

Espace des couleurs RVB

1. Introduction

Ce document explique le principe de la représentation des couleurs dans les images RVB, et définit la conversion en Teinte-Saturation-Luminance (TSL).

2. Synthèse additive des couleurs et codage RVB

Une grande partie des couleurs que nous percevons peut être obtenue en superposant trois couleurs indépendantes (rouge, vert et bleu) avec des intensités relatives variables. Si on note R , V et B les trois couleurs primaires utilisées, une couleur C peut s'écrire comme la combinaison linéaire :

$$\vec{C} = r\vec{R} + v\vec{V} + b\vec{B} \quad (1)$$

Lorsque les coefficients (r, v, b) sont positifs, il s'agit d'une synthèse additive. La plus grande étendue de couleur est obtenue par cette méthode lorsque les couleurs primaires utilisées sont des lumières monochromatiques. Même dans ce cas, les couleurs spectrales (couleurs monochromatiques) ne sont pas accessibles par synthèse additive. Par exemple, pour une couleur spectrale située dans la partie bleu-verte du spectre, il faut d'abord l'additionner à du rouge pour la rendre représentable par un mélange additif de vert de de bleu. On écrira alors :

$$\vec{C} + \alpha\vec{R} = v\vec{V} + b\vec{B} \quad (2)$$

En posant $\alpha = -r$, on retrouve la relation (1) mais avec un coefficient r négatif. D'une manière générale, la relation (1) permet d'obtenir toutes les couleurs à condition que les primaires soient monochromatiques et que les coefficients puissent prendre des valeurs négatives.

Les écrans d'ordinateur (cathodique, LCD), et d'une manière générale les systèmes de formation d'image numérique, fonctionnent sur le principe du mélange additif. Chaque pixel de l'écran est constitué de trois cellules, une verte, une bleue et une rouge. L'intensité lumineuse émise par ces cellules est ajustée pour produire la couleur voulue. Comme il s'agit d'une synthèse additive, les couleurs proches du spectre (couleurs très saturées) ne peuvent pas être obtenues sur ces écrans. Les couleurs des primaires sont variables d'un écran à l'autre. Le meilleur résultat est obtenu lorsque les primaires sont très saturées.

Les images numériques destinées à l'affichage sur ces écrans sont codées en RVB. Elles comportent une couche rouge, une couche verte et une couche bleue. Chaque couche, le plus souvent codée sur 8 bits (valeurs de 0 à 255) représente le niveau d'intensité qui doit être délivré par la cellule correspondante sur l'écran.

En réalité, la reproduction fidèle des couleurs sur un écran est compliquée par le fait que tous les écrans ne fonctionnent pas tout à fait avec les mêmes primaires (il y a aussi des différences de restitution des contrastes). Pour résoudre ce problème, on associe aux images numériques un profil de couleur (par exemple le profil sRGB) qui précise les primaires associées à l'image. Il est possible de déterminer les primaires d'un écran donné en réalisant un étalonnage ; on obtient ainsi le profil colorimétrique de l'écran. Lorsqu'une

image s'affiche sur l'écran, un logiciel de conversion tient compte des différences de profil entre l'image numérique et l'écran, et effectue la conversion pour que les couleurs s'affichent correctement.

