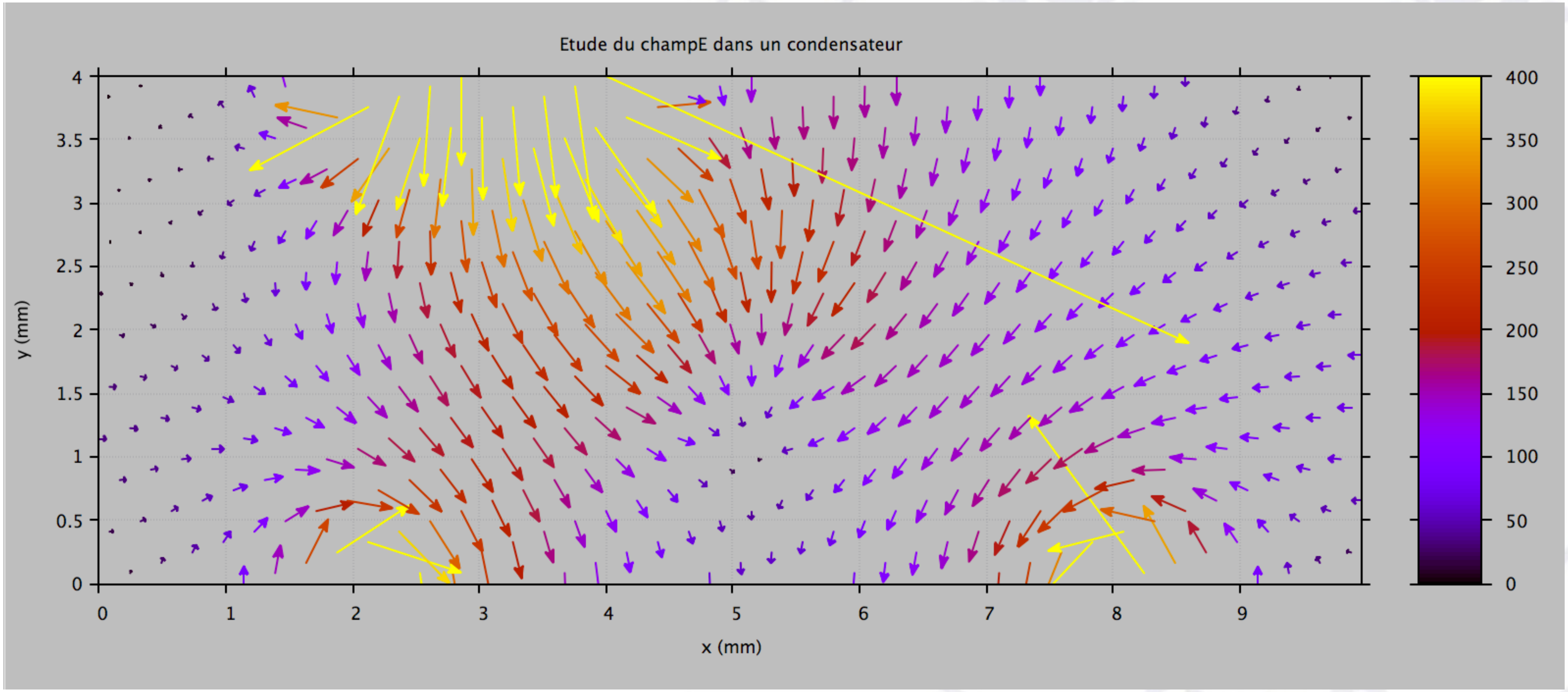
```
1. Progressive difference (when on a south or west edge):
dy y(x+h) - y(x)
--(x) = -------------
dx h
2. Regressive difference (when one is on a north or east edge):
dy y(x) - y(xh)
--(x) = -------------
dx h
3. Centered difference (when in the center of the capacitor):
dy y(x+h) - y(xh)
--(x) = -------------
dx 2h
```

- Display of a graph representing the E field in the capacitor using gnuplot, generated with files

