



Manuel d'utilisation de MIMOSA



développements :	M.-A. Drouin, N. Huret, R. Thiéblemont		
Rédigé par :	M. -A. Drouin, N. Huret		
Maintenance :	C. Boonne		
Version : 8.1	20/05/2010	Création version 8.0	
	19/09/2011	Modification de la partie II. suite à la modification des scripts Modification de la partie IV. suite à la modification des programmes IDL Ajout de la partie V. Autres outils	
	05/12/2011	Ajout du cas du compilateur INTEL dans la partie I.1. Pré requis	
	17/12/2012	Mise à jour code Mimosa, version 8.1	

Notations	2
Introduction	2
I. Installation & compilation	3
I.1. Pré requis	3
I.1.1. Installation de <i>jasper</i>	3
I.1.2. Installation de <i>grib_api</i>	3
I.2. Dossier de travail	4
I.2. Compilation de MIMOSA	5
II. Récupération des données d'initialisation	6
II.1. Données en niveaux pression (<i>mimosa_ecmwf_ECMR.sh</i>)	7
II.2. Données en niveaux modèle (<i>mimosa_ecmwf_grib.sh</i>)	8
III. Utiliser MIMOSA	9
III.1. Namelist run (L12 à L53)	11
III.2. Namelist grid (L57 à L85)	11
III.3. Namelist config (L89 à L123)	12
III.4. Namelist output (L127 à L151)	12
III.5. Démarrer une simulation	12
III.6. Format des fichiers	12
III.6.1. Fichiers de champs de PV, température et vent	12
III.6.2. Fichiers Stations	13
IV. Outils IDL de lecture et traçage des sorties MIMOSA	13
IV.1. Liste des programmes fournis	14
IV.2. Lecture des fichiers binaires	14
IV.3. Traçage des contours	14
IV.4. Traçage des surfaces de volume	16

V. Autres outils.....	18
V.1. Scripts mimosa.sh.....	18
V.2. Script convert_PNG_to_VIDEO.sh.....	19
Annexe	21
Exemples de tracés avec mimosa_plot_pv.....	21
Exemples de tracés avec mimosa_plot_t.....	23
Exemples de tracés avec mimosa_plot_wind.....	25

Notations

ASCII	American Standard Code for Information Interchange
AVC	Advance Video Coding
ECMWF	European Centre for Medium-Range Weather Forecasts
GRIB	GRIdded Binary
MARS	Meteorological Archival and Retrieval System
MIMOSA	Modélisation Isentrope du transport Méso-échelle de l'Ozone Stratosphérique par Advection
mp4	Extension officielle du format MPEG-4
MPEG	Moving Picture Experts Group
PV	Vorticité Potentielle (pvu, $1 \text{ pvu} = 1 \text{ K.kg}^{-1}.\text{m}^2.\text{s}^{-1}$)
x264	Bibliothèque libre permettant de coder des flux vidéo MPEG-4 AVC

Références

Bibliothèque IDL Coyote	http://www.idlcoyote.com/
ECaccess/ECtrans	http://www.ecmwf.int/services/ecaccess/guide/
ffmpeg	http://ffmpeg.org/
grib_api	http://www.ecmwf.int/publications/manuals/grib_api/index.html
imagemagick	http://www.imagemagick.org/script/index.php
jasper	http://www.ece.uvic.ca/~frodo/jasper/
Lait PV	Lait, 1993
MARS	http://www.ecmwf.int/publications/manuals/mars/guide/index.html
MIMOSA	Hauchecorne et al., 2002

Introduction

MIMOSA est un modèle haute résolution d'advection de vorticité potentielle développé en FORTRAN par A. Hauchecorne (Hauchecorne et al., 2002). Il est initialisé à un instant t à partir des données (champs de vents horizontaux U et V , température et pression) de l'ECMWF sur une grille orthogonale centrée sur le pôle nord. MIMOSA calcule puis advecte la vorticité potentielle sur des surfaces isentrope avec une résolution de $1/3$ ou $1/6$ de degré en latitude et en longitude.

MIMOSA a été mis à jour pour permettre l'utilisation de données d'entrée ECMWF en niveaux modèle.

Ce document décrit les prés requis nécessaires à l'utilisation de la nouvelle version de MIMOSA ainsi que comment réaliser une simulation avec les différents types d'entrée.

I. Installation & compilation

I.1. Pré requis

Pour utiliser MIMOSA, il est nécessaire qu'un compilateur FORTRAN soit installé. Le modèle a été testé avec GFORTRAN (version 4.3) et PGF95 (version 8 à 11).

Pour utiliser MIMOSA en niveaux modèle, il est nécessaire d'installer les bibliothèques *grib_api* et *jasper*. Ces deux bibliothèques sont disponibles gratuitement et sont compatibles avec la plupart des compilateurs FORTRAN.

Il est à noter que pour la plupart des distributions linux, il est possible d'installer ces librairies par l'intermédiaire du gestionnaire de paquet de votre distribution. Par exemple pour les distributions utilisant des paquets au format rpm, ces bibliothèques sont disponibles sur le dépôt rpmforge. La version de jasper ne varie pas depuis quelques années mais ce n'est pas le cas de la bibliothèque *grib_api*. La version la plus récente est toujours celle disponible sur le site de l'ECMWF (MIMOSA a été testé avec les versions 1.8.0 à 1.9.9). Il faut alors installer les paquets suivants :

- Distribution 32 bits
 - jasper.i386 (version 1.900.1)
 - grib_api.i686
- Distribution 64 bits
 - Jasper.x86_64 (version 1.900.1)
 - grib_api.x86_64

I.1.1. Installation de *jasper*

Au moment de la rédaction de ce manuel la bibliothèque jasper est en version 1.900.1. La bibliothèque peut être téléchargée à l'adresse <http://www.ece.uvic.ca/~mdadams/jasper/#download>. Il faut suivre les étapes suivantes pour installer jasper:

Dans cette description *install_dir_jasper* est le dossier où on veut installer la librairie

```
> unzip jasper-1.900.1.zip
> mkdir install_dir_jasper
> cd jasper-1.900.1
> ./configure --prefix=install_dir_jasper
> make
> make install
```

I.1.2. Installation de *grib_api*

Au moment de la rédaction de ce manuel, la bibliothèque *grib_api* est en version 1.9.9. Elle peut être obtenue à l'adresse http://www.ecmwf.int/products/data/software/download/grib_api.html. Pour l'installer, il faut suivre les étapes suivantes :

Dans cette description *install_dir_jasper* est le dossier où est installé la bibliothèque jasper. Si vous avez installé jasper en utilisant le gestionnaire de paquet de votre distribution linux, il n'est pas nécessaire d'ajouter l'option `--with-jasper=install_dir_jasper`.

Install_dir_grib est le dossier où on veut installer *grib_api*.

```
> gunzip grib_api-1.9.9.tar.gz
> tar xf grib_api-1.9.9.tar
> mkdir install_dir_grib
> cd grib_api-1.9.9
```

```
> ./configure --prefix=install_dir_grib --with-jasper=install_dir_jasper
> make
> make check
> make install
```

1.1.3. Cas particulier du compilateur INTEL fortran

La bibliothèque grib_api n'a pas été prévu par défaut pour être compilé avec les compilateurs INTEL. Avant la compilation, il faut donc forcer l'exécutable configure à utiliser ces compilateurs en tapant la commande suivante :

```
> ./configure F77=ifort FC=ifort CC=icc --prefix=install_dir_grib --with-jasper=install_dir_jasper
```

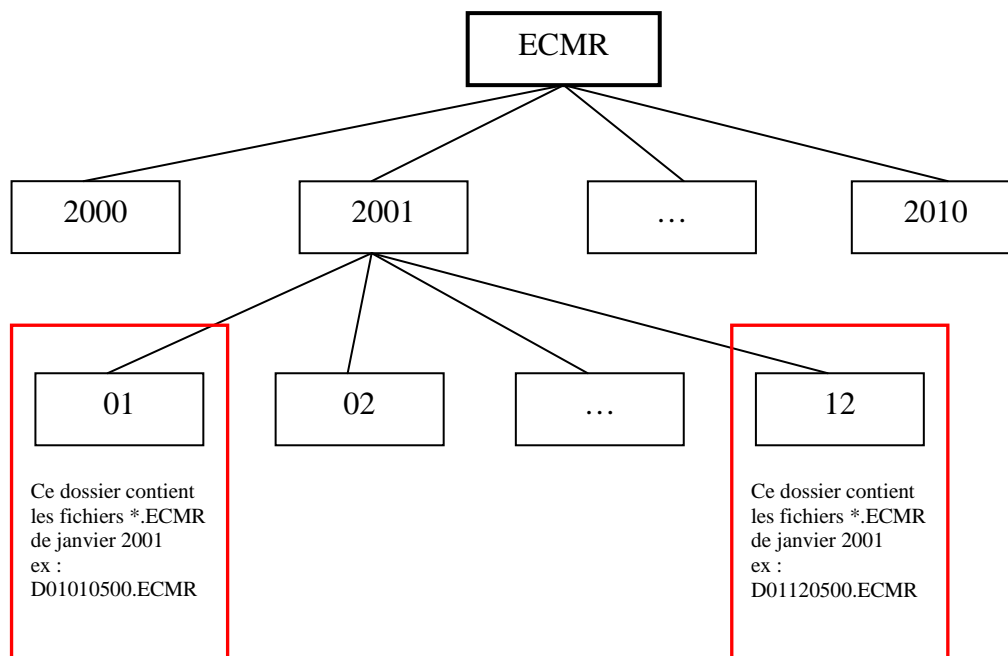
Ensuite, il suffit de suivre les mêmes étapes que dans la partie précédente. Quelques warnings apparaîtront durant la compilation mais ne généreront pas la création de la bibliothèque.

Le cas où on ne dispose pas du compilateur INTEL C, n'a pas été testé.