3. Espace des couleurs et triangle de Maxwell

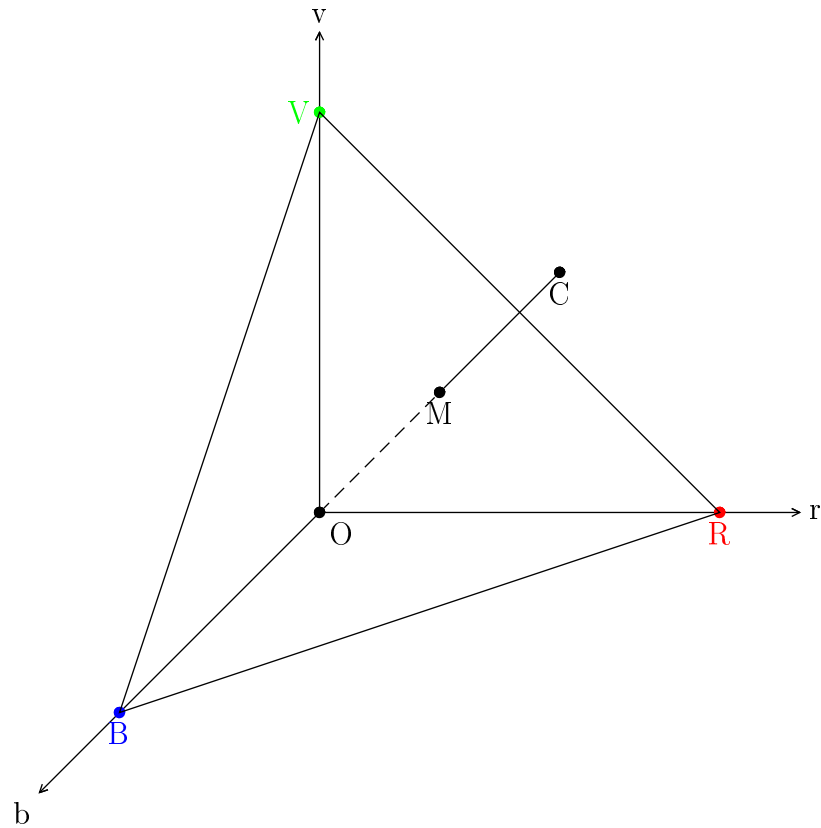
On se place dans le cadre de la synthèse additive des couleurs. Les coefficients réels (r, v, b) sont positifs. Par convention, on peut les choisir dans l'intervalle $[0, 1]$. Lorsque la couche est codée sur 8 bits, les coefficients sont convertis en nombres entiers compris entre 0 et 255.

Pour représenter géométriquement les couleurs, on représente les primaires par une base orthonormée $(\vec{R}, \vec{V}, \vec{B})$ et on place le noir à l'origine O . L'ensemble des couleurs est alors contenu dans un cube de côté 1. Le point $P(1, 1, 1)$ est le sommet du cube opposé à O .

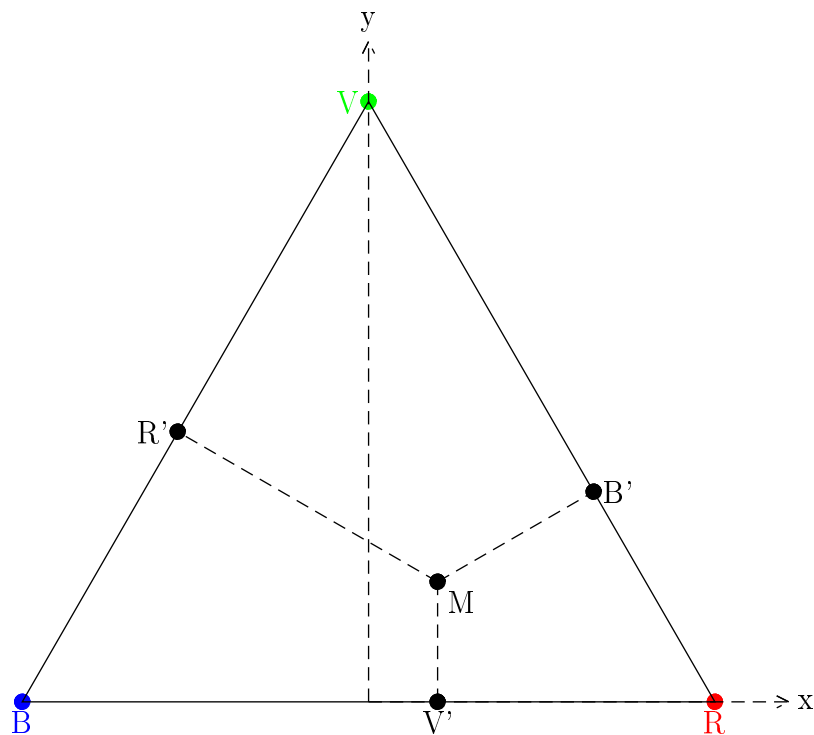
Voir [Espace des couleurs RVB et triangle de Maxwell](#) pour une représentation tridimensionnelle de ce cube.