- Electric field in the capacitor when rho is 0

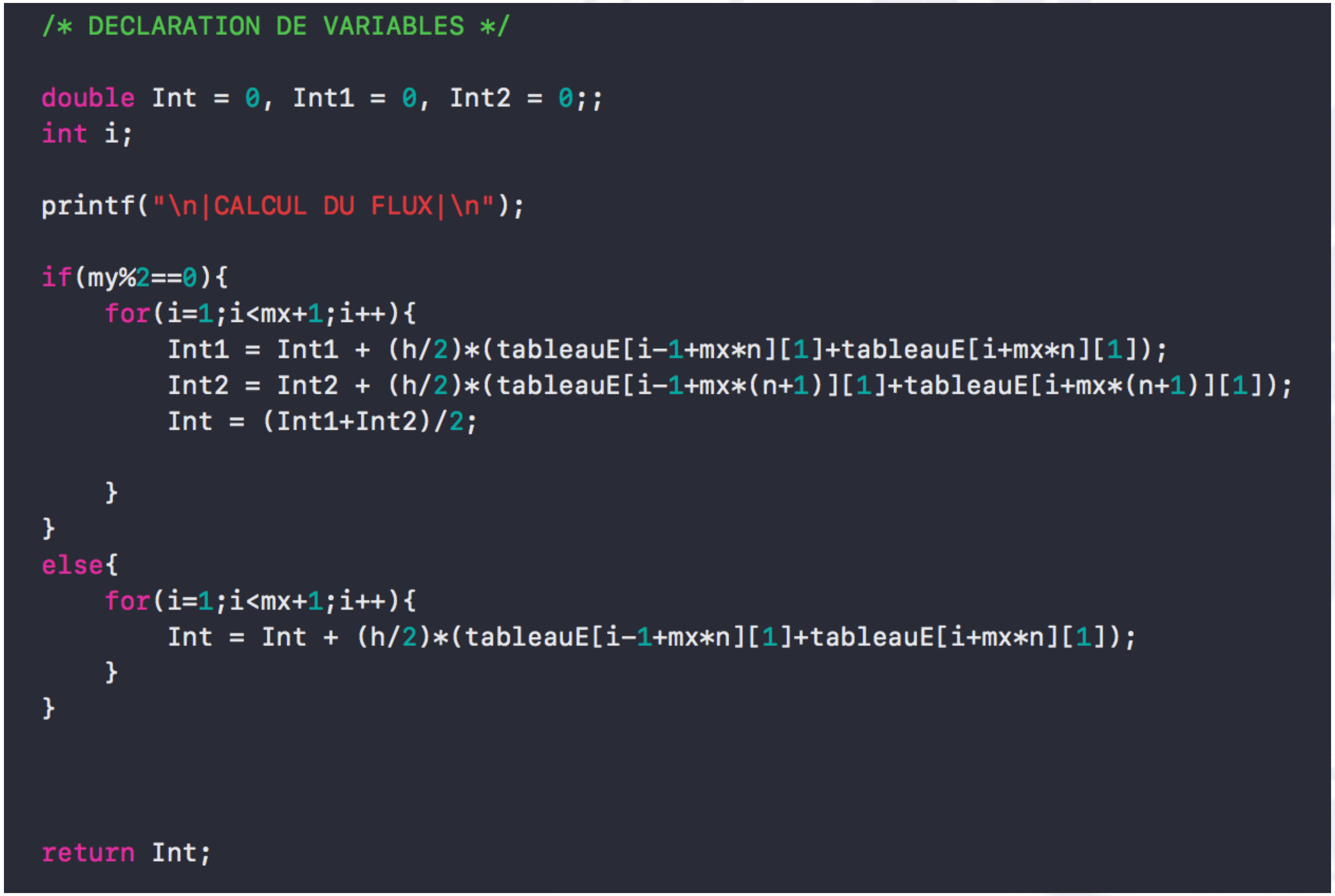


- Electric field in the capacitor when rho is Q/S where Q = -7.446149e-11 C/m (calculated using question 5 with my = 50)