1.2. Dossier de travail

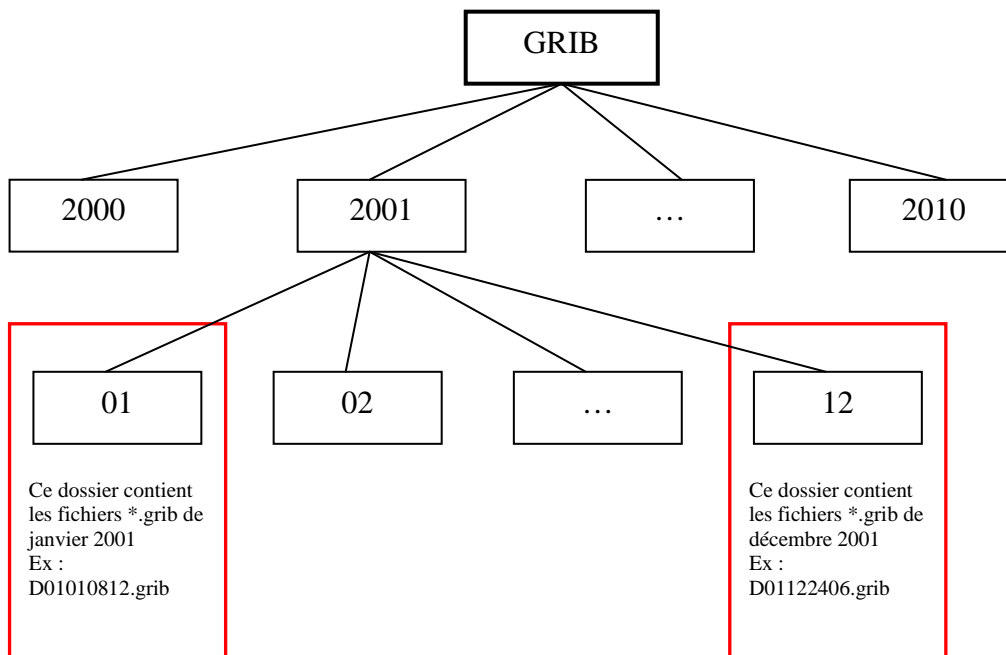
Pour fonctionner, il est nécessaire que certains dossiers soient présents. Ces dossiers dépendent du type de simulation qui doit être réalisé.

Si on veut réaliser une simulation initialisée avec des données en niveaux pression, le dossier où la simulation va être exécutée doit contenir un dossier ECMR qui va contenir les données de la manière suivante :



Les fichiers ECMR ont un format DYYMMDDHH.ECMR.

Si on veut réaliser une simulation à partir initialisée avec des données en niveaux modèle, le dossier où la simulation va être réalisée doit contenir un dossier GRIB qui va contenir les données de la manière suivante.



Les fichiers au format GRIB ont un format DYYMMDDHH.grib.

Le dossier où on exécute la simulation doit aussi contenir le dossier dans lequel vont être stocké les résultats de MIMOSA. Ce dossier aura la forme *RUNXX* où *XX* est le numéro de la simulation choisi dans pour la variable *nrun* dans *input.namelist* (cf. partie III.4). Par exemple si *nrun=6*, les sorties MIMOSA seront enregistrés dans le dossier *RUN06*.

1.2. Compilation de MIMOSA

La compilation de MIMOSA est réalisée grâce au Makefile. Dans l'archive fournie, quatre makefile sont disponibles. Un par compilateur avec lequel la compilation du modèle a été testée (gfortran version 4.1.2 et 4.4.0, pgf95 et ifort). Ils sont nommés *makefile.compileur*. L'exemple ci-dessous présente le cas du compilateur pgf95.

```

1  FF          = pgf95
2  DEBUG      = no
3  SRCDIR     = sources
4  INCPATH    = -I/usr/local/lib64/grib_api-1.9.9/pgf-11.8/include
5  GRIBPATH   = -L/usr/local/lib64/grib_api-1.9.9/pgf-11.8/lib
6  JASPERPATH = -L/usr/lib64
7  ifeq ($(DEBUG),yes)
8    FFLAGS = -fast -g $(INCPATH) -Minfo -Mconcur -V
9  else
10   FFLAGS = -fast $(INCPATH) -Mconcur
11 endif
12
13 LDFLAGS    = $(FFLAGS) $(GRIBPATH) $(JASPERPATH) -lgrib_api_f90 -lgrib_api -lm -ljasper
14 EXEC      = mimosa.x
15 SRC       = $(SRCDIR)/constantes.f90 \
16           $(SRCDIR)/interfaces_mod.f90 \
17           $(SRCDIR)/stations.f90 \
18           $(SRCDIR)/initgrid.f90 \
19           $(SRCDIR)/njour.f90 \
20           $(SRCDIR)/njour_i.f90 \
21           $(SRCDIR)/readecmr.f90 \
22           $(SRCDIR)/readgrib.f90 \
23           $(SRCDIR)/pvp.f90 \
24           $(SRCDIR)/pvp.f90 \
  
```

```

25     $(SRCDIR)/ainterp2d.f90 \
26     $(SRCDIR)/deplacement.f90 \
27     $(SRCDIR)/indgrille.f90 \
28     $(SRCDIR)/xyinterp.f90 \
29     $(SRCDIR)/lisse2d.f90 \
30     $(SRCDIR)/regrid2.f90 \
31     $(SRCDIR)/poids.f90 \
32     $(SRCDIR)/raccord.f90 \
33     $(SRCDIR)/diffusion.f90 \
34     $(SRCDIR)/mimosa.f90
35 OBJ      = $(SRC:.f95=.o)
36
37 all: $(EXEC)
38 ifeq ($(DEBUG),yes)
39     @echo "Generation en mode debug"
40 else
41     @echo "Generation en mode release"
42 endif
43
44 mimosa.x: $(OBJ)
45     @$$(FF) -o $$@ $$^ $(LDFLAGS)
46
47 $(SRCDIR)/%.o: $(SRCDIR)/%.f90
48     @echo $$@
49     @$$(FF) -o $$@ -c $$< $(FFLAGS)
50
51 .PHONY: clean veryclean
52
53 clean:
54     @rm -rf $(SRCDIR)/*.o *.mod
55
56 veryclean: clean
57     @rm -rf $(EXEC)
58
59

```

Avant de pouvoir compiler MIMOSA, il est nécessaire de fournir les chemins vers les bibliothèques jasper et grib_api dans le makefile correspondant au compilateur que vous allez utiliser :

- **INCPATH** : le chemin vers le module décrivant les sous-routines de la bibliothèque grib_api
- **GRIBPATH** : le chemin vers la bibliothèque grib_api
- **JASPERPATH** : le chemin vers la bibliothèque jasper

Il est aussi nécessaire de choisir d'activer ou non les options pour le débogueur en attribuant la valeur **yes** ou **no** à la variable **DEBUG**.

Une fois ces étapes passées, il ne reste plus qu'à lancer le makefile avec la commande :

```
> make -f makefile.compilateur
```

Le makefile.gfortran devrait être utilisable avec n'importe quelle version de gfortran supérieure à la 4.1.2.

II. Récupération des données d'initialisation

Les scripts permettant la récupération des fichiers d'initialisation de MIMOSA sont disponibles dans le dossier TOOLS/ECMWF de l'archive contenant les sources de MIMOSA. Ce dossier contient deux fichiers :

- *mimosa_ecmwf_ECMR.sh* : Ce script permet de préparer les fichiers au format ASCII en niveaux pression
- *mimosa_ecmwf_grib.sh* : Ce script permet de préparer les fichiers archivés au format GRIB1/2 en niveaux modèle

La récupération des données contenant les champs servant à l'initialisation de MIMOSA (température, vent zonal, vent méridien) se fait sur le serveur ECGATE de l'ECMWF.

Ces scripts contiennent dans leur entête des commandes destinées au système de gestion de queue sur les machines de l'ECMWF (lignes commençant par #@). Il est important d'adapter les chemins des lignes « #@ output » et « #@ error » à votre compte. Dans ces chemins il faut adapter l'UID (3 caractères) de l'utilisateur. Dans l'exemple ci-dessous l'UID est « aut ». Pour plus d'information sur la signification se référer à la documentation du serveur ECGATE.

II.1. Données en niveaux pression (mimosa_ecmwf_ECMR.sh)

Ce script permet le téléchargement des champs T, U et V au format ASCII. Il peut être utilisé pour récupérer les données issues de modèles opérationnel, ERA-INTERIM ou ERA-40. Le programme FORTRAN inclus dans ce script qui réalise la conversion des champs au format GRIB vers le format ASCII est compatible avec les données au format GRIB2 de l'ECMWF et ne nécessite aucune modification en fonction des caractéristiques des champs voulus.

Pour lancer une préparation de données, il est seulement nécessaire de compléter les variables contenues dans la partie « Settings » qui apparaît ci-dessous:

```

1  #!/bin/bash
2  #
3  #@ shell          = /usr/bin/bash
4  #@ class          = express
5  #@ job_name       = Mimosa_grib
6  #@ output         = /scratch/ms/fr/aut/${job_name}.${host}.${jobid}.out
7  #@ error          = /scratch/ms/fr/aut/${job_name}.${host}.${jobid}.out
8  #@ environment    = COPY_ALL
9  #@ notification   = error
10 #@ job_cpu_limit  = 00:55:00,00:54:55
11 #@ wall_clock_limit= 06:00:00,05:59:00
12 #@ queue
13
14 #-----
15 # Settings
16 #-----
17
18 # Characteristics of ECMWF Insp, T, U and V fields to retrieve
19 YEAR1=2011      # Start year
20 MONTH1=4        # Start month
21 DAY1=1          # Start Day
22 YEAR2=2011      # End Year
23 MONTH2=4        # End month
24 DAY2=30         # End day
25 M_LEVELS=60     # Number of vertical levels
26 M_STEP=6        # Hour between 2 files ( only accept 6 or 12)
27 M_RESOL=1.000   # lon/lat grid resolution
28 M_LONG_MAX=359.000 # max lon of the grid (360 - M_RESOL)
29 M_CLASS=ei      # Class of model (od, ei, e4)
30 M_STREAM=oper   #
31 M_TYPE=an       # type of data (an, fc, ...)
32
33 # What to do with the files created (save on $SCRATCH and/or send them using ECTRANS)
34 M_ECTRANS=0     # 1 to send files using ECTRANS else 0
35 M_SCRATCH=1     # 1 to save the file in directory $SCRATCH/MIMOSA/YEAR else 0
36 # If use of ECTRANS (see ECTRANS manual)
37 GATEWAY=ecaccess.ecmwf.int # ECTRANS gateway
38 DESTINATION=girafe_MIMOSA # ECTRANS DESTINATION

```

L'exemple d'entête ci-dessus permet la récupération des champs T, U et V sur les 37 niveaux pression du modèle ERA-INTERIM entre le 31 mai 2011 et le 1^{er} juin 2011 toutes les 6 heures sur une grille de 2° de résolution en latitude et en longitude :

- **YEAR1** est l'année du premier fichier voulu
- **MONTH1** est le mois du premier fichier voulu

- **DAY1** est le jour du premier fichier voulu
- **YEAR2** est l'année du dernier fichier voulu
- **MONTH2** est le mois du dernier fichier voulu
- **DAY2** est le jour du dernier fichier voulu
- **M_LEVELS** est la liste des niveaux pressions voulus. Attention à ne pas oublier les « / » entre chaque niveau et à ce que les niveaux apparaissent dans l'ordre décroissant.
- **M_STEP** est la durée en heure entre deux fichiers. Cette variable ne peut prendre que deux valeurs 6 ou 12
- **M_RESOL** est la résolution en latitude et en longitude de la grille voulue
- **M_CLASS** permet de définir de quel modèle ECMWF proviendront les données (opérationnel, ERA-INTERIM ou ERA-40)
- **M_LONG_MAX** est la valeur maximale de la longitude de la grille (360-M_RESOL)
- **M_STREAM** identifie le système de prévision utilisé pour générer les données
- **M_TYPE** définit si on veut des données analysées ou prévisionnelles.

