
Journal de l'OSGeo

Le Journal de la Fondation Open Source Geospatial

Volume 1 / Mai 2007

Dans ce volume

Développement de logiciels Open Source

Introduction à Mapbender, deegree, openModeller ...

Comprendre les relations spatiales

Examen de la spécification du Web Processing Server (WPS)

Interaction des logiciels - GRASS-GMT, Tikiwiki, PyWPS, GRASS-R ...

Mises à jour des logiciels

Actualités, et plus ...





**2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOSS4G) CONFERENCE**
VICTORIA CANADA  SEPTEMBER 24 TO 27, 2007

FOSS4G - Ouverture des Inscriptions à la Conférence

Nous sommes heureux de vous annoncer l'ouverture des inscriptions en ligne à la Conférence Free and Open Source Software for Geospatial 2007 (FOSS4G 2007). FOSS4G est l'évènement annuel qui réunit les personnes et les sociétés qui créent, utilisent, et gèrent des logiciels géospatiaux open source. Inscrivez-vous dès maintenant en ligne.¹

Inscrivez-vous avant la date limite du 27 Juillet, pour économiser sur les frais d'inscription! Tirez profit de l'opportunité que FOSS4G 2007 vous offre, de construire un réseau avec les autres professionnels des données géospatiales, de renouveler d'anciennes relations, et d'en créer de nouvelles.

Pour les dernières mises à jour, l'inscription et/ou la soumission d'une présentation, visitez le site web de la conférence.²

OPPORTUNITES D'EXPOSITION & DE SPONSORING

Concernant les opportunités d'exposition et de sponsoring, lisez la page des partenaires³ ou contac-

tez Paul Ramsey, Président de la Conférence par email.⁴

SOUMETTRE UNE PRESENTATION

Vous pouvez soumettre une présentation en ligne.⁵ La date limite pour les soumissions est le 29 Juin 2007.

Les présentations FOSS4G durent 25 minutes, avec 5 minutes de questions/réponses à la fin. Les présentations concernent l'utilisation ou le développement de logiciels géospatiaux opensource. Tout le monde peut soumettre une proposition de présentation et participer à la conférence comme présentateur. Plus d'informations sont disponibles sur la page des présentations sur le site web.

Nous espérons vous voir à Victoria, au Canada en Septembre!

¹Inscription en ligne : <http://www.foss4g2007.org/register/>

²Site web de la conférence : <http://www.foss4g2007.org/>

³Page des partenaires : <http://foss4g2007.org/sponsors>

⁴Email Paul Ramsey : pramsey@foss4g2007.org

⁵Soumettez une présentation sur <http://www.foss4g2007.org/presentations/>

Étude d'intégration

Créer des cartes prêtes à être publiées avec GRASS et GMT

par Dylan Beaudette, traduit par Yves Jacolin

Aperçu

La publication de cartes imprimables de haute qualité est un composant important de la plupart des analyses géographiques. Bien que GRASS n'ait pas été écrit pour la production de matériels imprimables, les commandes du type `ps.map` et plus récemment un wrapper bien utile `G-ps.map`, lui permettent de sortir des cartes postscripts de haute qualité. Cependant, la syntaxe et les possibilités plutôt limitées de l'approche `ps.map` produisent souvent une carte qui doit être finalisée dans un logiciel de PAO (Publication Assisté par Ordinateur) tel que Inkscape, Scribus, ou The GIMP. La fonctionnalité composer une carte trouvée dans QGIS peut être une bonne alternative dans un proche futur, mais la version actuelle manque de flexibilité, de stabilité, et de la possibilité d'exporter des données dans des formats tels que EPS ou PDF. Avec de telles options limitées pour la production de cartes à imprimer de grande qualité dans un processus open source, de nombreuses personnes ont bricolé ensemble le nécessaire requis pour

interfacer GRASS avec de nombreux programmes comme le package Generic Mapping Tools. Le projet GMT est très bien décrit par ses auteurs :

GMT est une collection open source d'une soixantaine d'outils pour la manipulation de jeux de données Cartesian et géographiques (comprenant les filtres, l'ajustement de la direction, la création de grilles, la projection etc.) et la production dans des fichiers PostScript Encapsulé (EPS) d'illustrations qui vont du simple graphe x-y en passant par des cartes de contour à la surbrillance artificielle des surfaces et des vues en perspective en 3D. GMT gère environ 30 projections et transformations cartographiques et est fourni avec des données telles que la ligne des côtes, les rivières, les limites politiques. GMT est développé et maintenu par Paul Wessel et Walter H. F. Smith avec l'aide d'une équipe de volontaires, et est géré par la Fondation Nationale des Sciences. Il est diffusé sous la licence GNU General Public.

