In [ ]:

```

```

Effect of Maxiter in SA (RH100)

In [1]:

```julia
# Load modules
using Plots
using NCDatasets
using Statistics: mean
using DelimitedFiles
using PrettyTables
using PaddedViews
using Dates
using Printf

```

In [95]:

```julia

# Useful functions
function get_var(file_name, var_name, t_spinup, nan_fill_value);
    ds = NCDataset(file_name, "r");
    var = ds[var_name][:];
    data=var[:, :, :, t_spinup:end]; # cut out the spinup;
    replace!(data, NaN=>nan_fill_value)
    return data
    none
end

function get_zonal_mean(file_name, var_name, nan_fill_value);
    ds = NCDataset(file_name, "r");
    var = ds[var_name][:];
    data_mean = mean( var[:,:,:,:], dims=1); # lon, lat,lev, time
    replace!(data_mean, NaN=>nan_fill_value)
    return data_mean[1,:,:,:]
    none
end

function get_slice(
                   file_name, var_name, nan_fill_value,
                   lon_i1, lon_12, lat_i1, lat_i2, lev_i1, lev_i2, t_i1, t_i2

    ds = NCDataset(file_name, "r");
    var = ds[var_name][:];
    data = (var[lon_i1:lon_12, lat_i1:lat_i2, lev_i1:lev_i2, t_i1:t_i2]); # c
    replace!(data, NaN=>nan_fill_value)
    return data
    none
end

function get_coords(file_name)
    ds = NCDataset(file_name, "r");
    lon = ds["long"][:];
    lat = ds["lat"][:];
    lev = ds["level"][:] / 1e3; # height in kilometers
    time = ds["time"][:];  # time
    close(ds)
    return lon, lat, lev, time
    none
end

function get_min_max(var);
    vmax = maximum(filter(!isnan,var))
    vmin = minimum(filter(!isnan,var))
    #vmax = findmax(var)
    #vmin = findmin(var)
    return vmin,vmax
    none
```

```
end

function get_short_expname(fname, var_code_1)
    driver = fname
    exp_name =  split(split(driver,var_code_1)[2],"_")[1]
    return exp_name
end

```

Out[95]:

```
get_short_expname (generic function with 1 method)
```

In [4]:

```

# Specify needed directories and filenames: these will be replaced automatica
CLIMA_ANALYSIS = "/central/scratch/elencz/output/hier_RH100_q_active_np128_re
CLIMA_NETCDF = "/central/scratch/elencz/output/SA_crash_data/100RHmaxiter"
CLIMA_LOGFILE = "/central/scratch/elencz/output/hier_RH100_q_active_np128_rel

# Get the current and previous  GCM netcdf file names in the CLIMA_NETCDF dir
fnames = filter(x -> occursin(".nc", x), readdir( CLIMA_NETCDF ) );


# set file name
filename = "$CLIMA_NETCDF/"fnames[1]

# print data info:
ds = NCDataset(filename, "r")


```

Out[4]:

```
NCDataset: /central/scratch/elencz/output/SA_crash_data/100RHmaxiter/ctrl_hier_R
H100_q_active_np128_relax60_diffn_none_remove_q_none_AtmosGCMDefault_2020-08-24T
15.19.01.998.nc
Group: /

Dimensions
   long = 361
   lat = 181
   level = 31
   time = 18

Variables
  long   (361)
    Datatype:    Float64
    Dimensions:  long
    Attributes:
     units                = degrees_east
     long_name            = longitude

  lat    (181)
```

In [5]:

```
fnames
```

Out[5]:

```
2-element Array{String,1}:
 "ctrl_hier_RH100_q_active_np128_relax60_diffn_none_remove_q_none_AtmosGCMDefaul
t_2020-08-24T15.19.01.998.nc"
 "hier_RH100_q_active_np128_relax60_10maxiter_diffn_none_remove_q_none_AtmosGCMD
efault_2020-08-25T01.31.59.292.nc"
```

In [6]:

```
# get coordinates
lon, lat, lev, time = get_coords(filename);

nan_fill_value = -9999
nexp = size(fnames);
```