Soit $C(r, g, b)$ un point représentant une couleur. Lorsqu'on multiplie tous les coefficients (r, g, b) par une même constante, on ne change pas la qualité d'une couleur, que l'on appelle sa chromaticité, mais seulement sa luminosité. Ainsi, toutes les couleurs de la droite OC sont perçues avec la même chromaticité. En particulier, les points de la diagonale OP sont des points neutres, c'est-à-dire des gris, obtenus par un mélange égal des trois primaires. Le point $O(0, 0, 0)$ est le noir, le point $P(1, 1, 1)$ est le blanc.

Considérons le plan d'équation $r + v + b = 1$ qui passe par les points $R(1, 0, 0)$, $V(0, 1, 0)$ et $B(0, 0, 1)$. Ces 3 points définissent dans ce plan un triangle appelé triangle de Maxwell. L'intersection de la droite OC avec ce triangle est le point M . La position de ce point dans le triangle suffit à déterminer la chromaticité de la couleur. D'un point de vue chromatique, toutes les couleurs sont donc représentables dans le triangle de Maxwell. Les coordonnées trichromatiques (r, v, b) vérifient alors $r + v + b = 1$. Il s'en suit que ces coordonnées sont en fait les coordonnées barycentriques du point M par rapport aux trois sommets du triangle.



Le triangle de Maxwell est isocèle, de côté $\sqrt{2}$ et de hauteur $\sqrt{3/2}$. On lui applique un changement d'échelle pour que sa hauteur devienne 1. La longueur d'un côté est alors $2/\sqrt{3}$.



Un repère a été ajouté afin de repérer un point M par ses coordonnées (x, y) , qui s'expriment en fonction des coordonnées trichromatiques par :

$$x = \frac{r - b}{\sqrt{3}} \quad (3)$$

$$y = v \quad (4)$$

Soit B' le projeté orthogonal de M sur le côté RV . Les coordonnées de ce point sont :

$$x_{B'} = x + \frac{\sqrt{3}}{2}b \quad (5)$$

$$y_{B'} = y + \frac{b}{2} \quad (6)$$

Le point R' projeté orthogonal de M sur le côté BV a pour coordonnées :

$$x_{R'} = x - \frac{\sqrt{3}}{2}r \quad (7)$$

$$y_{R'} = y + \frac{r}{2} \quad (8)$$

Lorsque la hauteur du triangle isocèle est 1, les coordonnées barycentriques sont les distances du point M aux côtés, c'est-à-dire :