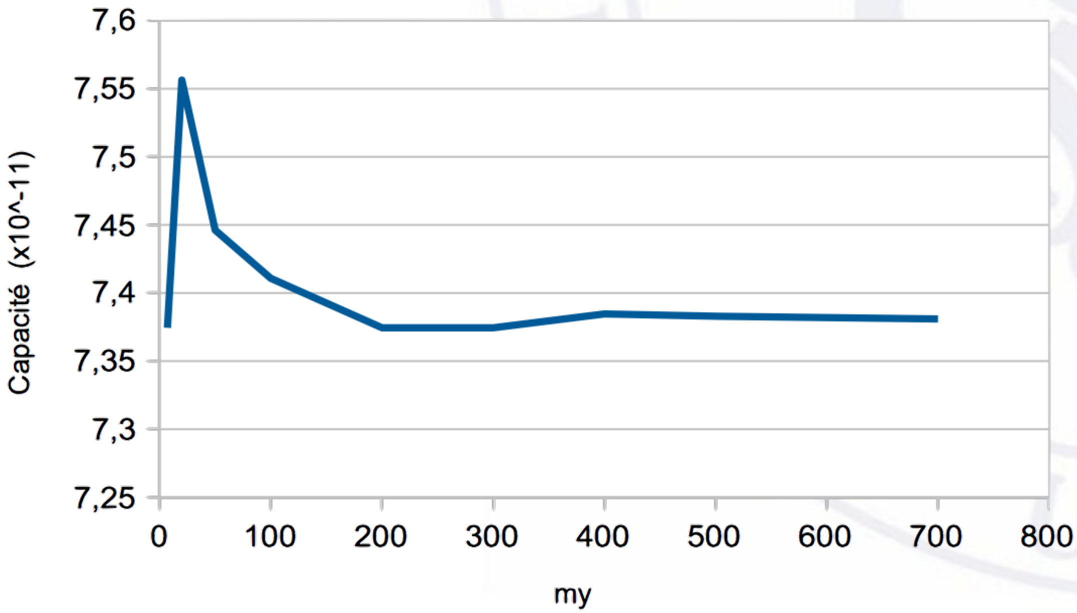
QUESTION 5: capacity

- The flux of E = integral(E.ds) = Q/eps.
- The trapezium method was used



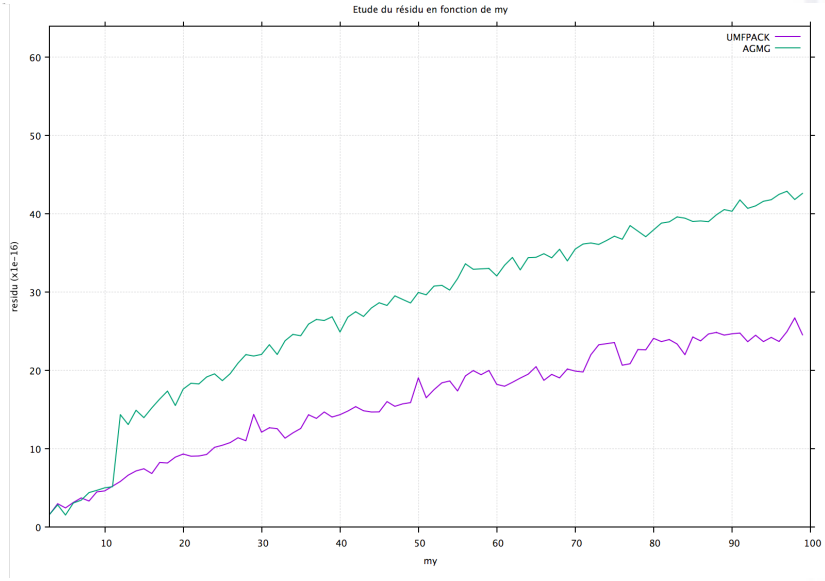
```
/* DECLARATION DE VARIABLES */

double Int = 0, Int1 = 0, Int2 = 0;;
int i;

printf("\n|CALCUL DU FLUX|\n");

if(my%2==0){
    for(i=1;i<mx+1;i++){
        Int1 = Int1 + (h/2)*(tableauE[i-1+mx*n][1]+tableauE[i+mx*n][1]);
        Int2 = Int2 + (h/2)*(tableauE[i-1+mx*(n+1)][1]+tableauE[i+mx*(n+1)][1]);
        Int = (Int1+Int2)/2;

    }
}
else{
    for(i=1;i<mx+1;i++){
        Int = Int + (h/2)*(tableauE[i-1+mx*n][1]+tableauE[i+mx*n][1]);
    }
}



return Int;
```

