ColorTEXrgb - ColorTEXcmyk : guide d'utilisation des macros de mise en couleur

Manuel de Référence

Version 2.9

Christophe Cérin
Université de Paris Sud
LRI bat 490
91405 Orsay - France
mail : cerin@lri.lri.fr
69-41-70-82

Petite histoire de ces macros

Dans un article récent [1], nous présentions les avantages d'utilisation de la couleur dans le traitement de texte TEX au travers de deux jeux de macros : ColorTEXrgb et colorTEXcmyk fonctionnant pour les imprimantes couleur PostScript des constructeurs QMS, Océ, Hitachi, Seiko.

Dans cet article, nous discutions de la couleur en général, comment la spécifier, la représenter, puis, nous dégagions des classes de problèmes qui appellent des classes de solutions communes, des algorithmes communs. Cela nous conduisait à proposer un ensemble minimal de primitives permettant de résoudre les problèmes identifiés suivants : action sur une entité syntaxique (lettre, mot, phrase, paragraphe) et action sur une entité graphique : vecteur ou flèche de l'environnement **picture** de LATEX.

Au moyen des constructeurs de base `\begincolor` et `\endcolor`, nous arrivions à réaliser des dégradés de couleur sur des chaîne de caractères : la couleur d'une lettre dépend de celle de ses voisines ou, plus généralement, de paramètres spécifiant l'interaction des couleurs.

Cet article présente la version 2.9 des macros qui bénéficient de l'ajout de primitives comme la création de boîtes couleur contenant du texte dans une couleur différente. Vous trouverez également des macros pour écrire du texte sur un fond dégradé et des primitives pour dessiner des courbes de splines en couleur et en particulier des contours fermés quelconques, des cercles hachurés ainsi que la possibilité de placer du texte le long d'une courbe.

L'implantation des constructeurs initiaux proposée est réalisée en partie avec PostScript, par l'intermédiaire de la commande `special` dans la syntaxe reconnue par le translateur `dvips` de Tom Rokicki, et en TEX. L'ensemble des macros est écrit pour fonctionner avec LATEX. Deux jeux sont disponibles : un pour les imprimantes du type LaserWriter (colorTEXrgb), l'autre pour les imprimantes du type de la QMS ColorScript (ColorTEXcmyk).

Modèle RGB

Ce module décrit l'ensemble des primitives qui spécifient la couleur au moyen des opérateurs RGB. Il est constitué d'une partie permettant de déclarer des variables, d'une partie permettant d'effectuer des initialisations sur ces variables et d'une partie qui décrit la manipulation de la couleur sur des chaînes de caractères. On propose ensuite quelques macros de confort qui déclarent des couleurs prédéfinies puis nous terminons par des exemples d'utilisation et une présentation de macros pour réaliser des boîtes dont le fond est un dégradé de couleurs.

Déclarations

rouge=int
contient la valeur de la composante rouge.

vert=int
contient la valeur de la composante vert.

bleu=int
contient la valeur de la composante bleu.

increrouge=long
contient la valeur de l'incrément du rouge.

increvert=long
contient la valeur de l'incrément du vert.

increbleu=long
contient la valeur de l'incrément du bleu.

modulo=long
valeur du nombre de lettres dans la même couleur.

maxrouge=int
contient la valeur de la composante rouge maximale.

maxvert=int
contient la valeur de la composante verte maximale.

maxbleu=int
contient la valeur de la composante bleue maximale.

Primitives d'initialisation des couleurs

initcolor{rouge}{vert}{bleu}
{maxrouge}{maxvert}{maxbleu}{modulo}
Procédure d'initialisation des couleurs, des couleurs maximales et du paramètre «modulo» permettant de spécifier le nombre de caractères devant être imprimés dans la même couleur.

initinc{rouge}{vert}{bleu}
Procédure d'initialisation de chaque couleur par des valeurs de type `int`.

Couleur sur les chaînes

fboxcolor{texte}{rouge}{vert}{bleu}
Boite dont le contour est tracé en couleur. La macro est inspirée de la macro LaTeX `\fbox` et elle utilise les variables `\fboxrule` et `\fboxsep`. Elle ne peut s'utiliser que sous LaTeX.

boxcolor{texte}{rouge}{vert}{bleu}
Boite dont le contenu est en couleur. Cette macro est inspirée de la macro LaTeX `\fbox` et elle utilise les variables LaTeX `\fboxrule` et `\fboxsep`. De plus, on utilise une nouvelle boîte `\@myboxb` et une nouvelle dimension `\@mydimb` afin d'éviter certains conflits de variables si l'on effectue un `\fbox{}` de la boîte en couleur.

underlinecolor{texte}{rouge}{vert}{bleu}
Le texte `texte` est souligné en couleur.

