Documentation of NCAR CCMOB Program Modules

David L. Williamson
Linda M. Bath
Richard K. Sato
Thomas A. Mayer
Michael L. Kuhn

ATMOSPHERIC ANALYSIS AND PREDICTION DIVISION
and
SCIENTIFIC COMPUTING DIVISION
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
BOULDER, COLORADO
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>4</td>
</tr>
<tr>
<td>Model Flow Diagram</td>
<td>6</td>
</tr>
<tr>
<td>List of Print Utility Routines</td>
<td>7</td>
</tr>
<tr>
<td>List of Utility Routines to Reset Variables</td>
<td>8</td>
</tr>
<tr>
<td>List of I/O Utility Routines</td>
<td>9</td>
</tr>
<tr>
<td>List of Dataset Utility Routines</td>
<td>10</td>
</tr>
<tr>
<td>List of General Utility Routines</td>
<td>11</td>
</tr>
<tr>
<td>Alphabetical List of Routines</td>
<td>12</td>
</tr>
<tr>
<td>Alphabetical List of DECKS</td>
<td>15</td>
</tr>
<tr>
<td>Alphabetical List of COMMON DECKS</td>
<td>18</td>
</tr>
<tr>
<td>Description of Routines (Alphabetical)</td>
<td></td>
</tr>
<tr>
<td>Description of COMMON Blocks (Alphabetical)</td>
<td></td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

We would like to thank the European Centre for Medium Range Weather Forecasts and its members for graciously making their adiabatic spectral model code available to us to serve as a basis for the CCM, and for their advice and suggestions related to adding the desired physical parameterizations. We would like to give special thanks to U. Cubasch for his efforts and cooperation in providing NCAR with its first copies of the code and documentation, and for his assistance in the early stages of our efforts. We would also like to thank Jane Raese, Holly Howard, and Mary Niemczewski for typing the manuscript.
1. INTRODUCTION

This report describes the program modules (main program, subroutines, functions, and common blocks) included in the NCAR Community Climate Model CCMOB library, except those involved with the radiation parameterization. These radiation routines are currently being converted to the CCMOB library standards and will be documented in the future. The governing equations and algorithms defining the model are described mathematically in a companion NCAR Technical Note Description of the NCAR Community Climate Model (CCMOB) (Williamson, 1983). The basic code design and structure is described in a second companion report Users' Guide to NCAR CCMOB (Sato et al., 1983). This Users' Guide also includes instructions for modifying and running the model. The present report describing the individual subroutines is designed to be a bridge between these two reports and the actual code. For example, with subroutines dealing with the various physical parameterizations, the primary difference between this report and the Description ... is one of algebra, while the difference between this report and the code itself is primarily in the symbols involved in each.

The subroutines are described in alphabetical order with a standard format. The routine name and calling arguments (if present) are listed on the left of the first line, followed on the next line by the UPDATE DECK which contains the subroutine. This title information is followed by a brief description of the purpose of the subroutine. The arguments are then presented in the order of the calling statement. Each is followed by one or both of the symbols [in] or [out] indicating whether the variable serves as input to the routine, output from the routine, or
both, with a description of the argument in each case. For many rou-
tines the brief initial description of the routine is then augmented by
additional details of the algorithms involved. Numbered comments in
this section refer to the corresponding numbered comments in the code
itself. Finally, the recent programming history of the routine is indi-
cated. This represents the history since the acquisition of the basic
code from ECMWF.

The COMMON blocks are described following the subroutines, again in
alphabetical order. The COMMON statement is given on the first line
followed by the UPDATE COMDECK containing it. The variable list is
given in the order of the COMMON statement, and values of the variables
are provided when they are relatively constant during the course of a
run. The subroutines in which the variables are defined are indicated.

The code described in this report is based on an adiabatic, invis-
cid version of the spectral model developed at the European Centre for
Medium Range Weather Forecasts (ECMWF) and was largely generated by
A.P.M. Baede, M. Jarraud, and U. Cubasch of the ECMWF. Some routines in
that spectral model were adopted from the ECMWF grid point model written
by D.M. Burr ridge, J. Hasseler, and J.K. Gibson. The spectral aspects as
well as the basic structure of this code have been retained. Although
not indicated in the history of the routines, some spectral aspects of
the code could probably be traced back to Copenhagen (Eliasen et al.,
1970; Machenhauer and Rasmussen, 1972; and Machenhauer and Daley, 1972)
and the common ancestry of many spectral models. Physical parameteriza-
tions and numerical approximations matching those in CCMOA were added to
this adiabatic code. The physical parameterizations include the radiation and cloud routines developed at NCAR (Ramanathan et al., 1983) and convective adjustment, stable condensation, vertical diffusion, surface fluxes and surface energy balance prescription developed at GFDL (Smagorinsky, 1963; Manabe et al., 1965; Smagorinsky et al., 1965; and Holloway and Manabe et al., 1971). The parentage of some of the code associated with these prescriptions could probably be traced back to GFDL. The vertical and temporal finite differences match those of the Australian spectral model (Bourke et al., 1977). This combination is designated as the NCAR CCMOB.
REFERENCES


Model flow diagram. Calling sequence is top to bottom and within a horizontal block left to right.
LIST OF PRINT UTILITY ROUTINES
[all write to unit NOUT(=6)]

