Conf-940115-27

PNL-SA-23046

INTERACTIVE CALCULATOR FOR USE WITH MULTI-DIMENSIONAL NETCDF FILES

D. W. Denbo[a]

January 1994

Presented at the
10th International Conference on Interactive
Information & Processing Systems for Meteorology,
Oceanography & Hydrology
January 23-28, 1994
Nashville, Tennessee

Work supported by
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

(a) Pacific Northwest Laboratory
Sequim, Washington

INTERACTIVE CALCULATOR FOR USE WITH MULTI-DIMENSIONAL NETCDF FILES

Donald W. Denbo

Pacific Northwest Laboratory
Sequim, Washington

1. INTRODUCTION

The interactive exploration of numerical results and observations has become increasingly difficult as the size of these data sets becomes larger. The scientist needs to be able to easily extract subsets of very large data sets and use these subsets in algebraic expressions to test hypotheses about the relationships between variables. The netCDF calculator (*nccalc*) satisfies many of these needs by providing several built-in functions and a framework that helps the scientist manipulate four-dimensional data sets written using the netCDF (Rew and Davis, 1990) data format.

2. THE CALCULATOR

nccalc is a simple programmable interpreter for floating point, string, and field expressions. It has C-style control flow, function definition, and the usual built-in numerical functions. *nccalc* is written in C and uses lex, a lexical analyzer, and yacc, a parser generator. This allows a systematic and consistent syntax to be implemented easily. The eps library (Denbo and Zhu, 1993), developed at the Pacific Marine Environmental Laboratory of the National Oceanic and Atmospheric Administration (NOAA/PMEL), has been used to provide a simplified and consistent interface to netCDF.

nccalc has five different data types: scalar (double precision floating point), string, slab (a specification of a four-dimensional hyper-slab), dbase (a pointer to a netCDF file and its associated attributes), and fields (a sub-sample of a four-dimensional data set). The syntax for each of the types is similar; however, not all operations are valid for all data types. The following discussion of operators and built-in functions will be restricted to the field variable type.

The standard "calculator" operators (division, multiplication, addition, subtraction and exponentiation) are available for fields and are computed pointwise. *nccalc* will automatically regrid fields that are not on identical grids for any dimension and have a length greater than one. For example, two subsets from three-dimensional fields (x,y,z) that have 15 x values, 10 y values and 1 z value, can have different z values, but if the x and y dimensions are not identical, the fields will be regridded.

In addition to the 5 basic operators, *nccalc* has over 30 built-in functions that operate on fields. These functions include transcendental functions (exp, log, log10, and sqrt), the multi-dimensional Fourier transform, finite difference functions, and statistical functions (ave, sum, min, max). Several of these functions allow the user to select for which dimensions the function will be applied. For example, if fld is a three-dimensional field (x,y,z), then min(fld,[x=*,y=*]) will return a one-dimensional field that contains the minimum value of fld at each z value.

nccalc can have multiple netCDF files open for both reading and writing. Global and variable attributes can be modified within *nccalc*. The user can easily use fields from two separate input files and write the results to a third output file.

nccalc can be called from within other programs. The connection is currently only one way, with *nccalc* transferring one- and two-dimensional fields and two component vectors to the calling program. The graphics program PPLUS (Denbo, 1993) has been connected to *nccalc* to provide interactive graphics to be made of the *nccalc* computations.

3. EXAMPLES

Three examples are presented below demonstrating the use and power of *nccalc*. These examples are from the analyses of numerical experiments in deep oceanic convection. In each example the *nccalc* commands are presented with explanations followed by a graphic produced by PPLUS.

