

Csv avec Scilab

Comment extraire les valeurs d'un fichiers de mesures .csv pour tracer la courbe caractéristique de ces valeurs avec Scilab.

On dispose d'un fichier de données test1.csv au format Coma Separated Values, valeurs séparées par des virgules, généré lors d'une expérience de physique par un logiciel d'EXAO. Les données sont en colonnes, les séparateurs de colonnes sont des virgules et le séparateurs décimaux des points, comme il est de coutume dans le monde anglo-saxon.

Voici le début de ce fichier qui compte en réalité 2483 lignes, résultat de 2483 mesures :

```
-4.93200030E-3,2.0000000,  
-4.92800027E-3,2.1599998,  
-4.92400024E-3,2.2400000,  
-4.92000021E-3,2.3199999,  
-4.91600018E-3,2.4800000,  
-4.91200015E-3,2.4800000,  
-4.90800058E-3,2.5599999,
```

...

La première colonne représente le temps, et la seconde la tension électrique.

Un examen attentif montre que l'interface de mesure a mis une virgule après la seconde colonne, ce qui fait que notre fichier est un tableau de 2483 lignes et 3 colonnes.

La troisième colonne est simplement vide.

-4.93200030E-3	2.0000000	
-4.92800027E-3	2.1599998	
-4.92400024E-3	2.2400000	
-4.92000021E-3	2.3199999	
-4.91600018E-3	2.4800000	
-4.91200015E-3	2.4800000	
-4.90800058E-3	2.5599999	

Tout d'abord, il faut permettre à Scilab de lire le fichier [test1.csv](#).

Par facilité, il a été placé avec d'autres fichiers csv de résultats de mesures dans un répertoire aaacsv facilement repérable.

Le chemin pour y accéder est donc :

```
/home/mesfichiers/aaacsv/test1.csv
```

C'est une adresse de fichier sous GNU/Linux.

Pour les autres systèmes, le principe est toujours le même : savoir où l'on range ses fichiers et savoir en écrire le chemin d'accès.

On peut maintenant utiliser la fonction `csvRead` :
`M=csvRead("/home/jcm/aaacsv/test1.csv")`

On obtient un tableau, une matrice, à trois colonnes et 2483 lignes.
En voici le début :

```
M =  
- 0.0049320003  2.          Nan  
- 0.00492800027 2.1599998 Nan  
- 0.00492400024 2.24       Nan  
- 0.00492000021 2.3199999 Nan  
- 0.00491600018 2.48       Nan  
- 0.00491200015 2.48       Nan  
...
```

La troisième colonne contient des variables logiques par défaut, mais ce n'est pas gênant. Nous n'allons pas utiliser cette colonne.

Pour avoir le vecteur temps, on extrait la première colonne de valeurs.

```
->t=M(:,1)
```

```
t =  
- 0.0049320003  
- 0.00492800027  
- 0.00492400024  
- 0.00492000021  
- 0.00491600018  
- 0.00491200015  
- 0.00490800058  
etc ...
```

De même pour avoir le vecteur des tensions électriques.

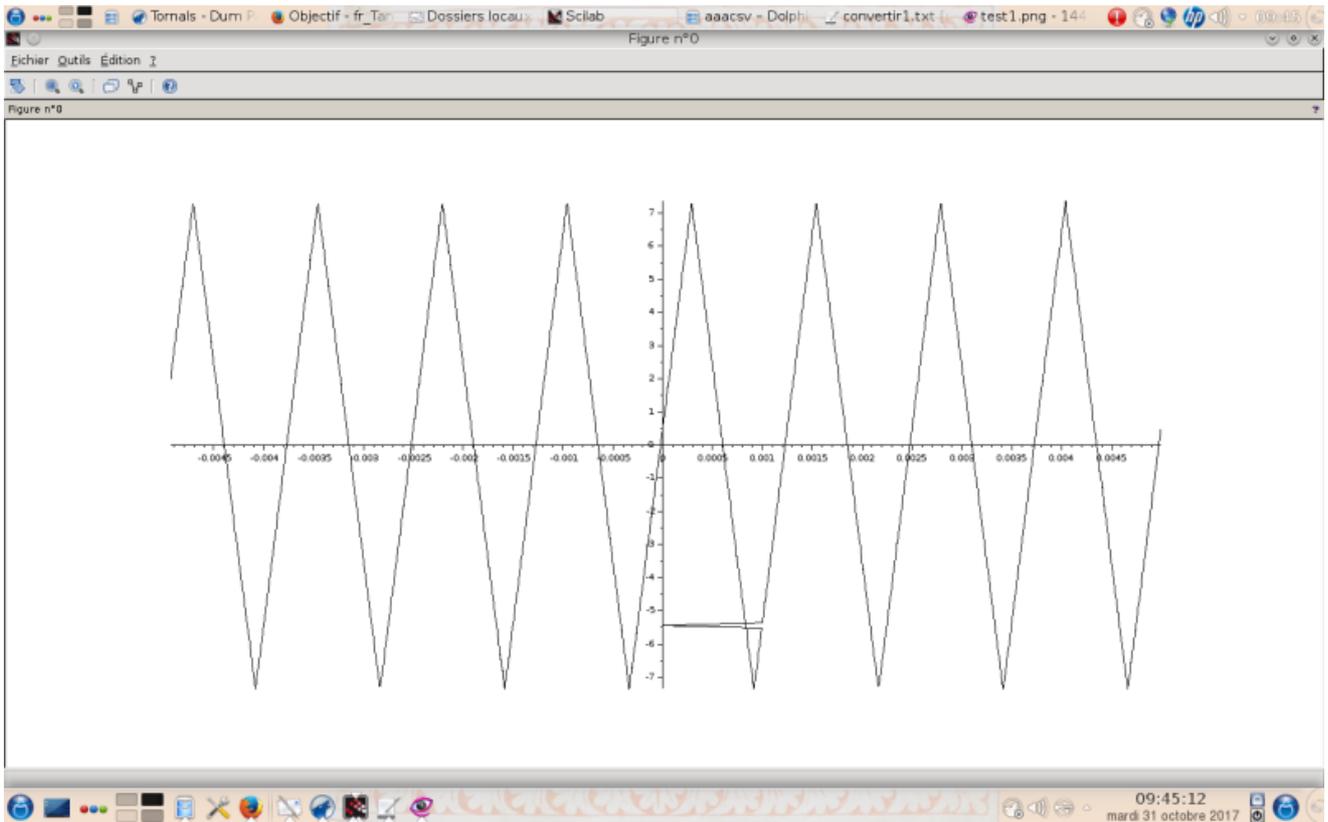
```
-->u=M(:,2)
```

```
u =  
2.  
2.1599998  
2.24  
2.3199999  
2.48  
2.48  
etc ...
```

On peut alors tracer la courbe :

-->plot2d(t,u)

L'interface permet ensuite de l'exporter au format png.



Donc en résumé, avec les points-virgules de fin de ligne pour ne pas être encombré par l'affichage des données :

```
M=csvRead("chemin de mon fichier .csv");  
t=M(:,1);  
u=M(:,2);  
plot2d(t,u)
```