Cet article décrit les efforts combinés de plusieurs personnes (David Finlayson, Hamish Bowman, Brent Wood, moi-même, et d'autres) pour créer le moyen d'interfacer GRASS à GMT. Plusieurs permutations des commandes du style *.out.gmt ont été créées, en utilisant à la fois des scripts Python et Bash, pour réaliser l'export nécessaire, la configuration, et l'exécution des commandes GMT. Après avoir lu cet article vous devriez avoir une idée générale pour composer de simples cartes dans GMT, avec des scripts modèles pour l'utilisation de données exportées de GRASS. L'article suivant présentera des exemples spécifiques à Spearfish (le jeu de données standard de GRASS) de création de cartes avec GMT.

GMT

Les applications des Outils de Cartographie Géométrique (GMT) peuvent être installées sur n'importe quelle plate-forme soit par des binaires préparés ou en compilant le code source. Les outils GMT peuvent aussi être installés à partir de votre système de gestion de paquets de votre distribution (par exemple aptitude), cependant cette méthode n'est pas recommandée puisque le paquet a de grandes chances de ne pas être à jour. Si vous êtes sur un système d'opération de type UNIX, compiler à partir des sources est relativement aisé, et assure que vous avez une version à jour. En supposant que vous avez un ensemble d'outils de développement en fonctionnement (gcc, make, etc.) à aller sur la page 'Download'⁶, et suivez les instructions. GMT est distribué avec un manuel compréhensible, des cours pour créer des cartes, et des données pour des cartes à différentes échelles. Une fois que vous avez une copie fonctionnelle de GMT installée sur votre machine, et que vous avez téléchargé les données d'exemple⁷ vous pouvez suivre les exemples de cet article. Cette archive contient également un ensemble de scripts approfondis dans plusieurs exemples dans cet article. Le script `template.sh` peut être utilisé pour démarrer pour une approche automatisée pour exporter et tracer des données Raster et vecteur de GRASS avec des outils de GMT. Remarquez que ce script est un point de démarrage et est loin d'être complet.

Les commandes GMT sont lancées depuis la console (avec un jeu complexe de paramètres, switches et d'autres arguments) et envoient les fragments Postscript qui en résultent vers la sortie standard. Le comportement par défaut de toutes les ap-

plications GMT peut être ajusté avec la commande `gmtset`. Cette commande est habituellement utilisée avant toute commande qui produit un rendu, ainsi des éléments clé tels que la taille du papier, l'espace des polices etc. sont établis. La première commande de traçage lancée (généralement `psbasemap`) crée le fichier de sortie initial avec l'opérateur standard ">", tandis que chaque appel suivant ajoute à ce fichier avec l'opérateur d'ajout double ">>". Le premier paramètre `-K` est utilisé avec toutes les commandes GMT avant la dernière commande, ainsi le fichier Postscript en sortie n'est pas finalisé prématurément. Nous utiliserons des données échantillons d'une application Mapserver pour illustrer divers aspects d'utilisation de GMT, avec une idée de comment coupler cela avec GRASS et GMT.

Application échantillon

```
# définir quelques paramètres globaux :
gmtset ANNOT_FONT_PRIMARY Times-Roman \
HEADER_FONT_SIZE 16 \
ANNOT_FONT_SIZE_PRIMARY 12 \
LABEL_FONT_SIZE 14 \
BASEMAP_TYPE plain \
PLOT_DEGREE_FORMAT DF \
PAPER_MEDIA letter+

# la carte échantillon est centrée sur l'est des USA
pscoast -JB-116/36/30/42/7i \
-R-125/-108/31/44 -B5 \
-Gwhite -W0.5p \
-A250 -Dh -Na -Xc -Yc -P -K > query_centers.eps

# Trace les points mapxy :
psxy sample_data/mapxy_locations.latlong -J -R \
-Sc0.075c -W1/1/200/1 -G1/200/1 -O -K >> query_centers.eps

# trace les points ka-map :
psxy sample_data/ka-map_locations.latlong -J -R \
-Sc0.075c -W1/255/1/1 -G255/1/1 -O >> query_centers.eps
```

L'exemple ci-dessus crée un graphique de l'Est des États-Unis, avec des points rouges et verts symbolisant l'activité spécifique à l'endroit d'un site Internet. Le cadre de la carte est créé avec la commande `psbasemap`, les lignes de côtes et limites politiques sont créées par `pscoast`, et les points sont créés avec la commande `psxy`. La localisation des points est stockée comme des paires simples de longitude-latitude au format texte, tandis que la ligne de côte est dessinée à partir du jeu de données inclus dans GMT. Le résultat peut être vu à la Figure 1.

