

Peindre avec Chebyshev (v.1.0)

manuel.luque27@gmail.com

24/05/2020

1 Présentation

C'est en quelque sorte la suite de : [Représentation du polynôme de Chebyshev de première espèce en coordonnées polaires](#) où des images remarquables sont obtenues avec les polynômes de Chebyshev.

Dans son livre "*The Mathematica GuideBook for Symbolics*", (2006 Springer Science+Business Media, Inc.), Michael Trott écrit page 862 :

« We could go on now and make some nice pictures involving various combinations of the orthogonal polynomials. Here is a contour plot that contains the two Chebyshev polynomials $T_7(z)$ and $U_8(z)$. »

et donne le code de l'exemple et l'image obtenue, comme elle me semblait plaisante j'ai eu envie de la réaliser avec PSTricks, mais j'ai d'abord testé le code avec Mathematica et là j'ai été un peu refroidi par la lenteur de l'exécution, le message Running ...) s'affichant pendant plus de 20 min avant d'avoir l'image et j'ai d'abord pensé qu'il était inutile de tenter l'expérience avec PSTricks, je l'ai fait quand même pour être sûr que c'était irréalisable ou pour le moins trop lent pour être exploitable. À ma grande surprise PSTricks s'est révélé bien moins lent que Mathematica (1 min), ce qui m'a amené à écrire une commande afin de tester tous les cas esthétiquement intéressants. Pour être complètement honnête il faut dire que PSTricks ne peut pas traiter tous les cas comme le fait Mathematica (certes avec une certaine lenteur) par exemple le choix du degré de chaque polynôme sera limité avec PSTricks (à tester, 10 semble la limite) ainsi que le pas des calculs (le nombre de points), limité lui par la taille des tableaux admissibles par postscript. Signalons aussi que le temps de calcul est lié au degré des polynômes.

2 La commande `\psPaintingWithChebyshev[options]` et ses options

Tout d'abord le code de Michael Trott :

```
ContourPlot[Re[ChebyshevT[7, I ChebyshevT[8, x + I y]], {x, -3/2, 3/2}, {y, -0.35, 0.35},  
Contours -> {0}, PlotPoints -> 222, FrameTicks -> None]
```

Que l'on peut traduire ainsi, dans le cas général (on est dans \mathbb{C}) :

$$\text{ChebyshevT}(n, i \text{ChebyshevT}(m, x + iy))$$

dont on prend la partie réelle (ou imaginaire)¹. Elle est codée ici en postscript, entre les crochets, par exemple :

```
\psContourPlot[function={10 [0 1] 8 [x y] ChebyshevT Imul ChebyshevT ReZ}] (-1, -1) (1, 1)
```

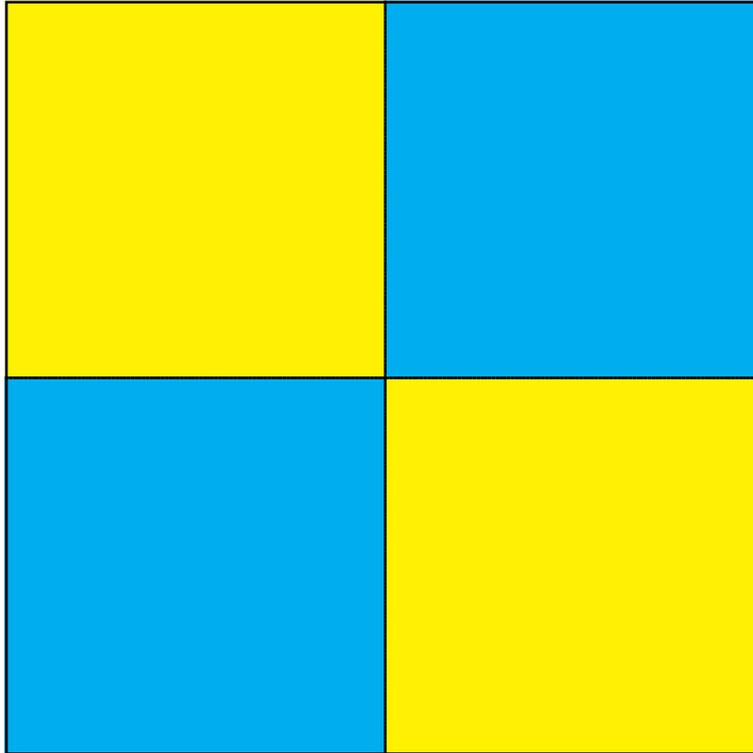
Voici les options de la commande permettant de transposer à PSTricks, l'ingénieuse idée de Michael Trott (les valeurs par défaut sont indiquées) :

1. `[m=1]` , `[n=1]` : degré des polynômes.
2. `[ImaginaryPart]` : pour choisir la partie imaginaire (la partie réelle est prise par défaut).

