Universitat Autònoma de Barcelona
Facultat de Ciències

Pràctica 1

Autors:

Gerard Lahuerta & Ona Sánchez

1601350 — 1601181

28 d'Octubre del 2022

Índex

1 Resolució apartat 1

Demostrem que $P'(t)=0$, on $P(t)=\int_0^1 u(x,t)dx$, mitjançant la següent expressió:

$$\frac{d}{dt}\int_a^b u(x,t)dx = Flux(b) - Flux(a) \tag{1}$$

Sabem, a més, que el sistema $u(x,t)$ defineix una circumferència de perímetre igual a 1, pel que $u(0,t)=u(1,t)$ (condició inicial de sistema); és a dir, $Flux(1)=Flux(0)$. (2)

Demostrem ara que $P'(t)=0$:

$$P(t)=\int_0^1 u(x,t)dx \Longrightarrow P'(t)=\frac{d}{dt}\int_0^1 u(x,t)dx \underset{\underset{(1)}{\uparrow}}{=} Flux(1)-Flux(0) \underset{\underset{(2)}{\uparrow}}{=}$$
$$= Flux(0)-Flux(0)=0 \Longrightarrow \boxed{\text{P'(t) = 0}}$$

Com que el recorregut de les partícules és una circumferència (per tant, una corba tancada), el nombre de partícules que passen per un punt sempre serà igual a les que surten del mateix (ja que la quantitat de partícules en la circumferència o en un punt de la mateixa serà constant per a un valor de t donat), per a qualsevol punt de l'interval $x\in[0,1]$ ja que és on està definida la circumferència, és 1-periòdica[1].

[1] Una funció és 1-periòdica si $f(x)=f(x+1)\ \forall x\in\mathbb{R}$.

2 Resolució apartat 2

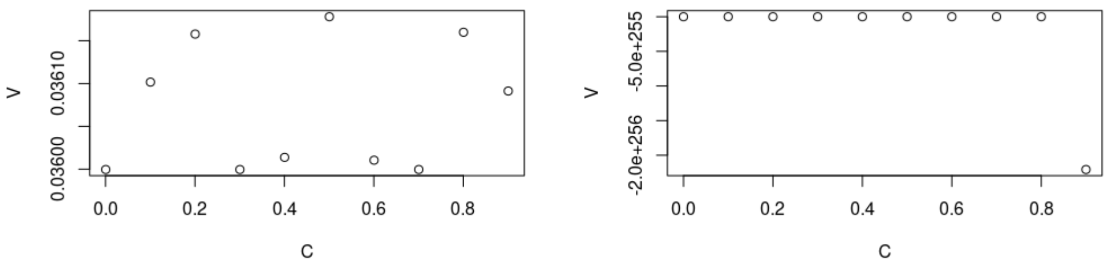
Exposem ara el codi programat per a obtenir la solució numèrica de la *EDP*:

```
u0 <- function(x) {pmax(-(x-0.2)*(x-0.8),0)}
cf <- function(x) {2 + cos(x * 2 * pi)}
M <- 500
dx <- 1/M
mu <- 1/2 #1/4
s <- 1
dt <- mu * dx ^ s
tfinal <- 1
U = c()
Unou = c()
for (m in 1:M) {
  U = c(U, u0(m*dx))
  Unou <- append(Unou, 0)
}

t = 0
res = c()
tim = c()
while (t <= tfinal) {
  for (m in 1:M) {
    if (cf(m) > 0) {
      if (m > 1) {
        Unou[m]=(1-dt/dx*(2*cf(m)-cf(m-1)))*U[m]+dt/dx*cf(m)*U[m-1]
      }
      else{
        Unou[m]=(1-dt/dx*(2*cf(m)-cf(M-1)))*U[m]+dt/dx*cf(m)*U[M-1]
      }
    }
    else{
      if (m < M) {
        Unou[m]=(1+dt/dx*(2*cf(m)-cf(m+1)))*U[m]-dt/dx*cf(m)*U[m+1]
      }
      else{
        Unou[m]=(1+dt/dx*(2*cf(m)-cf(0)))*U[m]-dt/dx*cf(m)*U[0]
      }
    }
  }
  U = Unou

  P = 0
  for (i in 1:M) {
    P = P + U[i]*dx
  }

  res = append(P, res)
  tim = append(t, tim)
  t = t + dt
}
k = floor(1/(dt*10))
comp = c(0:11)*k
C = c()
V = c()
for (i in 1:length(res)) {
  if (i \%in\% comp) {
    C = c(C,tim[i])
    V = c(V, res[i])
  }
}

plot(C, V)

xv <- (0:(M-1))*dx
cv <- cf(xv)
A <- matrix(rep(0,M*M), nrow=M)
for(i in 1:M) {
  ifelse( cv[i]>0,
          A[i,i] <- 1 - dt/dx * (2*cv[i]-cv[ifelse(i-1>=1,i-1,M)]),
          A[i,i] <- 1 + dt/dx * (2*cv[i]-cv[ifelse(i+1<=M,i+1,1)]) )
}
for(i in 2:M) {
  ifelse( cv[i]>0,
          A[i,i-1] <- cv[i]*dt/dx,
          A[i,i-1] <- 0 )
}
for(i in 1:(M-1)) {
  ifelse( cv[i]>0,
          A[i,i+1] <- 0,
          A[i,i+1] <- -cv[i]*dt/dx )
}
if(cv[1]>0) {
  A[1,M] <- cv[i]*dt/dx
  A[M,1] <- 0
} else {
  A[1,M] <- 0
  A[M,1] <- -cv[i]*dt/dx
}

vapsA <- eigen(A)$values
plot(as.complex(vapsA), xlim=c(-1.5,1.5),ylim=c(-1.5,1.5),pch=20, xlab="Re", ylab="Im")
lines(exp(-(0+1i)*seq(0, 2*pi, by=0.1)), col="purple")
```