Une explication complète des diverses options des lignes de commandes peut être trouvée dans la page de manuel de chaque commande. De plus, un

⁶http://gmt.soest.hawaii.edu/gmt/gmt_download.html

⁷http://169.237.35.250/~dylan/GRASS/newsletter/sample_data.tar.gz

exemple plus complet de GMT réalisé en dehors de GRASS peut être trouvé sur le site de l'auteur ⁸. Dans l'exemple ci-dessus, les paramètres de projection et de région sont définis manuellement par l'utilisateur. Des éléments de formattage additionnels (tel que le paramètre `-P` pour un affichage en portrait) doivent être définis manuellement et définis en ligne de commande. La génération automatique d'une carte dépend alors, en partie, d'une approche dynamique pour définir les options passées aux commandes GMT. Un langage de script, capable d'extraire les informations de région de GRASS à partir d'une session de GRASS en cours est une approche idéal pour automatiser cette tâche.

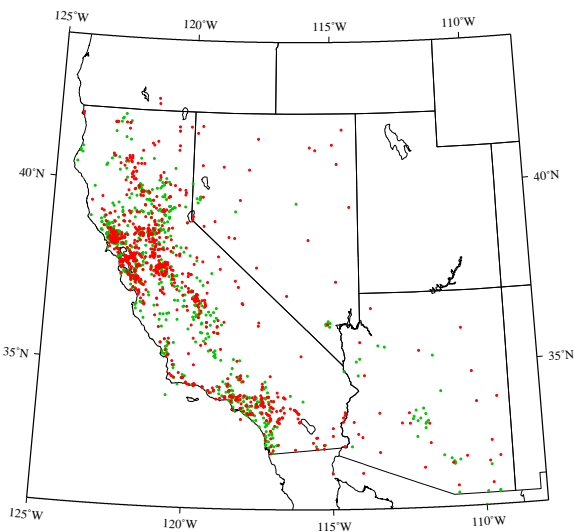


FIG. 1 – Localisations de requête, symbolisé par une application cartographie active. Les points verts représentent une application Mapserver basée sur LandView, les points rouges représentent une application Mapserver basée sur Ka-map.

⁸<http://casoilresource.lawr.ucdavis.edu/drupal/node/102>

⁹On peut trouver plus d'informations sur l'interpréteur de script GRASS ici http://grass.itc.it/gdp/html_grass63/g.parser.html

Coupler GRASS et GMT

L'approche générale pour interfacier GRASS et GMT peut être résumée comme suit :

- collecter des informations à partir d'utilisateurs sur :
 - les données en entrées
 - les options de formattage
 - les options du rendu
- collecter des paramètres géographiques à partir de la région GRASS
- calculer les paramètres clés de mise en page
 - information sur la projection
 - limites d'échelle
 - intervalles des échellons
- exporter les données raster de GRASS vers le format `.gd` de GMT
- exporter les données vecteurs de GRASS vers le format ASCII de GMT
- lancer les commandes GMT avec les paramètres ci-dessus

Collecter des informations à partir de l'utilisateur

Le parseur de script GRASS fournit une approche aisée à la création de commandes GRASS personnalisées, sans la nécessité de comprendre le langage C. Un script habile et rigoureux peut être utilisé pour parser des détails sur le jeu de données de GRASS comme des informations sur les paramètres de la région en cours aux commandes GMT, en utilisant la syntaxe familière et la structure de l'interface d'un module GRASS. Avec ces options sauveées dans des variables locales il est possible de modifier les paramètres par défaut tels que la taille du papier, la taille des polices et l'orientation du texte en lançant le programme `gmtset` avec nos variables locales substituées par l'interpréteur de script ⁹.

Collecter des informations géographiques

Extraire des informations clé des paramètres de la région en cours (pour une utilisation des paramètres plus tard dans GMT) peut être réalisé de plusieurs manières. Parmi celles-ci une implique les outils `g.region` et de traitement de texte, grâce à Bruce Raup pour la suggestion élégante de `eval`. Remarquez que tous les exemples suivants nécessitent que vous soyez dans une session GRASS active.

```

# necessite d'etre dans une sessions GRASS active
# pour que cela fonctionne
#
# étendu du secteur : utilise avec le parametre -R
# resolution : utilise avec le parametre -I
#
# approche elegante pour extraire l'etendue et
# la resolution du secteur comme suggere par
# Bruce Raup - National Snow and Ice Data Center
#
region_assignments=`g.region -g`
eval $region_assignments

# definis les parametres pour les commandes GMT
region="-R$w/$e/$s/$n"
inc="-I$wres/$nsres"

# extrait les valeurs d'etendues pour calculer
# le rapport de la carte
extent_assignments=`g.region -ge`
eval $extent_assignments

# preserve le rapport a partir des coordonnÃ'es UTM E,N :
aspect_ratio=`echo $ns_extent $ew_extent
| awk '{printf("%f", $1 / $2)}'`

# calcul la longueur de la carte base sur
# le rapport orginel
map_length=`echo $map_width $aspect_ratio
| awk '{printf("%f", $1 * $2)}'`

# calcul la localisation y (dans l'unite de la feuille)
# pour le champ titre de la page
field_sheet_title_y=`echo $map_length
| awk '{printf("%f", $1 + 0.33)}'`

# compile la chaine de projection, avec
# les variables width/length
# projection linaiere, largeur en pouce / hauteur en pouce
projection_string="X${map_width}i/${map_length}i"