$$r = MR' \quad (9)$$

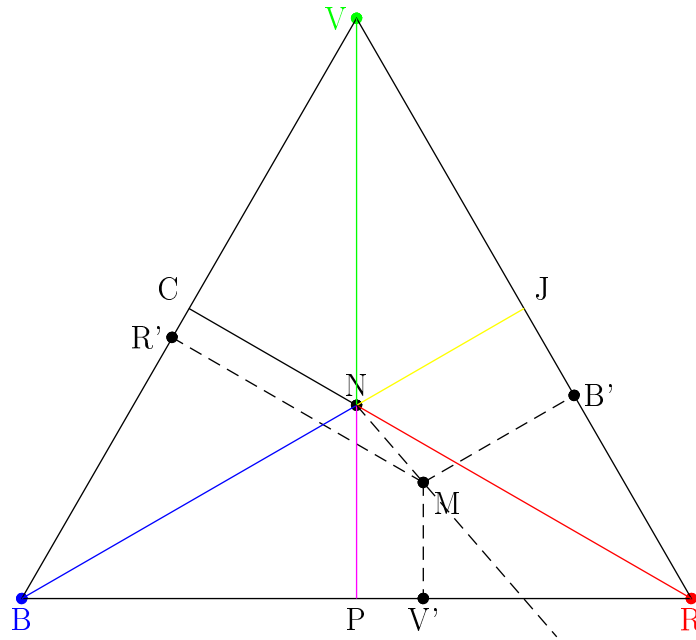
$$v = MV' \quad (10)$$

$$b = MR' \quad (11)$$

Lorsque $r = v = b = 1/3$ le point M se trouve à égale distance des trois côtés, au centre du triangle.

4. Teinte et saturation

Soit $N(1/3, 1/3, 1/3)$ le point neutre, centre du triangle de Maxwell. En joignant ce point à ses projections sur les trois côtés, on définit dans le triangle trois zones, que l'on appellera zones rouge, verte et bleue. Les projections sont notées J (jaune), C (cyan) et P (pourpre ou magenta). Il s'agit des couleurs complémentaires du bleu, du rouge et du vert.



Pour les points M de la zone rouge (polygone $NJRP$), la composante trichromatique maximale est r , car la distance MR' est supérieure à MB' et à MV' .

Les points de la droite NM correspondent à des couleurs perçues comme ayant la même teinte avec des degrés de saturation différents. Par exemple, les points de la droite NR ont la même teinte que la primaire R . Les points de cette droite proches de N ont une faible saturation alors que ceux proches de R ont une forte saturation. Pour la teinte définie par la droite NM représentée sur la figure, le point le plus saturé est à l'intersection avec le côté BR (il s'agit d'une teinte pourpre à dominante rouge).

La direction de la droite NM suffit donc à définir la teinte d'une couleur. Considérons le cas d'un point M situé dans la zone rouge. On a alors $\max(r, v, b) = r$. La distance algébrique entre le point M et la droite NR est :

$$d_{M,NR} = \frac{v - b}{\sqrt{3}} \quad (12)$$

Lorsque cette distance est négative, le point M est dans le triangle NPR (comme sur la figure). On définit dans ce cas la teinte en formant le rapport avec la distance de M à la droite NB :

$$T_{M \in (NPR)} = \frac{d_{M,NR}}{d_{M,NB}} = \frac{v - b}{r - v} \quad (13)$$

Lorsque M est sur le segment NP (teinte magenta), la teinte est $T = -1$. Lorsqu'il est

sur le segment NR (teinte rouge), $T = 0$. De manière analogue, on définit la teinte pour un point situé dans le triangle NRJ :

$$T_{M \in (NRJ)} = \frac{d_{M,NR}}{d_{M,NP}} = \frac{v-b}{r-b} \quad (14)$$

Cette teinte varie de 0 pour le rouge à 1 pour le jaune (points du segment NJ).

Afin de représenter la teinte sur un cercle, on la convertit en un angle en degrés. On adopte par convention l'angle zéro pour la teinte rouge (segment NR). On obtient ainsi pour un point de la zone rouge :

$$T_{M \in (NPR)} = 360 + 60 \frac{v-b}{r-v} \quad (15)$$

$$T_{M \in (NRJ)} = 60 \frac{v-b}{r-b} \quad (16)$$

Pour savoir si le point M se trouve dans le triangle NPR ou NRJ , il suffit de connaître $\min(r, v, b)$.

Soit θ l'angle entre la droite NM et la droite NR . Les bornes de variation de cet angle sont les mêmes que ceux de la teinte (360 – 60 à 0 dans le triangle NPR , 0 à 60 dans le triangle NRJ). Toutefois, la teinte n'est pas cet angle.