PAGE advances to new page
BLINES(K) writes K blank lines
MESSAGE(KMESS) writes a 40 character message (KMESS)
HVAR(KNAME,KVALUE) writes an 8 character label (KNAME) and a Hollerith variable (KVALUE)
IVAR(KNAME,KVALUE) writes an 8 character label (KNAME) and an integer variable (KVALUE)
RVAR(KNAME,PVALUE) writes an 8 character label (KNAME) and a real variable (PVALUE)
RARRAY(KNAME,PA,KDIM) writes an 8 character label (KNAME) and KDIM values of a real array (PA)
BLDUMP(INAME,IADR,ILEN) writes an 8 character label (INAME) and ILEN values of the BUF array starting with BUF(IADR+1) along with the current values of NROW and NSTEP
WECOEF(PCOEF,KMBIG,KNBIG) writes spectral coefficients (PCOEF) of an even field for \( n < \text{KNBIG} \) and \( m < \text{KMBIG} \) for vertical levels 1 to NLEV.
WOECEOF(PCOEF,KMBIG,KNBIG) writes spectral coefficients (PCOEF) of an odd field for \( n < \text{KNBIG} \) and \( m < \text{KMBIG} \) for vertical levels 1 to NLEV.
REPTHD(KCLASS,KSUB,KPOINT) writes the three integer arguments to identify the CLASS, SUBROUTINE and POINT.
<table>
<thead>
<tr>
<th>Routine Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESETH(KA,KDIM,KVALUE)</td>
<td>resets KDIM elements of the Hollerith array KA to KVALUE</td>
</tr>
<tr>
<td>RESETI(KA,KDIM,KVALUE)</td>
<td>resets KDIM elements of the integer array KA to KVALUE</td>
</tr>
<tr>
<td>RESETL(KLA,KDIM,KLVAL)</td>
<td>resets KDIM elements of the logical array KLA to KLVAL</td>
</tr>
<tr>
<td>RESETR(PA,KDUM,PVALUE)</td>
<td>resets KDUM elements of the real array PA to PVALUE</td>
</tr>
<tr>
<td>COPYBC(KDISA,KDISB,KNO)</td>
<td>copies KNO words from BUF(KDISA) to BUF(KDISB)</td>
</tr>
</tbody>
</table>
LIST OF I/O UTILITY ROUTINES

- **READBF(KUNIT,PA,KLEN)**: initiates BUFFER IN on KUNIT of PA(1) to PA(KLEN)
- **READER(KDV,PA,KLEN,KREC)**: initiates BUFFER IN on KDV of PA(1) to PA(KREC). If KREC=1, rewind KDV prior to BUFFER.
- **WRITBF(KUNIT,PA,KLEN)**: initiates BUFFER OUT on KUNIT of PA(1) to PA(KLEN)
- **WRITER(KDV,PA,KLEN,KREC)**: initiates BUFFER OUT on KDV of PA(1) to PA(KREC). If KREC=1, rewind KDV prior to BUFFER.
- **CHECKR(KDV)**: checks for successful buffer operation on KDV
- **CHKBF(KUNIT,KERR)**: checks previous buffer operation on KUNIT, returns KERR=0 if successful
<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUNDN(IUNIT)</td>
<td>Returns a CRAY dataset name FTnn for integer unit number IUNIT.</td>
</tr>
<tr>
<td>POSUN(IUM,NR)</td>
<td>Skips (reads past using READBF) NR+1 records on unit IU.</td>
</tr>
<tr>
<td>ATTACH(IF,IU,IA,ICY,IPASWD,IMVN)</td>
<td>Acquires dataset IA and copies to local unit IU.</td>
</tr>
<tr>
<td>REQUEST(IFAIL,IU,IBSZ)</td>
<td>Calls ASSIGN for unit IU.</td>
</tr>
<tr>
<td>RETURN(IF,IU)</td>
<td>Calls RELEASE for unit IU.</td>
</tr>
<tr>
<td>CATALOG(IF,IU,IA,ICY,IID)</td>
<td>Calls SAVE to make dataset on unit IU permanent, PDN=IA, ED=ICY.</td>
</tr>
<tr>
<td>SAVDISP(IUNIT,LTAPE,IPASWD,IMVN)</td>
<td>Saves and disposes copies of dataset on unit IUNIT. PDN,SDN=LTAPE.</td>
</tr>
</tbody>
</table>
LIST OF GENERAL UTILITY ROUTINES

SIGMA(N,A,IA,B,IB) (FUNCTION) Computes sum of N elements of A or scalar dot product of A and B over N elements.