Une fois les données préparées, il y a deux possibilités :

- Sauvegarder les données sur la zone ECSCRATCH du compte ECMWF utilisé dans un dossier nommé *MIMOSA/annee_du_premier_fichier*. Si ces dossiers n'existent pas le script va les créer.
- Envoyer les fichiers créés en utilisant le système ECTTRANS de l'ECMWF. Dans ce cas il faut aussi fournir le nom de la GATEWAY et de la DESTINATION. La GATEWAY est en général *ecaccess.ecmwf.int*. La destination doit être configurée en fonction de chaque utilisateur (voir la documentation du système ECTTRANS)

Ces options sont activées si on leur attribut la valeur et désactivé si on leur attribut 0. Il n'y a pas de contrôle dans le script si au moins une des variables ECSCRATCH ou ECTTRANS a la valeur 1. Si aucune n'est activée les données seront supprimées à la fin de la préparation.

Une fois configuré pour exécuter le script, il suffit de lancer la commande :

```
lsubmit mimosa_ecmwf_ECMR.sh
```

II.2. Données en niveaux modèle (mimosa_ecmwf_grib.sh)

Le script *mimosa_ecmwf_grib.sh* est fait pour récupérer les champs T, U et V mais en niveau modèle et au format GRIB. Il a le même fonctionnement que le script pour récupérer les fichiers ECMR.

```

1 #-----
2 # Settings
3 #-----
4
5 # Characteristics of ECMWF Insp, T, U and V fields to retrieve on MARS
6 YEAR1=2011          # Start year
7 MONTH1=5           # Start month
8 DAY1=31            # Start Day
9 YEAR2=2011         # End Year
10 MONTH2=6          # End month
11 DAY2=1            # End day
12 M_LEVELS=91       # Number of vertical levels
13 M_STEP=6          # Hour between 2 files ( only accept 6 or 12)
14 M_RESOL=2.000     # lon/lat grid resolution
15 M_LONG_MAX=358.000 # max lon of the grid (360 - M_RESOL)
16 M_CLASS=od        # Class of model (od, ei, e4)
17 M_STREAM=oper     #
18 M_TYPE=an         # type of data (an, fc, ...)
19
20 # What to do with the files created (save on $SCRATCH and/or send them using ECTTRANS)
21 M_ECTTRANS=1      # 1 to send files using ECTTRANS else 0
22 M_SCRATCH=1      # 1 to save the file in directory $SCRATCH/MIMOSA/YEAR else 0
23 # If use of ECTTRANS (see ECTTRANS manual)
24 GATEWAY=ecaccess.ecmwf.int # ECTTRANS gateway

```


La seule différence avec le script précédent concerne la variable M_LEVELS. Dans ce cas elle ne reçoit pas la valeur des niveaux voulus mais le nombre de niveaux du modèle :

- 60 pour les modèle ERA-INTERIM et ERA-40
- 91 pour le modèle opérationnel

III. Utiliser MIMOSA

Avant le lancement d'une simulation, il est nécessaire de configurer la simulation à l'aide du fichier *input.namelist*. L'utilisation de ce fichier permet de changer la configuration d'une simulation sans pour autant devoir recompiler MIMOSA. La plupart des paramètres sont accessibles dans ce fichier. Ce fichier regroupe quatre namelist qui doivent impérativement apparaître dans cet ordre dans le fichier dans le cas contraire la simulation échouera ou retournera des résultats aberrant.

- run
- grid
- config
- output

On peut voir ci-dessous un exemple de ce fichier.

```

1  !-----!
2  !                                     !
3  !                                     !
4  !                                     !
5  !                                     !
6  !                                     !
7  !                                     !
8  !                                     !
9  !                                     !
10 !                                     !
11 !                                     !
12 !                                     !
13 !                                     !
14 !-----!
15 !
16 ! CHARACTERISCS OF THE RUN
17 !
18 &run
19 !-----
20 ! ZONE defines the geographical area :
21 !   -> 1 for the Northern Hemisphere [-10N, 90N]
22 !   -> 2 for the Southern hemisphere [-90N, 10N]
23 !   -> 3 for the both Hemisphere    [-90N, 90N]
24 Zone = 3
25 !-----
26 ! INTYPE defines the type of input files :
27 !   -> 1 for ASCII isobaric files (*.ECMR)
28 !   -> 2 for GRIB encoded model levels files (*.grib)
29 intype = 2
30 !-----
31 ! IAND, MOISD, JOURD, IHEURED define the starting date of the simulation
32 !   -> Year (YY)
33 iand   = 11
34 !   -> Month (MM)
35 moisd  = 3
36 !   -> Day (DD)
37 jourd  = 1
38 !   -> Hour (HH)
39 iheured = 0
40 !-----
41 ! IANF, MOISF, JOURF, IHEUREF define the final date of the simulation
42 !   -> Year (YY)
43 ianf   = 11
44 !   -> Month (MM)
45 moisf  = 3
46 !   -> Day (DD)
47 jourf  = 10
48 !   -> Hour (HH)
49 iheuref = 0
50 !-----
51 ! INITPV defines if the simulation is new or a restart

```

```

52 ! -> 1 if initialization is need
53 ! -> 0 if a ph* file from a previous run should be read
54 initpv = 1
55 !-----
56 ! TETA defines the isentropic surface (K)
57 ! -> shouldn't be greater than 950K for isobaric input files (intype = 1)
58 teta = 1500
59 /
60 !
61 ! CHARACTERISCS OF ECMWF GRID
62 !
63 &grid
64 !-----
65 ! NX, NY and NP define the number of points of the ECMWF grid
66 ! -> Number of grid points along longitudes
67 nx = 320
68 ! -> Number of grid points along latitudes
69 ny = 161
70 ! -> Number of pressure levels (intype = 1) or number or model levels (intype = 2)
71 np = 60
72 !-----
73 ! PRES allows to define the pressure levels of isobaric file (intype = 1)
74 ! -> this variable is not needed for GRIB files (intype = 2)
75 ! -> If there is more than 50 levels, declaration of pres variable in mimosa.f95
76 Pres(1) =
77 1000.,975.,950.,925.,900.,875.,850.,825.,800.,775.,750.,700.,650.,550.,500.,450.,400.,350.,300.,250.,22
78 5.,200.,175.,150.,125.,100.,70.,50.,30.,20.,10.,7.,5.,3.,2.,1.
79 !-----
80 ! PASLAT and PASLONG define the horizontal resolution of the ECMWF grid
81 ! -> resolution along latitude
82 paslat = 1.125
83 ! -> resolution along longitude
84 paslong = 1.125
85 !-----
86 ! LATMINECMR and LATMAXECMR define the minumum and maximum latitude of ECMWF grid
87 ! -> minimum latitude
88 latminecmmr = -90
89 ! -> maximum latitude
90 latmaxecmmr = 90
91 /
92 !
93 ! CONFIGURATION OF THE SIMULATION
94 !
95 &config
96 !-----
97 ! NDEG defines the number of MIMOSA grid points per degree of latitude and longitude
98 ! -> value should be 3 or 6
99 ndeg = 3
100 !-----
101 ! NLIS2D defines the number of points use for the smooth
102 Nlis2D = 8
103 !-----
104 ! NHGRID defines the number of hours between two call to regrid
105 nhgrid = 6
106 !-----
107 ! NWRITE defines the number of hour between two outputs of PV files
108 nwrite = 6
109 !-----
110 ! NHMOD defines the number of hours between two ECMWF files
111 nhmod = 12
112 !-----
113 ! NHRELAX defines the number of hours of relaxation time
114 nhrelax = 240
115 !-----
116 ! NPRHMOD defines the hour of the first ECMWF file
117 nprhmod = 0
118 !-----
119 ! NPRWRITE defines the first hour of PVF file
120 nprwrite = 0
121 !-----
122 ! Explicit diffusion
123 ! INDIFEXPL defines if explicit diffusion is activated

```

```

124 ! -> 0 no explicit diffusion
125 ! -> 1 explicit diffusion
126 indifexpl = 0
127 ! DIFF defines the value of the explicit diffusion if it is activated
128 diff = 4050
129 /
130 !
131 ! OUTPUT OF THE SIMULATION
132 !
133 &output
134 !-----
135 ! NRUN defines the name of the directory where MIMOSA outputs will be saved
136 ! -> if nrun = 5, files will be saved in RUN05 directory
137 ! -> if nrun = 16, files will be saved in RUN16 directory
138 nrun = 16
139 !-----
140 ! NWTEMP defines the time between two output of temperature or wind
141 nwtemp = 12
142 !-----
143 ! WIND_OUT defines if wind horizontal components files will be saved
144 ! -> 0 no output of wind files
145 ! -> 1 output of wind files
146 Wind_out = 1
147 !-----
148 ! T_OUT defines if temperature files will be saved
149 ! -> 0 no output of temperature files
150 ! -> 1 output of temperature files
151 t_out = 1
152 !-----
153 ! STATIONS_OUT defines if PV and temperature and PV profiles at stations files will be saved
154 ! -> 0 no output of stations files
155 ! -> 1 output of stations files
156 stations_out = 1
157 /

```

Dans le fichier input.namelist, la plupart des variables sont commentées. Dans les parties suivantes, on décrit les initialisations de variables qui nécessitent une valeur particulière.