Vertical Slices

In [7]:

```
#Slices
```

```
t_slice = 18
z_slice = 5
var_name = "v"

# vertical
lon_i1, lon_12, lat_i1, lat_i2, lev_i1, lev_i2, t_i1, t_i2 = [1, length(lon),



# control
hs_c = get_slice( filename, var_name, nan_fill_value, lon_i1, lon_12, lat_i1,

p_ctrl = contourf( lon, lat, (hs_c[:,:,1,1])', title="$var_name (ctrl relax=6
#
# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  hs = get_slice( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value, lon_i1
  exp_name = get_short_expname(fnames[i], "relax60_")
  z_in_km = lev[z_slice]
  one_plot = contourf( lon, lat, ( (hs[:,:,1,1]-hs_c[:,:,1,1])*1.0 )', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(800, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))



display(fig)




```

```
contours not sorted in ascending order
GKS: Possible loss of precision in routine SET_WINDOW
GKS: Rectangle definition is invalid in routine SET_WINDOW

InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
 [15] gr_display(::Plots.Plot{Plots.GRBackend}, ::String) at /home/elencz/.julia
/packages/Plots/jpF9l/src/backends/gr.jl:678
 [16] _show(::Base.GenericIOBuffer{Array{UInt8,1}}, ::MIME{Symbol("image/svg+xm
l")}, ::Plots.Plot{Plots.GRBackend}) at /home/elencz/.julia/packages/Plots/jpF9l
/src/backends/gr.jl:1968
 [17] show(::Base.GenericIOBuffer{Array{UInt8,1}}, ::MIME{Symbol("image/svg+xm
l")}, ::Plots.Plot{Plots.GRBackend}) at /home/elencz/.julia/packages/Plots/jpF9l
```

```
/src/output.jl:215
 [18] sprint(::Function, ::MIME{Symbol("image/svg+xml")}, ::Vararg{Any,N} where
N; context::Nothing, sizehint::Int64) at ./strings/io.jl:105
 [19] sprint at ./strings/io.jl:101 [inlined]
 [20] _ijulia_display_dict(::Plots.Plot{Plots.GRBackend}) at /home/elencz/.julia
/packages/Plots/jpF9l/src/ijulia.jl:53
 [21] display_dict at /home/elencz/.julia/packages/Plots/jpF9l/src/init.jl:73 [i
nlined]
 [22] display(::IJulia.InlineDisplay, ::Plots.Plot{Plots.GRBackend}) at /home/el
encz/.julia/packages/IJulia/DrVMH/src/inline.jl:95
 [23] display(::Any) at ./multimedia.jl:323
 [24] top-level scope at In[7]:27
```

In [8]:

```
mean( (hs[:,:,1,1]-hs_c[:,:,1,1])*1.0 )
```

```
UndefVarError: hs not defined

Stacktrace:
 [1] top-level scope at In[8]:1
```

Zonal means

In [9]:

```
# get_zonal_means
t_slice = 18
var_name = "u"

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name (ctrl re

# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  zm = get_zonal_mean( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value);
  exp_name = get_short_expname(fnames[i], "relax60_")
  one_plot = contourf( lat, lev, (zm[:,:,t_slice]-zm_c[:,:,t_slice])', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(1000, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))
display(fig)

```

```
contours not sorted in ascending order
GKS: Rectangle definition is invalid in routine SET_WINDOW
```

```
InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
```

In [10]:

```julia
# get_zonal_mean: T
t_slice = 18
var_name = "temp"

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name (ctrl re

# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  zm = get_zonal_mean( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value);
  exp_name = get_short_expname(fnames[i], "relax60_")
  one_plot = contourf( lat, lev, (zm[:,:,t_slice]-zm_c[:,:,t_slice])', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(1000, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))
display(fig)


```

```
contours not sorted in ascending order
GKS: Rectangle definition is invalid in routine SET_WINDOW
```

```
InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
 [15] gr_display(::Plots.Plot{Plots.GRBackend}, ::String) at /home/elencz/.julia
```

In [ ]:

In [11]:

```julia
# get_zonal_mean: qt
t_slice = 18
var_name = "qt"

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name (ctrl re

# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  zm = get_zonal_mean( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value);
  exp_name = get_short_expname(fnames[i], "relax60_")
  one_plot = contourf( lat, lev, (zm[:,:,t_slice]-zm_c[:,:,t_slice])', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(1000, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))
display(fig)
```

```
contours not sorted in ascending order
GKS: Rectangle definition is invalid in routine SET_WINDOW
```

```
InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
 [15] gr_display(::Plots.Plot{Plots.GRBackend}, ::String) at /home/elencz/.julia
/packages/Plots/jpF9l/src/backends/gr.jl:678
 [16] show(::Base.GenericIOBuffer{Array{UInt8,1}}, ::MIME{Symbol("image/svg+xm
```

In [12]:

```julia
# get_zonal_mean: ql
t_slice = 18
var_name = "ql"

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name (ctrl re

# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  zm = get_zonal_mean( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value);
  exp_name = get_short_expname(fnames[i], "relax60_")
  one_plot = contourf( lat, lev, (zm[:,:,t_slice]-zm_c[:,:,t_slice])', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(1000, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))
display(fig)
```

```
contours not sorted in ascending order
GKS: Rectangle definition is invalid in routine SET_WINDOW
```

```
InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
 [15] gr_display(::Plots.Plot{Plots.GRBackend}, ::String) at /home/elencz/.julia
/packages/Plots/jpF9l/src/backends/gr.jl:678
```

In [13]:

```julia
# get_zonal_mean: qi
t_slice = 18
var_name = "qi"

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name (ctrl re

# Make anomaly plots and save them in an array
plot_array = Any[p_ctrl]; # can type this more strictly
for i in 2:nexp[1]
  zm = get_zonal_mean( "$CLIMA_NETCDF/"fnames[i], var_name, nan_fill_value);
  exp_name = get_short_expname(fnames[i], "relax60_")
  one_plot = contourf( lat, lev, (zm[:,:,t_slice]-zm_c[:,:,t_slice])', title=
  push!(plot_array,one_plot); # make a plot and add it to the plot_array
end

fig=plot(plot_array... , layout=(1, 2), size=(1000, 400) )
#savefig(fig, string("$CLIMA_ANALYSIS/plot_$var_name","_hovmoller_sens.pdf"))
display(fig)
```

```
contours not sorted in ascending order
GKS: Rectangle definition is invalid in routine SET_WINDOW
```

```
InexactError: trunc(Int64, NaN)

Stacktrace:
 [1] trunc at ./float.jl:703 [inlined]
 [2] round at ./float.jl:367 [inlined]
 [3] _broadcast_getindex_evalf at ./broadcast.jl:631 [inlined]
 [4] _broadcast_getindex at ./broadcast.jl:614 [inlined]
 [5] getindex at ./broadcast.jl:564 [inlined]
 [6] macro expansion at ./broadcast.jl:910 [inlined]
 [7] macro expansion at ./simdloop.jl:77 [inlined]
 [8] copyto! at ./broadcast.jl:909 [inlined]
 [9] copyto! at ./broadcast.jl:864 [inlined]
 [10] copy at ./broadcast.jl:840 [inlined]
 [11] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle
{1},Nothing,typeof(round),Tuple{Base.RefValue{Type{Int64}},StepRangeLen{Float64,
Base.TwicePrecision{Float64},Base.TwicePrecision{Float64}}}}) at ./broadcast.jl:
820
 [12] gr_colorbar_colors(::Plots.Series, ::Tuple{Float64,Float64}) at /home/elen
cz/.julia/packages/Plots/jpF9l/src/backends/gr.jl:486
 [13] gr_draw_colorbar(::Plots.GRColorbar, ::Plots.Subplot{Plots.GRBackend}, ::T
uple{Float64,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/packages/Plots
/jpF9l/src/backends/gr.jl:527
 [14] gr_display(::Plots.Subplot{Plots.GRBackend}, ::Measures.Length{:mm,Float6
4}, ::Measures.Length{:mm,Float64}, ::Array{Float64,1}) at /home/elencz/.julia/p
ackages/Plots/jpF9l/src/backends/gr.jl:1846
```