MINV(A,N,D,L,M) Inverts a general NxN matrix A.
## ALPHABETICAL LIST OF ROUTINES

<table>
<thead>
<tr>
<th>ROUTINE</th>
<th>DECK</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM</td>
<td>SPECTR</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td>ACQHIS</td>
</tr>
<tr>
<td></td>
<td>ATTACH</td>
</tr>
<tr>
<td></td>
<td>BASIC</td>
</tr>
<tr>
<td></td>
<td>BLDUMP</td>
</tr>
<tr>
<td></td>
<td>BLINES</td>
</tr>
<tr>
<td></td>
<td>BLKDAT</td>
</tr>
<tr>
<td></td>
<td>BSSLZR</td>
</tr>
<tr>
<td></td>
<td>BUDGET</td>
</tr>
<tr>
<td></td>
<td>CATALOG</td>
</tr>
<tr>
<td></td>
<td>CCLDUMP</td>
</tr>
<tr>
<td></td>
<td>CHECKR</td>
</tr>
<tr>
<td></td>
<td>CHKBF</td>
</tr>
<tr>
<td></td>
<td>CHKSM3</td>
</tr>
<tr>
<td></td>
<td>COMHES</td>
</tr>
<tr>
<td></td>
<td>COMLDR</td>
</tr>
<tr>
<td></td>
<td>CONCAT</td>
</tr>
<tr>
<td></td>
<td>COND</td>
</tr>
<tr>
<td></td>
<td>CONST</td>
</tr>
<tr>
<td></td>
<td>CONVAD</td>
</tr>
<tr>
<td></td>
<td>COPYBC</td>
</tr>
<tr>
<td></td>
<td>COTROL</td>
</tr>
<tr>
<td></td>
<td>CUBSPL</td>
</tr>
<tr>
<td></td>
<td>DADADJ</td>
</tr>
<tr>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td></td>
<td>DATCOM</td>
</tr>
<tr>
<td></td>
<td>DATINI</td>
</tr>
<tr>
<td></td>
<td>DAYTIM</td>
</tr>
<tr>
<td></td>
<td>DOMADJ</td>
</tr>
<tr>
<td></td>
<td>DTRADS</td>
</tr>
<tr>
<td></td>
<td>DYN</td>
</tr>
<tr>
<td></td>
<td>ENDRUN</td>
</tr>
<tr>
<td></td>
<td>ESTABL</td>
</tr>
<tr>
<td></td>
<td>FFLDLST</td>
</tr>
<tr>
<td></td>
<td>GAUAW</td>
</tr>
<tr>
<td></td>
<td>GLBAVG</td>
</tr>
<tr>
<td></td>
<td>GRCALC</td>
</tr>
<tr>
<td></td>
<td>GRMULF</td>
</tr>
<tr>
<td></td>
<td>HORDIF</td>
</tr>
<tr>
<td></td>
<td>HVAR</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>IGTSEQ</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td>INIDAT</td>
</tr>
<tr>
<td></td>
<td>INITAL</td>
</tr>
<tr>
<td>ROUTINE</td>
<td>DECK</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>IVAR</td>
<td>CLASSU.57</td>
</tr>
<tr>
<td>JOBTIM</td>
<td>CLASSU.200</td>
</tr>
<tr>
<td>LABRUN</td>
<td>C1SI.2</td>
</tr>
<tr>
<td>LEG</td>
<td>LEG.2</td>
</tr>
<tr>
<td>LINEMS</td>
<td>LINMS.2</td>
</tr>
<tr>
<td>FUNCTION</td>
<td></td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td></td>
</tr>
<tr>
<td>LUNDN</td>
<td>SAVDISP.61</td>
</tr>
<tr>
<td>MADADJ</td>
<td>MADADJ.1.2</td>
</tr>
<tr>
<td>MASTER</td>
<td>COS0.2</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>CLASSU.2</td>
</tr>
<tr>
<td>MINV</td>
<td>MINV.2</td>
</tr>
<tr>
<td>MODIFY</td>
<td>COS2.2</td>
</tr>
<tr>
<td>FUNCTION</td>
<td></td>
</tr>
<tr>
<td>NEWROW</td>
<td>NEWROW.2</td>
</tr>
<tr>
<td>NORSOU</td>
<td>NORSOU.2</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td></td>
</tr>
<tr>
<td>NUDATE</td>
<td>NUDATE.2</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>OUTPUT.2</td>
</tr>
<tr>
<td>PAGE</td>
<td>CLASSU.23</td>
</tr>
<tr>
<td>PB2GFD</td>
<td>GFDLPB2.2</td>
</tr>
<tr>
<td>PHCS</td>
<td>LGNDRE.90</td>
</tr>
<tr>
<td>PHYS</td>
<td>PHYS.2</td>
</tr>
<tr>
<td>POINTS</td>
<td>POINTS.2</td>
</tr>
<tr>
<td>POSUN</td>
<td>UTIL.100</td>
</tr>
<tr>
<td>PRESET</td>
<td>PRESES.2</td>
</tr>
<tr>
<td>PRICOM</td>
<td>PRICOM.2</td>
</tr>
<tr>
<td>PRNTHD</td>
<td>PRNTHD.2</td>
</tr>
<tr>
<td>PRSCHD</td>
<td>PRSCHD.2</td>
</tr>
<tr>
<td>PURGE</td>
<td>UTIL.77</td>
</tr>
<tr>
<td>QNEGAT</td>
<td>UTIL.137</td>
</tr>
<tr>
<td>QREIG</td>
<td>QREIG.2</td>
</tr>
<tr>
<td>RARRAY</td>
<td>CLASSU.88</td>
</tr>
<tr>
<td>RDHDR</td>
<td>RDHDR.2</td>
</tr>
<tr>
<td>READBF</td>
<td>READBF.2</td>
</tr>
<tr>
<td>READR</td>
<td>READR.2</td>
</tr>
<tr>
<td>REORD</td>
<td>LGNDRE.169</td>
</tr>
<tr>
<td>REP THD</td>
<td>CLASSU.108</td>
</tr>
<tr>
<td>REQUEST</td>
<td>UTIL.37</td>
</tr>
<tr>
<td>RESET H</td>
<td>CLASSU.181</td>
</tr>
<tr>
<td>RESET I</td>
<td>CLASSU.162</td>
</tr>
<tr>
<td>RESET L</td>
<td>CLASSU.216</td>
</tr>
<tr>
<td>RESET R</td>
<td>CLASSU.143</td>
</tr>
<tr>
<td>RESUME</td>
<td>RESUME.2</td>
</tr>
<tr>
<td>RETURN</td>
<td>UTIL.65</td>
</tr>
<tr>
<td>RVAR</td>
<td>RVAR.2</td>
</tr>
<tr>
<td>SAVDISP</td>
<td>SAVDISP.2</td>
</tr>
<tr>
<td>SCAN 1</td>
<td>SCAN1.2</td>
</tr>
<tr>
<td>SCAN 2</td>
<td>SCAN2.2</td>
</tr>
<tr>
<td>SDS</td>
<td>SDS.2</td>
</tr>
<tr>
<td>SETTAU</td>
<td>SETTAU.2</td>
</tr>
<tr>
<td>FUNCTION</td>
<td></td>
</tr>
<tr>
<td>SIGMA</td>
<td>UTIL.117</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td></td>
</tr>
<tr>
<td>SPLITF</td>
<td>SPLITF.2</td>
</tr>
<tr>
<td>ROUTINE</td>
<td>DECK</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>STARTN</td>
<td>STRTN.2</td>
</tr>
<tr>
<td>STATS</td>
<td>STATIS.2</td>
</tr>
<tr>
<td>STEPON</td>
<td>STPN.2</td>
</tr>
<tr>
<td>TCALC</td>
<td>TCALC.2</td>
</tr>
<tr>
<td>TESEND</td>
<td>DUMMIES.2</td>
</tr>
<tr>
<td>TIMCPU</td>
<td>DUMMIES.82</td>
</tr>
<tr>
<td>TIMESM</td>
<td>TIMESM.2</td>
</tr>
<tr>
<td>TRUNC</td>
<td>DATIN.435</td>
</tr>
<tr>
<td>TSTEP</td>
<td>TSTEP.2</td>
</tr>
<tr>
<td>VIRDIF</td>
<td>VIRDIF.2</td>
</tr>
<tr>
<td>WECOEF</td>
<td>WECOEF.2</td>
</tr>
<tr>
<td>WOCOEFS</td>
<td>WOCOEFS.2</td>
</tr>
<tr>
<td>WRITBF</td>
<td>WRITBF.2</td>
</tr>
<tr>
<td>WRITER</td>
<td>WRITER.2</td>
</tr>
<tr>
<td>WRTHDR</td>
<td>WRTHDR.2</td>
</tr>
<tr>
<td>WSHIST</td>
<td>WSHIST.2</td>
</tr>
</tbody>
</table>
### ALPHABETICAL LIST OF DECKS

