User guide for NCLscripts

timeseries.ncl – profiles.ncl – cross_sections.ncl – spectra.ncl

- *timeseries.ncl* draws line plots from time_series-netCDF data produced by PALM
- prompt to run the script:

ncl timeseries.ncl [parameter_value=value] ['parameter_string="string" ']

- *profiles.ncl* draws profile line plots from profiles-netCDF or 3d-netCDF data produced by PALM
- prompt to run the script:

ncl profiles.ncl [parameter_value=value] ['parameter_string="string" ']

- *cross_sections.ncl* draws contour, isoline or vector plots (of a 2-dimensional vector) from 2d- or 3d-NetCDF data produced by PALM; instantaneous or time-averaged xy-, xz-, yz- or 3D-data ca n be used
- possibility of overlaying different plot types
- prompt to run the script:

ncl cross_sections.ncl [parameter_value=value] ['parameter_string="string" ']

- *spectra.ncl* draws NCL spectra plots from spectra-netCDF data produced by PALM
- prompt to run the script:

ncl spectra.ncl [parameter_value=value] ['parameter_string="string" ']

PARAMETER LIST

parameter_value/ 'parameter_string'		value	used by	default	meaning
'file_1'	required	STRING [e.g.“path/name.nc”]	all		name and location of input file
'format_out'	optional	STRING ["ngcm", "ps", "eps", "epsi", “pdf" or "x11"]	all	“x11”	format of output file
'file_out'	optional	STRING [e.g. “path/name”]	all	“test”	name and location for output file
start_time_step	optional	DFLOAT [e.g. 0.5 - unit is hour]	all	first time step of input file	first time step of plot; you don't have to indicate a precise time value of the input file, your *start_time_step* value will be rounded if not exactly existent on the input file
end_time_step	optional	DFLOAT [e.g. 2.5 - unit is hour]	all	last time step of input file	last time step of plot; you don't have to indicate a precise time value of the input file, your *end_time_step* value will be rounded if not exactly existent on the input file
no_columns	optional	INTEGER	all	1	number of plots in one row

parameter_value/ 'parameter_string'		value	used by	default	meaning
no_lines	optional	INTEGER	all	2	number of plots in one column
'var'	optional	STRING [e.g. “,ws2,pt,”]	all	all variables	names of variables that shall be plotted; be sure to have one comma before and after every variable name (also before the first and after the last one!)
combine	optional	INTEGER [0 or 1]	profile	0	plot with more than one variable will be switched on [1] or off [0]
number_comb	required if combine =1	INTEGER [2 or 3]	profile	0	number of variables that shall be plotted together in one plot; 2 or 3 is possible
'c_var'	required if combine =1	STRING [e.g. “,umax,vmax,”]	profile		names of variables that shall be plotted together in one plot
dash	optional	INTEGER [0 or 1]	profile & spectra	0	use of different line patterns [1] or continous lines [0]
black	optional	INTEGER [0 or 1]	profile & spectra	0	colored [0] or black and white [1] plots
min_z	optional	FLOAT [e.g. 100 - unit is m]	profile	minimum height of input file	minimum height of profiles
max_z	optional	FLOAT [e.g. 500 - unit is m]	profile	maximum height of input file	maximum height of profiles
over	optional	INTEGER [0 or 1]	time & profile	0	defined overlaying of the standard variables will be switched on [1] or off [0]; this feature cannot be used with statistic regions
vector	optional	INTEGER [0 or 1]	cross	0	vector plots for one vector will be switched on [1] or off [0]
'vec1'	required if vector = 1	STRING [e.g. “,u,”]	cross		name of variable for first component of vector for vector plot
'vec2'	required if vector = 1	STRING [e.g. “,v,”]	cross		name of variable for second component of vector for vector plot
'plotvec'	optional	STRING [e.g. “,u,v,”]	cross		variables where a vector plot shall overlay if desired
ref_mag	optional	FLOAT [e.g 0.1]	cross	0.05	value of referenced vector with defined length [can be seen in legend]
'mode'	optional	STRING [“Fill”, “Line” or “Both”]	cross	“Fill”	contour plots [“Fill”], isoline plots [“Line”] or both [“Both”] will be drawn