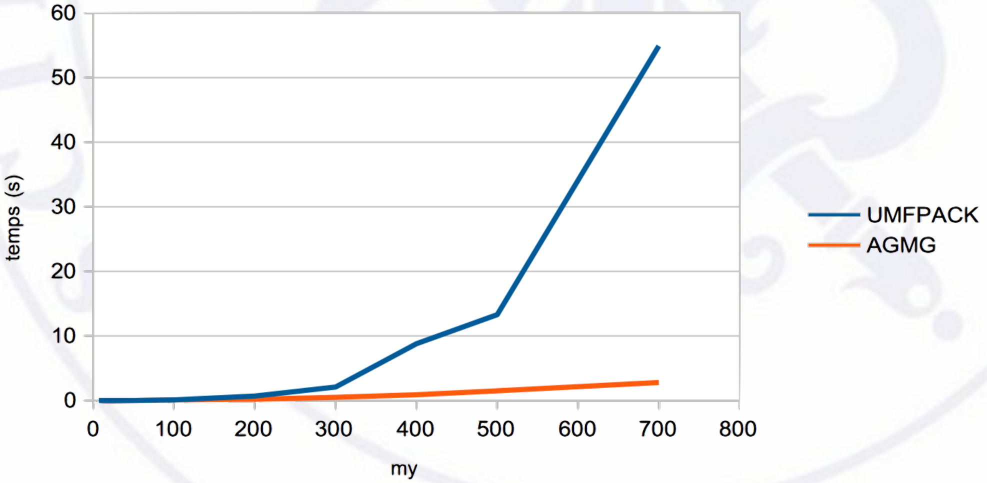
- Flux of E = integral(E.ds) = Q/eps -> Q = flux at center . eps
- The more h ↘ (my ↗), the more precise C is because E $_{glob}$ prop to h ^{2}
- C does not depend on V, therefore on Ut and on Ub either:
  - CV = Q
  - Q prop to flux therefore to E
  - E prop to grad(V)
  - So Q is linearly dependent on V