dentdescie{texte}
Cette procédure prend une chaîne de caractères et imprime les « modulo » premiers caractères dans la couleur spécifiée par les trois composantes rouge, vert, bleu définies au moyen de la primitive `\initcolor` ; puis décrémente chaque composante d'une valeur spécifiée au moyen de la primitive `\initinc` et imprime encore « modulo » caractères dans cette nouvelle couleur jusqu'à ce que « modulo » atteigne une valeur maximale. Dans ce cas, on recommence l'affichage des « modulo » premiers caractères suivants à partir de la couleur initiale spécifiée par les trois premiers paramètres de la primitive `\initcolor`.

echelon{texte}
Cette procédure prend une chaîne de caractères et imprime les « modulo » premiers caractères dans la couleur spécifiée par les trois composantes rouge, vert, bleu définies au moyen de la primitive `\initcolor` ; puis incrémente chaque composante d'une valeur définie au moyen de la primitive `\initinc` et imprime encore « modulo » caractères dans cette nouvelle couleur jusqu'à ce que « modulo » atteigne une borne max. Dans ce cas, on recommence l'affichage des « modulo » caractères suivants en décrémentant les couleurs jusqu'aux valeurs initiales qui sont sauvegardées dans les variables `\sauvxxx`. La variable globale `\sens` initialisée à 1 détermine un éclaircissement progressif des couleurs alors que `\sens` $= 0$ détermine un noircissement progressif des couleurs.

degrade{texte}
Cette macro prend une chaîne de caractères puis répartit uniformément

les gammes de couleurs définies par la primitive `\initcolor` sur l'ensemble de la chaîne.

longueur[texte]
Cette macro calcule la longueur d'une chaîne de caractères. Le résultat est rangé dans la variable globale `\lg`. Le paramètre doit être entre crochets.

Comment créer des effets spéciaux ?

Voici des macros qui effectue des dégradés concentriques sur des cercles ou linéaires sur des boîtes. L'utilisateur peut placer du texte (éventuellement en couleur) au milieu des cercles ou des boîtes et passer des boîtes TEX comme paramètres aux procédures.

CercleCouleur{texte}{max}{min}{couleur}{nb-nuances}
Macro qui trace des cercles degradés sous le texte «texte»en commençant par un grand cercle de couleur «max»jusqu'au petit cercle de couleur «min»en tracant «nb-nuances»nuances de couleur «couleur». Si le paramètre de la couleur vaut 1 on effectue un dégradé en rouge, si le paramètre de la couleur vaut 2 on effectue un dégradé en vert, si le paramètre de la couleur vaut 3 on effectue un dégradé en bleu. Remarquer que «nb-nuances»doit être > 0 pour éviter des divisions par 0 !

CercleGeneral{texte}{rouge}{vert}{bleu}
{maxrouge}{maxvert}{maxbleu}{nb-nuances}
Cette macro trace des cercles degradés sous le texte «texte»en commencant par un grand cercle de couleur «(rouge, vert, bleu)»jusqu'au petit cercle de couleur «(maxrouge, maxvert, maxbleu)»en tracant «nb-nuances»nuances de couleur. Cette macro est une généralisation de la précédente que l'on conserve pour raison historique. De plus, l'utilisateur peut spécifier un décalage pour le centre du cercle au moyen des variables `\xtrans` et `ytrans`. On peut alors obtenir l'effet d'une boule dégradée (cf exemples). Remarquer que «nb-nuances»doit être > 0 pour éviter des divisions par 0 !

RectangleDegrade{texte}{rouge}{vert}{bleu}
{maxrouge}{maxvert}{maxbleu}{nb-nuances}
Cette macro trace des rectangles dégradés sous le texte «texte»en commencant par un rectangle de couleur «(rouge, vert, bleu)»jusqu'au rectangle de couleur «(maxrouge, maxvert, maxbleu)»en tracant «nb-nuances»rectangles de couleur differentes. Pour choisir le sens du dégradé (de gauche à droite ou de haut en bas), vous utiliserez un booléen représenté en TEX par `\ifhorizontal`. Si vous souhaitez un dégradé de couleurs disposées de haut en bas, vous écrirez `\horizontaltrue` ; si vous souhaitez une disposition de couleurs de la gauche vers la droite,

vous écrirez `\horizontalfalse` avant l'appel à `\RectangleDegrade`. Remarquer que «nb-nuances»doit être > 0 pour éviter des divisions par 0 !