La saturation est définie par :

$$S_{M \in (NPR)} = \frac{r-v}{r} \quad (17)$$

$$S_{M \in (NRJ)} = \frac{r-b}{r} \quad (18)$$

Ainsi dans le triangle NPR la saturation varie de 0 en N à 1 sur le segment PR .

Pour la zone verte, la teinte est définie de manière analogue, mais avec un angle variant de 60 à 180 degrés :

$$T_{M \in (NVJ)} = 120 + 60 \frac{b-r}{v-b} \quad (19)$$

$$T_{M \in (NVC)} = 120 + 60 \frac{b-r}{v-r} \quad (20)$$

Enfin pour la zone bleue, la teinte varie de 180 à 300 degrés :

$$T_{M \in (NCB)} = 240 + 60 \frac{r-v}{b-r} \quad (21)$$

$$T_{M \in (NBP)} = 240 + 60 \frac{r-v}{b-v} \quad (22)$$

5. Conversion RGB/TSL

La teinte et la saturation ont été définies ci-dessus à partir des composantes (r, g, b) dans le triangle de Maxwell. Pour une couleur quelconque, il faut en plus une information sur la luminance. La manière la plus simple est de définir la luminance comme le

maximum de r, g, b . Pour résumé, voici l'algorithme qui permet de calculer la teinte (T), la saturation (S) et la luminance (L) :

$$Max = \max(r, g, b) \quad (23)$$

$$Min = \min(r, g, b) \quad (24)$$

$$C = Max - Min \quad (25)$$

$$L = Max \quad (26)$$

$$S = \frac{C}{L} \quad (27)$$

$$T = 60 \frac{v - b}{C} \quad [360] \quad \text{si } Max = r \quad (28)$$

$$T = 120 + 60 \frac{b - r}{C} \quad \text{si } Max = v \quad (29)$$

$$T = 240 + 60 \frac{r - v}{C} \quad \text{si } Max = b \quad (30)$$

Si $C = 0$, il s'agit d'une couleur neutre (trois couleurs égales) pour laquelle la teinte n'est pas définie et la saturation nulle. Les valeurs de S et L sont dans l'intervalle $[0, 1]$. La teinte est un angle dans l'intervalle $[0, 360]$, mais il est parfois ramené dans l'intervalle $[0, 1]$.

Voyons à présent la transformation inverse, de TSL vers RGB. On a bien sûr $C = LS$ et $Min = L - C$. Si la teinte T est dans l'intervalle $[300, 360]$, on a $r = L$ et $v = Min = L - C$. La relation (28) nous permet alors d'obtenir b :

$$b = v + \frac{360 - T}{60} C \quad (31)$$

Si T est dans l'intervalle $[0, 60]$ alors $r = L$, $b = Min = L - C$ et

$$v = b + \frac{T}{60} C \quad (32)$$

On procède de manière similaire pour les autres cas. Le tableau suivant résume la transformation :

$$\begin{array}{llll}
 T \in [300, 360] & r = L & v = L - C & b = v + C(360 - T)/60 \\
 T \in [0, 60] & r = L & b = L - C & v = b + C(T/60) \\
 T \in [60, 120] & v = L & b = L - C & r = b + C(120 - T)/60 \\
 T \in [120, 180] & v = L & r = L - C & b = r + C(T - 120)/60 \\
 T \in [180, 240] & b = L & r = L - C & v = r + C(240 - T)/60 \\
 T \in [240, 300] & b = L & v = L - C & r = v + C(T - 240)/60
 \end{array} \quad (33)$$

6. Application

Nous programmons des fonctions python permettant de tracer le triangle de Maxwell, de le colorier, de placer une couleur (r, v, b) et d'effectuer la conversion RVB/TSV.