<table>
<thead>
<tr>
<th>DECK</th>
<th>ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN.2</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>ACQHIS</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>BUDGET</td>
<td>ACQHIS</td>
</tr>
<tr>
<td>COS0.2</td>
<td>BUDGET</td>
</tr>
<tr>
<td>COS1.2</td>
<td>MASTER</td>
</tr>
<tr>
<td>COS2.2</td>
<td>BASIC</td>
</tr>
<tr>
<td>COS3.2</td>
<td>MODIFY</td>
</tr>
<tr>
<td>CLS1.2</td>
<td>CONTROL</td>
</tr>
<tr>
<td>CLCLOUD.2</td>
<td>MAILRUN</td>
</tr>
<tr>
<td>CHECKR.2</td>
<td>CHECKR</td>
</tr>
<tr>
<td>CHKBF.2</td>
<td>CHKBF</td>
</tr>
<tr>
<td>CHKSM3.2</td>
<td>CHKSM3</td>
</tr>
<tr>
<td>CLASS.2</td>
<td>BLKDAT</td>
</tr>
<tr>
<td>CLASSU.2</td>
<td>MESSAGE</td>
</tr>
<tr>
<td>.23</td>
<td>PAGE</td>
</tr>
<tr>
<td>.40</td>
<td>BLINES</td>
</tr>
<tr>
<td>.57</td>
<td>IVAR</td>
</tr>
<tr>
<td>.73</td>
<td>HVAR</td>
</tr>
<tr>
<td>.88</td>
<td>RARRAY</td>
</tr>
<tr>
<td>.108</td>
<td>REPTHD</td>
</tr>
<tr>
<td>.124</td>
<td>DAYTIM</td>
</tr>
<tr>
<td>.143</td>
<td>RESETR</td>
</tr>
<tr>
<td>.162</td>
<td>RESETI</td>
</tr>
<tr>
<td>.181</td>
<td>RESETH</td>
</tr>
<tr>
<td>.200</td>
<td>JOBTIM</td>
</tr>
<tr>
<td>.216</td>
<td>RESETL</td>
</tr>
<tr>
<td>CONCAT.2</td>
<td>CONCAT</td>
</tr>
<tr>
<td>CONDENS.2</td>
<td>COND</td>
</tr>
<tr>
<td>CONSTS.2</td>
<td>CONST</td>
</tr>
<tr>
<td>COPYBC.2</td>
<td>COPYBC</td>
</tr>
<tr>
<td>CONVAD.2</td>
<td>CONVAD</td>
</tr>
<tr>
<td>DATAS.2</td>
<td>DATA</td>
</tr>
<tr>
<td>DATCOM.2</td>
<td>DATCOM</td>
</tr>
<tr>
<td>DATIN.2</td>
<td>DATINI</td>
</tr>
<tr>
<td>.435</td>
<td>TRUNC</td>
</tr>
<tr>
<td>DRAADJ.2</td>
<td>DRAADJ</td>
</tr>
<tr>
<td>DTRADS.2</td>
<td>DTRADS</td>
</tr>
<tr>
<td>DUMMIES.2</td>
<td>TESEND</td>
</tr>
<tr>
<td>.19</td>
<td>ENDRUN</td>
</tr>
<tr>
<td>.82</td>
<td>TIMCPU</td>
</tr>
<tr>
<td>DYNAM.2</td>
<td>DYN</td>
</tr>
<tr>
<td>ESATUR.2</td>
<td>ESTABL</td>
</tr>
<tr>
<td>FDLST.2</td>
<td>FDLST</td>
</tr>
<tr>
<td>GFDPB2.2</td>
<td>PB2GFD</td>
</tr>
<tr>
<td>GLBAVG.2</td>
<td>GLBAVG</td>
</tr>
<tr>
<td>DECK</td>
<td>ROUTINE</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
</tr>
<tr>
<td>GRCALC.2</td>
<td>GRCALC</td>
</tr>
<tr>
<td>GRMULT.2</td>
<td>GRMULT</td>
</tr>
<tr>
<td>HORDIF.2</td>
<td>HORDIF</td>
</tr>
<tr>
<td>IGSEQ.2</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>INIDAT.2</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>INITIAL.2</td>
<td>INITIAL</td>
</tr>
<tr>
<td>LEG.2</td>
<td>LEG</td>
</tr>
<tr>
<td>LGNDRE.2</td>
<td>GAUAW</td>
</tr>
<tr>
<td>.59</td>
<td>BSSLZR</td>
</tr>
<tr>
<td>.90</td>
<td>PHCS</td>
</tr>
<tr>
<td>.169</td>
<td>REORD</td>
</tr>
<tr>
<td>.198</td>
<td>BLDUMP</td>
</tr>
<tr>
<td>LINMS.2</td>
<td>LINEMS</td>
</tr>
<tr>
<td>MADADJ1.2</td>
<td>MADADJ</td>
</tr>
<tr>
<td>MADADJ2.2</td>
<td>DOMADJ</td>
</tr>
<tr>
<td>MINV.2</td>
<td>MINV</td>
</tr>
<tr>
<td>NEWROW.2</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>NORSOU.2</td>
<td>NORSOU</td>
</tr>
<tr>
<td>NUDATE.2</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>OUTPUT.2</td>
<td>OUTPUT</td>
</tr>
<tr>
<td>PHYS.2</td>
<td>PHYS</td>
</tr>
<tr>
<td>POINTS.2</td>
<td>POINTS</td>
</tr>
<tr>
<td>PRESESP.2</td>
<td>PRESET</td>
</tr>
<tr>
<td>PRICOM.2</td>
<td>PRICOM</td>
</tr>
<tr>
<td>PRNTHD.2</td>
<td>PRNTHD</td>
</tr>