III.1. Namelist run (L12 à L53)

C'est dans cette partie que sont définies les variables qui caractérisent la simulation.

La variable *zone* sert à définir la zone géographique sur laquelle va être réalisée la simulation. En fonction de la zone choisie le nom des fichiers de sorties va varier.

- si *zone* = 1
 - o fichiers de sortie du type pvn*, tn*, un*, vn*
- si *zone* = 2
 - o fichiers de sortie du type pvs*, ts*, us*, vs*
- si *zone* = 3
 - o fichiers de sortie du type pvg*, tg*, ug*, vg*

III.2. Namelist grid (L57 à L85)

Cette namelist permet de définir la grille des données ECMWF qui vont être lu par MIMOSA. La seule variable qui peut poser problème est *pres*. Celle-ci ne nécessite d'être défini uniquement dans le cas où la variable *intype* = 1 (L23), c'est-à-dire que l'on utilise une initialisation de MIMOSA en niveaux pression. Dans ce cas la variable *pres* doit contenir la liste des niveaux de pression par ordre **décroissant**, chaque valeur étant séparée par une virgule.

Dans le fichier source mimosa.f95, la variable *pres* a été définie pour recevoir jusqu'à 50 valeurs. Si dans le futur l'ECMWF augmente le nombre de niveaux pression de leur modèle, il faudra augmenter la taille de ce tableau. Une augmentation de la taille de ce tableau n'impliquerait pas d'autres changements dans les sources de MIMOSA.

III.3. Namelist config (L89 à L123)

Dans cette namelist, on peut activer ou non la diffusion explicite dans MIMOSA, variable *indifexpl* (L120) pour l'activer et variable *diff* (L122) pour définir la valeur de la diffusion explicite. L'utilisation de la diffusion explicite a été intégrée mais nécessite encore d'être testée.

III.4. Namelist output (L127 à L151)

Cette namelist permet de choisir dans quel dossier les sortie MIMOSA vont être écrites et quelles paramètres vont être écrits.

La variable *nrun* sert à choisir dans quel dossier vont être écrits les résultats de MIMOSA. Par exemple si on définit *nrun* = 7, les résultats seront écrits dans le dossier RUN07. Mais MIMOSA ne peut créer ce dossier s'il n'existe pas. Donc avant de démarrer la simulation, il faut créer le dossier.

La variable *nwtemp* permet de choisir à quel pas de temps vont être écrits les champs de température et de vent. La valeur minimum conseillée est la valeur du pas de temps entre deux fichiers ECMWF car si cette valeur est inférieure, on n'aura pas de différence entre les sorties au pas de temps *t* et les sorties au temps intermédiaire entre deux fichiers ECMWF.

Les sorties de champs de vents (*wind_out*) U et V ne se font pas sur la grille à la résolution MIMOSA mais une grille à la résolution ECMWF.

La variable *stations_out* (L150) permet de choisir si on veut en sortie la valeur de la vorticité potentielle à différentes stations définies dans le fichier *stations.inc*. Etant donné que ces stations sont réparties sur les hémisphères nord et sud, il n'est possible d'activité ces sorties que dans le cas où la variable *zone* = 3 (namelist run).

III.5. Démarrer une simulation

Pour démarrer la simulation, il suffit de taper dans le dossier où se trouve l'exécutable *mimosa.x* créé par le makefile :

```
> ./mimosa.x
```

III.6. Format des fichiers

III.6.1. Fichiers de champs de PV, température et vent

Les fichiers de sortie de MIMOSA pour les champs de vorticité potentielle (*pv**), de température (*t**) et de vents (*u** et *v**) sont au format binaire FORTRAN (option *unformatted*).

Ces fichiers sont composés de deux parties :

- un entête contenant les caractéristiques de la simulation
 - o c'est un tableau de 30 entiers codés sur 4 octets, on donne la description des éléments ci-dessous :

Elément du tableau	Description	Elément du tableau	Description
0	Année	15	Pas de temps en heure
1	Mois	16	Heure du premier fichier ECMWF
2	Jour	17	Nombre de points par degré de la grille MIMOSA
3	Heure	18	Fréquence d'écriture en heure

4	Année de l'initialisation	19	Nombre de points de la grille MIMOSA en longitude
5	Mois l'initialisation	20	Nombre de points de la grille MIMOSA en latitude
6	Jour l'initialisation	21	Pas de temps en heure
7	Heure l'initialisation	22	Nombre heure avant regrillage
8	Niveau isentrope	23	Relaxation (heures)
9	Limite ouest de la grille ECMWF	24	Non attribué
10	Limite sud de la grille ECMWF	25	Non attribué
11	Limite est de la grille ECMWF	26	Non attribué
12	Limite nord de la grille ECMWF	27	Non attribué
13	Nombre de points de la grille ECMWF en longitude	28	Non attribué
14	Nombre de points de la grille ECMWF en latitude	29	Non attribué

- les données du champ correspondant
 - o C'est un tableau de flottant codés sur 4 octets et de dimension header[19]xheader[20]

III.6.2. Fichiers Stations

Ces fichiers sont codés au format ASCII. Ils sont composés de deux parties :

- Un entête d'une ligne contenant :
 - o Le code de la station en 3 caractères
 - o La latitude de la station (°N)
 - o La longitude de la station (°E)
 - o La température potentielle (K)
- N-ligne contenant à 12 heures tous les jours de la simulation:
 - o L'année
 - o Le jour julien modifié
 - o Le mois
 - o Le jour
 - o L'heure
 - o La vorticité potentielle (pvu)
 - o La température (K)

IV. Outils IDL de lecture et traçage des sorties MIMOSA

L'ensemble des programmes de traçage a été réalisé en langage IDL au LPC2E en se basant sur les programmes du pôle CNES-INSU/CNRS ETHER.

Toutes les procédures de post traitements sont disponibles dans le dossier TOOLS/IDL de l'archive fournie. Pour les procédures traçant des contours les résultats sont enregistrés dans des fichiers PNG 24 bits. Chaque procédure contient un entête décrivant les paramètres d'entrée, de sortie ainsi que ceux qui sont optionnels. La plupart du temps un exemple est fourni.

Toutes les procédures traçant des cartes utilisent les bibliothèques coyote et catalyst de D. Fanning (fournie dans le dossier TOOLS/IDL). Avant de pouvoir utiliser les programmes IDL il faut donc ajouter au PATH IDL le chemin vers ce dossier avec la commande :

- Sur une machine WINDOWS si le dossier se trouve dans « C:\Documents and settings\user\coyote »
 - o IDL> !PATH = Expand_Path('+ C:\Documents and settings\user\coyote\ ; '+ C:\Documents and settings\user\catalyst\ ') + ';' + !PATH
- Sur une machine UNIX si le dossier se trouve dans « /home/user/coyote »
 - o !PATH = Expand_Path('/home/user/coyote/ :+ /home/user/catalyst/') + ':' + !PATH

IV.1. Liste des programmes fournis

Huit procédures et fonctions IDL sont fournis dans le dossier TOOLS/IDL de l'archive :

- **colorbar.pro** : ce programme permet d'ajouter une colorbar sur les tracés.
- **mimosa_3d_votes.pro** : ce programme permet à partir de simulations MIMOSA réalisées à plusieurs niveaux isentropes de tracer la surface à une valeur choisie par l'utilisateur du volume de vorticit  potentielle sur une projection du p le nord ou sud.
- **mimosa_plot_pv.pro** : ce programme permet de tracer les contours de vorticit  potentielle   partir des fichiers pv* produit par MIMOSA.
- **mimosa_plot_t.pro** : ce programme permet de tracer les contours de temp rature   partir des fichiers t* produit par MIMOSA.
- **mimosa_plot_wind.pro** : ce programme permet de tracer les contours et la direction des champs de vent   partir des fichiers u* et v* produit par MIMOSA.
- **Mimosa_read_bin.pro** : ce programme permet de lire les fichiers binaires produits par MIMOSA.
- **mimosa_read_list.pro** : ce programme permet de lire les listes de fichiers   tracer par les programmes mimosa_plot*.
- **str_month.pro** : cette fonction retourne le nom du mois (en anglais)   partir de son num ro.

Les principaux programmes vont  tre d crits bri vement dans les parties suivantes.

IV.2. Lecture des fichiers binaires

Le fichier servant   la lecture des sorties binaires de MIMOSA appel  *MIMOSA_READ_BIN* est le m me quelque soit le param tre qui doit  tre lu (vorticit  potentielle, temp rature ou champs de vent).

Il est utilis  de la mani re suivante si vous souhaitez lire le fichier pvn11040312.0850 situe dans le dossier MIMOSA/RUN00 :

```
IDL> inFile = '/home/madrouin/MIMOSA/RUN00/pvn11040312.0850'
IDL> MIMOSA_READ_BIN, inFile, DATA=data, HEADER=header, DIMX=nLon, DIMY=nLat
```

Les variables contiendront les donn es suivantes :

- Data : le champ lu dans le fichier (vorticit  potentielle, temp rature, vent zonal ou vent m ridien) sur une grille de taille nLon X nLat
- Header : les donn es contenues dans l'ent te du fichier binaire
- DIMX : la dimension de la grille MIMOSA selon la longitude
- DIMY : la dimension de la grille MIMOSA selon la latitude

IV.3. Tra age des contours

Trois programmes de tra age ont  t  cr s pour tracer les contours des champs sur une carte. Ces codes appellent le programme *READ_MIMOSA_BIN* pour la lecture des donn es avant tra age. Les trois programmes principaux sont faits pour tracer :

- *MIMOSA_PLOT_PV* pour les champs de vorticit  potentielle
- *MIMOSA_PLOT_T* pour les champs de temp rature
- *MIMOSA_PLOT_WIND* pour les champs de vent

Ces programmes par défaut réalisent un traçage sur une carte en projection azimutale centrée sur le pôle nord et le sauvegarde dans une image PNG 24 bit.

Ils utilisent aussi le programme permettant d'ajouter une barre de couleur discrétisée (*colorbar.pro*) développée par S. Masson de l'IPSL. Deux autres fonctions sont utilisées, *READ_MIMOSA_LIST* pour lire les listes contenant les fichiers et *STR_MONTH* pour convertir le numéro d'un mois en lettre.