Comment déssiner des courbes dans l'environnement «picture»?

bezierps liste *endbezierps*
{wd}{dash1}{dash2}{red}{green}{blue}
Cette macro trace une courbe de bezier en couleur avec les points du paramètre `liste` qui est un tableau de tableaux au sens PostScript. Le paramètre `wd` contient la largueur des traits. Lorsque l'on veut tracer des courbes pointillées, il faut positionner le booléen `\ifdashps` à vrai (`\dashpstrue`) et à ce moment `dash1` spécifie la forme des pointillés (voir la commande Postscript `setdash`) et `dash2` définit le déplacement (à partir du début du trait) après lequel débutera le tracé. Si on a positionnée `\dashpsfalse` alors la courbe est continue. Attention ; les unités sont des millimètres et il faut un blanc avant `\endbezierps`. Enfin, l'ordre des parametrès est :

```
\bezierps [ [ x0 y0 x1 y1 x2 y2 x3 y3 ]...]
\endbezierps {}{}{}{}{}
```

avec les points x_0, y_0, x_1, y_1 définissant la première tangente à la courbe et les points x_2, y_2, x_3, y_3 définissant la deuxième tangente à la courbe.

CercleHachure{rayon}{wdrayon}{wd} {*angle*}{*r*}{*g*}{*b*}{*space*}
Cette macro trace un cercle de rayon `rayon` mm et d'épaisseur `wdrayon` mm, puis remplit ce cercle avec des hachures de largueur = `wd` mm inclinées de `angle` degrès. La couleur est specifiée par les paramètres r, g, b. Le paramètre `space` donne l'espacement entre les hachures. Les angles doivent être compris entre 0 et 180 degrès.

CarreHachure{ht}{wd1}{wd2} {*angle*}{*r*}{*g*}{*b*}{*space*}
Cette macro trace un carré de hauteur `2*ht` mm et d'épaisseur `wd1` mm, puis remplit ce cercle avec des hachures de largueur = `wd2` mm inclinées de `angle` degrès. La couleur est specifiée par les paramètres r, g, b. Le paramètre `space` donne l'espacement entre les hachures. Les angles doivent être compris entre 0 et 180 degrès.

RotObj{angle}{texte}
Pour mettre du texte en rotation ou tout autre chose, programmer avec la macro `\RotObj`. Le point à partir duquel la rotation se fait peut varier suivant l'appel à certaines macros : par défaut ce point est fixé au premier caractère du texte. Par exemple, l'appel `\mc` possitionne ce point au milieu de la chaîne, l'appel `\fc` à la fin, et l'appel `\dc` au début. Il y a seulement création d'une boite de largeur nulle, donc il n'y a pas

de place de réservée pour l'objet. Voici un exemple de programmation : `\mc\RotObj{45}{\Huge\bf Exemple}`.

CourbeRempliee liste *endcourbe*
{wd}{dash1}{dash2}{red}{green}{blue}
Trace une courbe fermée au moyen de courbes de Bezier avec les points du paramètre `liste` qui est un tableau de tableaux. Le paramètre `\wd` contient la largueur des traits du contour. Lorsque l'on veut tracer des courbes pointillées, il faut positionner le booléen `\ifdashps` à vrai (`\dashpstrue`) et à ce moment `dash1` spécifie la forme des pointillés (voir la commande Postscript `setdash`) et `dash2` définit le déplacement (à partir du début du trait) après lequel débutera le tracé. Si on a positionnée `\dashpsfalse` alors la courbe est continue. (cf fonction `\bezierps`). Attention, les unités sont des millimetres et il faut un blanc avant `\endcourbe`. L'ordre des paramètres est le même que celui de la fonction `\bezierps`. Le remplissage est effectué avec les couleurs spécifiées par les trois derniers paramètres. Le contour est tracé dans la couleur actuelle définie par les composantes `\rouge, \vert, \bleu`. Le contour est fermé si le booléan `\ifclosepath` est positionné à vrai (`\closepathtrue`). Enfin, tous les points sont en positions relatives par rapport à x_0, y_0 contrairement aux points définissant les tangentes dans la macro `\bezierps` et il faut respecter les blancs convenablement :

```
\CourbeRempliee [ [ x0 y0 x1 y1 x2 y2 x3 y3 ]...]
\endcourbe{}{}{}{}{}
```

TraceTexte{texte}{x_0}{y_0}{x_1}{y_1}{x_2}{y_2}{x_3}{y_3}
Cette macro répartit pas à pas chaque caractère de la chaîne `texte` le long de la courbe spécifiée par les 4 points $x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3$ (cf macro `bezierps`). La formule mathématique de la courbe de Bezier est dérivée de la paire d'équation cubique paramétrique de paramètre $t\,(0 \leq t \leq 1)$ suivante :

$$\begin{array}{rcll} x(t) & = & a_x t^3 + b_x t^2 + c_x t + x_0 & \quad \textit{coordonnées en } x \\ y(t) & = & a_y t^3 + b_y t^2 + c_y t + y_0 & \quad \textit{coordonnées en } y \end{array}$$

Les points de contrôle sont définis par :

$$\begin{array}{ll} x_1 = x_0 + c_x/3 & y_1 = y_0 + c_y/3 \\ x_2 = x_1 + (c_x + b_x)/3 & y_2 = y_1 + (c_y + b_y)/3 \\ x_3 = x_0 + c_x + b_x + a_x & y_3 = y_0 + c_y + b_y + a_y \end{array}$$

Pour chaque lettre à positionner, les différentes valeurs du paramètre t sont fonction de la largeur des caractères et de la tantente à la courbe.

