UNIFIED MODEL DOCUMENTATION PAPER NO C5

CONTROL OF MEANS CALCULATIONS

by

N S GRAHAME

Version No 2, DATED 20/8/92
Revised by R A Stratton

Model Version 2.7

Climate Research
Meteorological Office
London Road
BRACKNELL
Berkshire
RG12 2SY
United Kingdom

This document has not been published. Permission to quote from it must be obtained from the Head of Climate Prediction at the above address.

Modification Record		
Document version	**Author**	**Description....................**

1. Introduction

This document describes the purpose and structure of the system designed for time-meaned calculations in the Unified Model, and is referred to in the following text by its task code, C5. The subroutines written to execute this task are designated as ‘control’ rather than‘meteorological’ and adhere to the standards set down in ‘Unified Model Documentation Paper No.3’. A "running mean" method is used to calculate means in contrast to the method previously employed in the climate model. Instantaneous data obtained at the end of each restart period are accumulated and stored in partial sum dumps which reside on disk. Time-meaned values are subsequently derived from the partial sums via normalisation.

Production of time-meaned dumps on disk will occur only in integrations using a 360-day year i.e. climate simulations, and are of limited use in operational forecast mode. Switches will be set in the top level programs to reflect this.

2. Overview

The "infra-structure" required to produce time-meaned dumps is set up in the preliminary job (see Unified Model Documentation Paper No.1). Mean dump frequencies are specified by the user in the following form

period_1_freq/ period_2_freq/ period_3_freq/ etc.

and are stored in the history file (see task H for more details). Period_1_frequency is expressed as an integral multiple of the restart dump frequency; the latter being the fundamental unit for means calculations. Subsequent means are expressed as multiples of the preceding frequency. C5 will deal with a maximum number of 4 meaning periods provided the above criteria are satisfied. Thus, as an example of the above, 10/3/3/4 would represent 10-day, 30-day, 90-day and 360-day means being requested if the restart dump frequency were daily. (N.B. If less than 4 period means are required, the redundant figures should be set to zero).

A partial sum dump consists of accumulating fields of data (including time-series) stored in a format identical to that used for instantaneous data in a restart dump. Note, however, that the data is not prefixed by the header records found in the latter. Since partial sum dumps are somewhat auxiliary, this information need not be stored but can be accessed when needed once a restart dump has been read into memory. The filenames of partial sum dumps are assigned to a dedicated set of unit numbers and relate to the frequencies mentioned above. The naming convention is as follows:

Period 1 partial sums

runid.apstmp1	atmosphere
runid.opstmp1	ocean

Periods 2-4 partials sums

runidm.spa or
runidm.spb

where

runid	is the five character experiment
m	is an indicator (=a for atmosphere dump; =o for ocean dump)
p	indicates which period ie 2, 3 or 4

Note each of the periods 2-4 has two files a and b. Once set, these filenames remain fixed throughout an integration (unlike restart dumps).

A time-meaned dump has an **identical** format to that of a restart dump and each unique filename is generated within the program using information stored in the history file, the form of which is given in the Unified Model Documentation paper 7, on file naming conventions.

Once the main job is executing, partial sum data will be read in, processed and written to the appropriate dump when necessary. The instantaneous dump will be written out to a Solid State Device (SSD) via a high-speed I/O transfer link whenever means need to evaluated from partial sum data in order to release space in memory, and restored when all means processing has been completed. It suffices to say that C5 will deal with all data transfers when necessary in a way which should be transparent to the user.

The procedure for creating time-meaned dumps is straightforward and general, and can be summarised as follows:
Accumulating data in a ‘period_n’ partial sum dump is normalised at the ‘period_n_frequency’ and written out to a ‘period_n’ mean dump. The normalised data is then used to update a ‘period_n+1’ partial sum dump. When the process is complete, the ‘period_n’ partial sum data is removed from disk to save space.

The removal of time-meaned dumps on disk will be determined by C0 interacting with C5.

In the event of a normal restart, the initialisation job will ensure that the partial sum data is set up correctly to continue the accumulation process. In the event of a failure within the means program itself, a flag will be set in the history file, and any outstanding calculations involving partial sums
and/or means will be completed before the run continues.

According to the above, means production will be invoked once the integration is started, if any mean dump frequency is non-zero. Under certain circumstances, this may not be desirable since the first set of means produced will relate to periods calculated from that start time. For example, a climate run starting from initial data for 15th May, would produce 30-day means which would straddle two months if the frequencies were set to the values quoted earlier. In such cases it would be advantageous to initiate means production once the integration reaches 1st June. In order to accommodate this a reference time for means production can be set in the history file. The means
control program will then calculate the offset between the reference time and model start time. If a ‘period_n’ frequency is a multiple of the offset then that particular ‘period_n’ accumulation/meaning process is activated. The test is then repeated against model time, as the integration continues, until all of the prescribed periods have been activated. Thus in the above example, the reference time could be set to 1st June and the meaning program would remain
dormant until that time. If meaning is required from the outset, the reference time should be set to a default value of zero.

The "running mean" method described here will be accessed by a subroutine call in the main job. This has not been the method employed in climate integrations prior to the Unified Model, where time-meaned dumps are created by running a subtask called from the control program (for details see Met.O.20 Model Documentation). The decision to use the former method is primarily based on
disk space considerations whilst acknowledging that there will be increased overheads in I/O. Partial sums are read in and out of memory from disk on a field by field basis using BUFFER IN/BUFFER OUT commands. Details of the asynchronous nature of data transfers to and from memory are shown in Appendix A.

