

travail Pracma

vincent obertelli

12/7/2020

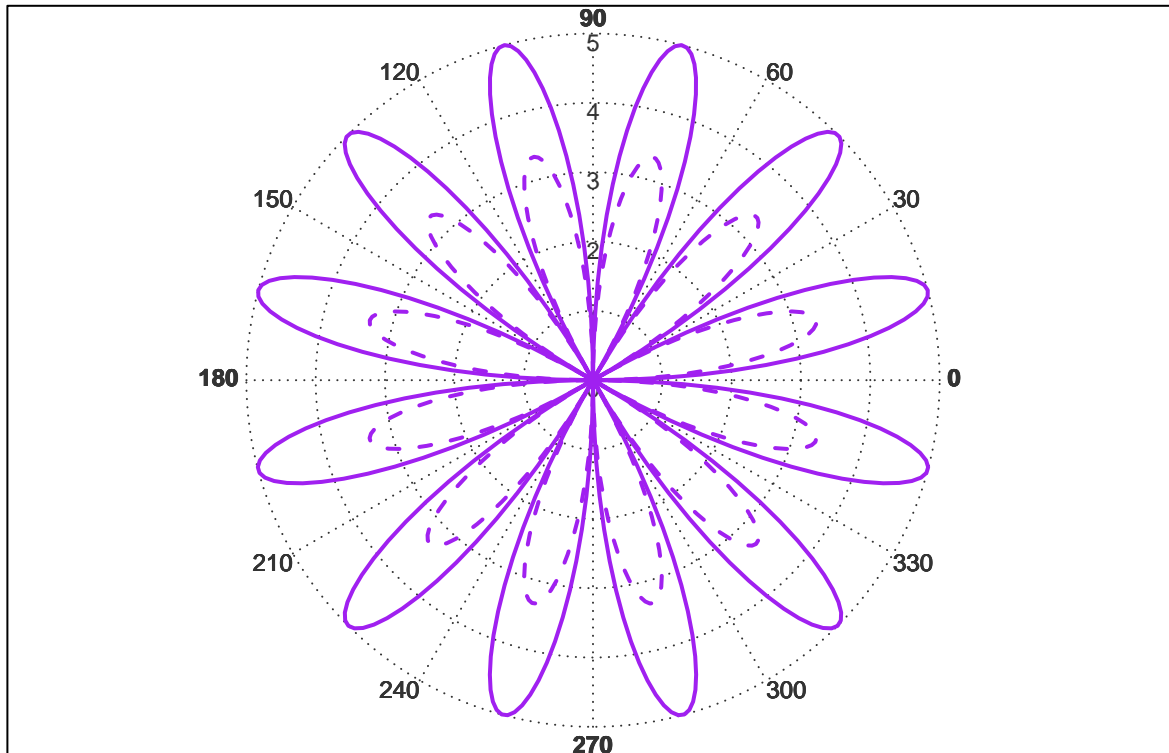
Maintenant nous allons voir la librairie Pracma. Cette librairie propose un large panel de fonction, allant de l'analyse numérique en passant par différentes équations (algèbre numéraire, optimisation numérique, ...)

Pour commencer, nous allons installer la bibliothèque : `install.packages("pracma")`

Pour commencer, voila ce qui est possible en utilisant la librairie Pracma. On peut travailler sur des problèmes mathématiques complexes et proposer des visuels pour mieux les comprendre.

```
library(pracma)
t=seq(0,2*pi, length=360)
f=function(t){5*sin(6*t)}
par(mar=c(1,1,2,1))
polar(t,f(t),grcol="gray20",bxcol="black",col="purple",
lwd=2,main=expression(5*sin(6*theta) ~ ~"and" ~ ~
2(5*sin(6*theta))/3))
polar(t,2*f(t)/3,col="purple",lty=2,lwd=2,add=TRUE)
```

$5\sin(6\theta)$ and $2(5\sin(6\theta))/3$



Maintenant, nous allons voir les fonctions de base de cette librairie :

abm3pc. Cette fonction est une fonction d'approche prédictive. Elle contient les arguments suivants :

