Analyse de pracma par Jean Souris

Wenjun ZHAO

12/22/2020

I. Introduction

Vous pouvez retrouver mon travail sur mon Github, **https://github.com/claudia0524/PSBX**.

Pour ce dossier, nous avons étudier le travail de Jean Souris, étudiant en M2DM au sein de PSB.

Nous avons trouvé son dossier sur son Github : **https://github.com/jeansouris/PSBX**

Maintenant, nous allons établir mes 5 critères d'évaluations, qui seront les mêmes pour tout les dossiers que j'ai vais étudier pour ce devoir:

Les cinq critères d'évaluation

1. Rmd se comporte bien à l'exécution

2. Les aspects intéressant, didactique, complet

3. La qualité Rmarkdown, la qualité de l'écriture

4. Didactique, conformité aux exigences vues plus haut et comporte du calcul symbolique et du calcul numérique

5. La qualité du LaTeX et des illustrations, la qualité de l'écriture, le choix des ressource internet, la compréhension personnelle des concepts

II. Synthèse du travail en question

Pracma signifie "Practical Numerical Maths Functions" et est issue de la fonction "MATLAB". Cette fonction est utilisée pour exprimer des langages mathématiques tels que des analyses numériques, le développement de fonctions afin de trouver leurs solution(s), voire même obtenir leur représentation numérique.

Dans cette démonstration, nous allons nous intéresser aux fonctions d'Einstein ; célèbre physicien et mathématicien que tout le monde connait. En revanche, je ne connais pas les raisons pour lesquelles les fonctions que je vais vous ˆrésenter ci-dessous se nomment ainsi.(Souris, n.d.)

Nous allons maintenant étudier le code de Jean Souris dans le chapitre suivant.

III. Extrait commenté des parties de code

```
library(pracma)
```

Notre première fonction "y1":

Expression de notre "x1" :

Ensuite, pour les 2 premières fonctions qui vont suivre, nous allons définir un seul et même "x" :

```r
x1 <- seq(-4, 4, len = 300)
```

Nous venons donc de définir un environnement pour x ; notre “x” est donc compris sur l’intervalle allant de -4 à 4 tel que

$$x \in [-4; 4]$$

ou encore

$$-4 \leq x \leq 4$$

Je suppose que le “len” se trouvant dans la sequence correspond au nombre de points composant la courbe que vous allez voir, car en diminuant ce chiffre, la courbe devenait obtu, mais cette information est à vérifier.Après l’observation de mon fichier en classe, il semblerait que ce soit correcte ; en rajoutant que sur l’intervalle défini, la fonction est représentée par 300 valeurs différentes, défini donc grâce au “len” étant équivalent à 300.

Expression de notre “y1” :

Nous allons maintenant définir notre “y1” pour la première fonction :

```r
y1 <- einsteinF(1, x1)
```

Composition de notre formule y :

Ce “y1” se compose, comme toutes les formules que nous allons utiliser, de la base de donnée “einsteinF”, qui, si j’ai bien compris, possède les 4 formules des fonctions que nous allons voir dans cette démonstration.Ici, le “1” qui suit “einsteinF” correspond au numero de la fonction que nous allons utiliser ; donc la première dans notre cas. En second argument, nous pouvons voir “x1” que nous avons défini plus haut.

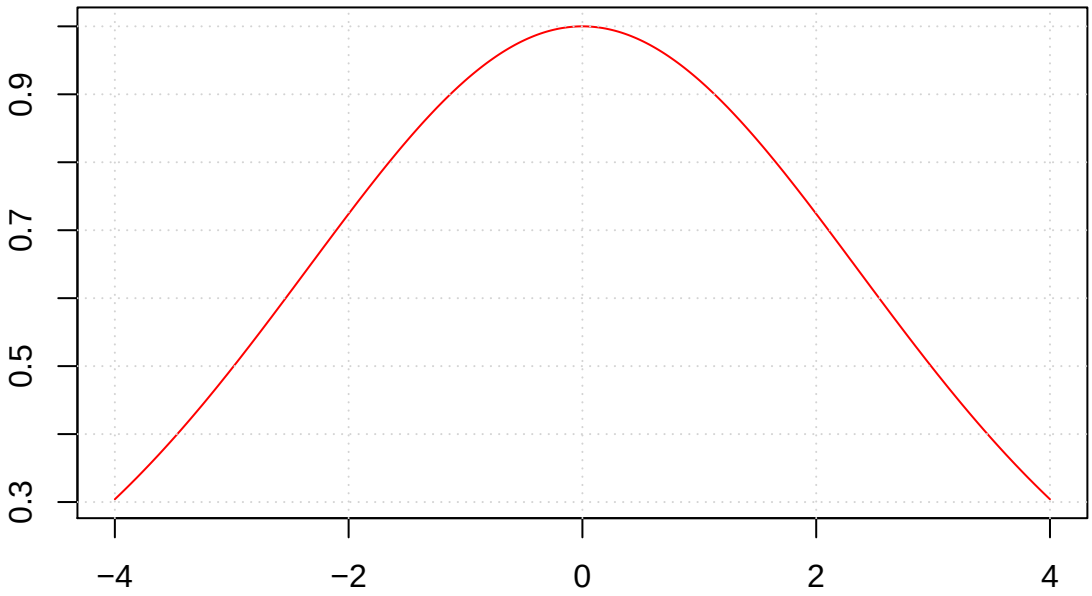
Cette fonction se présente sous cette forme en version écrite :