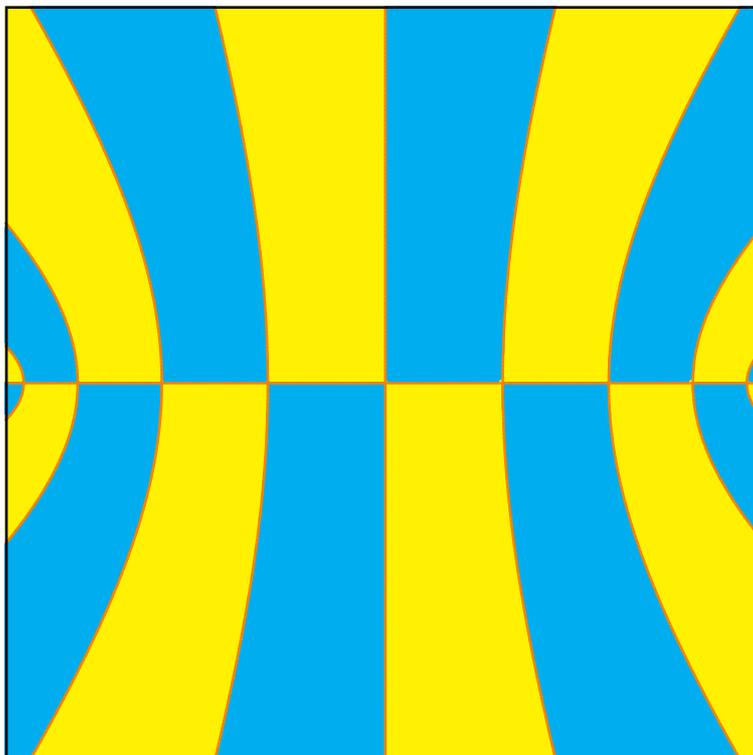
Les options du package "pst-contourplot" restent valables.

¹D'autres combinaisons sont évidemment possibles.