Pour fonctionner, ces programmes ont besoins d'une liste des fichiers qu'ils vont devoir lire puis tracer. Pour créer ces listes il faut se placer dans le dossier RUNXX où les sorties MIMOSA ont été enregistrées et exécuter les commandes bash suivantes:

- pour les fichiers de vorticité potentielle

```
$ ls pv* | wc -l > list_mimosa_idl.in
$ ls pv* >> list_mimosa_idl.in
```

- pour les fichiers de température

```
$ ls t* | wc -l > list_temp_idl.in
$ ls t* >> list_temp_idl.in
```

- pour les fichiers de vents

```
$ ls u* | wc -l > list_vent_idl.in
$ ls u* >> list_vent_idl.in
```

Il est à noter que le nom du fichier contenant la liste dépend du choix de l'utilisateur puisque le nom est fourni en paramètre d'entrée de la procédure IDL.

Les procédures de traçage IDL reçoivent au minimum en entrée :

- le nom du fichier contenant la liste de fichiers à tracer
- le chemin du dossier où se trouvent les fichiers de sortie MIMOSA

Elles acceptent les options suivantes :

- **OUTPUTDIR** : le dossier dans lequel les images vont être enregistrées. Si cette option n'est pas utilisée, les fichiers seront enregistrés dans le dossier contenant les fichiers de sortie MIMOSA.
- **LEVELS** : Les niveaux des contours pour le champ à tracer. Dans le cas où cette option n'est pas utilisée, les niveaux sont déterminés à partir des valeurs minimum et maximum du champ du premier fichier à tracer
- **NO_FILL_LEVELS** : Cette option permet de fournir une liste de contours blanc sur le tracé. Si elle n'est pas activée, un contour noir est ajouté pour chaque niveau fourni dans l'option LEVELS
- **CCOLORS** : les numéros des couleurs de la palette à utiliser pour les contours
- **NPALETTE** : la palette de couleurs à utiliser pour le tracer des contours. Cette valeur est comprise entre 0 et 40 (voir la documentation IDL). Quelle que soit la palette choisie le fond des tracés sera blanc et les traits en noir.
- **LAT_RANGE** : spécifie la gamme de la latitude à tracer
- **LON_RANGE** : spécifie la gamme de longitude à tracer
- **SOUTH_AZIMUTHAL** : si cette option est activée, elle permet de réaliser le tracé de contour sur une carte avec une projection azimutale centrée sur le pôle sud.
- **CYLINDRICAL** : si cette option est activée, elle permet de réaliser le traçage sur une carte en projection cylindrique.

1500 K

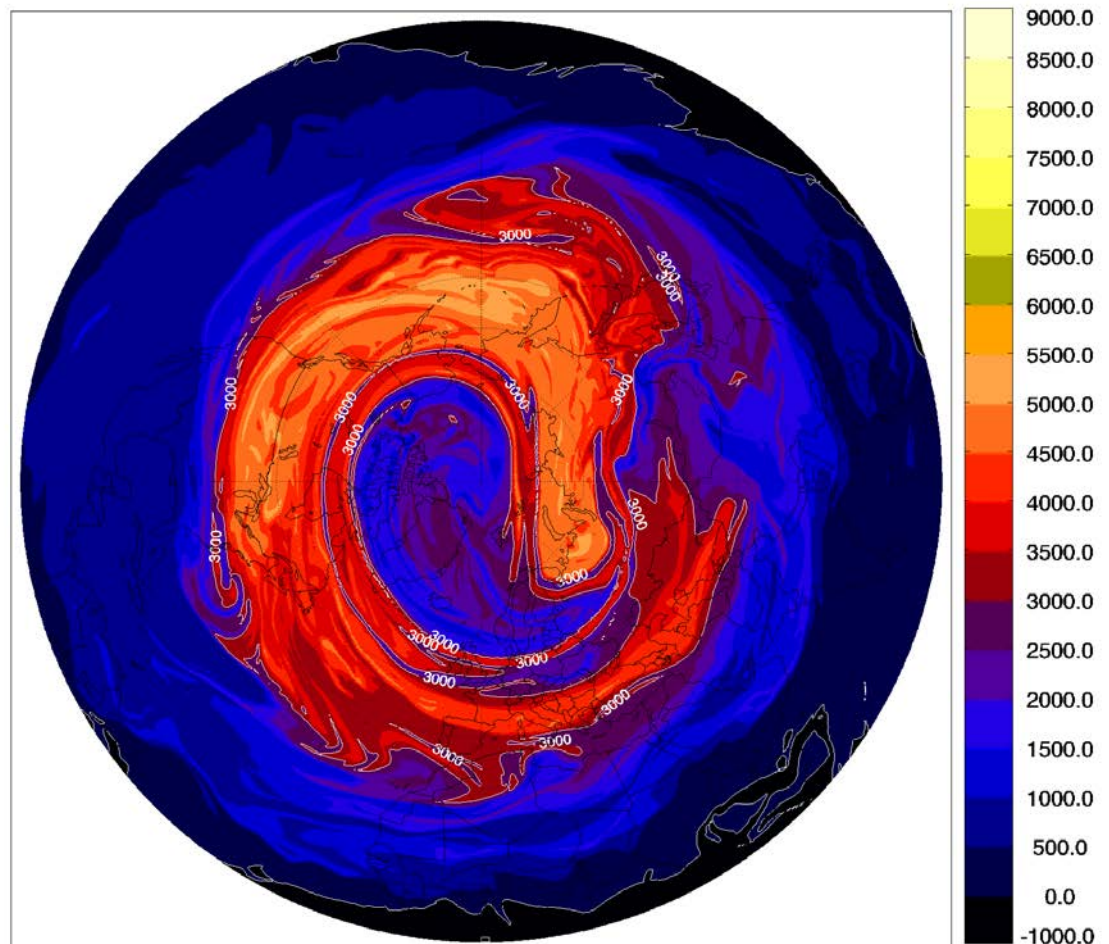


Figure 1: exemple de carte de vorticité potentielle tracée avec `mimosa_plot_pv` sur le niveau isentropie $\theta = 1500\text{K}$ à partir d'une simulation MIMOSA en niveau modèle

En annexe de ce document apparaissent des exemples de sorties de ces programmes avec les options utilisées.

Il faut noter que pour ces programmes de traçage, lorsque que l'on choisit un choix automatique des niveaux des contours, le programme utilise le premier fichier à tracer pour déterminer les niveaux. Il est donc possible que les niveaux choisis par le programme ne soient pas adaptés à tous les pas de temps.

IV.4. Traçage des surfaces de volume

Le programme `mimosa_3d_vortex` permet à partir de simulations de `mimosa` réalisées pour plusieurs surfaces isentropes de tracer la surface d'un volume pour une valeur choisie. Il permet de tracer ces surfaces sur projection azimutale centrée sur le pôle nord ou le pôle sud et de les enregistrer dans une image PNG 24 bit.

De manière à pouvoir tracer ce volume, ce programme commence par calculer la vorticité potentielle de Lait qui permet de normaliser les champs de vorticité potentielle (Lait, 1993).

A partir d'une date de départ, d'une date de fin et d'un pas de temps, cette fonction va rechercher dans un dossier donné tous les fichiers qui ont la même date.

Ce programme requiert au moins les paramètres suivant :

- Le chemin du dossier contenant les sorties MIMOSA.
- Le chemin du dossier où les images PNG vont être enregistrées
- L'année du premier fichier
- Le mois du premier fichier
- Le jour du premier fichier
- L'heure du premier fichier

- L'année du dernier fichier
- Le mois du dernier fichier
- Le jour du dernier fichier
- L'heure du dernier fichier
- Le pas de temps entre deux fichiers en heures
- La valeur de température potentielle θ_0 qui va être utilisée pour normaliser la température potentielle avec la formule $laitPV = PV \left(\frac{\theta}{\theta_0} \right)^{\frac{9}{2}}$
- La valeur de laitPV pour laquelle la surface va être tracée.
- L'hémisphère sur laquelle va être tracée le volume ('N' pour l'hémisphère nord et 'S' pour l'hémisphère sud)

Ce programme accepte les options suivantes :

- **LAT_MIN** : ce mot clé permet de choisir pour quelle latitude minimum va être calculées la surface. Par défaut le calcul se fera entre 60°N et 90°N pour un tracé sur l'hémisphère NORD et entre -60°N et -90°N pour un tracé sur l'hémisphère sud.
- **RESOLUTION** : Ce mot clé permet de choisir la résolution de l'image de sortie. Par défaut, la valeur est [2048,1536].