<tr>
<td>PRSCHD.2</td>
<td>PRSCHD</td>
</tr>
<tr>
<td>QREIG.2</td>
<td>QREIG</td>
</tr>
<tr>
<td>.18</td>
<td>COMHES</td>
</tr>
<tr>
<td>.64</td>
<td>COML</td>
</tr>
<tr>
<td>READBF.2</td>
<td>READBF</td>
</tr>
<tr>
<td>READR.2</td>
<td>READR</td>
</tr>
<tr>
<td>RESUME.2</td>
<td>RESUME</td>
</tr>
<tr>
<td>RDHDR.2</td>
<td>RDHDR</td>
</tr>
<tr>
<td>RVAR.2</td>
<td>RVAR</td>
</tr>
<tr>
<td>SAVDISP.2</td>
<td>SAVDISP</td>
</tr>
<tr>
<td>.61</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>SCAN1.2</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>SCAN2.2</td>
<td>SCAN1</td>
</tr>
<tr>
<td>SDS.2</td>
<td>SDS</td>
</tr>
<tr>
<td>SETTAU.2</td>
<td>SETTAU</td>
</tr>
<tr>
<td>SPLITP.2</td>
<td>SPLITP</td>
</tr>
<tr>
<td>STATIS.2</td>
<td>STATS</td>
</tr>
<tr>
<td>STPN.2</td>
<td>STEPON</td>
</tr>
<tr>
<td>STRTN.2</td>
<td>STARTN</td>
</tr>
<tr>
<td>TCALC.2</td>
<td>TCALC</td>
</tr>
<tr>
<td>TIMESM.2</td>
<td>TIMESM</td>
</tr>
<tr>
<td>TSTEP.2</td>
<td>TSTEP</td>
</tr>
<tr>
<td>UTIL.2</td>
<td>ATTACH</td>
</tr>
<tr>
<td>DECK</td>
<td>ROUTINE</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>UTIL.37</td>
<td>REQUEST</td>
</tr>
<tr>
<td>.53</td>
<td>CATALOG</td>
</tr>
<tr>
<td>.65</td>
<td>RETURN</td>
</tr>
<tr>
<td>.77</td>
<td>PURGE</td>
</tr>
<tr>
<td>.100</td>
<td>POSUN</td>
</tr>
<tr>
<td>.117</td>
<td>FUNCTION SIGMA</td>
</tr>
<tr>
<td>.137</td>
<td>SUBROUTINE QNEGAT</td>
</tr>
<tr>
<td>VIRDIF.2</td>
<td>VIRDIF</td>
</tr>
<tr>
<td>WECOEF.2</td>
<td>WECOEF</td>
</tr>
<tr>
<td>WOCOEF.2</td>
<td>WOCOEF</td>
</tr>
<tr>
<td>WRITBF.2</td>
<td>WRITBF</td>
</tr>
<tr>
<td>WRITER.2</td>
<td>WRITER</td>
</tr>
<tr>
<td>WRTHDR.2</td>
<td>WRTHDR</td>
</tr>
<tr>
<td>WSHIST.2</td>
<td>WSHIST</td>
</tr>
<tr>
<td>COMMON BLOCKS</td>
<td>UPDATE COMDECK</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>blank COMMON</td>
<td>BLANCOM1.2</td>
</tr>
<tr>
<td>COMMON</td>
<td>BLANCOM2.2</td>
</tr>
<tr>
<td>COMMON</td>
<td>BLANCOM4.2</td>
</tr>
<tr>
<td>COMMON/COMBAS/</td>
<td>COMBAS.2</td>
</tr>
<tr>
<td>/COMBUD/</td>
<td>COMBUD.2</td>
</tr>
<tr>
<td>/COMCOL/</td>
<td>COMCOL.2</td>
</tr>
<tr>
<td>/COMDAD/</td>
<td>COMDAD.2</td>
</tr>
<tr>
<td>/COMDDP/</td>
<td>COMDDP.2</td>
</tr>
<tr>
<td>/COMEST/</td>
<td>COMEST.2</td>
</tr>
<tr>
<td>/COMEXP/</td>
<td>COMEXP.2</td>
</tr>
<tr>
<td>/COMFFT/</td>
<td>COMFFT.2</td>
</tr>
<tr>
<td>/COMGRD/</td>
<td>COMGRD.2</td>
</tr>
<tr>
<td>/COMHD1/</td>
<td>COMHD1.2</td>
</tr>
<tr>
<td>/COMHKP/</td>
<td>COMHKP.2</td>
</tr>
<tr>
<td>/COMIMP/</td>
<td>COMIMP.2</td>
</tr>
<tr>
<td>/COMIOC/</td>
<td>COMIOC.2</td>
</tr>
<tr>
<td>/COMISP/</td>
<td>COMISP.2</td>
</tr>
<tr>
<td>/COMLEG/</td>
<td>COMLEG.2</td>
</tr>
<tr>
<td>/COMMAP/</td>
<td>COMMAP.2</td>
</tr>
<tr>
<td>/COMPBL/</td>
<td>COMPBL.2</td>
</tr>
<tr>
<td>/COMPHO/</td>
<td>COMPHO.2</td>
</tr>
<tr>
<td>/COMPH1/</td>
<td>COMPH1.2</td>
</tr>
<tr>
<td>/COMPRTS/</td>
<td>COMPRTS.2</td>
</tr>
<tr>
<td>/COMSDS/</td>
<td>COMSDS.2</td>
</tr>
<tr>
<td>/COMSPE/</td>
<td>COMSPE.2</td>
</tr>
<tr>
<td>/COMSTA/</td>
<td>COMSTA.2</td>
</tr>
<tr>
<td>/COMTAP/</td>
<td>COMTAP.2</td>
</tr>
<tr>
<td>/COMTRU/</td>
<td>COMTRU.2</td>
</tr>
<tr>
<td>/COMVDF/</td>
<td>COMVDF.2</td>
</tr>
<tr>
<td>/COMZERO/</td>
<td>COMZERO.2</td>
</tr>
</tbody>
</table>
SUBROUTINE ACQHIS
*DECK ACQHIS.2