$$E1(x) = \frac{x^2 e^x}{(e^x - 1)^2}$$

Représentation de notre “y1” :

Sa représentation graphique est de cette forme :

```r
plot(x1, y1, type = "l", col = "red",
     xlab = "", ylab = "", main = "Fonction d'Einstein 1 E1(x)")
grid()
```



Explication des paramètres utilisés :

Maintenant étudions comment est écrite cette courbe (ce même modèle se répètera pour toutes les prochaines fonctions)

plot() :

Les arguments chiffrés : Premièrement, nous avons la fonction plot(), qui nous défini dans un premier temps sur quel intervalle va se définir notre courbe, et qui accessoirement la trace ; donc naturellement "x1" et "y1".

Modification visuelle de notre courbe : Dans un second temps, nous définirons comment nous voulons que la courbe s'affiche. Nous utilisons donc "l" pour la définir en tant que ligne. Si nous voulions des points, nous aurions pu utiliser "p" par exemple. Ensuite vient la couleur de cette courbe, défini par "col = 'red' ".

Enfin, pour définir les noms de nos axes, nous pouvons utiliser"xlab" et "ylab", qui dans cet exemple ne sont pas définis mais que nous pouvons renommer à notre guise. Puis, pour donner un titre à notre graphique, nous utilisons la fonction "main".

grid() : Afin d'avoir une meilleure idée des echelles ou des coordonnées de certains points, nous utilisons la fonction grid() afin de tracer une grille.

Etude de la courbe E(1) :

En regardant la représentation graphique de notre courbe, nous pouvons constater qu'elle a la forme d'une parabole inversée, avec pour axe de symétrie 0 pour y y étant égal à

Notre seconde fonction "y2":

Maintenant, passons à la seconde fonction d'Einstein.

Le "y" que nous allons utiliser se présente sous cette forme, avec "eisteinF" suivi d'un 2 pour appliquer la seconde fonction si vous avez bien suivi.

```
y2 <- einsteinF(2, x1)
```

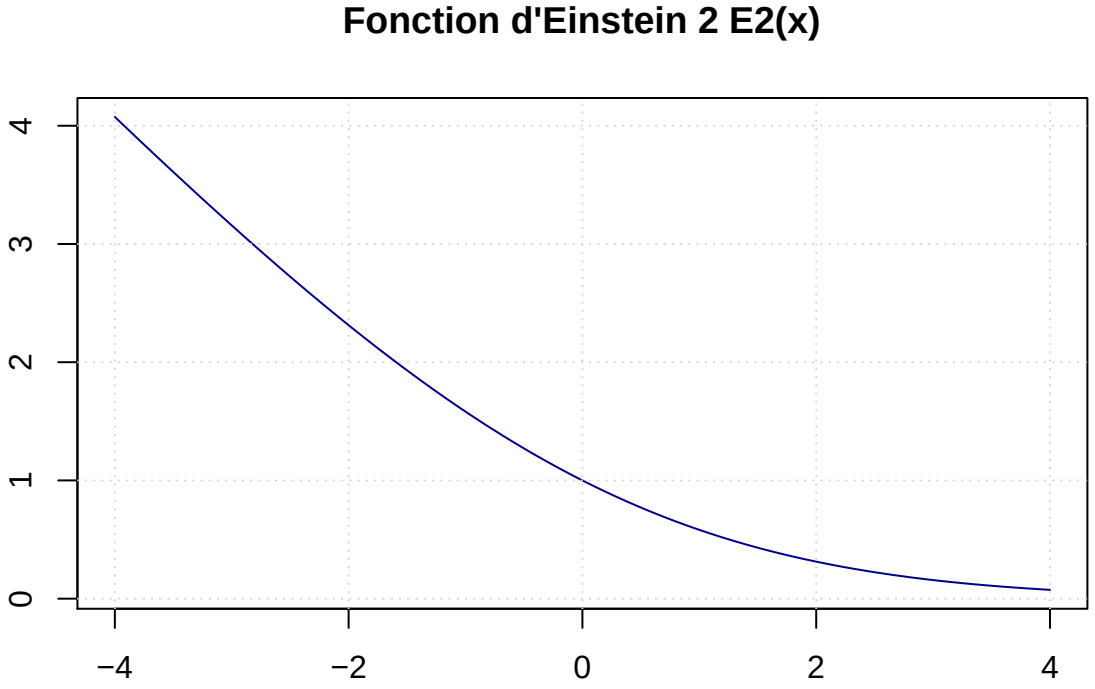
Cette fonction se présente sous cette forme en version écrite :