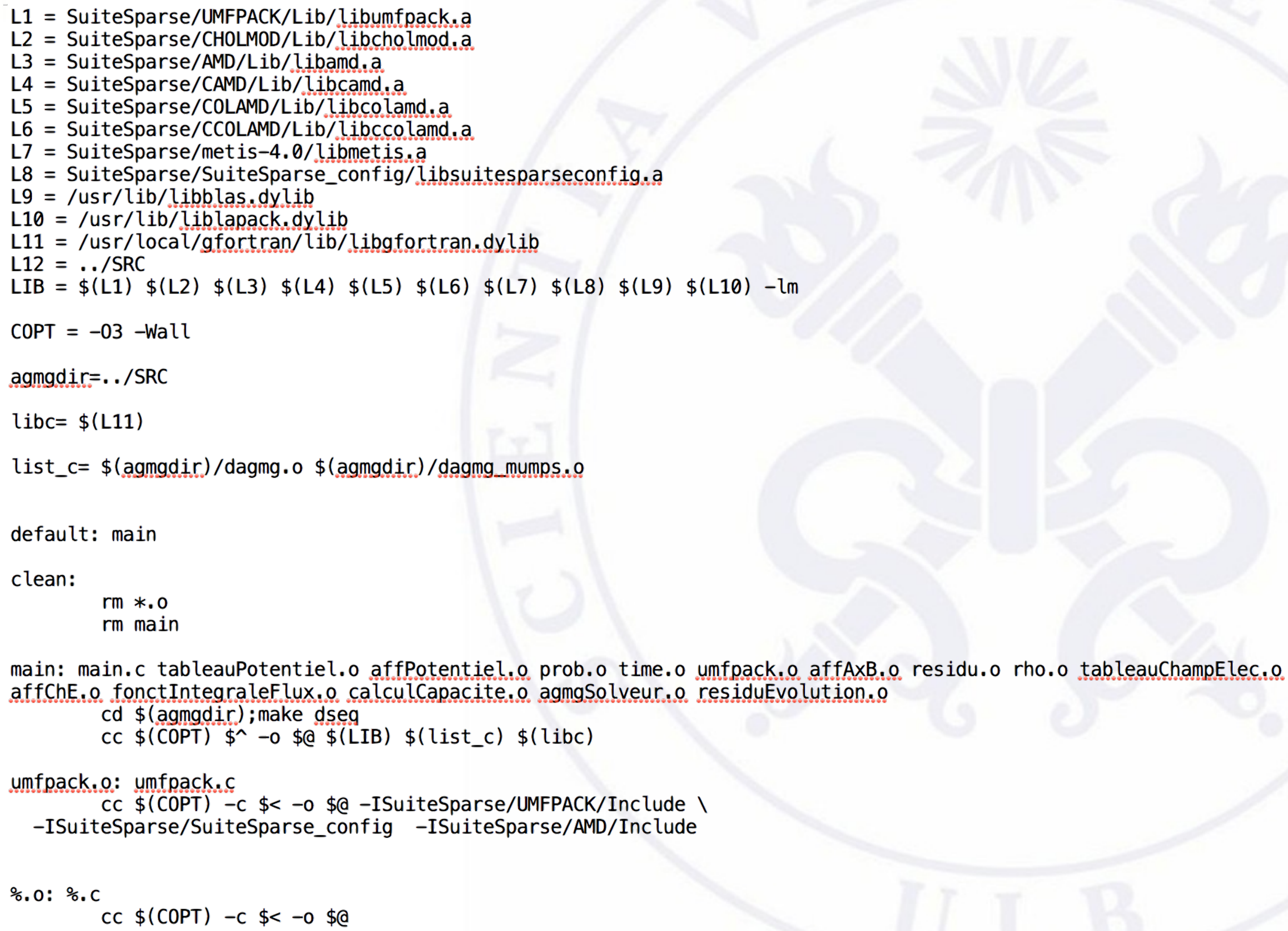
QUESTION 6: AGMG

- Comparison of times and residuals of the 2 solvers
- Residual and time as a function of my





The Makefile



```
L1 = SuiteSparse/UMFPACK/Lib/libumfpack.a
L2 = SuiteSparse/CHOLMOD/Lib/libcholmod.a
L3 = SuiteSparse/AMD/Lib/libamd.a
L4 = SuiteSparse/CAMD/Lib/libcamd.a
L5 = SuiteSparse/COLAMD/Lib/libcolamd.a
L6 = SuiteSparse/CCOLAMD/Lib/libccolamd.a
L7 = SuiteSparse/metis-4.0/libmetis.a
L8 = SuiteSparse/SuiteSparse_config/libsuitesparseconfig.a
L9 = /usr/lib/libblas.dylib
L10 = /usr/lib/liblapack.dylib
L11 = /usr/local/gfortran/lib/libgfortran.dylib
L12 = ../SRC
LIB = $(L1) $(L2) $(L3) $(L4) $(L5) $(L6) $(L7) $(L8) $(L9) $(L10) -lm

COPT = -O3 -Wall

agmgdir=../SRC

libc= $(L11)

list_c= $(agmgdir)/dagmg.o $(agmgdir)/dagmg_mumps.o


default: main

clean:
	rm *.o
	rm main

main: main.c tableauPotentiel.o affPotentiel.o prob.o time.o umfpack.o affAxB.o residu.o rho.o tableauChampElec.o affChE.o fonctIntegraleFlux.o calculCapacite.o agmgSolveur.o residuEvolution.o
	cd $(agmgdir);make dseq
	cc $(COPT) $^ -o $@ $(LIB) $(list_c) $(libc)

umfpack.o: umfpack.c
	cc $(COPT) -c $< -o $@ -ISuiteSparse/UMFPACK/Include \
  -ISuiteSparse/SuiteSparse_config  -ISuiteSparse/AMD/Include


%.o: %.c
	cc $(COPT) -c $< -o $@
```

Improvements

- Changed double board [Ex, Ey] -> 2 different boards -> performance improvement
- Remove menu, matrix display
- Remove unnecessary #includes
- Add in the gnuplot file of the electric field: set size ratio -1
- Remove bounds adjustment in prob.c file
- Not calculate capacity when rho is zero
- Reduce the number of arguments of some functions