3 Resolució apartat 3

Representem ara per a temps $t \in \{0, 0.1, \cdots 0.9, 1\}$ el valor de P prenent $\Delta t = \frac{\Delta x}{4}$ i $\Delta t = \frac{\Delta x}{2}$.
Els paràmetres de la simulació són:

- $u0(x) = max(-(x-0.2)(0.8-x); 0)$
- $c(x) = 2 + cos(2x)$
- $M = 500$
- $tfinal = 1$
- $\Delta x = \frac{1}{M}$



Valor de P amb $\Delta t = \frac{\Delta x}{4}$ Valor de P amb $\Delta t = \frac{\Delta x}{2}$

Figura 1: Gràfic del valor de P en funció del temps Δt

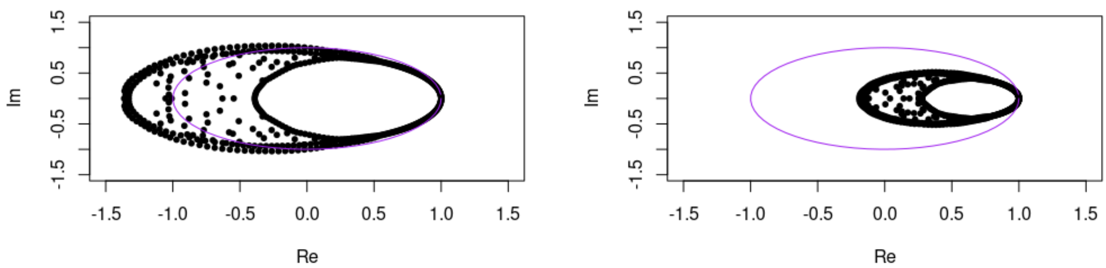
S'observa com per a $\Delta t = \frac{\Delta x}{2}$, P obté valors molt elevats i no aproxima bé els valors reals, per contra de $\Delta t = \frac{\Delta x}{4}$ que oscil·la els valors aproximats al voltant del 0.

Concluïm doncs que el mètode amb $\Delta t = \frac{\Delta x}{2}$ no convergeix al valor real, per contra del $\Delta t = \frac{\Delta x}{4}$ que si ho fa.

Els valors obtinguts al mètode, quan $\Delta t = \frac{\Delta x}{4}$ tenen sentit amb els resultats obtinguts a l'apartat 1 ja que estan aproximant una recta sense pendent propera al 0; pel que els valors reals deuen ser constants en el temps (resultat de l'apartat 1).

4 Resolució apartat 4

Exposem ara els valors propis de la matriu definida anteriorment al programa com A per als dos valors de Δt estudiats:



Valors propis amb $\Delta t = \frac{\Delta x}{2}$

Valors propis amb $\Delta t = \frac{\Delta x}{4}$

Figura 2: Gràfic dels valors propis de la matrius A

S'observa com tots els valors propis de la matriu tenen norma menor a 1 si $\Delta t = \frac{\Delta x}{4}$ i que alguns son major que 1 si $\Delta t = \frac{\Delta x}{2}$.

Podem doncs conjecturar que si la norma dels valors propis de la matriu és menor o igual a 1, el mètode convergeix al valor real de la EDP, però si no es compleix la condició, el mètode no convergirà als valors reals.