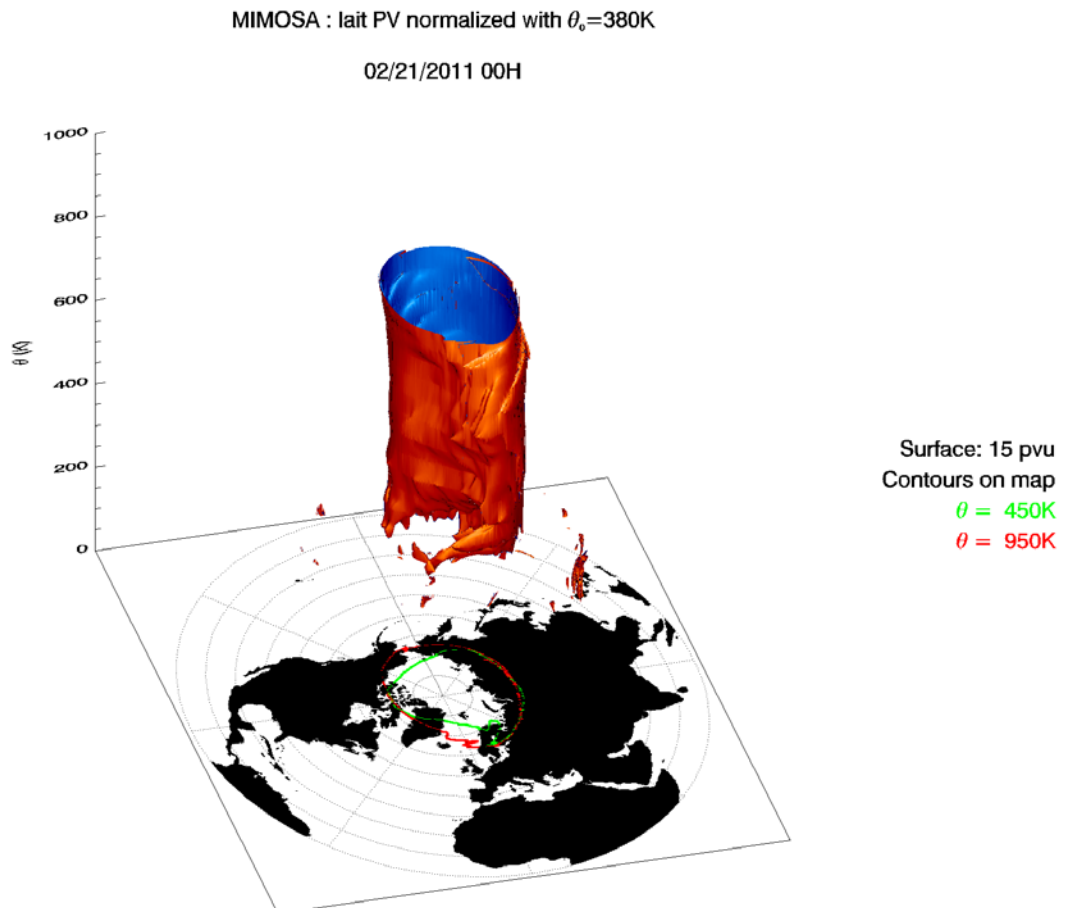


Figure 2: Exemple de tracé de mimosa_3d_vortex

V. Autres outils

V.1. Scripts *mimosa.sh*

Ce script permet de lancer des simulations sur plusieurs surfaces isentropes pour une même période de temps. A la fin des simulations, il créera des listes des fichiers de vorticité potentielle, de température et de vent au format nécessaire pour l'utilisation des programmes de traçage de contour IDL.

On retrouve au début de ce script les mêmes options que celles décrites dans la partie III.1 à III.4. La seule différence réside dans la déclaration des valeurs de θ voulues (L19 ci-dessous)

```
1 #-----
2 # Parameters start
3 #-----
4
5 ## RUN
6 ZONE=3
7 INTYPE=1
8 # Start date
9 SYEAR=11
10 SMONTH=4
11 SDAY=10
12 SHOUR=0
13 # End date
14 EYEAR=11
15 EMONTH=4
16 EDAY=20
17 EHOURL=12
18 INITPV=1
19 declare -a THETA=( 850 )
20
21 ## GRID
22 NX=320
23 NY=161
24 NP=37
25 PRES="1000.,975.,950.,925.,900.,875.,850.,825.,800.,775.,750.,700.,650.,600.,550.,500.,450.,400.,350.,3
26 00.,250.,225.,200.,175.,150.,125.,100.,70.,50.,30.,20.,10.,7.,5.,3.,2.,1."
27 PASLAT=1.125
28 PASLONG=1.125
29 LATMIN=-90
30 LATMAX=90
31
32 ## CONFIG
33 NDEG=3
34 NLIS2D=8
35 NHGRID=6
36 NWRITE=6
37 NHMOD=12
38 NHRELAX=240
39 NPRHMOD=0
40 NPRWRITE=0
41 INDIFEXPL=0
42 DIFF=4050
43
44 #OUTPUT
45 NRUN=0
46 NWTEMP=12
47 WINDOUT=1
48 TOUT=1
49 STATIONS=0
```

Après avoir configuré le script et avant de le lancer, il est nécessaire de le rendre exécutable avec la commande

```
> chmod u+x mimosa.sh
et de l'exécuter en tapant :
> ./mimosa.sh
```

V.2. Script *convert_PNG_to_VIDEO.sh*

Ce script permet de créer une animation à partir d'images PNG générées par les programmes décrits précédemment. Ce script permet de créer des animations au format mp4 encodé en x264. Il est situé dans le dossier TOOLS/BASH de l'archive.

Pour pouvoir utiliser ce script, il est en général nécessaire d'installer ffmpeg (ffmpeg.org) qui ne fait pas parti des packages installés de base avec la plupart des distributions linux et parfois imagemagick (www.imagemagick.org) qui est en général lui installé de base. Les vidéos créées par ce script sont facilement lisible par le logiciel VLC (videolan.org) par exemple.

Pour utiliser ce script, il suffit de remplir la partie « settings » qui apparaît ci-dessous :

```
1 #-----
2 # Settings
3 #-----
4
5 # Directory were the PNG images are stored
6 M_INDIR="/home/madrouin/test/IDL_Plot3D"
7 # Directory were the movie will be saved
8 M_OUTDIR="/home/madrouin/test/IDL_Plot3D"
9 # Name Pattern of the images which will converted to movie
10 M_PATTERN="mimosa_3D_15"
11 # Name of the movie (without extension)
12 M_VNAME=test_mimosa_vortex
13 # Number of frames per seconds
14 # each images is copied 5 times so for example
15 # if you want 3 of your images per seconds in your movie M_FPS=3images*5copies=15
16 # For compatibility with most of video player it is advise to never use a framerate below 10 FPS
17 M_FPS=10
18 # Resolution of the video
19 # It doesn't matter if your images have a bigger resolution ffmpeg will take care of the conversion
20 # horizontal resolution
21 M_XSIZE=1024
22 # vertical resolution
23 M_YSIZE=768
```

Description des paramètres de configuration du script :

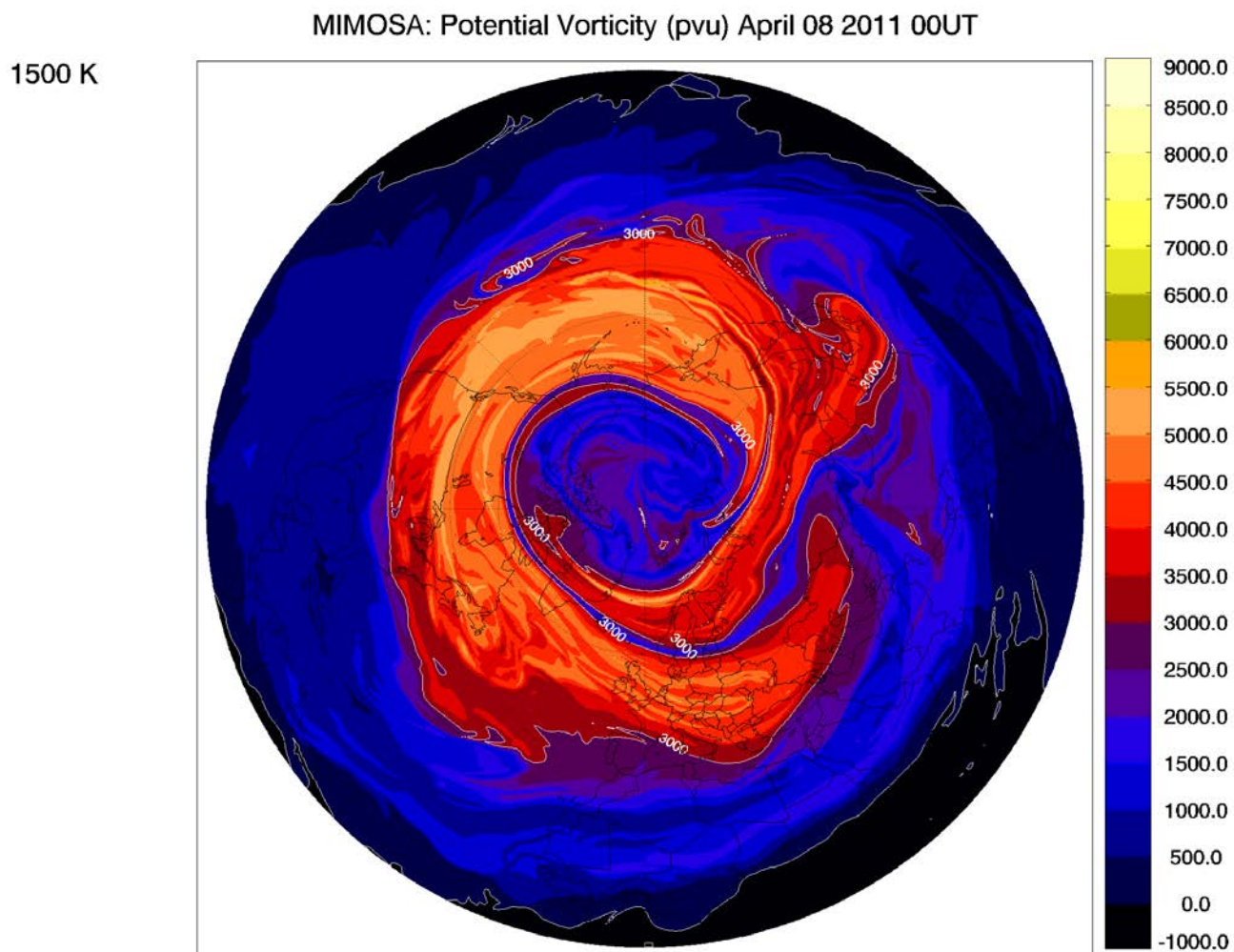
- **M_INDIR** : est le chemin du dossier contenant les images PNG que l'on veut utiliser pour créer l'animation
- **M_OUTDIR** : est le chemin du dossier dans lequel l'animation va être sauvegardée. Pendant le traitement des images, un dossier « tmp » sera créé dans ce dossier pour enregistrer les copies d'images nécessaires à la création de la vidéo et le supprimera à la fin du traitement.
- **M_PATTERN** : permet de définir la partie commune des noms des fichiers à traiter. Par exemple pour des images créées avec mimosa_3d_vortex pour une surface de 15 pvu, tous les fichiers auront en commun la partie « mimosa_3D_vortex_15 ».
- **M_VNAME** : permet de définir le nom voulu pour l'animation. Il n'est pas nécessaire d'ajouter l'extension .mp4, le script s'en occupe.
- **M_FPS** : permet de définir le nombre d'image par seconde. Pour des raisons de compatibilité avec la plupart des lecteurs vidéo, il est déconseillé de mettre une valeur inférieure à 10. Dans l'utilisation qui est faite de ce script au LPC2E, on voulait en général afficher 2 images par seconde, le script réalise donc 5 copies de chaque image de façon à atteindre 10 images par seconde mais d'en afficher seulement 2 différentes. De plus cette méthode n'augmente pas de manière significative la taille finale de l'animation.
- **M_XSIZE** et **M_YSIZE** : Ces paramètres permettent de définir la résolution voulue pour l'animation. Pour des conversions d'images produites par les programmes décrits dans la

partie précédentes il est conseillé de garder un rapport de 4/3 pour M_XSIZE/M_YSIZE de façon à éviter toute déformation. Le changement de résolution est géré par ffmpeg.

