

Script de test des aérosols utilisés pour les simulations CMIP

Variable à ajuster

```
ntasks=2      # Nombre de processus mip
nthreads=1    # Nombre de taches openMP
```

La fonction main

Elle contient la succession de l'appel aux principales fonctions et est appelée en fin de script

```
function main {
```

Préparation de la configuration avec aérosols

```
setting_up_aerosols
```

Lancement des simulations Inca préindustrielle et SP+Inca actuelle

```
mkdir -p log
for sim in SP Inca ; do
    if [ ! -d $sim ] ; then
        echo Running simulation $sim with log in log/$sim.$$
        simulation $sim > log/$sim.$$ 2>&1 &
    fi
done
```

On attend que les deux simulations lancées avec & soient finies grace à **wait** avant de créer le script de tracer

```
wait
echo Simulations finies
plot_figures
echo On peut tracer avec ferret et le script trace.jnl
}
```

Configuration des aérosols

- 1) Passe à rrtm pour le rayonnement si ce n'était pas le cas en éditant directement init.sh Puis relance init.sh après avoir sauvegardé les anciens répertoires **INITIAL**, **SIMU1** et **PRODO** dans **OLDRAD**/
- 2) Fait appel à la fonction **utils/install_aerosols.sh** qui crée un répertoire **PRODaer** à partir de **PRODO**

```
function setting_up_aerosols {
if [ ! -d PRODaer ] ; then
    mkdir OLDRAD
    mv INITIAL SIMU1 PRODO OLDRAD/
    sed -i '' -e 's/^rad=.*$/rad=rrtm/' init.sh
    ./init.sh
    utils/install_aerosols.sh
    sed -i '' -e 's:^(~#)*./enchaine.*$:.*/enchaine.sh:' \
    -e 's/^ntasks=.*$/ntasks='$ntasks'/' \
    -e 's/^nthreads=.*$/nthreads='$nthreads'/' \
```

```

    PRODaer/enchaine.sh
fi
}

```

Lancement des simulations

Fait appel à `enchaine.sh`, le script d'enchainement des simulations mensuelles de TUTORIAL

```

function simulation {
case $1 in
    Inca) flag_aerosol=6 ;;
    SP) flag_aerosol=7 ;;
    *) echo Cas non prévu ; exit 1 ;;
esac
cp -r PRODaer $1
cd $1
sed -i'' -e 's/^flag_aerosol=.*$/flag_aerosol='$flag_aerosol'/' DEF
    /config.def
sed -i'' -e 's/^flag_rrtm=.*$/flag_rrtm=1/' -e 's/^NSW=.*$/NSW=6/'
    DEF/physiq.def
./enchaine.sh
}

```

Préparation d'un script de visualisation ferret

Ce script `trace.jnl` n'est pas lancé automatiquement.

```

function plot_figures {
var=od550aer # Variable pour les tracers

```

2 cas rad) Partie rad non automatique gardée pour mémoire SP) comparaison Inca et SP

```

diff=SP
case $diff in
    rad) ctl=PRODaer0
        bis=PRODaerInca
        ctl_name="Inca Old rad"
        bis_name="RRTM" ;;
    SP)  ctl=Inca
        bis=SP
        ctl_name="Inca"
        bis_name="SP" ;;
esac

```

Concaténation des fichiers `histday.nc`

```

for sim in $ctl $bis ; do
    cd $sim ; if [ ! -f histday.nc ] ; then cdo mergetime histday.2*
        histday.nc ; fi ; cd -
done

```

Ouverture des fichiers et calcul des valeurs intégrées

```

cat > trace.jnl <<eod
go white
use $ctl/histday.nc
use $bis/histday.nc
let odyr=od550aer[i=@ave,j=@ave,l=@ave]
let odjan=od550aer[i=@ave,j=@ave,l=1:30@ave]
list "Annuel : PI, PD=PI+CC, PI+CCjan, CC, CCjan",odyr[d=1],odyr[d
    =2],odyr[d=1]+odjan[d=2]-odjan[d=1],odyr[d=2]-odyr[d=1],odjan[d
    =2]-odjan[d=1]
eod

```

Boucle sur les localisations de tracers (y compris global)

```

views=( ul ur ll lr )
iv=0
stats="SouthAmerica Dakar Manaus Paris"
stats="Glob NorthAfrica EastAsia NorthAmerica"
for stat in $stats ; do
ave=""
case $stat in
Glob)    max=0.14 ; ave='[i=@ave,j=@ave]' ;;
Dakar)   max=1.    ; lat=15. ; lon=-15. ;;
Manaus)  max=0.05 ; lat=-3.1 ; lon=-60. ;;
Paris)   max=0.3   ; lat=48. ; lon=2.   ;;
SouthAmerica) max=0.1 ; lat=-19.3 ; lon=298. ;;
NorthAfrica)  max=0.8 ; lat=3.5 ; lon=22.5 ;;
EastAsia)     max=1. ; lat=30. ; lon=114. ;;
NorthAmerica) max=0.2 ; lat=40.1 ; lon=277.5 ;;
esac
if [ $stat != Glob ] ; then
    reg="/x=$lon/y=$lat"
fi
cat>> trace.jnl<<eod
let varm=od550aer$ave
set v ${views[$iv]}
plot$reg/d=1/vlim=0:$max/line=7/title="AOD $stat $ctl_name" varm
plot$reg/d=2/o/line=8/title="$bis_name" varm
plot$reg/d=2/o/line=9/title="DIFF" varm[d=2]-varm[d=1]
eod
iv=$(( $iv + 1 ))
done
}
main

```