Par exemple, supposons que l'on cherche à positionner le caractère `c` de la figure .4 page 18. On connait la valeur t' et les coordonnées (x', y') du point sur la courbe, obtenu au moyen des équations paramétriques, qui nous ont servi à positionner le caracère `a`. Pour calculer la tangente, nous cherchons la valeur de t'' correspondant à un déplacement le long de la courbe égal à la longueur du caractère `c`. Soit (x'', y'') les coordonées du point sur la courbe associé à t''. On pose :

$$tan(\alpha) = \frac{y' - y''}{x' - x''} \qquad (.1)$$

Le caractère est positionné à partir de (x', y') en le faisant pivoté de l'angle α. On continue ce même traitement avec le prochain caractère de la chaîne à partir du point (x'', y''). Les conditions initiales sont : le premier caractère à positionner est le premier caractère de la chaîne, $t = 0$, l'angle est déterminé au moyen du procédé décrit plus haut et ne correspond pas à la tangente définie par les points $x_0, y_0, x_1, y1$ qui est la tangente au point x_0, y_0 (un caractère a une largeur non nulle).

La variable `\BlancInterLettre` vous permet de rajouter de l'espace entre les caractères. Par défaut, sa valeur est fixée à 1mm (on rappelle que l'unité de tous les paramètres numériques doit être le mm).

L'implantation est réalisée en TEX pour ce qui est de l'application récursive des calculs et de l'obtention de la largeur des caractères. Les calculs numériques sont implantés, quant à eux, en PostScript. L'utilisation successive, dans le corps de la macro, de plusieurs ordres `\special` pour insérer du code PostScript nécessite des traitements particuliers car l'implantation du `\special` introduit des ordres de positionnement, en plus du code PostScript que l'on souhaite inséré, qui perturbent le bon fonctionnement. La macro a donc été testée et ne fonctionne pour l'instant qu'avec le translateur `dvips 5.4`. De petits ajustements de ces macros sont nécessaires pour des versions antérieures.

ColleTexte{texte}{x_0}{y_0}{x_1}{y_1}{x_2}{y_2}{x_3}{y_3}
Cette macro répartit la chaîne `texte` le long de la **totalité** de la courbe de Bezier spécifiée par les points $x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3$. Le calcul des différentes valeurs du paramètre t est plus facile que dans le cas de la macro `TraceTexte` : la nouvelle valeur du paramètre est obtenue en augmentant la dernière valeur du paramètre de la quantité $\frac{1}{\mid texte - 1 \mid}$ (attention : une croissance linéaire de t n'entraine pas nécessairement un espacement identique entre chaque caractère du fait de l'équation cubique paramétrique utilisée). Un exemple est présenté à la figure .4 page 18.

Quelques macros de confort

Afin de faciliter la manipulation de textes, de graphiques en couleur nous proposons quelques raccourcis.

\icone
Equivalent à un `\item` et place sur la feuille un point en bleu.

\ictwo
Equivalent à un `\item` et place sur la feuille un diamand en bleu.

\bcrouge
Remplace la couleur courante par la couleur rouge.

\bcbleu
Remplace la couleur courante par la couleur bleue.

\bcjaune
Remplace la couleur courante par la couleur jaune.

\bcvert
Remplace la couleur courante par la couleur vert.

\bcmauve
Remplace la couleur courante par la couleur mauve.

\bcmagenta
Remplace la couleur courante par la couleur magenta.

\bcorange
Remplace la couleur courante par la couleur orange.

\bcvj
Remplace la couleur courante par la couleur vert-jaune.

\bcblanc
Remplace la couleur courante par la couleur blanche.

\ec
Substitue la couleur sauvegardée à la couleur courante.

Pour changer de couleur, vous disposez également de macros qui portent les noms des primitives qui permettent de spécifier une couleur sous l'environement X11 : `\MediumAquamarine`, `coral`, `\DarkOliveGreen`, `MediumOrchid`... Reportez-vous au fichier `colorrgb.tex` pour obtenir toutes les redéfinitions.