Acquires and positions history tape for restart or regeneration run.

ALGORITHM:

0.5 For regeneration run, calls ATTACH to acquire history tape and copy it to a local unit.

0.6 Positions to save information at end of history tape, calls SPLITF to copy this information to the appropriate files. Calls SDS to read restart values into COMMON. Rewinds NDATA and returns.

1.05 For a normal restart, calls ATTACH to acquire and copy history tape.

1.1 Positions to beginning of history file before current history file.

HISTORY:

9 OCT 81, R. Sato, NCAR.
Modified, 13 JAN 83, T. Mayer, NCAR.
SUBROUTINE ATTACH(IF, IU, IA, ICY, IPASWD, IMVN)
*DECK UTIL.2

Acquires dataset IA and copies it to local unit IU.

ARGUMENTS:

IF : Error flag (not currently used).
IU : [in] Local unit onto which the acquired dataset is copied.
IA : [in] Dataset to be acquired.
IPASWD: [in] Write password (Default is WRITE)
IMVN : [in] Dedicated MSD reel (Default is blank)

ALGORITHM:

If error returned from ACQUIRE then error message printed and program halted.

HISTORY:

8 SEP 81, R.K. Sato, NCAR.
SUBROUTINE BASIC
*DECK COS1.2

Initializes /COMBAS/ and /COMDDP/.
Prints "NCAR CCMOB" on unit NOUT (=6).
Calls subroutine MODIFY.

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE BLDUMP(INAME, IADR, ILEN)
*DECK LGNDRE.198

Prints on unit NOUT the input arguments IADR and ILEN along with the current values of NROW and NSTEP. Calls RARRAY to print ILEN values of B array starting at the (IADR+1)st value.

ARGUMENTS:

INAME: [in] Hollerith variable to be printed
IADR : [in] offset with B array to values to be printed
ILEN : [in] number of values to be printed

ALGORITHM:

Calls RARRAY

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE BLINES(K)
*DECK CLASSU.40

Writes K blank lines on unit NOUT (=6).

ARGUMENT:

K: [in] number of blank lines to be written

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE BLKDAT
*DECK CLASS.2

Establishes, through DATA statements, values for following labelled common blocks:

COMMON/TABLES/
COMMON/PRESF/
COMMON/LATDAT/
COMMON/SSALB/
COMMON/ADDRAD/

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE BSSLZR(BES,N)
*DECK LGNDRE.59

Returns N zeros or (for N > 50) approximate zeros of the Bessel function $J_0$.

ARGUMENTS:

BES(n): [out] array of zeros from smallest to largest
N: [in] number of zeros desired

ALGORITHM:

First 50 zeros are specified by a data statement, remaining by extrapolation.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE BUDGET(PTIME,KCALL)

*DECK BUDGET.2

Monitors the global energy and moisture budgets. Computes selected global average quantities. Prints these global averages and various conversion terms that have been accumulated in the physical parameterization subroutines during the course of the integration. All quantities are in kJoules/m².

ARGUMENTS:

PTIME: [in] current time (in hours) for print out

KCALL: [in] = 1 calculate initial values of global integrals and zero out accumulation variables

≠ 1 calculate current values of global integrals. Print global integrals, conversion terms, and deficit values.

GLOBAL AVERAGE QUANTITIES:

Initial values are denoted by the same symbol with a suffix 0.

\[
\text{HAIR} = \frac{C_p 10^{-3}}{21g} K \sum_{k=1}^{J} \sum_{j=1}^{I} \sum_{i=1}^{I} \left( T_{n-1} - p_{n-1} - w_j \Delta \sigma_k \right)
\]

\[
\text{HWAVA} = \frac{L 10^{-3}}{21g} K \sum_{k=1}^{J} \sum_{j=1}^{I} \sum_{i=1}^{I} \left( q_{n-1} - p_{n-1} - w_j \Delta \sigma_k \right)
\]

\[
\text{HST} = \text{(not used)}
\]

\[
\text{KIN} = \frac{10^{-3}}{41g} K \sum_{k=1}^{J} \sum_{j=1}^{I} \sum_{i=1}^{I} \left( (u_{n-1}^2 + v_{n-1}^2) \right) p_{n-1} w_j \Delta \sigma_k
\]

\[
\text{KINZ} = \frac{10^{-3}}{41g} K \sum_{k=1}^{J} \sum_{j=1}^{I} \sum_{i=1}^{I} \left( \frac{u_{n-1}}{I} \right)^2 (I * 10^5) w_j \Delta \sigma_k
\]
WSOIL = (not used)
SNOW = (not used)
HSOIL = (not used)