[espaceRVB.py](#)

```
import math
```



```
import numpy
from matplotlib.pyplot import *
```

La fonction suivante trace le triangle de Maxwell :

```
def maxwell():
    a = 1.0/math.sqrt(3)
    scatter([a,0,-a],[0,1,0],s=40,c=[(1,0,0),(0,1,0),(0,0,1)])
    plot([a,0,-a,a],[0,1,0,0], 'k-')
    axis([-0.7,0.7,-0.2,1.2])
```

La fonction suivante place un point dans le triangle, en fonction de ses composantes trichromatiques :

```
def point(rvb):
    somme = rvb[0]+rvb[1]+rvb[2]
    r = rvb[0]*1.0/somme
    v = rvb[1]*1.0/somme
    b = rvb[2]*1.0/somme
    scatter([(r-b)/math.sqrt(3)],[v],s=50,c=(r,v,b))
```

La fonction suivante colorie l'intérieur du triangle de Maxwell :

```
def fill_maxwell():
    Nlignes=300
    Ncol=300
    img = numpy.zeros((Nlignes,Ncol,4))
    dx = 2.0/(Ncol-1)
    dy = 1.0/(Nlignes-1)
    for i in range(Ncol-1):
        for j in range(Nlignes-1):
            x = -1.0+i*dx
            y = j*dy
            v = y
            r = (x+1-v)/2.0
            b = 1.0-v-r
            if (r>=0) and (r<=1.0) and (v>=0) and (v<=1.0) and (b>=0) and (b<=1.0):
                img[j][i] = numpy.array([r,v,b,1.0])
            else:
                img[j][i] = numpy.array([1.0,1.0,1.0,0.0])
    a = 1.0/math.sqrt(3)
    imshow(img,origin='lower',extent=[-a,a,0.0,1.0])
```

La fonction suivante effectue la conversion de RVB vers TSV :

```
def rvb2tsl(rvb):
    r=rvb[0]*1.0
    v=rvb[1]*1.0
    b=rvb[2]*1.0
    somme = r+v+b
    r=r/somme
    v=v/somme
    b=b/somme
    Max = max(r,v,b)
    Min = min(r,v,b)
    C=Max-Min
    L = Max
    if L==0:
        return [0,0,0]
    S = C/L
    if C==0:
        return [0,0,L]
    if max==r:
        T = 60.0*(v-b)/C % 360
    elif max==v:
        T = 120.0+60.0*(b-r)/C
    else:
        T = 240.0+60.0*(r-v)/C
    return [T,S,L]
```

La fonction suivante effectue la conversion de TSV vers RVB :

```
def tsl2rvb(tsl):
    T=tsl[0]*1.0
    S=tsl[1]*1.0
    L=tsl[2]*1.0
    C=L*S
    Min = L-C
    if (T>300) and (T<=360):
        r = L
        v = Min
        b = v+C*(360.0-T)/60
    elif (T>=0) and (T<=60):
        r = L
        b = Min
        v = b+C*(T/60)
    elif (T>60) and (T<=120):
        v = L
        b = Min
        r = b+C*(120.0-T)/60
    elif (T>120) and (T<=180):
        v = L
        r = Min
```

```

        b = r+C*(T-120.0)/60
    elif (T>180) and (T<=240):
        b = L
        r = Min
        v = r+C*(240.0-T)/60
    else:
        b = L
        v = Min
        r = v+C*(T-240.0)/60
    return [r,v,b]

```

La fonction suivante colorie un disque en utilisant la teinte comme angle et la saturation comme rayon :

```

def disqueTSL():
    Ncol = 300
    Nlignes = 300
    img = numpy.zeros((Nlignes,Ncol,4))
    dx = 2.0/(Ncol-1)
    dy = 2.0/(Nlignes-1)
    rad2deg = 180.0/math.pi
    for i in range(Ncol):
        for j in range(Nlignes):
            x=-1.0+i*dx
            y=-1.0+j*dy
            r = math.sqrt(x*x+y*y)
            if r<1.0:
                if x==0:
                    if y>0:
                        a = 90.0
                    else:
                        a = -90.0
                else:
                    a = math.atan(y/x)*rad2deg
            if x<0:
                a = a+180.0
            a = a % 360
            rvb = tsl2rvb([a,r,1.0])
            img[j][i] = numpy.array([rvb[0],rvb[1],rvb[2],1.0])
        else:
            img[j][i] = numpy.array([1.0,1.0,1.0,0.0])
    imshow(img,origin='lower',extent=[-1,1,-1,1])