Modèle CMYK

Dans le modèle Cyan-Magenta-Yellow-Black (CMYK), les primitives d'initialisation des couleurs, de couleur sur les chaînes nécessitent un paramètre supplémentaire. Par exemple :

`\underlinecolor{texte}{rouge}{vert}{bleu}` est remplacé par `\underlinecolor{texte}{cyan}{magenta}{yellow}{black}`.

Pour toute information complémentaire concernant l'utilisation des macros dans ce modèle, le lecteur est prié de se reporter aux sources des macros (fichier `colorcmyk.tex`).

Exemples d'utilisation dans le modèle RGB

Pour mettre en couleur le mot « bonjour » dans cette phrase, il suffit de baliser le mot par `\begincolor{255}{0}{0}` (ou `\bcrouge`) et `\endcolor` (ou `\ec`) pour obtenir bonjour . De la même façon, on peut baliser des traits, des flêches de l'environnement `\picture`

Par exemple, le trait qui suit est codé de la manière suivante :

```
\bcbleu %un trait bleu de longueur 8cm
\hrule height 2pt width 8cm %et d'epaisseur 2pt
\ec
```

Voici un exemple d'utilisation de la macros `\echelon` avec initialisation des couleurs et des incréments de chaque couleur. Les couleurs sont initialisées avec la valeur 20 et on désire qu'elles croissent jusqu'à la valeur 210 par saut de 40 entre chaque caractères (modulo = 1).

```
{% debut de groupe
\Huge\bf
% rouge=bleu=vert=20
% maxrouge=maxvert=maxbleu=210
% chaque caractere change de couleur
\initcolor{20}{20}{20}{210}{210}{210}{1}
% increrouge=increvert=increbleu=40
\initinc{40}{40}{40}
\echelon{la\ fonction\ echelon}
}% fin du groupe
```

la fonction echelon

Pour construire des boîtes dont le fond est en couleur, et le texte sur deux colonnes dans une autre couleur, nous programmons :

```
\setbox2=\vbox{\hsize=0.485\textwidth
Dans un article r\'ecent \cite{cc}, nous pr\'esentions
les avantages d'utilisation de la couleur dans le
traitement de texte \TeX\  au travers de deux jeux
de macros~: Color\TeX rgb et color\TeX cmyk fonctionnant
pour les imprimantes couleur PostScript des constructeurs
QMS, Oc\'e, Hitachi, Sei\-ko.

identifi\'es suivants~: action sur une entit\'e
syntaxique (lettre, mot, phrase, paragraphe) ; et action
sur une entit\'e graphique~: vecteur ou fl\`eche de
l'environnement {\bf picture} de \LaTeX .
\par
}% fin de la bo\^\i te numero 2
% on annule le blanc interbo\^\i te
\splittopskip=0pt
% on coupe la bo\^\i te 2 en deux selon la hauteur
% et on range dans la bo\^\i te 1
\setbox1=\vsplit2 to 0.5\ht2%
\setbox1=\vbox{\unvbox1}%
% on pose sur la feuille les deux bo\^\i tes
\hbox{\boxcolor{\box1}{200}{255}{255}\quad\vrule\quad
      \boxcolor{\box2}{50}{255}{255}}
```

On remarque que le texte est collé sur le bord de la boîte en couleur ; pour ajouter du blanc, il faut utiliser la variable LaTeX `\fboxsep`.

Dans un article récent [1], nous présentions les avantages d'utilisation de la couleur dans le traitement de texte TeX au travers de deux jeux de macros : ColorTeXrgb et colorTeXcmyk fonctionnant pour les imprimantes couleur PostScript des constructeurs QMS, Océ, Hitachi, Seiko.

Dans cet article, nous discutions de la couleur en général, comment la spécifier, la représenter, puis, nous dégagions des classes de problèmes qui appellent des classes de solutions communes, des algorithmes communs. Cela nous conduisait à proposer un ensemble minimal de primitives permettant de résoudre les problèmes identifiés suivants : action sur une entité syntaxique (lettre, mot, phrase, paragraphe) ; et action sur une entité graphique : vecteur ou flèche de l'environnement **picture** de LaTeX.

Nous présentons maintenant le codage d'un tableau avec les pourtours en magenta. On notera l'utilisation de la macro `\halign` avec la séquence `\bcmagenta\tvc#\endcolor&` ce qui provoque un passage à la couleur magenta pour tous les traits verticaux séparant les colonnes. Pour une explication détaillée de la séquence `#&` dans l'environnement d'un `\halign`, le lecteur est invité à se reporter à [7], pages 126 et suivantes.

```
$$\vbox{\offinterlineskip
\def\cc#1{\kern .2em\hfill\raise 1.5mm%
                \hbox{#1}\hfill\kern .2em}
\def\tvi{\vrule height 24pt depth 2pt width 0pt}
\def\tvc{\tvi\vrule width 1mm}
\halign{\bcmagenta\tvc#\endcolor&%
&\cc{$\displaystyle#$}&\bcmagenta\tvc#\endcolor\cr
\noalign{\bcmagenta\hrule height 1mm\endcolor}
&\omit\cc{\bf Fonction}&&\omit\cc{\bf Primitive}&\cr
\noalign{\bcmagenta\hrule height 1mm\endcolor}
     height 34pt depth 30pt&{ 1 \over 1 + x^2 }&&{\rm
         Arctg}\,(x)+C&\cr
\noalign{\bcmagenta\hrule height 1mm\endcolor}
}}$$
```