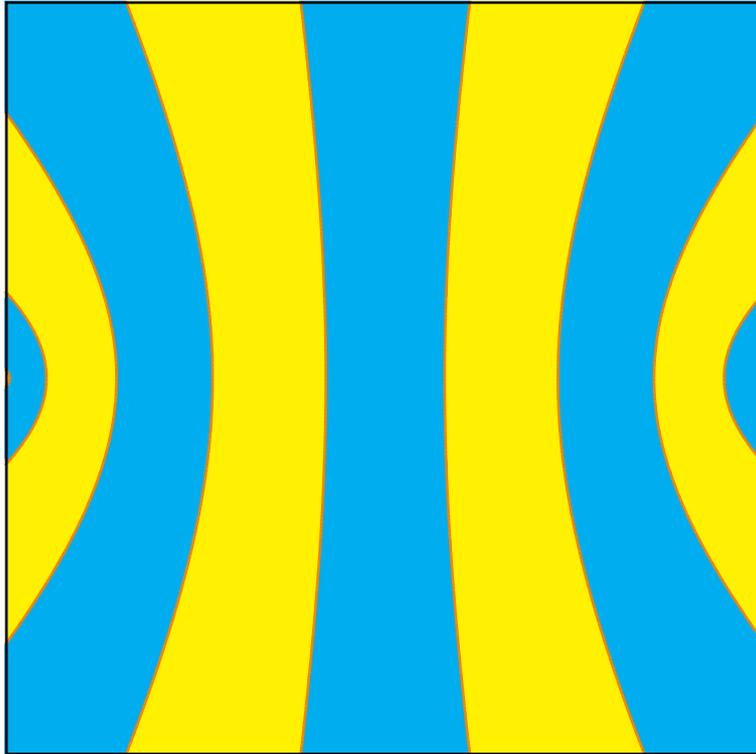
3 Exemples



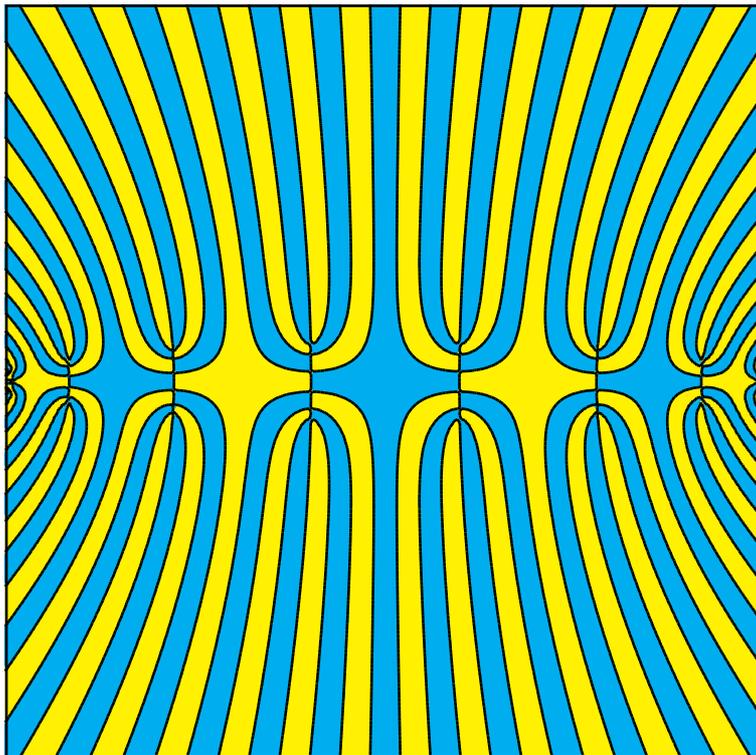
```
\begin{pspicture*}(-5,-5)(5,5)
\psframe*[linecolor=yellow](-5,-5)(5,5)
\psframe(-5,-5)(5,5)
\psset{linewidth=1pt,fillcolor=cyan}
\psPaintingWithChebyshev[m=2,n=1]
\end{pspicture*}
```



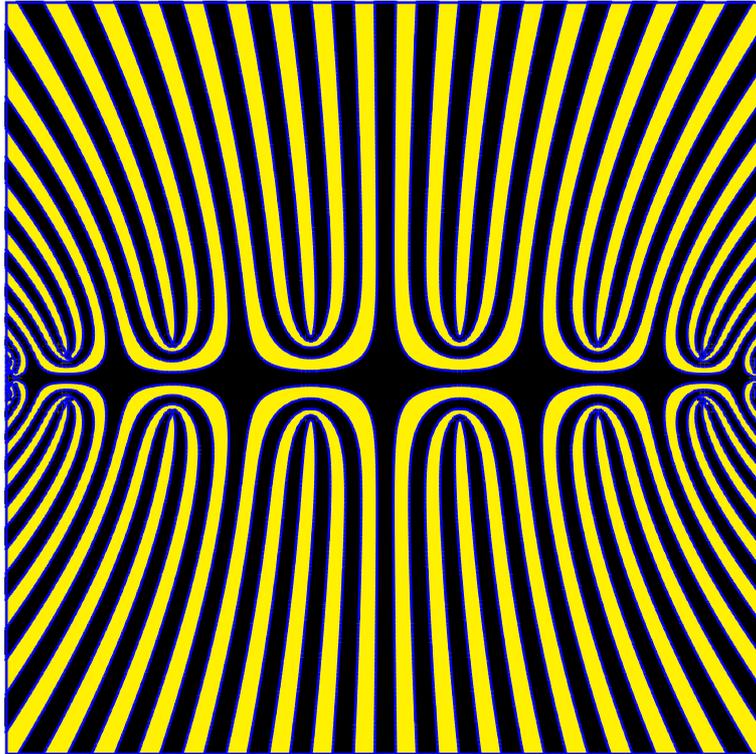
```
\psPaintingWithChebyshev[m=10,n=1,linecolor=orange]
```



```
\psPaintingWithChebyshev [m=10,n=1,ImaginaryPart,linecolor=orange]
```