f, fonction différentiel

a, b, l'intervalle

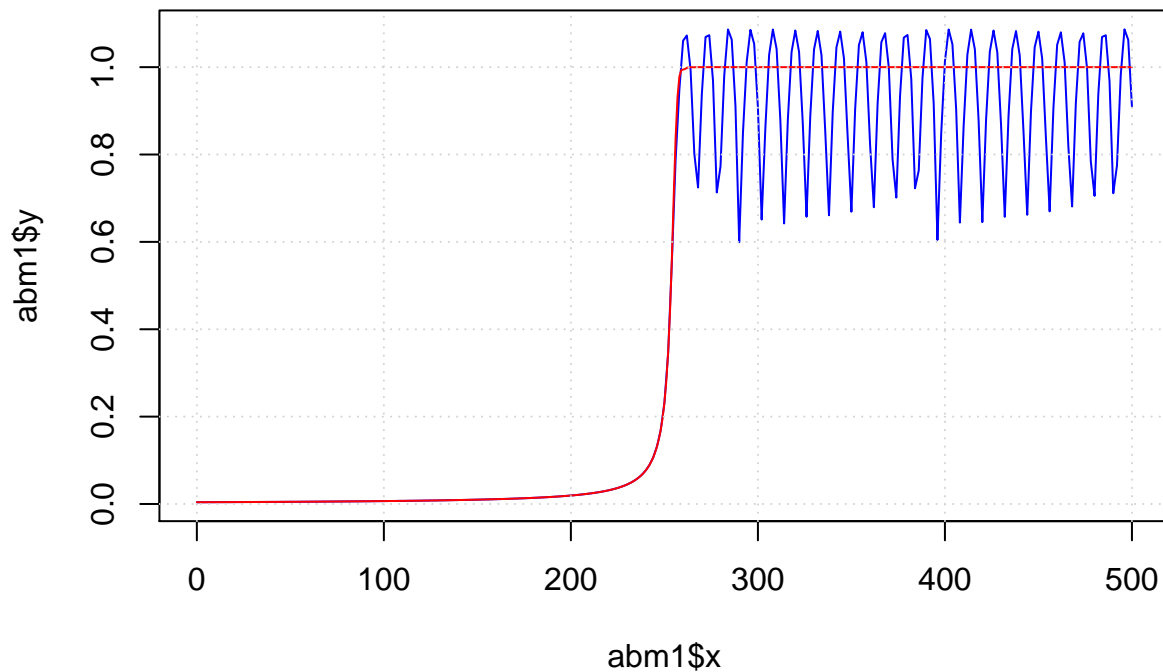
Y0, valeur de début

n, nombre de valeur

..., paramètres additionnels

Maintenant, voici la fonction : `abm3pc(f, a, b, y0, n = 50, ...)`

```
library(pracma)
# y' = y^2 - y^3, y(0) = d, 0 <= t <= 2/d, d = 0.01
f <- function(t, y) y^2 - y^3
d <- 1/250
abm1 <- abm3pc(f, 0, 2/d, d, n = 1/d)
abm2 <- abm3pc(f, 0, 2/d, d, n = 2/d)
## Not run:
plot(abm1$x, abm1$y, type = "l", col = "blue")
lines(abm2$x, abm2$y, type = "l", col = "red")
grid()
```



Nous allons désormais voir la fonction **agmean**. Cette fonction permet de calculer la moyenne.

On l'utilise de cette manière : `agmean(a, b)`.

`a` et `b` sont des vecteurs de nombres réels ou complexes de même longueur.

Voilà un exemple de comment utiliser cette fonction :

```
library(pracma)
agmean(1, sqrt(2))$agm - 0.834626841674073186
```

```
## [1] 0.3635134
```

maintenant, nous allons voir **accumarray**. Cette fonction permet de grouper des éléments d'un data set et d'appliquer une fonction pour chaque groupe.

Voilà à quoi ressemble l'expression : `accumarray(subs, val, sz = NULL, func = sum, fillval = 0)`

subs, vecteur ou matrice

val, vecteur numérique

sz, taille du array

func, la fonction à appliquer aux nombres vecteurs

fillval, valeur utilisées pour remplir le array

a, valeur numérique

```
library(pracma)
## exemple
val = 101:105
subs = as.matrix(c(1, 2, 4, 2, 4))
accumarray(subs, val)
```

```
##      [,1]
## [1,] 101
## [2,] 206
## [3,]   0
## [4,] 208
```

Enfin, nous allons voir les fonctions **and** et **or**. Comme leur noms l'indique, il s'agit de “et” et “ou”.

```
library(pracma)
A <- matrix(c(0.5, 0.5, 0, 0.75, 0,
              0.5, 0, 0.75, 0.05, 0.85,
              0.35, 0, 0, 0, 0.01,
              0.5, 0.65, 0.65, 0.05, 0), 4, 5,
            byrow=TRUE)

B <- matrix(c(0, 1, 0, 1, 0,
              1, 1, 1, 0, 1,
              0, 1, 1, 1, 0,
              0, 1, 0, 0, 1), 4, 5, byrow=TRUE)

and(A, B)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   0   1   0   1   0
## [2,]   1   0   1   0   1
## [3,]   0   0   0   0   0
## [4,]   0   1   0   0   0
```

```
or(A, B)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   1   1   0   1   0
## [2,]   1   1   1   1   1
## [3,]   1   1   1   1   1
## [4,]   1   1   1   1   1
```