Fonction	**Primitive**
$\frac{1}{1}+x^2$	$\mathrm{Arctg}\,(x)+C$

Pour coder une boîte en couleur avec un cadre dans une autre couleur :

```
{\Large
\fboxsep=0pt% on initialise le blanc
\fboxrule=3pt% la dimension du cadre est de 3pt
\fboxcolor{\fboxsep=8pt\fboxrule=0pt
    \boxcolor{du texte en noir}%
            {255}{255}{0}}{255}{0}{0}
}
```

du texte en noir

Pour tracer un cercle avec des dégradés concentriques qui partent de la couleur rouge jusqu'au noir en décalant le centre (effet boule dégradée), on programme :


```
\setbox100=\vbox{\Huge\bf
\halign{\quad#&\quad#&\quad#\cr
\begincolor{255}{255}{255}
& LRI& \cr
LRI& LRI& LRI\cr
& LRI& \cr}
\endcolor}
\xtrans=50pt% on decale le centre de 50pt vers la droite
\ytrans=50pt% et 50pt vers le haut
\CercleGeneral{\copy100}{255}{0}{0}{0}{0}{0}{10}
```

Pour tracer une boîte dont le contenu est dégradé, on programme :

Dans un article récent, nous présentions les avantages d'utilisation de la couleur dans le traitement de texte TEX au travers de deux jeux de macros : ColorTEXrgb et colorTEXcmyk fonctionnant pour les imprimantes couleur PostScript des constructeurs QMS, Océ, Hitachi, Seiko.

```
\setbox10=\vbox{\hsize=0.6\textwidth\bf
\begincolor{255}{255}{255}% texte blanc
Dans un article r\'ecent, nous pr\'esentions
les avantages d'utilisation de la couleur dans le
traitement de texte \TeX\  au travers de deux jeux
de macros~: Color\TeX rgb et color\TeX cmyk fonctionnant
pour les impri\-man\-tes couleur PostScript des constructeurs
QMS, Oc\'e, Hitachi, Seiko.

\par
}
\horizontaltrue % des degrades de haut en bas
\centerline{%
\RectangleDegrade{\copy10}{255}{255}{0}{0}{0}{0}{50}}
```

Pour tracer les courbes de bezier des figures .1 et .2, on pourra marquer avec `Pictex` ou `TexCad` les points définissant les tangentes, puis récupérer la valeur de ces points sous éditeur et programmer :

```
\begin{figure}[th]
\begin{center}
\unitlength=1.00mm
\linethickness{0.4pt}
\begin{picture}(74.00,61.00)
\put(0.00,0.00){\framebox(74.00,61.00)[t]{}}
\put(3.97,10.00){\vector(1,0){28.10}}
....
% debut de la premiere courbe de bezier (celle de gauche)
\put(0,0){\bezierps [ [ 2.06 36.98 6.03 39.00 10.95 39.00
 14.03 29.02][14.03 29.02 19.05 16.06 23.02 16.98 30.00 13.97]]
 \endbezierps{.4}{3}{2}{255}{0}{0}}
```

```
% fin de la premiere courbe de bezier (celle de droite)
% debut de la 2 courbe de bezier (celle de gauche)
\put(0,0){\bezierps [[33.97 39.00 37.94 39.00 40.00 37.94
42.06 34.92] [42.06 34.92 44.92 26.98 50.95 16.98 66.98 12.06]]
\endbezierps{.4}{3}{2}{255}{0}{0}}
% fin de la 2 courbe de bezier
% je place un cercle hachure
\put(6.03,38){\CercleHachure{4.5}{.5}{.1}{60}{255}{60}{255}{2}}
\put(42,35){\CarreHachure{4}{.6}{.1}{135}{255}{0}{255}{1}}
....
\put(6.03,54){\makebox(0,0)[cc]{$f(x)$}}
\put(42,54){\makebox(0,0)[cc]{$f(x)$}}
\end{picture}
\caption{{\it extremum particulier}}
\label{extre}
\end{center}
\end{figure}
```