```

Une approche plus robuste impliquerait la liaison GRASS-Python pour extraire directement les paramètres de la région. Une fois que les données de la région ont été extraites et sauvees dans des variables locales, nous pouvons réaliser les calculs nécessaires pour obtenir les autres paramètres tels que l'échelle de la carte, le rapport de l'aspect de la carte, la taille de la carte sur le papier, etc. Les paramètres de la région sont utilisés pour le paramètre *de traçage de la région* (-R\$w/\$e/\$s/\$n), souvent utilisé seulement dans la première fonction de traçage. Toutes les fonctions de traçages suivantes peuvent hériter de ces valeurs en définissant des paramètres vides -J et -R en fonction du paramètre -O (superposition).

Les projections des cartes dans GMT

L'approche traditionnelle pour utiliser les outils GMT implique le traçage de données, encodé en paire longitude-latitude, dans une projection définie

avec le paramètre -J. Les coordonnées sont transformées au fur et à mesure que la sortie Postscript est générée. Comme noté dans l'exemple de code ci-dessus, une projection Alber Equal Area centrée sur (116°W, 36°N), en ayant des parallèles standards à 30°N et 42°N, et qui a 7 pouces de large est encodé par -JB-116/36/30/42/7i. Lisez le manuel de GMT pour une liste complète de projections gérées et de leurs paramètres.

Lorsqu'on travaille dans GRASS, les données sont habituellement (mais pas toujours) dans une projection géographique et donc le mode de projection *linear* doit être utilisé. Une fois qu'une largeur et une longueur appropriées de la carte (dans ce cas en utilisant des pouces) ont été établies les paramètres de la projection linéaire sont encodées par -JX\${map_width}i/\${map_length}i. Remarquez que les unités métriques ou impériales peuvent être utilisées dans GMT et passer de l'un à l'autre implique seulement l'ajout d'un 'c' pour centimètre et 'i' pour pouces ([NdT] inches en anglais) après le nombre.

Exporter et tracer des données raster GRASS

Exporter une donnée raster GRASS vers un format compatible GMT peut être réalisé avec une combinaison de r.out.bin et xyz2grd (inclu dans la distribution GMT). L'utilisation correcte de xyz2grd nécessite des connaissances dans les données raster en entrée : c'est à dire entier ou virgule flottante, etc. Pour une cellule raster exportée avec r.out.bin, le paramètre -ZTLh pourrait être utilisé pour renseigner xyz2grd que le flux de données en entrées est de type entier court orientée. Pour les cartes FCELL et DCELL les paramètres correspondants pourraient être respectivement -ZTLf ou -ZTLd. Le paramètre -F instruit xyz2grd pour utiliser l'enregistrement des pixels des grilles des cellules, le même enregistrement utilisé pour les rasters GRASS. Une simple déclaration CASE peut être utilisée dans nos exemples de scripts pour créer automatiquement l'appel à xyz2grd. Un paramètre optionnel dans notre exemple de script peut être utilisé pour sauter cette étape pour faciliter la récupération des autres paramètres sans perdre de temps de traitement pour exporter les données raster. L'auteur a trouvé que définir la région par une résolution entière entraîne l'opération la plus fiable de xyz2grd.

```

# obtenir le type de la carte d'un raster donnee
MAP_TYPE=`r.info -t "$output_raster" | cut -f2 -d'|`

# exporter le raster base sur son type,

```

```
# ses parametres de regionde l'exemple ci-dessus
# et les parametres de resolution de l'exemple plus haut
case "$MAP_TYPE" in
CELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLh -F -N-9999 ;;
FCELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLf -F -N-9999 ;;
DCELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLD -F -N-9999 ;;
esac
```

Une fois que les données rasters ont été exportées au format .grd nous sommes presque prêts pour leur utilisation avec les commandes de dessins de GMT telles que `psimage`. Cette commande peut soit utiliser un seul fichier grille, avec un fichier de palette de couleurs (.cpt), soit avec des canaux de couleurs séparées rouge, vert, bleu ¹⁰.

Exporter les données des tables de couleur GRASS

Pour l'instant il n'y a pas de méthode *simple* disponible pour convertir les informations de la table des couleurs de GRASS en un fichier de palette de couleurs fiable pour GMT. Les développements existants sont capables d'approcher le format .cpt, cependant l'auteur n'a pas été capable d'utiliser avec succès ces fichiers générés automatiquement. Un script de conversion fonctionnel (de préférence écrit en un langage de script de haut niveau tel que Python) sera posté prochainement sur le wiki de GRASS dès qu'il sera complet ¹¹.