3.1 Example 1: Vertical Temperature Flux

The vertical temperature flux is a quantitative measure of the advection of temperature in the vertical

Corresponding author address: Donald W. Denbo, Pacific Northwest Laboratory, Battelle/Marine Sciences Laboratory, 1529 W. Sequim Bay Road, Sequim, Washington 98382. Internet: dwd@mystery.pnl.gov

direction. The vertical derivative of this quantity determines whether temperature is increasing or decreasing locally. Vertical temperature flux is computed by multiplying the horizontally demeaned vertical velocity by the horizontally demeaned temperature. In this example, the vertical temperature flux is computed for the entire model domain at model day 1.5 and then horizontally averaged to create bulk temperature flux values at each depth (Figure 1). A *nccalc* script to compute this follows:

```
(1)   openr "c65-3.2.cdf"
(2)   slb=[x=*,y=*,z=*,t=1.5]
(3)   wp=prime(w[slb],[x=*,y=*])
(4)   tp=prime(t[slb],[x=*,y=*])
(5)   ave(wp*tp,[x=*,y=*])*1e4
```

(1) The netCDF model output file *c65-3.2.cdf* is opened for reading. (2) A slab *slb* is created for all x, y, and z values at time 1.5. (3) The *prime* function is called for all vertical velocities defined by *slb* (*prime* subtracts the mean calculated over all x and y's from each point) and the result is assigned to *wp*. (4) The *prime* function is called for all potential temperatures defined by *slb* and the result is assigned to *tp*. (5) The average of the product *wp* × *tp* is computed over all x and y's and then multiplied by 1000. The averaging process reduces the number of dimensions in the result by the number of dimensions that the average is computed over, thus the result is a function of depth only.

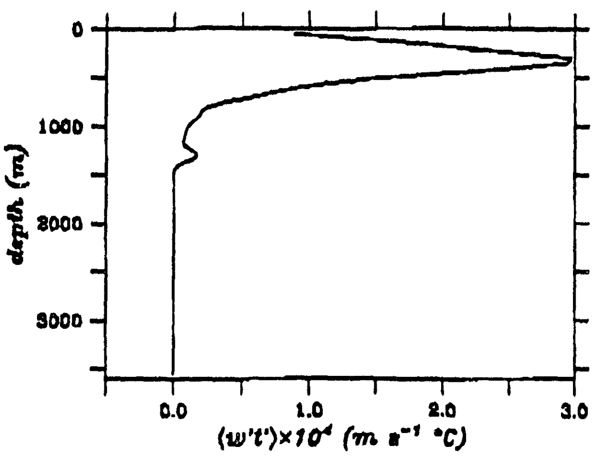

Figure 1. Horizontal average of vertical temperature flux (w't') at model day 1.5.

3.2 *Example 2: Vertical Temperature Flux*

The second example is similar to the first, but the horizontal range has been reduced and all times are computed. The changes in the vertical temperature flux over time provide the scientist with important information related to the onset of deep oceanic convection and its temporal structure (Figure 2). The *nccalc* script to compute vertical temperature flux as a function of depth and time follows:

```
(1)   openr "c65-3.2.cdf"
(2)   sl=[x=1000:2000,y=0:1000,z=0:2000,t=*]
(3)   wp=prime(w[sl],[x=*,y=*])
(4)   tp=prime(t[sl],[x=*,y=*])
(5)   ave(wp*tp,[x=*,y=*])*1e4
```

(1) The netCDF model output file *c65-3.2.cdf* is opened for reading. (2) A slab *sl* is created for x from 1000 to 2000 meters, y from 0 to 1000 meters, z from 0 to 2000 meters, and all available values of time. (3) The *prime* function is called for all vertical velocities defined by *sl* (*prime* subtracts the mean calculated over all x and y's, but not over time, from each point) and the result is assigned to *wp*. (4) The *prime* function is called for all potential temperatures defined by *sl* and the result is assigned to *tp*. (5) The average of the product *wp* × *tp* is computed over all x and y's and then multiplied by 1000. In this example, the averaging reduces the dimensions, leaving the result as a function of depth and time.

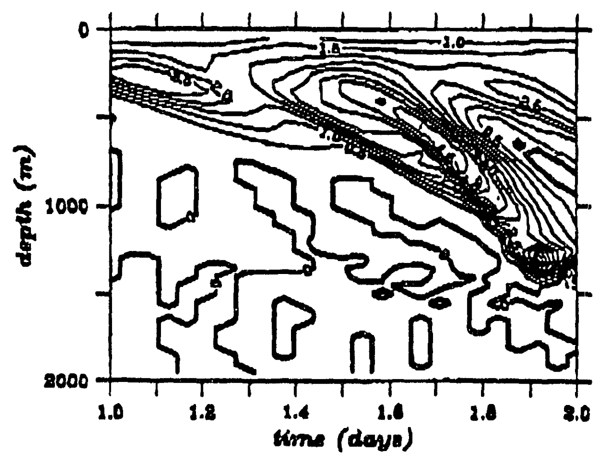

Figure 2. Horizontal average of vertical temperature flux ×10⁴ (m s⁻¹ °C).