CONVERSION TERMS:
see /COMBUD/ for description of conversion terms and for list of subroutines where they are accumulated

DEFICITS:

DHDEF = (HAIR-HAIRO) - (DHCVHP+DHRAD+DHCDST+DHCDCU+DHFLUX)
DKDEF = (KIN-KINO) - (DKCVKP+DKDISS+DKDIFF)
DWDEF = (HWAVA-HWAVAO) - (DWCDST+DWCDCU+DWFLUX) - (DWVCU+DWMIHC)

HISTORY:
1 MAR 81, ECMWF.
SUBROUTINE CATALOG(IF,IU,IA,ICY,IID)
*DECK UTIL.53

Saves data set on unit IU as permanent data set with name IA.

ARGUMENTS:

IF : [out] error flag (not used)
IU : [in] unit number of local data set to be saved
IA : [in] permanent data set name
ICY: [in] permanent data set edition number
IID: not used

HISTORY:

3 SEP 81, R.K. Sato, NCAR.
SUBROUTINE CHECKR(KDV)
*DECK CHECKR.2

    Checks for successful buffer operation on unit KDV.
    Aborts run for unsuccessful operation.

ARGUMENT:

KDV: [in] unit number to be checked

HISTORY:

  8 MAY 78, A.P.M. Baede, ECMWF.
SUBROUTINE CHKBF(KUNIT,KERR)
*DECK CHKBF.2

Checks previous BUFFER operation on unit KUNIT and returns status.

ARGUMENTS:

KUNIT: [in] unit number to be checked
KERR: [out] status of previous BUFFER operation
    = 0 successful
    = 1 EOF encountered
    = 2 parity error

HISTORY:

24 FEB 77, Jan Haseler, ECMWF.
SUBROUTINE CHKSM3(K1,K2,KLEN1,KLEN2,KSUM)
*DECK CHKSM3.2

Calculates a check sum from two parts of the BUF array for history records.

ARGUMENTS:

K1 : [in] offset in BUF array for first part of sum
K2 : [in] offset in BUF array for second part of sum
KLEN1: [in] length of first part of sum
KLEN2: [in] length of second part of sum
KSUM : [out] value of check word

ALGORITHM:

\[ KSUM = \sum_{k=1}^{KLEN1} BUF(K1+k) + \sum_{k=1}^{KLEN2} BUF(K2+k) \]

HISTORY:

1 SEP 79, U. Cubasch, ECMWF.
SUBROUTINE COMHES(A,N,K,L,INT)
*DECK QREIG.18

Reduces a general complex matrix to upper Hessenburg form using
stabilized elementary similarity transformations.

ARGUMENTS:

A: [in] = the complex matrix to be reduced to upper Hessenburg
form
    [out] the upper Hessenburg matrix as well as the multipliers
    used in the reduction in the lower triangular elements

N: [in] = order of matrix

K: [in] = lower index for balanced matrix

L: [in] = upper index for balanced matrix

INT: [out] vector indicating the rows and columns interchanged in
the reduction

ALGORITHMS:

For details see, "Path chart and Documentation for the EISPACK
Package of Matrix Eigensystem Routines" by B.S. Garbow and
J.J. Dongarra, Argonne National Laboratory.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE COMLR(H,W,NC,NFT)
*DECK QREIG.64

Calculates the eigenvalues of a complex upper Hessenburg matrix using the modified LR method.

ARGUMENTS:

H : [in] complex upper Hessenburg matrix
W : [out] the complex eigenvalues of H
NC : [in] order and dimension of the matrix
NFT: [in] = unit number for write messages

ALGORITHMS:

For details see, "Path chart and Documentation for the EISPACK Package of Matrix Eigensystem Routines" by B.S. Garbow and J.J. Dongarra, Argonne National Laboratory.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE CONCAT
*DECK CONCAT.2

Concatenates the three local datasets containing restart data onto
the restart dataset, VSN = NSVSN2, unit = NSRE. The three files
are written in the order NRA1, NRB1 and NSDS.

HISTORY:

10 DEC 82, L. Bath, NCAR.
SUBROUTINE COND
*DECK CONDENS.2

Simultaneously adjusts temperature and mixing ratio of a point that is initially supersaturated so that it is just saturated.

ALGORITHM:

1. Initialize displacements of variables
   set local offsets to BUF array to adjust temperature and mixing ratio at time n

2. Condensation-calculations
   RAIN = 0

2.2 Calculate condensation-rate and new T and q values

   The following calculation through PREC is iterated twice

   \[ q^*_k = e^{\frac{\alpha_k e_{sk}}{(\alpha_k P_s - q^*_k e_{sk})}} \]

   where \( \alpha_k = .8 \) and the saturation vapor pressure \( e_{sk} \) is determined by a call to ESTABL

   \[ q_{CS_k} = \max\left\{ 0, q^*_k \mid \frac{(q/q^*_k - 1)}{[1+ \frac{L}{C_p} (\frac{dq^*_k}{dT})_k]} \right\} \]

   where

   \[ (\frac{dq^*_k}{dT})_k = \frac{\alpha_k P_s}{(\alpha_k P_s - \alpha_k e^*_s)} \frac{L}{R_{H2O} T_k 2} q^*_k \]

   \[ q_k = q_k - q_{CS_k} \]

   \[ T_k = T_k + q_{CS_k} \cdot \frac{L}{C_p} \]

   \[ QC_k = QC_k + q_{CS_k} \cdot \frac{p_s A_{sk}}{g} / 2 \Delta t \]
2.3 Calculate precipitation

\[ PRL = PRL + \sum_{k=1}^{K} \frac{P_s \Delta q_k}{g_{\text{H}_2\text{O}}} \]

HISTORY:

1 MAR 81, ECMWF.
10 Jan. 82, Modified by D. Williamson, NCAR.
SUBROUTINE CONST(KSTEP)
*DECK CONSTS.2

Initialize constants for /COMPHO/, /COMPH1/, /COMDAD/, /COMPBL/, /COMBUD/, /COMZER/, /COMEST/.