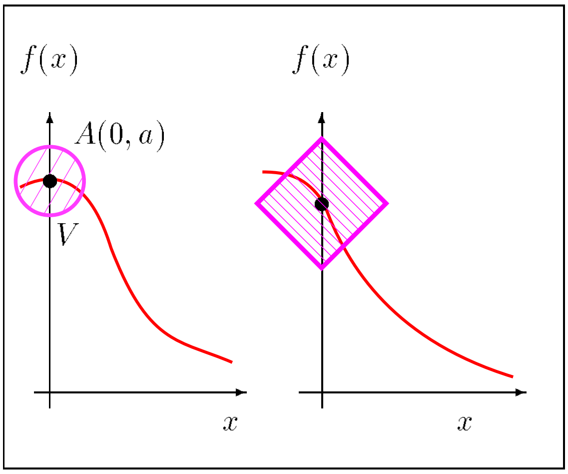

Figure .1 : *extremum particulier*

```
\begin{figure}[th]
\begin{center}
\unitlength=1.00mm
\linethickness{0.4pt}
\begin{picture}(74.00,61.00)
\put(0.00,0.00){\framebox(74.00,61.00)[t]{}}
....
% debut de la courbe de bezier
\put(0,0){\bezierps [
[ 16.03 14.92 20.00 16.03 26.98 39.05 33.97 43.02 ]
```

```
[ 33.97 43.02 42.06 40.95 43.97 20.95 47.94 19.05 ]
[ 47.94 19.05 52.06 20.95 54.92 49.05 56.98 52.06 ]
[ 56.98 52.60 60.00 49.05 62.06 32.06 64.92 17.94 ]
] \endbezierps{.4}{2}{.5}{255}{0}{0}}
....
\end{picture}
\caption{{\it extremums locaux et globaux}}
\label{extretwo}
\end{center}
\end{figure}
```

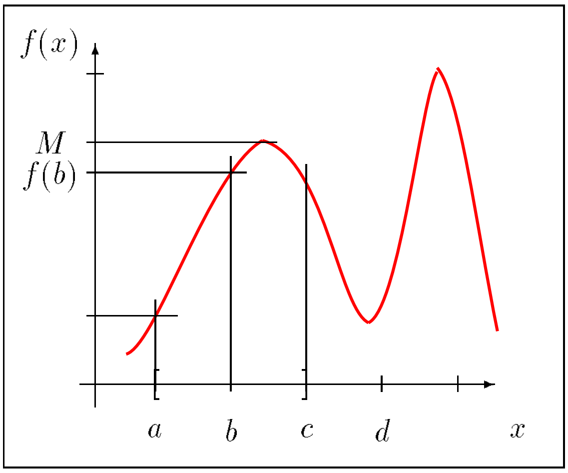

Figure .2 : *extremums locaux et globaux*

Pour tracer des objets quelconques au moyen de courbes de Bezier dont le fond est en couleur, nous programmons :

```
\begin{figure}
\begin{center}
\unitlength=1.00mm
\linethickness{0.4pt}
\begin{picture}(124,100)
\put(0,0){\framebox(124,100)[cc]{}}
\put(20,80){\makebox(0,0)[cc]{\Large\bf\LaTeX}}
\put(97,5){\makebox(0,0)[cc]{\small
          $\copyright - lri - 1991$}}
% Definition de la couleur du contour.
% Le trait est en pointille.
\rouge=255\vert=0\bleu=255\dashpstrue
```