$$E2(x) = \frac{x}{e^x - 1}$$

Représentation de notre "y2" :

Et sa représentation graphique est de cette forme :

```
plot(x1, y2, type = "l", col = "darkblue",
     xlab = "", ylab = "", main = "Fonction d'Einstein 2 E2(x)")
grid()
```



Grâce à cette représentation graphique, nous pouvons voir que la limite de la fonction quand x tend vers l'infini.

Expression de notre "x3" :

Maintenant, passons aux 2 dernières fonctions, et définissons donc un nouvel "x" que l'on nommera "x3"

```
x3 <- seq(0, 5, len =1000)
```

Ici, notre axe des abscisses sera défini sur l'intervalle

$$[0; 5]$$

Notre Troisième fonction "y3":

Voici notre "y3" :

```
y3 <- einsteinF(3, x3)
```

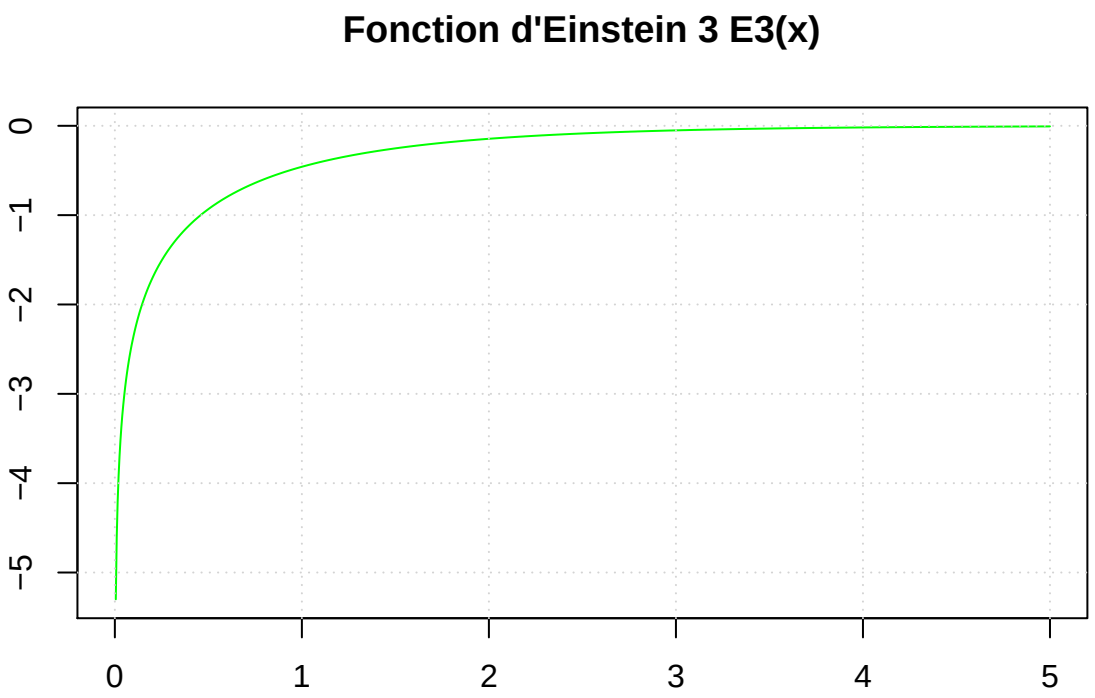
Cette fonction se présente sous cette forme en version écrite :

$$E3(x) = ln(1 - e^{-x})$$

Représentation de notre “y3” :

Et sa représentation graphique est de cette forme :

```
plot(x3, y3, type = "l", col = "green",
     xlab = "", ylab = "", main = "Fonction d'Einstein 3 E3(x)")
grid()
```



Grâce à cette représentation graphique, nous pouvons voir que la limite de la fonction quand x tend vers l’infini.