```

La fonction suivante trace dans le triangle de Maxwell une série de points avec des teintes régulièrement espacées et une saturation constante :

```

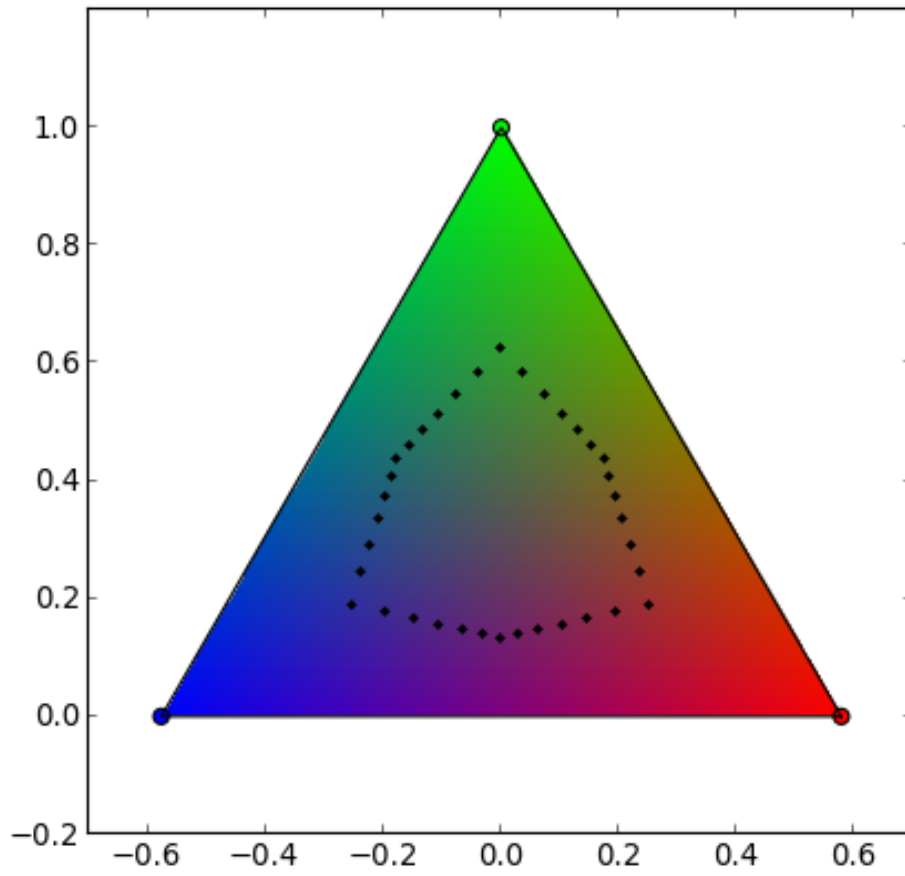
def pointsTSV(dT,S):

```

```
T = numpy.arange(0,360,dT)
x = numpy.zeros(T.size)
y = numpy.zeros(T.size)
for k in range(T.size):
    rvb = tsl2rvb([T[k],S,1.0])
    somme = rvb[0]+rvb[1]+rvb[2]
    x[k] = (rvb[0]-rvb[2])/math.sqrt(3)/somme
    y[k] = rvb[1]/somme
plot(x,y,"k. ")

import math
from matplotlib.pyplot import *
import espaceRVB
from random import random

figure(figsize=(6,6))
espaceRVB.maxwell()
espaceRVB.fill_maxwell()
espaceRVB.pointsTSV(10.0,0.7)
```



```
figure(figsize=(6,6))
espaceRVB.disqueTSL()
```