Dans le dossier TOOLS/EXAMPLES de l'archive une vidéo ainsi que les images qui ont servi à la générer est fournie de façon à pouvoir tester que l'utilisateur dispose de toutes les logiciels nécessaires pour créer les animations.

Annexe

Exemples de tracés avec mimosa_plot_pv

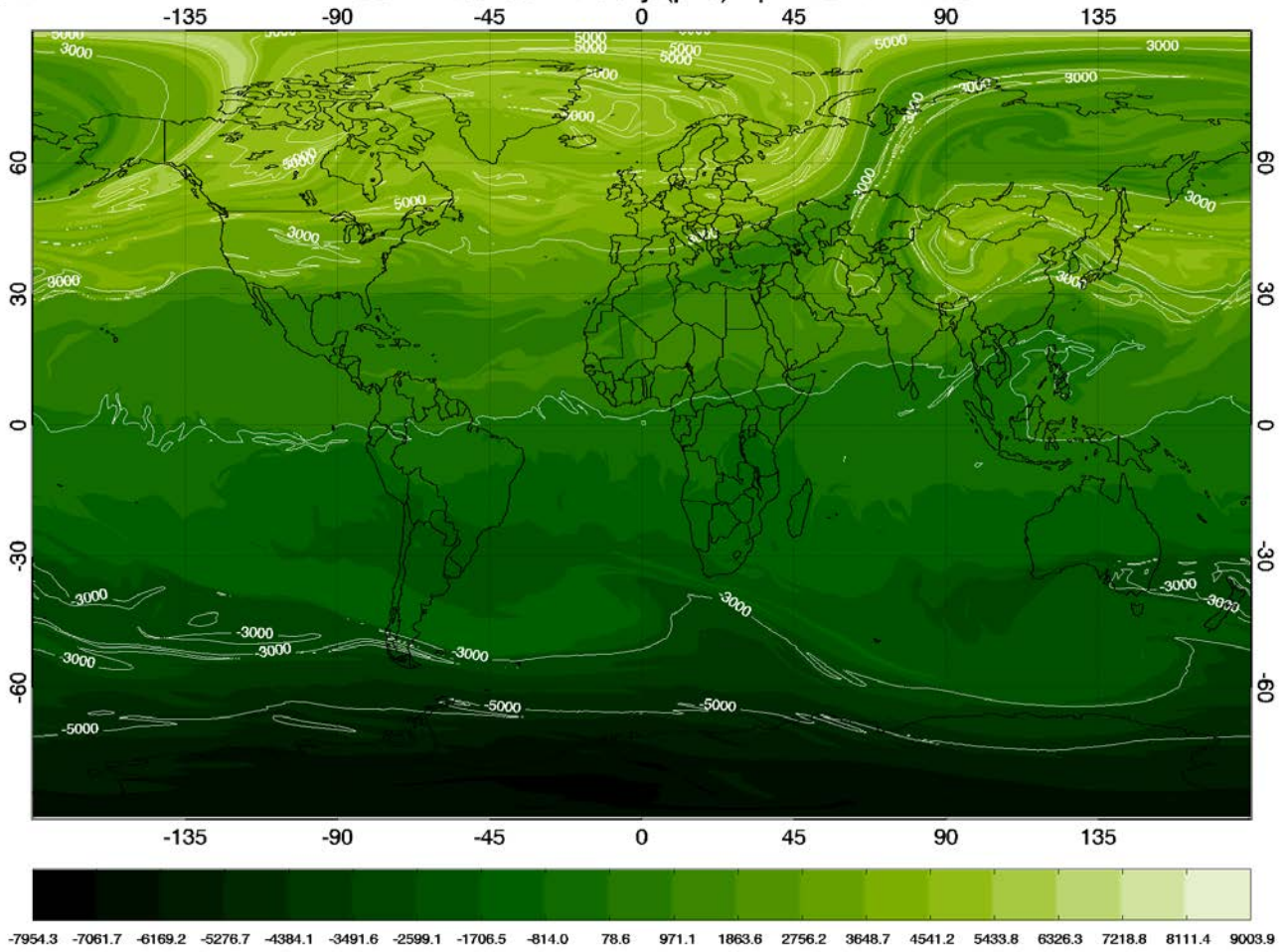


Pour créer cette image, on a seulement contraint, la valeur des niveaux de contours et le choix des couleurs avec les options :

- LEVELS = [-100, FINDGEN(19)*500.]
- NO_FILL_LEVELS = [0., 3000.]

1500 K

MIMOSA: Potential Vorticity (pvu) April 05 2011 00UT



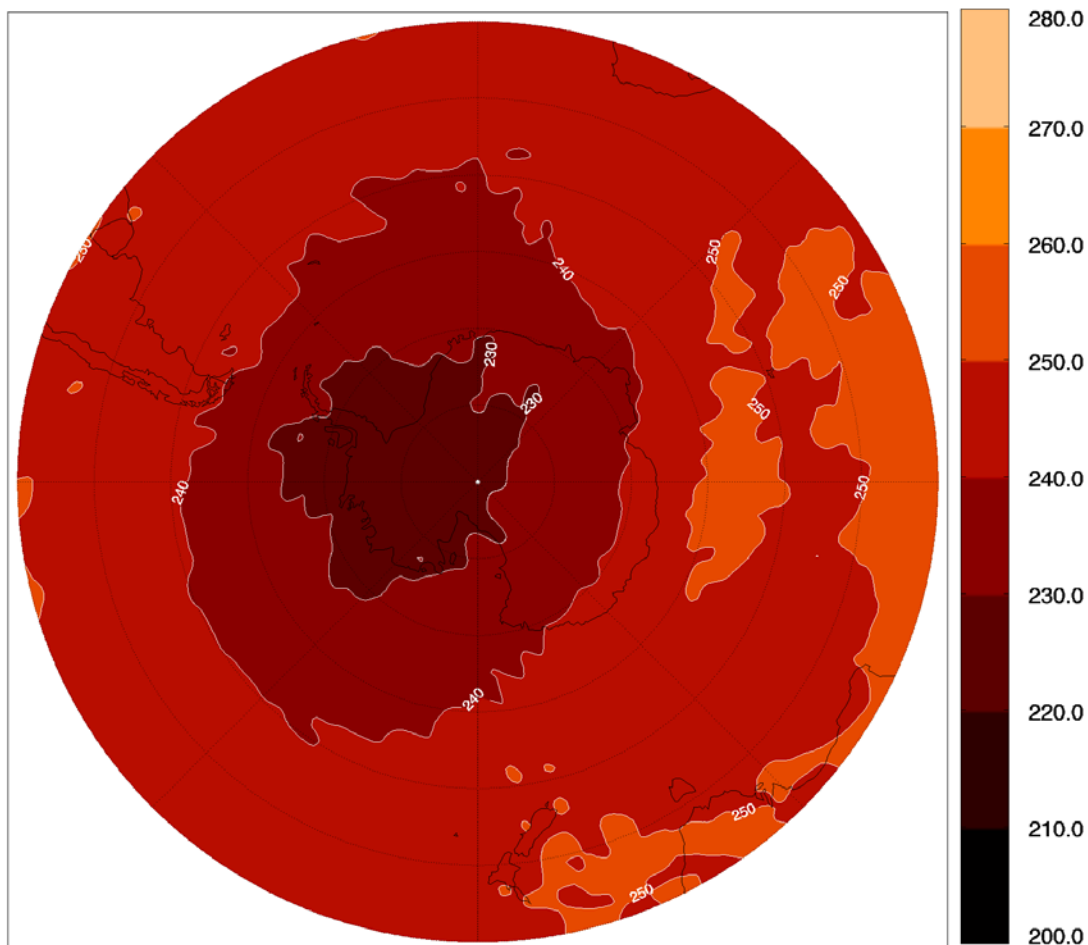
Pour cette image, on a laissé le programme choisir seule les couleurs et les niveaux des contours remplis, mais les options suivantes ont été activées :