Notre dernière fonction “y4”:

Voici la toute dernière fonction d’Einstein représenté par “y4” :

```
y4 <- einsteinF(4, x3)
```

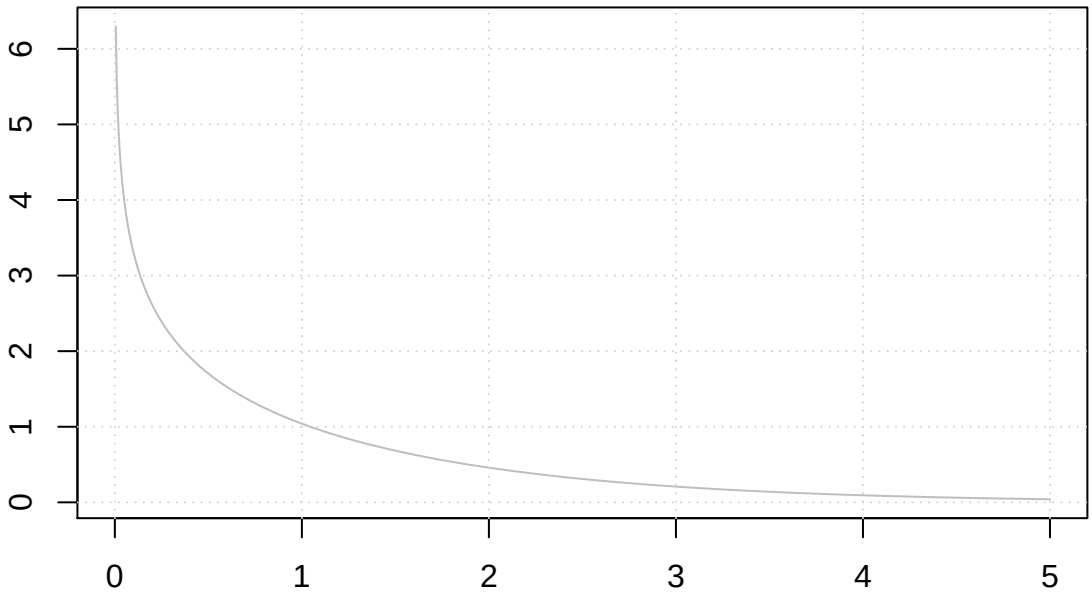
Qui se présente sous cette forme en version écrite :

$$E4(x) = \frac{x}{e^x - 1} - ln(1 - e^{-x})$$

Représentation de notre “y3” :

Et voici sa représentation graphique :

```
plot(x3, y4, type = "l", col = "grey",
     xlab = "", ylab = "", main = "Fonction d'Einstein 4 E4(x)")
grid()
```



IV. Evaluation du travail en question

1. Rmd se comporte bien à l'exécution (4/4)

2. Les aspects intéressant, didactique, complet (3/4)

3. La qualité Rmarkdown, la qualité de l'écriture (4/4)

4. Didactique, conformité aux exigences vues plus haut et comporte du calcul symbolique et du calcul numérique (3/4)

5. La qualité du LaTeX et des illustrations, la qualité de l'écriture, le choix des ressource internet, la compréhension personnelle des concepts (3/4)

V. Conclusion

En général, ce tracail de Jean Souris exécuse bien dans l'environement de R. Et il nous explique clairement comment fonctionner le package pracma dans un rmd. L'aspect est intéressant, complet et propre. La qualité Rmarkdown et la qualité de l'écriture sont ainsi bien. Didactique, conformité aux exigences vues plus haut. La qualité du LaTeX fonctionne bien aussi, et il peut montrer le but principal de package de pracma. Le mieux point est qu'il présente des ressources internet. Il utilise des calcul symbolique et du calcul numérique.

VI. Bibliographie

Souris, Jean. n.d. "Travail de Pracma." https://github.com/jeansouris/PSBX.