Since all information required to produce time-means is stored in partial sum dumps it is obviously important that these do not become corrupt at any stage during an integration. For this reason, there are two files set up on disk for each partial sum period; one to read data from and the other to write data to. Thus, a slight overhead in diskspace is incurred whilst partial sums are being
updated. Once all field data for any particular period has been updated and successfully transferred to disk, the file from which data has been read is temporarily deleted. The file will then be reused at the next update to receive data.

3. Programming aspects

Subroutine MEANCTL acts as the interface to the master control routine and has three arguments associated with it:

1) I_AO - Integer - Indicator for atmosphere/ocean
2) MEANLEV - Integer - 1-4 to indicate the mean period
3) ICODE - Integer - Return code
4) CMESSAGE - Character - Error message (if appropriate)

If MEANCTL completes successfully, ICODE will be passed as zero, otherwise a non-zero value will be returned with an appropriate message.

MEANCTL calls a further nine subroutines directly:

TRANSIN - transfers data from a dump on disk (or SSD) to memory, en masse.
TRANSOUT - transfers data from memory to a dump on disk (or SSD), en masse.
DUMPCTL - writes out from memory, header records and data to a dump on disk.
ACUMPS - (i) reads in field data from a partial sum dump on disk to a local array
(ii) arithmetically adds in data stored in main data block
(N.B. This could be instantaneous or time-meaned data - see next section)
(iii) writes out updated field data to the alternative partial sum dump on disk.
MEANPS - (i) reads in data from a 'completed' partial sum dump on disk to a local array.
(ii) normalises data using the specified meaning frequencies
(iii) copies meaned data to main data block.
(iv) sets up appropriate indicators in header record.
PRINTCTL - prints out diagnostics (e.g. zonal means) for data in main data block.
GET_NAME - works out the name for the partial sum file.
SETEXNER - sets P_exner from mean primary fields.
MEANDIAG - extracts mean diagnostics.

Two important points to note:

ACUMPS:
when a period_n partial sum dump contains no data, the read step is skipped and data stored in the main data block is copied into the local array to constitute initial data for the new partial sum.
ACUMPS and MEANPS:
integer fields and logical fields are not accumulated/normalised, they are simply copied from the main data block.

4. Logical Structure

4.1 Normal procedure in MEANCTL
C5 initially determines how many time-meaned dumps need to be produced (if any). The logical sub-process, C51, is then invoked. C51 simply controls the fundamental process of accumulating instantaneous data in a period_1 partial sum dump, and can be thought of as a default process which is executed each time the subroutine MEANCTL is called (i.e. at the restart frequency).

C51 comprises a single step:
Step 1. Call subroutine ACUMPS to create or update period_1 partial sum data on disk (using instantaneous data stored in the main data block)

If C5 indicates that mean dumps need to be produced, the second logical sub-process, C52, is invoked.

C52 comprises of the following steps:
Step 1. Call subroutine TRANSOUT to transfer instantaneous data to SSD.
Step 2. Call subroutine MEANPS to create period_1 time-meaned data in main data block.
Step 3.1 Call subroutine SETEXNER and then MEANDIAG to extract PP fields from mean data.
Step 3.2 Call subroutine PRINCTL to perform zonal mean diagnostics on time-meaned data.
Step 4. If period_2 frequency is non-zero, call subroutine ACUMPS to update or create period_2 partial sum dump using the period_1 time-meaned data (stored in the main data block).
Step 5. Call subroutine DUMPCTL to transfer period_1 time-meaned dump to disk.
Steps 2-5 are repeated for subsequent periods until the required processing is complete.
Step 6. Call subroutine TRANSIN to restore instantaneous data to memory.

MEANCTL concludes by setting a suitable 'completion code' in the history file ie RUN_MEANCTL_RESTART=0 if completed without errors. (in addition to passing ICODE=0 in the argument list)

An example of data stored in partial sum dumps is shown below for a random day (day 176) of a climate integration using four meaning frequencies defined by 10/3/3/4 in the history file.

period_1 partial sum dump - accumulated data for days 171 → 176
period_2 partial sum dump - 10-day mean data over days 151 → 160 +
10-day mean data over days 161 → 170

period_3 partial sum dump - 30-day mean data over days 91 → 120 +
30-day mean data over days 121 → 150
period_4 partial sum dump - 90-day mean data over days 1 → 90

Essentially, each time a period_n mean is produced, a period_n+1 partial sum dump is updated if it exists, or created if it doesn’t (assuming that n+1 is less than or equal to the number of mean periods specified).

4.2 Abnormal end in MEANCTL

In the event of a data transfer failure in partial sum accumulation, the file receiving the data will be deleted and the process of updating will need to start from scratch. Any failure within MEANCTL (whether creating/updating partial sum dumps or producing mean dumps) will generate appropriate flag settings in the history file, including an abnormal completion code which will terminate the model run. On restart, C5 will attempt to complete all outstanding processing for the timestep on which the failure occurred before proceeding. MEANCTL has been written such that the duplication of work after an abnormal failure is kept to a minimum.

APPENDIX A

For a partial sum dump on disk, the data is processed on a field-by-field basis in an asynchronous manner shown below:-

```
   |          FIELD K               FIELD K-1
   |          Read                  Unpack data (if 32-bit).
   |            in                  Process data (acc'm/normalise).
   |          data                  Pack data (if 32-bit).
   |          from                  Write out data to disk/ main data
   |          disk                  block.
   V
 time
```