Maxiter10

In [30]:

```julia
# Check showing that model is bit-consistent and exactly reproducible
var_name = "v"
one = get_slice( "$CLIMA_NETCDF/"fnames[1], var_name, nan_fill_value, lon_i1,
two = get_slice( "$CLIMA_NETCDF/"fnames[2], var_name, nan_fill_value, lon_i1,

println(one[10,40,:])

println(two[10,40,:])

println(mean(one))

println(mean(two))

println(NCDataset("$CLIMA_NETCDF/"fnames[1], "r") )

println(NCDataset("$CLIMA_NETCDF/"fnames[2], "r") )
```

```
[0.18795765955131827]
[0.18795765955131827]
-7.017834088101161e-5
-7.017834088101161e-5
NCDataset: /central/scratch/elencz/output/SA_crash_data/100RHmaxiter/ctrl_hier_R
H100_q_active_np128_relax60_diffn_none_remove_q_none_AtmosGCMDefault_2020-08-24T
15.19.01.998.nc
Group: /

Dimensions
   long = 361
   lat = 181
   level = 31
   time = 18

Variables
  long   (361)
    Datatype:    Float64
    Dimensions:  long
```

In [31]:

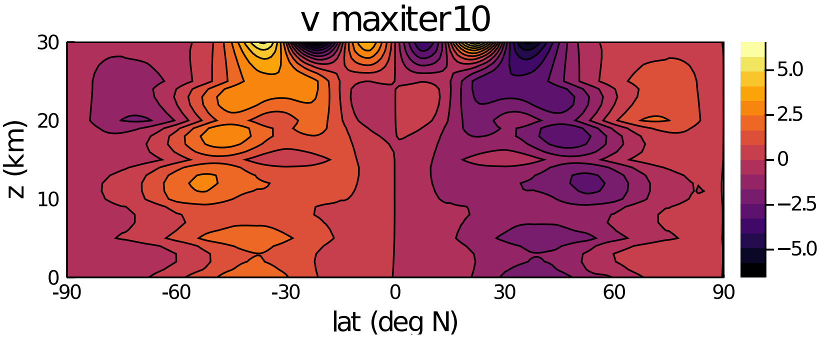
```
filename = "$CLIMA_NETCDF/"fnames[2]
```

Out[31]:

```
"/central/scratch/elencz/output/SA_crash_data/100RHmaxiter/hier_RH100_q_active_n
p128_relax60_10maxiter_diffn_none_remove_q_none_AtmosGCMDefault_2020-08-25T01.3
1.59.292.nc"
```

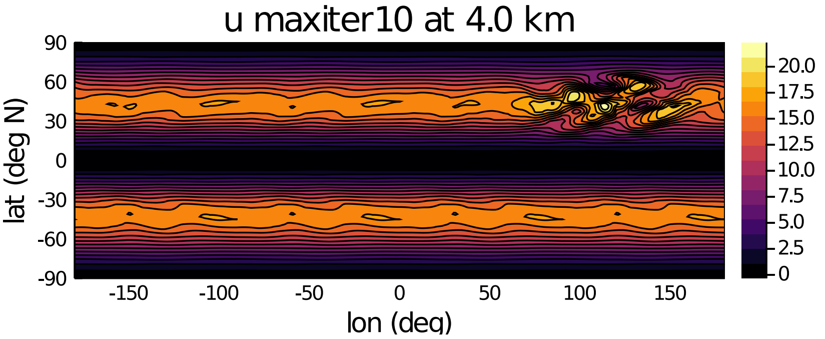
In [32]:

```
# Zonal Means
var_name = "v"
t_slice = 38

zm_c = get_zonal_mean(filename, var_name, nan_fill_value)
p_ctrl = contourf( lat, lev, (zm_c[:,:,t_slice])' , title="$var_name maxiter1
fig=plot(p_ctrl, size=(500, 200) )
display(fig)
```



In [33]:

```
#Slices
t_slice = 38
z_slice = 5
var_name = "u"

# vertical
lon_i1, lon_12, lat_i1, lat_i2, lev_i1, lev_i2, t_i1, t_i2 = [1, length(lon),



# control
hs_c = get_slice( filename, var_name, nan_fill_value, lon_i1, lon_12, lat_i1,
z_in_km = lev[z_slice]
p_ctrl = contourf( lon, lat, (hs_c[:,:,1,1])', title="$var_name maxiter10 at

fig=plot(p_ctrl, size=(500, 200) )
display(fig)