parameter_value/ 'parameter_string'		value	used by	default	meaning
'sort'	optional	STRING ["layer" or "time"]	cross & spectra	"layer"	defines the sequence of plots; either by time step ["time"] or by layer/height ["layer"]
'fill_mode'	optional	STRING ["AreaFill", "RasterFill" or "CellFill"]	cross	"AreaFill"	Style of filling the contour plots
shape	optional	INTEGER [0 or 1]	cross	1	aspect ratio of axis will be kept [1] or not [0]
xyc	required [one (only one!) of these parameters must be set to 1]	INTEGER [0 or 1]	cross	0	ouput of xy-cross sections will be switched on [1] or off [0]
xzc			cross	0	ouput of xz-cross sections will be switched on [1] or off [0]
yzc			cross	0	ouput of yz-cross sections will be switched on [1] or off [0]
xs	optional	cross: DFLOAT profile: FLOAT	cross & profile	first x value of input file	**cross**: start value of x-coordinate [m]; you don't have to indicate a precise x value of the input file, your input value xs will be rounded if not existent on the input file
					profile: minimum of x-range [dimension of input varaible]
xe	optional	cross: DFLOAT profile: FLOAT	cross & profile	last x value of input file	**cross**: end value of x-coordinate [m]
					profile: maximum of x-range [dimension of input varaible]
ys	optional	DFLOAT	cross	first y value of input file	start value of y-coordinate [m]
ye	optional	DFLOAT	cross	last x value of input file	end value of y-coordinate [m]
zs	optional	INTEGER	cross	first index of input file	first index of z-coordinate; due to grid stretching you have to indicate an index instead of meters
ze	optional	INTEGER	cross	last index of input file	last index for z-coordinate; due to grid stretching you have to indicate an index instead of meters
prof3d	optinal	INTEGER [0 or 1]	profile	0	*prof3d* must be switched on [1] if PALM 3d data and switched off [0] if PALM profile data is used
logy	optional	INTEGER [0 or 1]	profile & spectra	0	logarithmic scale for y-axis (and x-axis for spectra) will be switched on [1] or off [0]

parameter_value/ 'parameter_string'		value	used by	default	meaning
norm	optional	FLOAT [e.g. 100]	profile & spectra	1	value for normalising the y-axis
no_files	optional	INTEGER [1 to 12]	profiles	1	you can use up to 12 input files; the files must contain exactly the same variables and dimensionalities; identical variables will be drawn together; you cannot combine or overlay any variables; an extra legend (besides the time step legend) will be drawn where you can explicitily name the items
'name_legend_1'	optional	STRING	profiles	File 1	name of 1st item in legend
'file_2'	required if no_files > 1	STRING [e.g.“path/name.nc”]	profiles		name and location of 2nd input file
'name_legend_2'	optional	STRING	profiles	File 2	name of 2nd item in legend
'file_3'	required if no_files > 2	STRING [e.g.“path/name.nc”]	profiles		name and location of 3rd input file
'name_legend_3'	optional	STRING	profiles	File 3	name of 3rd item in legend
...	...	...	...	...	...
...	...	...	...	...	...
'file_12'	required if no_files > 11	STRING [e.g.“path/name.nc”]	profiles		name and location of 12th input file
'name_legend_12'	optional	STRING	profiles	File 12	name of 12th item in legend

REFERENCES

- instead of running the prompt with all parameters they can be written into the ascii file *.ncl_preferences* which will be read by the script; values/strings given in the prompt overwrite the list values/strings
- it is recommend to copy *.ncl_preferences* into your $home directory before modification
- *.ncl_preferences* runs for all three scripts, so some parameters will be ignored by the scripts and you don't need to consider them
- please check your strings and values in *.ncl_preferences* for correctness if the program aborts (it should not abort, if you use the original list with the default values); small differences may let abort the program (e.g. there must not be any blanks after the parameter strings); please be sure not to swap any rows in *.ncl_preferences*
- please check the type of your parameter values in the prompt if the program aborts (be sure to use the type that is indicated for all parameters; e.g. integer = **2**; float = **2.0**; dfloat = **2.0**; string = **“name”**)

EXAMPLES

Examples for *timeseries.ncl:*

1. minimal required prompt to get time serieses of all variables in the data file:
   ncl timeseries.ncl 'file_1="~/.../example_ts.nc"'

2. time serieses of all variables in the data file with four plots on one sheet and output to $home/time_out.pdf:
   ncl timeseries.ncl 'file_1="~/.../example_ts.nc"' 'format_out="pdf"' 'file_out="~/time_out"' no_columns=2 no_lines=2

Examples for *profiles.ncl:*

3. minimal required prompt to get profiles of all variables in the data file:
   ncl profiles.ncl 'file_1="~/.../example_pr.nc"'

4. one combined plot of two variables (pt, wpt):
   ncl profiles.ncl 'file_1="~/.../example_pr.nc"' 'var=",pt,wpt,"' combine=1 number_comb=2 'c_var=",pt,wpt,"'

Examples for *cross_sections.ncl:*

5. minimal required prompt to get cross sections of all variables in the data file for one cross section (xy in this example):
   ncl cross_sections.ncl 'file_1="~/.../example_3d_av.nc"' xyc=1

6. three vector plots of all variables in the data file; yz-cross section of all layers from time step 1 to 3 in one column
   ncl cross_sections.ncl 'file_1="~/.../example_3d_av.nc"' yzc=1 vector=1 'vec1="v"' 'vec2="w"' start_time_step=1 end_time_step=3 no_columns=1 no_lines=3

7. contour with isoline plots of all variables in the data file sorted by layer (xy cross section); vector plots will be overlayed on variable u:
   ncl cross_sections.ncl 'file_1="~/.../example_3d_av.nc"' xyc=1 'mode="Both"' vector=1 ' vec1=",u,"' 'vec2=",v,"' 'plotvec=",u,"' 'sort="layer"'

Examples for spectra*.ncl:*

8. minimal required prompt to get spetra plots of all variables in the data file:
   nlc spectra.ncl 'file_1="~/.../example_sp.nc"'