GMT fournit plusieurs outils pour créer manuellement un fichier de palette de couleur dans un domaine de données fourni par l'utilisateur, ou un domaine de données lu à partir d'un fichier grid de GMT. Bien que cela ne soit pas une approche idéale, il peut être utile quand une des palettes de couleur de GMT est appropriée. Dans le cas spécial où l'image exportée est en nuance de gris, la palette de couleur standard "gray.cpt" peut être utilisée pour créer un nouveau fichier cpt avec le programme `makecpt`.

```
# creer une palette de couleur fiable pour une image
# raster CELL en nuance de gris, les valeurs de grille
# sont des entiers dans un domaine compris de 0-255
makecpt -Cgray -T0/255/1 -V > doqq.cpt
```

Lorsque l'on travaille avec des canaux de couleurs qui ont déjà été séparés (par exemple, landsat ou d'autres images multispectrales) il est possible d'utiliser `psimage` sans un fichier de palette de couleurs, au lieu de définir des fichiers de grilles en entrée en rouge, vert et bleu. Cette approche est plus aisée lorsque l'on travaille avec des données qui ont été collectées dans des bandes séparées. Pour des données à une seule bande, la séparation des couleurs peut être créée avec `r.mapcalc`.

```
# obtenir les separations de couleurs pret pour GMT
```

```
r.mapcalc "some_raster.red = r#some_raster"
r.mapcalc "some_raster.blue = b#some_raster"
r.mapcalc "some_raster.green = g#some_raster"
```

Exporter et tracer des données vecteurs GRASS

Le format vecteur de GMT est basé sur un format ASCII basé sur des vertex, très simple, désigné au début pour stocker des données géométriques. Cependant, avec des scripts il est possible d'encoder des données attributs simples dans ce format pour la production de cartes thématiques. Une approche alternative implique que les données vecteurs qui ont été pré-filtrées avec un outil GRASS tel que `v.extract`, en plusieurs fichiers où chacun représente une classe spécifique. En utilisant la première approche (encoder les attributs dans un fichier vecteur GMT) la symbologie pour chaque point, ligne et polygone est encodée dans le fichier, tandis que la symbologie est définie avec des paramètres en ligne de commande pour la seconde approche.

Données points

Exporter des données ponctuelles est la conversion la plus simple de GRASS → Vecteur GMT. Dans la plupart des cas, où un ensemble de points doit être tracé dans GMT en utilisant une symbologie fixe, la sortie de `v.out.ascii` ou `v.out.ascii.db` peut être directement tracée avec la commande GMT `texttpsxy`. Notez que filtrer la sortie avec l'une ou l'autre commande avec des outils de traitement de texte UNIX est une méthode idéale pour convertir des attributs dans une symbologie ou pour ajuster les étiquettes.

```
# exporter d'un format ASCII de GRASS
v.out.ascii in=points fs=" " > points.xy
```

¹⁰Les définitions des fichiers de palette de couleur de GMT peuvent être trouvées sur http://www.soest.hawaii.edu/GMT/gmt/doc/html/GMT_Docs/node57.html

¹¹De plus amples tests et adaptations du script `r.out.gmt.py` de David Finlayson pourraient être l'approche la plus rapide).


```
# trace les points avec un contour blanc et rempli de
# bleu en utilisant un symbol circulaire de 0.125
# pouces de diametre en ajoutant le resultat Postscript
# a 'outfile.eps'
psxy points.xy -R -J -M -Sc0.125 -G255/1/1 \
-W1/255/255/255 -O -K >> output.eps
```

L'étiquetage des données points peut être réalisé d'abord en filtrant la sortie de `v.out.ascii` avec un simple script `awk`, puis en utilisant la sortie avec la commande `pstext`. Le format des étiquettes utilisé par GMT est documenté dans la page du manuel de `pstext`, cependant une simple explication est donnée dans l'échantillon de code ci-dessous.

```
# export du format de GRASS vers ASCII avec des
# attributs selectionnes (colonne 'ID'), sortie
# filtre avec awk, classement des colonnes,
# et insertion des etiquettes dans le formulaire :
# x_coord, y_coord, font_size, rotation_angle,
# font_number, label_offset, label_text
v.out.ascii.db in=points columns=ID | awk -F'|'
'{print $2, $3, 10, 0, 4, "BL", $4}' \
> points_with_labels.xy
```

```
# trace les textes des etiquettes a partir des
# geometries et des proprietes des etiquettes stockees
# dans 'points_with_labels.xy' les etiquettes seront
# bleu et distante de 0.1 cm dans les directions
# horizontale et verticales definie dans la colonne
# 5 du fichier en entree
pstext points_with_labels.xy -R -J -Dj0.1c/0.1c
-G0/0/255 -K -O >> outfile.eps
```