```



Anim: vertical slices

In [112]:

```
# Setup run-time environmnent
ENV["GKSwstype"] = "100"

# Get zero padded exp data
```

```
var_name_list = ["v","vort"]
t_spinup = 1
nvar = length(var_name_list)
var_array = Any[]; # can type this more strictly
t_nos = Any[];
diag_dts=Any[]
dummy = get_var("$CLIMA_NETCDF/"fnames[1], var_name_list[1], t_spinup, nan_fi
clims = ( get_min_max(dummy) )
for i in 1:nexp[1]
  for n in 1:nvar[1]
    data = get_var("$CLIMA_NETCDF/"fnames[i], var_name_list[n], t_spinup, nan_
    lon, lat, lev, time = get_coords("$CLIMA_NETCDF/"fnames[i]);
    push!(var_array,data); # make a plot and add it to the plot_array
    push!(t_nos,size(time)[1]); # make a plot and add it to the plot_array
    diag_dt_days =  (time[2] - time[1]).value / (1000*60*60*24) # get simtime
    push!(diag_dts,diag_dt_days);
  end
end
max_time_no=maximum(t_nos) # number of timesteps of the longest running exper



clims_list=[]
for n in 1:nvar[1]
    dummy = get_var( "$CLIMA_NETCDF/"fnames[1], var_name_list[n], 1, nan_fill_
    clim = ( get_min_max(dummy) )
    push!(clims_list,clim);
end


anim = @animate for t_i in 2:max_time_no
  plot_array = Any[]; # can type this more strictly
  ct = 0
  for i in 1:nexp[1]
    for n in 1:nvar[1]
      ct += 1
      var_array_pad = PaddedView(nan_fill_value, var_array[ct], (size(lon)[1]
      vs = var_array_pad[:,:,z_slice,t_i]
      title = var_name_list[n]
      if i ==1
        z_in_km = lev[z_slice]
        title = title*" at $z_in_km km" # could add name of commit here
      end
      clims=clims_list[n]
      one_plot = contourf( lon, lat, vs', title = title, xlabel="lon (deg)",
      push!(plot_array,one_plot); # make a plot and add it to the plot_array
    end
  end
  plot(plot_array..., layout=(nexp[1],nvar[1]), size=(1000, 400) )
end
mp4(anim, string("$CLIMA_ANALYSIS/plot_horizontal_slice_anim.mp4"), fps = 5)
#display(anim)


```

```
┌ Info: Saved animation to
│   fn = /central/scratch/elencz/output/hier_RH100_q_active_np128_relax60/analys
is/plot_horizontal_slice_anim.mp4
└ @ Plots /home/elencz/.julia/packages/Plots/jpF9l/src/animation.jl:104
```

Out[112]:

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

Anim: zonal means

In [111]:

```julia
# Zonal Means
var_name_list = ["u","temp"]
t_spinup = 1
nvar = length(var_name_list)
var_array = Any[]; # can type this more strictly
t_nos = Any[];
diag_dts=Any[]

for i in 1:nexp[1]
  for n in 1:nvar[1]
    data = get_var("$CLIMA_NETCDF/"fnames[i], var_name_list[n], t_spinup, nan
    lon, lat, lev, time = get_coords("$CLIMA_NETCDF/"fnames[i]);
    push!(var_array,data); # make a plot and add it to the plot_array
    push!(t_nos,size(time)[1]); # make a plot and add it to the plot_array
    diag_dt_days =  (time[2] - time[1]).value / (1000*60*60*24) # get simtime
    push!(diag_dts,diag_dt_days);
  end
end
max_time_no=maximum(t_nos) # number of timesteps of the longest running exper


clims_list=[]
for n in 1:nvar[1]
    dummy = get_var( "$CLIMA_NETCDF/"fnames[2], var_name_list[n], 1, nan_fill
    clim = ( get_min_max(dummy) )
    push!(clims_list,clim);
end


anim = @animate for t_i in 2:max_time_no
  plot_array = Any[]; # can type this more strictly
  ct = 0
  for i in 1:nexp[1]
    for n in 1:nvar[1]
      ct += 1
      var_array_pad = PaddedView(nan_fill_value, var_array[ct], (size(lon)[1]
      vs = mean(var_array_pad[:,:,:,t_i], dims=1)[1,:,:,1];
      title = var_name_list[n]
      if i ==1
        time_slice = t_i * diag_dts[ct]
        title = title*" @ $time_slice" # could add name of commit here
      end
      clims = clims_list[n]
      one_plot = contourf( lat, lev, vs', title = title, xlabel="lat (deg N)"
      push!(plot_array,one_plot); # make a plot and add it to the plot_array
    end
  end
    plot(plot_array..., layout=(nexp[1],nvar[1]), size=(1000, 400) )
end
mp4(anim, string("$CLIMA_ANALYSIS/plot_zonal_mean_anim.mp4"), fps = 5) # hide
#display(anim)
```

```
┌ Info: Saved animation to
│   fn = /central/scratch/elencz/output/hier_RH100_q_active_np128_relax60/analys
```