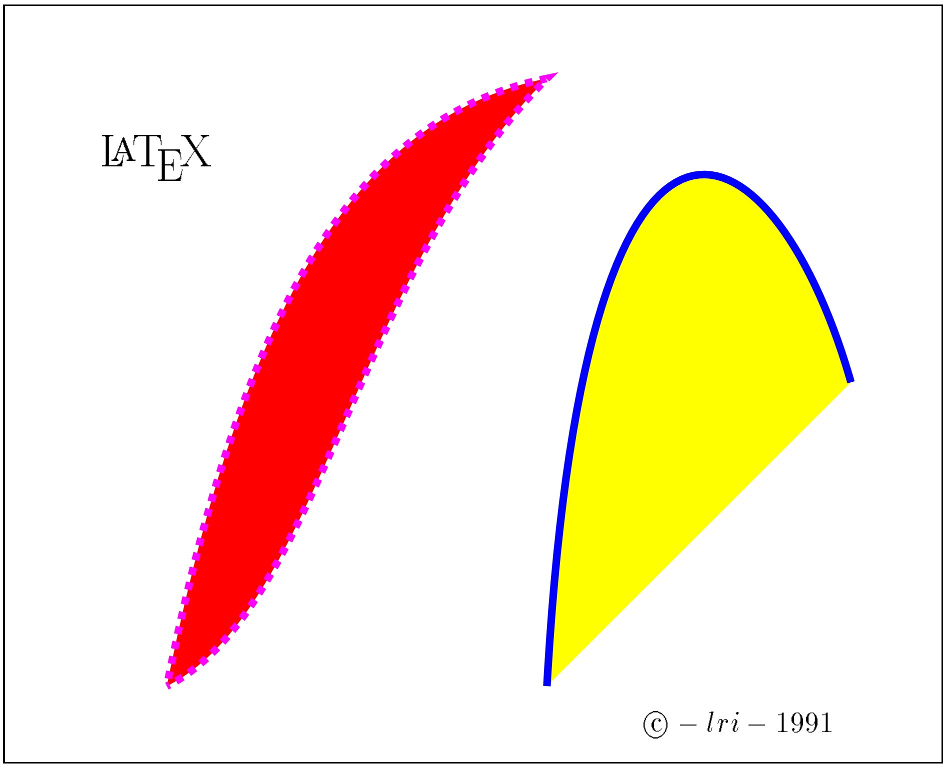

Figure .3 : *Des courbes remplies*

```
\put(0.00,0.00){\CourbeFermee [
[ 20 10 10 50 25 75 50 80 ]
[ 50 80 25 55  20 10 0 0 ]
] \endcourbe{1}{1}{1}{255}{0}{0}}
% Le contour n'est plus en pointille
\dashpsfalse\closepathfalse
\rouge=0\vert=0\bleu=255
\put(0.00,0.00){\CourbeFermee
   [ [ 70 10 5 90 30 75 40 40 ]
] \endcourbe{1}{1}{1}{255}{255}{0}}
\end{picture}
\caption{{\it Des courbes remplies}}
\label{f:remplie}
\end{center}
\end{figure}
```

qui produit le résultat de la figure .3

Pour placer du texte le long de courbes de Bezier il faut programmer, par exemple, qui produit le résultat de la figure .4 :

```
\Huge\bf
\BlancInterLettre=1% 1 mm entre les lettres
\begin{figure}[htp]
\begin{center}
\unitlength=1.00mm
\linethickness{0.4pt}
\begin{picture}(95,70)
\put(0,0){\TraceTexte{TraceTexte}
      {5}{65}{15}{35}{35}{35}{90}{65}}
\put(0,0){\ColleTexte{ColleTexte}
      {5}{5}{15}{35}{15}{35}{81}{5}}
\put(0.00,0.00){\framebox(95,70)[cc]{}}
\end{picture}
\caption{{\it courbes sur un texte}}
\label{fd:one}
\end{center}
\end{figure}
```

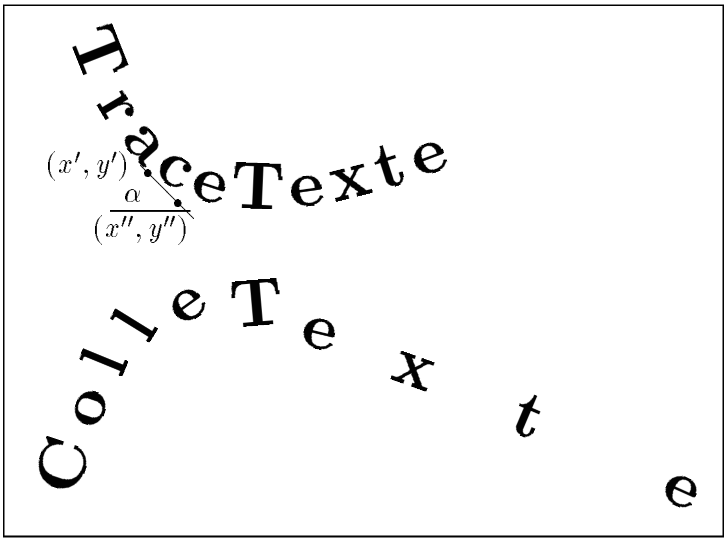

Figure .4 : *courbes sur un texte*

Bibliographie TEX

[1] Christophe Cérin et Benoît Lemaire *Vers l'introduction de la couleur dans TEX*, Cahiers GUTenberg numéro 5 – pages.

[2] Adobe Systems Incorporated. *PostScript Language Reference Manual*, Addison-Wesley. October 1986.

[3] Adobe Systems Incorporated. *PostScript Language Tutorial and Cookbook*, Addison-Wesley. March 1987.

[4] Adobe Systems Incorporated. *PostScript Language Program Design*, Addison-Wesley. February 1988.

[5] Donald E.Knuth. *The TEXbook*, Addison-Wesley. 1986.

[6] Leslie Lamport. LATEX *user's guide & reference Manual*, Addison-Wesley. 1986

[7] Raymond Seroul. *Le petit livre de TEX*, InterEdition – Mai 1989