Données ligne et polygone

Il n'y a pas pour l'instant de méthode directe pour convertir des données lignes et polygone GRASS en un format compatible avec GMT. Cependant, en utilisant une application externe appelée `shp2gmt`¹², il est possible de créer des fichiers vecteurs compatibles avec GMT en exportant d'abord les données GRASS en shapefile puis en les convertissant avec `shp2gmt`. Le texte issu de `shp2gmt` contient tous les attributs stockés dans le shapefile en entrée en fonction des coordonnées du vertex, et peut être facilement filtré avec `awk` pour inclure les paramètres de la symbologie. Un exemple de session GRASS illustrant ce type d'opération est listé ci-dessous.

```
# export de GRASS vers le format shapefile
v.out.ogr -e in=lines dsn=. olayer=lines
v.out.ogr -e in=polys dsn=. olayer=polys
```

```
# convertit du format shapefile vers GMT
shp2gmt lines.shp > lines.xy
shp2gmt polys.shp > polys.xy
```

¹²ce petit programme utile initialement écrit par (Frank Warmerdam), a été modifié par (Mark Fenbers) pour inclure des informations attributaires à partir d'une shapefile. Le code du `shp2gmt` modifié peut être trouvé à <http://www.arcknowledge.com/gmane.comp.gis.gmt.user/2004-01/msg00121.html>

```
# en option filtre le fichier vecteur GMT
# pour inclure la symbologie
awk '
{
# tous les segments lignes avec l'attribut
# 'some_value' sera dessine avec un pinceau
# rouge
if ($0 ~ /some_value/) printf "> -wred\n"
# ... ajoute plus d'amelioration pour les autres classes
# pour les lignes qui ne correspondent pas, pour les
# tracer mot pour mot (c.a.d. les donnees vertex)
else print
}
' lines.xy > lines-thematic.xy

# traces les lignes, avec la symbologie definie dans
# le fichier vecteur GMT
psxy lines-thematic.xy -R -J -O -K >> outfile.eps
```

L'étiquetage des objets lignes et polygones suit la même approche générale que l'étiquetage des objets ponctuels, par l'utilisation de `pstext`. Pour de simples cartes, une approche presque automatique pour étiqueter les données polygonales implique l'export des centroïdes avec `v.out.ascii.db`, comme listé au-dessus. Étiqueter les données lignes nécessite plus d'actions de la part de l'utilisateur, puisque chaque étiquette nécessite des coordonnées et un angle associés. Avec un peu de travail il est possible d'utiliser le fichier créé par `v.label` pour produire un ensemble compatible avec GMT d'instruction de placement d'étiquettes. Un exemple de script `awk`, utilisé dans une session GRASS, est présenté ci-dessus.

```
# creer le fichier d'etiquette GRASS :
# tourner les etiquettes pour correspondre aux segments
# de ligne
v.label -a map=trails column=trail_name labels=trails.lab

# converti le foramt d'etiquette GRASS vers le format GMT
awk '
BEGIN{FS="\n" ; RS="\n\n"}
{
split($1, e, ": ")
split($2, n, ": ")
split($15, r, ": ")
split($16, l, ": ")
# s'il n'y a pas de rotation, nous devons ajouter une
# roration de 0
if(NF == 16) {rotation = r[2] ; label = l[2]}
else {rotation = 0 ; label = r[2]}
# creation les instructions d'etiquettage GMT
print e[2], n[2], 10, rotation, 4, label
}
' DATABASE/LOCATION/MAPSET/paint/labels/trails.lab
> labels.xy
```



```
# dessin avec pstext
pstext labels.xy -R -J -Dj0.1c/0.1c -G0/0/255 -K -0
>> outfile.eps
```

Bien que l'approche au-dessus peut fonctionner pour de simples cartes, une intervention de l'utilisateur est habituellement requise pour des cartes plus complexes. Une approche plus robuste pour l'étiquetage des objets ponctuels, lignes et polygones avec des possibilités avancées telles que la détection des collisions des étiquettes est nécessaire pour une approche entièrement automatisée.