Out[111]:

In [ ]:

In [104]:

Out[104]: 2

In [ ]:

In [ ]:

In [ ]:

```
# Additional Anims: Moisture
```

In [137]:

```
# Setup run-time environment
ENV["GKSwstype"] = "100"

# Get zero padded exp data
var_name_list = ["qt","ql","qi"]
t_spinup = 1
nvar = length(var_name_list)
var_array = Any[]; # can type this more strictly
t_nos = Any[];
diag_dts=Any[]
dummy = get_var("$CLIMA_NETCDF/"fnames[1], var_name_list[1], t_spinup, nan_fi
clims = ( get_min_max(dummy) )
for i in 1:nexp[1]
  for n in 1:nvar[1]
    data = get_var("$CLIMA_NETCDF/"fnames[i], var_name_list[n], t_spinup, nan
    lon, lat, lev, time = get_coords("$CLIMA_NETCDF/"fnames[i]);
    push!(var_array,data); # make a plot and add it to the plot_array
    push!(t_nos,size(time)[1]); # make a plot and add it to the plot_array
    diag_dt_days =  (time[2] - time[1]).value / (1000*60*60*24) # get simtime
    push!(diag_dts,diag_dt_days);
  end
end
max_time_no=maximum(t_nos) # number of timesteps of the longest running exper



clims_list=[]
```

```julia
for n in 1:nvar[1]
    dummy = get_var( "$CLIMA_NETCDF/"fnames[1], var_name_list[n], 1, nan_fill
    clim = ( get_min_max(-dummy) )
    push!(clims_list,clim);
end


anim = @animate for t_i in 2:max_time_no
  plot_array = Any[]; # can type this more strictly
  ct = 0
  for i in 1:nexp[1]
    for n in 1:nvar[1]
      ct += 1
      var_array_pad = PaddedView(nan_fill_value, var_array[ct], (size(lon)[1]
      vs = var_array_pad[:,:,z_slice,t_i]
      title = var_name_list[n]
      if i ==1
        z_in_km = lev[z_slice]
        title = title*" at $z_in_km km" # could add name of commit here
      end
      clims=clims_list[n]
      one_plot = contourf( lon, lat, -vs', title = title, xlabel="lon (deg)",
      push!(plot_array,one_plot); # make a plot and add it to the plot_array
    end
  end
  plot(plot_array..., layout=(nexp[1],nvar[1]), size=(1000, 400) )
end
mp4(anim, string("$CLIMA_ANALYSIS/plot_horizontal_slice_anim.mp4"), fps = 5)
#display(anim)
```

```
┌ Info: Saved animation to
│   fn = /central/scratch/elencz/output/hier_RH100_q_active_np128_relax60/analys
is/plot_horizontal_slice_anim.mp4
└ @ Plots /home/elencz/.julia/packages/Plots/jpF9l/src/animation.jl:104
```

Out[137]:

In [138]:

```julia
# Zonal Means
var_name_list = ["qt","ql","qi"]
t_spinup = 1
nvar = length(var_name_list)
var_array = Any[]; # can type this more strictly
t_nos = Any[];
diag_dts=Any[]

for i in 1:nexp[1]
  for n in 1:nvar[1]
  data = get_var("$CLIMA_NETCDF/"fnames[i], var_name_list[n], t_spinup, nan
  lon, lat, lev, time = get_coords("$CLIMA_NETCDF/"fnames[i]);
  push!(var_array,data); # make a plot and add it to the plot_array
  push!(t_nos,size(time)[1]); # make a plot and add it to the plot_array
  diag_dt_days =  (time[2] - time[1]).value / (1000*60*60*24) # get simtime
  push!(diag_dts,diag_dt_days);
 end
end
max_time_no=maximum(t_nos) # number of timesteps of the longest running exper


clims_list=[]
for n in 1:nvar[1]
  dummy = get_var( "$CLIMA_NETCDF/"fnames[2], var_name_list[n], 1, nan_fill
  clim = ( get_min_max(-dummy) )
  push!(clims_list,clim);
end


anim = @animate for t_i in 2:max_time_no
 plot_array = Any[]; # can type this more strictly
 ct = 0
 for i in 1:nexp[1]
  for n in 1:nvar[1]
   ct += 1
   var_array_pad = PaddedView(nan_fill_value, var_array[ct], (size(lon)[1]
   vs = mean(var_array_pad[:,:,:,t_i], dims=1)[1,:,:,1];
   title = var_name_list[n]
   if i ==1
    time_slice = t_i * diag_dts[ct]
    title = title*" @ $time_slice" # could add name of commit here
   end
   clims = clims_list[n]
   one_plot = contourf( lat, lev, -vs', title = title, xlabel="lat (deg N)
   push!(plot_array,one_plot); # make a plot and add it to the plot_array
  end
 end
  plot(plot_array..., layout=(nexp[1],nvar[1]), size=(1000, 400) )
end
mp4(anim, string("$CLIMA_ANALYSIS/plot_zonal_mean_anim.mp4"), fps = 5) # hide
#display(anim)
```

```
┌ Info: Saved animation to
│   fn = /central/scratch/elencz/output/hier_RH100_q_active_np128_relax60/analys
is/plot_zonal_mean_anim.mp4
└ @ Plots /home/elencz/.julia/packages/Plots/jpF9l/src/animation.jl:104
```

Out[138]:

# Min and max

In [142]:

```
var_name_list = ["u","v","w","temp","qt","ql","qi"]
nvar = length(var_name_list)
for n in 1:nvar[1]
    dummy = get_var( "$CLIMA_NETCDF/"fnames[2], var_name_list[n], 1, nan_fill
    println(var_name_list[n])
    println(findmax(dummy))
    println(findmin(dummy))
end
```

```
u
(30.920331585820954, CartesianIndex(313, 143, 11, 38))
(-12.623126544116376, CartesianIndex(145, 66, 31, 13))
v
(24.221303137218932, CartesianIndex(303, 139, 1, 37))
(-18.64945652793599, CartesianIndex(290, 141, 1, 38))
w
(0.6579552789475613, CartesianIndex(290, 141, 1, 36))
(-1.117780806788111, CartesianIndex(299, 140, 1, 37))
temp
(310.00000000001245, CartesianIndex(289, 91, 1, 1))
(151.25123500341755, CartesianIndex(226, 101, 31, 11))
qt
(0.009602434464702807, CartesianIndex(44, 80, 1, 7))
(-2.6205702991723398e-5, CartesianIndex(288, 136, 6, 36))
ql
(0.002335920671977674, CartesianIndex(286, 135, 1, 33))
(-0.00044821793342733856, CartesianIndex(287, 134, 1, 35))
qi
(0.0028153314321946628, CartesianIndex(295, 141, 1, 37))
(-0.0003960787536779044, CartesianIndex(290, 135, 1, 34))
```

In [ ]:

```

```

In [ ]:

```
# Convert into this form later

@recipe function f(dummy::DummyType)
    @series begin
        seriestype := :contourf
        seriescolor --> :bluesreds

        [i+j for i in 1:10, j in 1:10]
    end
end
plot(DummyType(), color=:plasma)
```