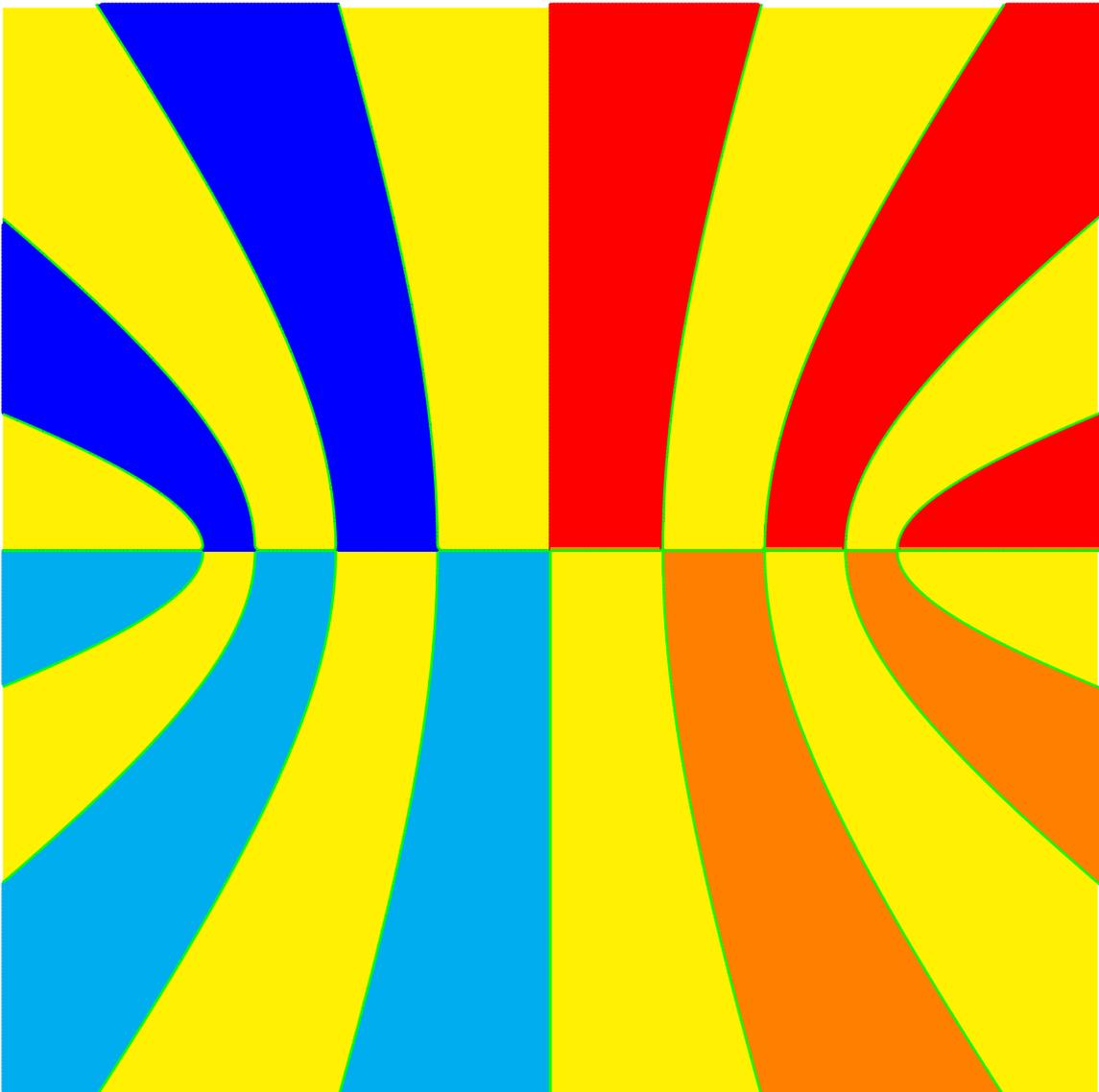
```
\begin{pspicture}(-5,-5)(5,5)
\psframe*[linecolor=yellow](-5,-5)(5,5)
\psset{linewidth=1pt,fillcolor=cyan}
\psPaintingWithChebyshev [ImaginaryPart,m=8,n=7]
\end{pspicture}
```



```
\psPaintingWithChebyshev [m=8,n=10]
```

4 Prolongements

Si le fichier "pst-operations-on-complex-numbers.pro" est dans votre répertoire de travail, vous pouvez utiliser le package "pst-contourplot" pour créer d'autres combinaisons des polynômes et d'autres tableaux.



```

\begin{pspicture}(-7.5,-7.5)(7.5,7.5)
\psframe*[linecolor=yellow](-7.5,-7.5)(7.5,7.5)
\psset{linecolor=green,fillcolor=cyan,unit=5,linewidth=1pt}
\psset{Fill,a=0.01,function={1 [0 1] 10 [x y] ChebyshevT Imul ChebyshevT ReZ}}
\psContourPlot(-1.5,-1.5)(0,0)
\psContourPlot[fillcolor=blue](-1.5,0)(0,1.5)
\psContourPlot[fillcolor=red](0,0)(1.5,1.5)
\psContourPlot[fillcolor=orange](0,-1.5)(1.5,0)
\end{pspicture}

```