Divers éléments de cartographie

Des éléments supplémentaires tels que le contexte de la carte (psbasemap avec psxy), l'échelle graphique (psbasemap), ou la légende (pslegend) peuvent être ajoutés après que les composants de la carte principale aient été ajoutés au fichier en sortie. Vérifier que la dernière commande GMT n'utilise pas le paramètre `-K` est toujours une bonne idée, afin de s'assurer que le fichier en sortie est finalisé. Par l'utilisation rigoureuse des paramètres `-0`, `-X`, et `-Y` il est possible de redéfinir la région du tracé pour chaque appel d'un programme GMT. Cette approche fonctionne bien pour créer une carte de situation de petite taille, dans une autre carte finalisée. En étendant le premier exemple, vous trouverez ci-dessous un script plus complet pour illustrer l'utilisation de psbasemap pour produire des éléments de la carte tels qu'une échelle graphique, une flèche indiquant le nord, en fonction de la création d'une carte de situation dans la carte principale.

```
# definition de parametres globaux :
gmtset ANNOT_FONT_PRIMARY Times-Roman \
HEADER_FONT_SIZE 16 \
ANNOT_FONT_SIZE_PRIMARY 12 \
LABEL_FONT_SIZE 14 \
BASEMAP_TYPE plain \
PLOT_DEGREE_FORMAT DF \
PAPER_MEDIA letter+

# carte echantillon centree sur l'est des USA
pscoast -JB-116/36/30/42/7i \
-R-125/-108/31/44 -B5 \
-Gwhite -W0.5p \
-A250 -Dh -Na -Xc -Yc -P -K
> query_centers_2.eps

# trace les points mapxy :
psxy sample_data/mapxy_locations.latlong -J -R \
-Sc0.075c -W1/1/200/1 -G1/200/1 -0 -K
>> query_centers_2.eps

# trace les points ka-map
psxy sample_data/ka-map_locations.latlong -J -R \
```

```
-Sc0.075c -W1/255/1/1 -G255/1/1 -0 -K
>> query_centers_2.eps
```

```
# ajout de la barre d'echelle
psbasemap -J -R -Lf-119/32.5/32/150k:"Kilometers": \
-0 -K -P -V >> query_centers_2.eps
```

```
# creer un contexte a la carte
pscoast -JB-116/36/30/42/7i \
-R-130/-100/25/50 \
-X0.25i -Y1.75i \
-Gwhite -W0.5p \
-A250 -Di -Na -P -K >> query_centers_2.eps
```

```
# ajoute la boite de la region
psxy context_box.xy -M -JB -R -W3/1/1/0 -P -0
>> query_centers_2.eps
```

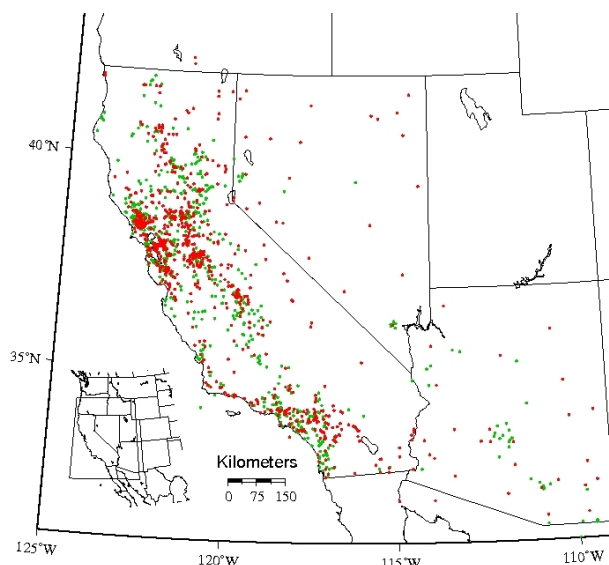


FIG. 2 – Ajout d'un contexte à la carte et d'une échelle graphique.

Limitations actuelles

L'approche actuelle par la console unix pour interfacer GRASS et GMT a plusieurs désavantages : manque de portabilité vers les systèmes non-unix, manque d'opérateurs mathématiques sophistiqués, et une dépendance avec des outils plus utiles tels que `awk`, `sed`, `grep`, etc. Ces désavantages sont particulièrement ennuyeux lors qu'on essaye d'automatiser la construction des éléments d'une carte tel que les intervalles ou les décorations basés sur l'échelle. Les exemples dans cet article utilisent des intervalles définis manuellement, cependant lors de la création de cartes à partir de GRASS (qui est habituellement dans un système de coordonnées projeté) cela prend du temps de rajouter les intervalles annotés ou non pour chaque carte. Regardez le script `tem-`

plate.sh inclus dans l'archive des données d'échantillon pour avoir une idée. Encore une fois, une intégration étroite avec GRASS via la liaison Python résoudreait le problème ci-dessus. En attendant qu'un développement entièrement en Python soit terminé, l'estimation des intervalles automatiques peut être simplifiée avec le script perl suivant.

```
#!/usr/bin/perl -w

# Le 1er argument est l'etendue maximale en unite de la carte
# divise par un nombre raisonnable d'annotations par bord
$x = $ARGV[0] / 10;

print round_up($x) , "\n";

# arrondi a une echelle raisonable
# trouve a :
# http://www.perlmonks.org/?node_id=599865

sub round_up {
    my $n = shift;
    my $scale = 10**int(log($n)/log(10));
    $n = 9 if $scale == 1; #magic for single digits
    if ($n > $scale) {
        $n = int($n/$scale+1)*$scale;
    }
    $n;
}
```