- NO_FILL_LEVELS = [-5000.,-3000.,0.,3000., 5000.]
- NPALETTE = 8
- LAT_RANGE = [-90.,90.]
- CYLINDRICAL = 1

Exemples de tracés avec mimosa_plot_t

MIMOSA: Temperature (K) April 06 2011 12UT

1500 K

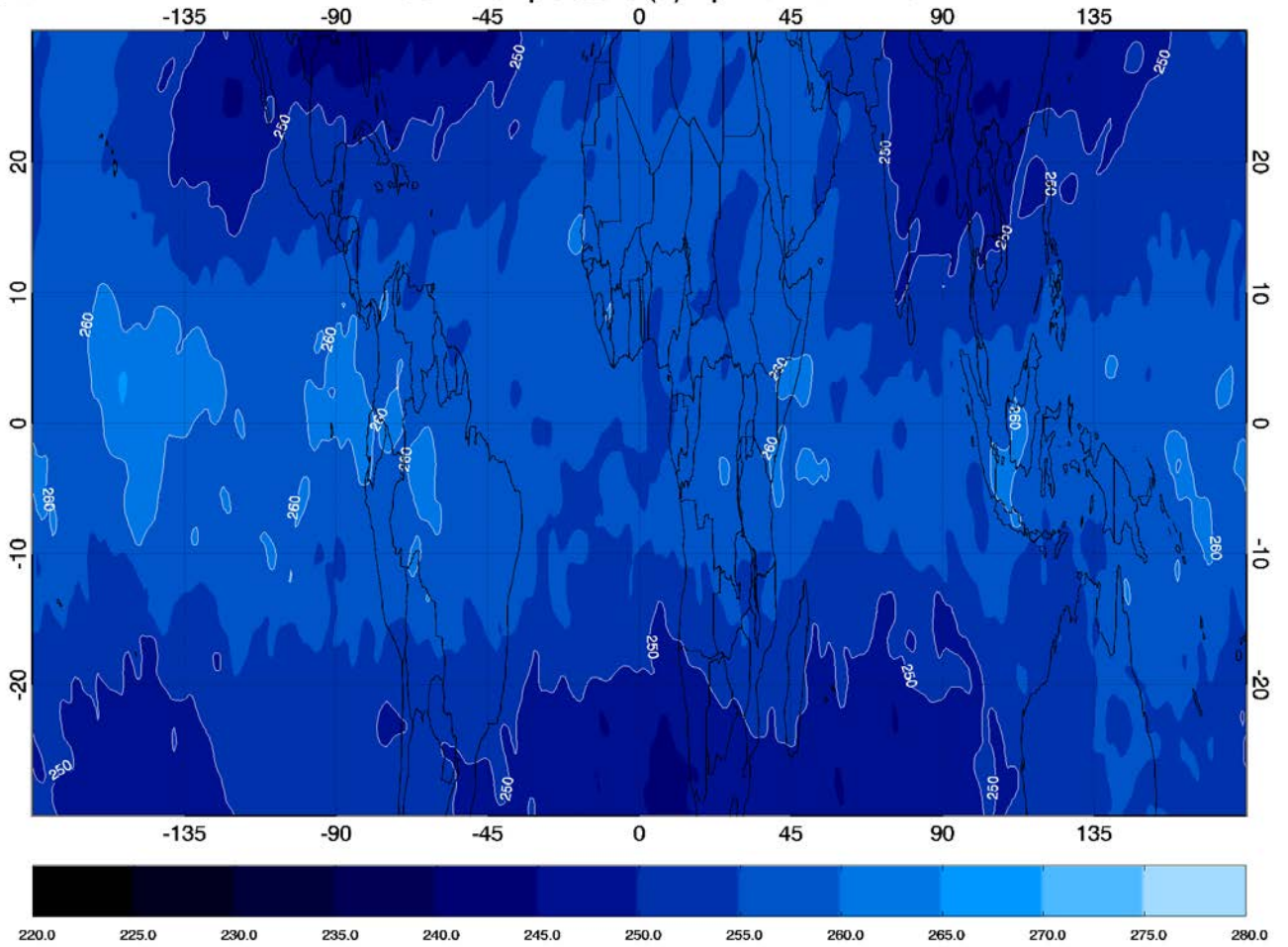


Pour cette image, les options suivantes ont été utilisées :

- LEVELS = 200.+FINDGEN(9)*10.
- NO_FILL_LEVELS = 230.+FINDGEN(3)*10.
- NPALETTE = 3
- /SOUTH_AZIMUTHAL

1500 K

MIMOSA: Temperature (K) April 01 2011 12UT



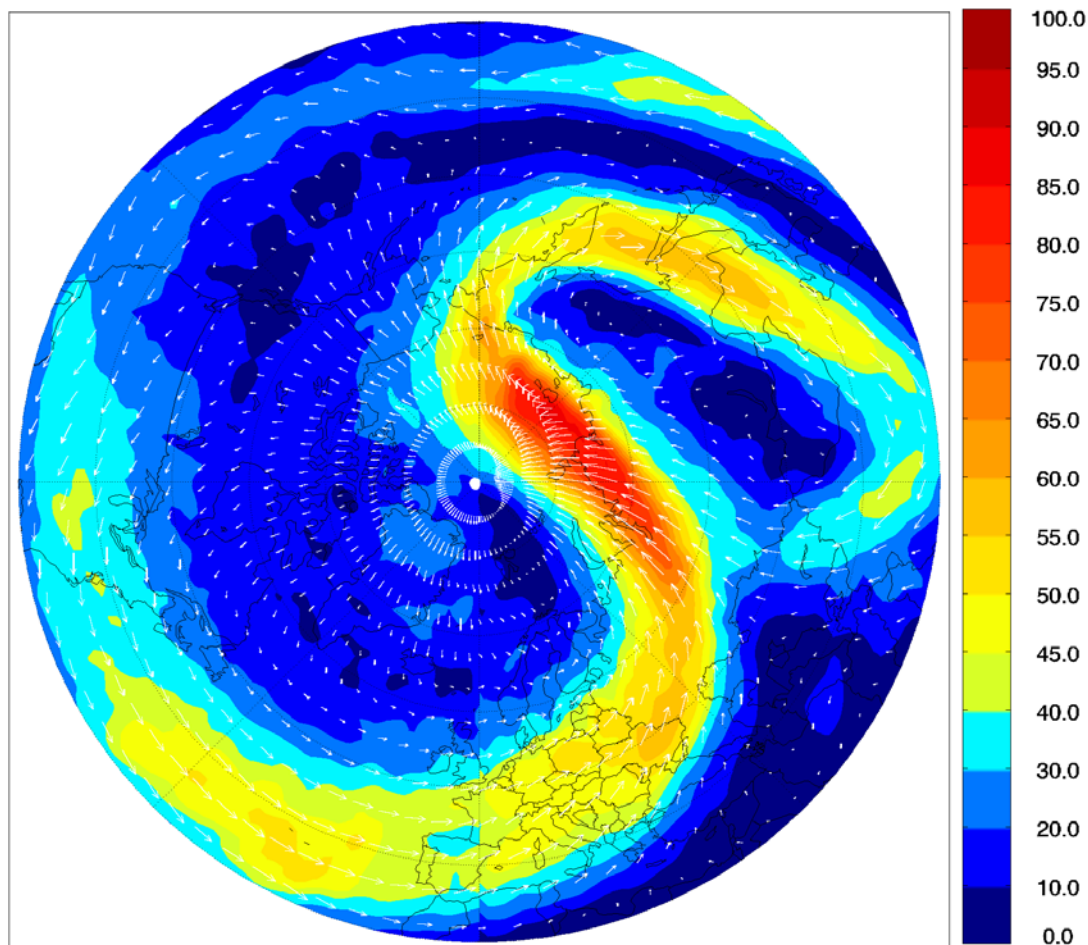
Pour cette image, on a utilisé les options :

- LEVELS = 220.+FINDGEN(13)*5.
- NO_FILL_LEVELS = 200.+FINDGEN(9)*10.
- NPALETTE = 1
- LAT_RANGE = [-30., 30.]
- /CYLINDRICAL

Exemples de tracés avec mimosa_plot_wind

MIMOSA: wind (m.s⁻¹) April 03 2011 12UT

1500 K

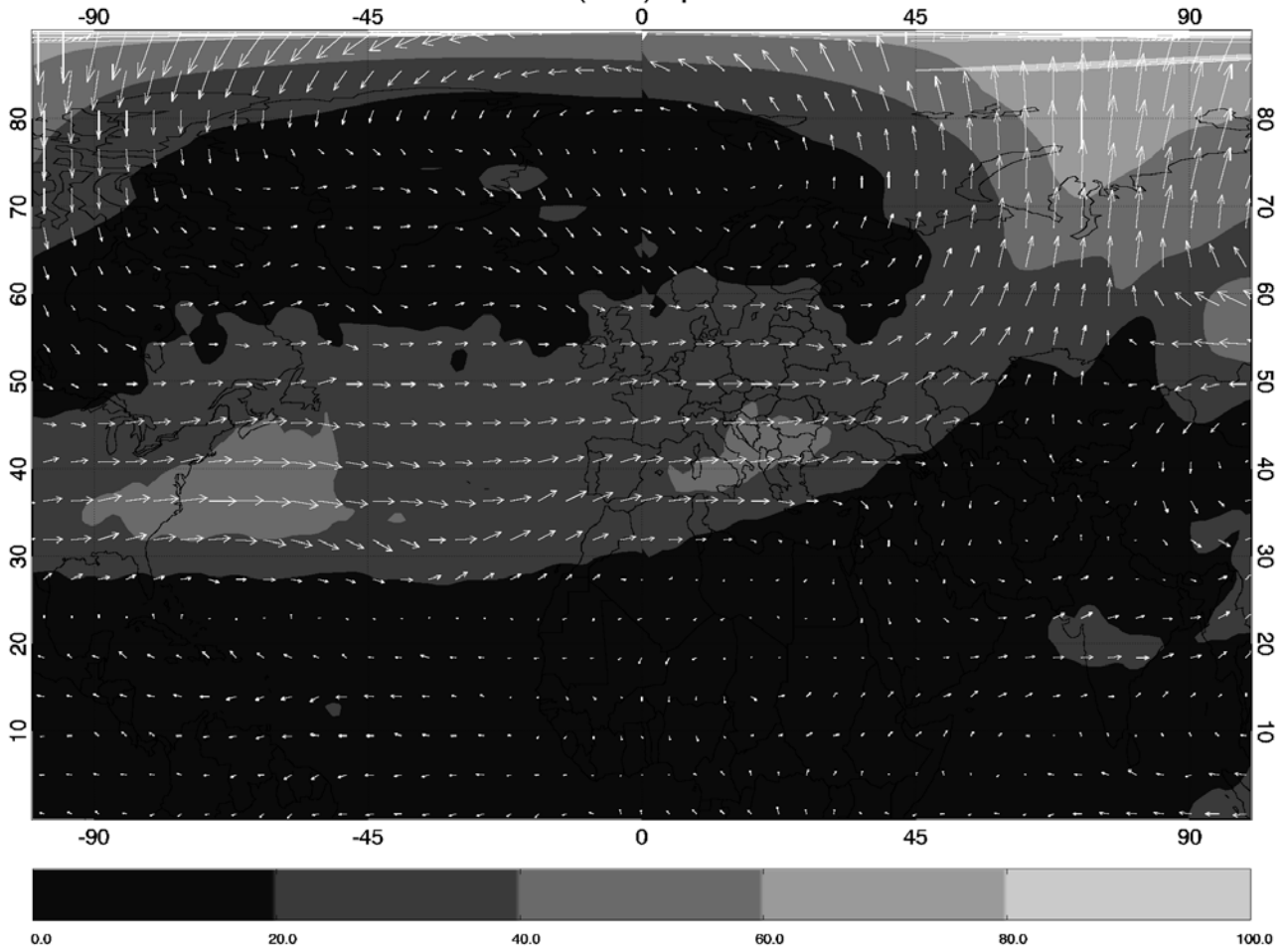


Les options suivantes ont été utilisées :

- LEVELS = [FINDGEN(5)*10.,45.+FINDGEN(12)*5.]
- CCOLORS = [INDGEN(4)*127/4, 152+INDGEN(12)*(255-152)/12]
- NPALETTE = 33

1500 K

MIMOSA: wind (m.s⁻¹) April 05 2011 00UT



Pour cette image, les options suivantes ont été utilisées :

- LEVELS = FINDGEN(6)*20.
- LAT_RANGE = [0., 90.]
- NPALETTE = 0
- LON_RANGE = [-100., 100.]
- /CYLINDRICAL