3.3 *Example 3: Fourier Transform*

The horizontal scales of the potential temperature structure can be determined by computing the horizontal wavenumber spectra. In this example, the vertical distribution of temperature variance is found by computing the spectra at each depth and then ensemble averaging the horizontal spectra in the y dimension to improve the accuracy of the spectral estimates. The common logarithm of the spectrum is then computed for display purposes (Figure 3). The *nccalc* script for these calculations follows:

```
(1)  openr "c65-3.2.cdf"
(2)  slxyz=[x=*,y=*,z=*,t=1.5]
(3)  t_fft=ave(fft(t[slxyz],[x=*]),[y=*])
(4)  log10(t_fft+1e-10)
```

(1) The netCDF model output file *c65-3.2.cdf* is opened for reading. (2) A slab *slxyz* is created for all x, y, and z values at time 1.5. (3) A one-dimensional Fourier transform is computed in the x direction for all potential temperature defined by *slxyz*. The transform is then averaged in the y direction (ensemble averaging) and the result stored in the field variable *t_fft*. (NOTE: a two-dimensional transform is computed by specifying a second direction in the second argument of the *fft*). (4) The common logarithm is computed for *t_fft* with a constant offset of 1e-10. This offset is required to guarantee that no value is identically equal to zero.

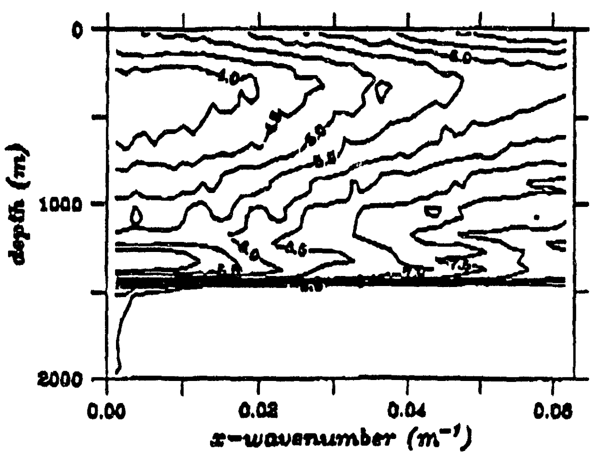

Figure 3. $\log_{10}$ of the horizontal Fourier transform of potential temperature.

4. FUTURE DIRECTIONS

nccalc has been in use for almost two years. During this time, *nccalc* has evolved to meet users' needs. The following features are planned for implementation in the near future.

Vectors: Many calculations can be made more intuitive by using vectors and vector functions. The functions to be implemented include divergence, curl, gradient, cross and dot products.

Grid generation: By creating empty fields the user will be able to break large computations into smaller pieces and assign the results to a subset of an empty field.

Programming interface: A more flexible and feature-laden application programming interface is planned. The new interface will allow the calling program to store and retrieve fields directly.

Keyboard I/O: The above API will also enable *nccalc* to have better keyboard I/O. The new I/O routines will enable a script to prompt for and read user input.

5. ACKOWLEDGEMENTS

This research was supported by the U.S. Department of Energy (DOE) under Contract DE-AC06-76RLO 1830. Battelle/Marine Sciences Laboratory is part of the Pacific Northwest Laboratory, operated for DOE by Battelle Memorial Institute.

6. REFERENCES

Denbo, D.W. 1993. PLOT PLUS Scientific Graphics System, Version 1.2c, Plot Plus Graphics, P.O. Box 4, Sequim, Washington.

Denbo, D.W., and W.H. Zhu, 1993. EPS Library Users Guide. NOAA/PMEL, Seattle, Washington.

Rew, R.K., and G.P. Davis, 1990. *The Unidata netCDF: Software for scientific data access.* Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology, Feb 5-9, 1990, Anaheim, CA.

DATE FILMED

6 / 1 / 94

END