Conclusion

Cet article résume les conditions actuelles de conversion des données GRASS au format GMT

pour la production de carte Postscript de grandes qualités. Bien que plusieurs modèles et exemples existent pour automatiser ce processus, pour la production de cartes complexes de considérables modifications sont nécessaires par l'utilisateur. Après des progrès suffisants sur des projets comme l'interface de GRASS et la liaison Python, il pourrait être possible de créer un système plus complet pour améliorer le processus de GRASS → GMT Il y a actuellement un effort mené par Brent Wood pour réviser le format vecteur de GMT qui pourrait simplifier le processus de cartographie thématique avec GMT. De plus Brent est en train de travailler sur l'inclusion de la gestion en écriture pour le format vecteur GMT dans OGR (GDAL). Une fois terminé, l'export vectoriel de GRASS vers GMT pourrait être aussi simple que `v.out.ogr format=GMT`. Le travail proposé par Wolf Bergenheim, sur un algorithme général de détection et de correction de collision des étiquettes pour `v.label`, pourrait être une excellente solution pour les tâches d'étiquetage à la fois dans GRASS et GMT. Nous verrons la prochaine fois un exemple plus en profondeur de création d'une carte complexe en utilisant des données de GRASS à partir de l'échantillon de données Spearfish.

Dylan Beaudette

Université de Californie à Davis

<http://casoilresource.lawr.ucdavis.edu>
debeaudette AT ucdavis.edu



**2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOSS4G) CONFERENCE**
VICTORIA CANADA  SEPTEMBER 24 TO 27, 2007

FOSS4G - Ouverture des Inscriptions à la Conférence

Nous sommes heureux de vous annoncer l'ouverture des inscriptions en ligne à la Conférence Free and Open Source Software for Geospatial 2007 (FOSS4G 2007). FOSS4G est l'évènement annuel qui réunit les personnes et les sociétés qui créent, utilisent, et gèrent des logiciels géospatiaux open source. Inscrivez-vous dès maintenant en ligne.¹³

Inscrivez-vous avant la date limite du 27 Juillet, pour économiser sur les frais d'inscription! Tirez profit de l'opportunité que FOSS4G 2007 vous offre, de construire un réseau avec les autres professionnels des données géospatiales, de renouveler d'anciennes relations, et d'en créer de nouvelles.

Pour les dernières mises à jour, l'inscription et/ou la soumission d'une présentation, visitez le site web de la conférence.¹⁴

OPPORTUNITES D'EXPOSITION & DE SPONSORING

Concernant les opportunités d'exposition et de sponsoring, lisez la page des partenaires ¹⁵ ou

contactez Paul Ramsey, Président de la Conférence par email.¹⁶

SOUMETTRE UNE PRESENTATION

Vous pouvez soumettre une présentation en ligne.¹⁷ La date limite pour les soumissions est le 29 Juin 2007.

Les présentations FOSS4G durent 25 minutes, avec 5 minutes de questions/réponses à la fin. Les présentations concernent l'utilisation ou le développement de logiciels géospatiaux opensource. Tout le monde peut soumettre une proposition de présentation et participer à la conférence comme présentateur. Plus d'informations sont disponibles sur la page des présentations sur le site web.

Nous espérons vous voir à Victoria, au Canada en Septembre!

¹³Inscription en ligne : <http://www.foss4g2007.org/register/>

¹⁴Site web de la conférence : <http://www.foss4g2007.org/>

¹⁵Page des partenaires : <http://foss4g2007.org/sponsors>

¹⁶Email Paul Ramsey : pramsey@foss4g2007.org

¹⁷Soumettez une présentation sur <http://www.foss4g2007.org/presentations/>

Rédacteur en chef :Tyler Mitchell - [tmitchell AT osgeo.org](mailto:tmitchell@osgeo.org)**Rédacteur, Actualité :**

Jason Fournier

Rédactrice, Étude de cas :

Micha Silver

Rédacteur, Zoom sur un projet :

Martin Wegmann

Rédacteur, Étude d'intégration :

Martin Wegmann

Rédacteur, Documents de programmation :

Landon Blake

Remerciements

Tous les relecteurs & le projet Actualités de GRASS

Le *journal de l'OSGeo* est une publication de la *Fondation OSGeo*. La base de ce journal, les sources du style $\LaTeX 2_{\epsilon}$ ont été généreusement fournies par l'équipe éditoriale de l'actualité de GRASS et R.



This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this licence, visit :

<http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

Journal en ligne : <http://www.osgeo.org/journal>

Site de l'OSGeo : <http://www.osgeo.org>

Contact postal pour l'OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8



ISSN 1994-1897