ARGUMENT:

KSTEP: [in] time step index
  if KSTEP = 0, 2Δt is divided by 2 in all constants to
  invoke a forward time step initially
  if KSTEP ≠ 0, 2Δt is used for centered time differences

ALGORITHM:

0.5 Initialize constants in /COMEST/
  see /COMEST/ description for definition of variables
1.0 Initialize constants in /COMPHO/
  see /COMPHO/ description for definition of variables
2.0 Initialize constants in /COMPH1/
  see /COMPH1/ description for definition of variables
4.0 Initialize constants in /COMDAD/
  see /COMDAD/ description for definition of variables
5.0 Initialize constants in /COMPBL/
  see /COMPBL/ description for definition of variables
6.0 Initialize constants in /COMBUD/
  see /COMBUD/ description for definition of variables
8.0 Initialize constants in /COMZER/
  see /COMZER/ description for definition of variables

HISTORY:

1 May 81, ECMWF.
SUBROUTINE CONVAD
*DECK CONVAD.2

Calls subroutines to perform convective adjustments, calculate stable condensation and eliminate negative mixing ratio values.

ALGORITHM:

1. Compute temperature
   \[ T_k^n = T_k^n + \tau_o^n \quad ; \quad k = 1, K \]

2.1 Do convective adjustment
   CALL DADADJ
   CALL MADADJ

2.6 Do large-scale condensation
   CALL COND

2.8 Adjust negative mixing ratios
   CALL QNEGAT(NQ)

3. Compute temperature derivation
   \[ T_k^{\prime n} = T_k^n - \tau_o^n \quad ; \quad k = 1, K \]

HISTORY:

1 MAR 81, ECMWF
SUBROUTINE COPYBC(KDISA,KDISB,KNO)
*DECK COPYBC.2

Copies data from one part of BLANK COMMON into another.

ARGUMENTS:

KDISA: [in] index of first word in BUF array of blank COMMON to be copied
KDISB: [in] index of first word in BUF array of blank COMMON to which the copy is to be written
KNO: [in] number of words copied

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE COTROL
*DECK  COS3.2

Controls basic model flow.
Calls appropriate initialization routines for initial run or for restart runs.
Calls subroutine STEPON to proceed with integration.

ALGORITHM:
1.3 Set Default values
   CALL PRESET
1.4 Read input control data from cards
   CALL DATA
1.6 Initialize model if not restart
   CALL INITAL
1.7 Reestablish model conditions if restart
   CALL RESUME
2.1 Proceed with integration in time
   CALL STEPON
3.0 Terminate run
   CALL OUTPUT
   CALL TSEND
   CALL ENDRUN

HISTORY:
19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE DADADJ
*DECK DRADADJ.2

Checks for instability with respect to dry adiabatic lapse rate and performs dry adiabatic adjustment if necessary.

ALGORITHM:

1. Displacement of variables
   
   set local offsets to BUF array to adjust temperature at time level \( n \)
   \[ \delta = .01 \]

2. Find grid points with unstable stratification

   \[
   ZDAD = \begin{cases}
   0 & \text{if } T_{k+1} - T_k < C_1(k+1)(T_{k+1} + T_k) + \delta \\
   1 & \text{otherwise}
   \end{cases} \\
   \text{for all } k=1,K-1
   \]

   where \( C_1(k+1) = \frac{k\Delta \alpha_{k+1/2}}{2\alpha_{k+1/2}} \)

3. Do dry adiabatic adjustment

   if \( ZDAD = 1 \), the column is stabilized from top to bottom so that for every layer

   \[ T_{k+1} - T_k < \gamma^* + \delta \]

   where \( \gamma^* = C_1(k+1)(T_k + T_{k+1}) \)

   if \( (T_{k+1} - T_k) \geq \gamma^* + \delta \)

   \[ T_{k+1} = C_3(k)T_k + C_4(k+1)T_{k+1} \]

   \[ T_k = C_2(k+1)T_{k+1} \]
where \( C_2(k) = \frac{1 - C_1(k)}{1 + C_1(k)} \)

\[
C_3(k) = \frac{\Delta \sigma_k}{\Delta \sigma_{k+1} + \Delta \sigma_k C_2(k+1)}
\]

\[
C_4(k+1) = \frac{\Delta \sigma_{k+1}}{\Delta \sigma_{k+1} + \Delta \sigma_k C_2(k+1)}
\]

The column is passed through a maximum of 15 times for each value of \( \delta \). If convergence is not reached by the 15th pass, the convergence criterion \( \delta \) is doubled, a message is printed to NOUT(=6), and the process is repeated. If \( \delta \) becomes > .1 and the column still has at least one unstable layer, the model stops.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE DATA
*DECK DATAS.2

Reads formatted input identification from input unit NREAD; reads control parameters from card image input via NAMELIST $NEWRUN. Sets up control arrays to control model history tape and diagnostic output.

ALGORITHM:

2.2 Reads four card images which are used to describe run

3.1 Reads NAMELIST NEWRUN whose variables are initially set in many different routines

3.15 Reads case title and name for history tape header

3.2 Basic run parameters

NESTEP specifies the final time step for the run. If NESTEP < 0, then NESTEP has been expressed in days and must be converted to number of time steps, i.e.,

\[ \text{NESTEP} = \frac{-\text{NESTEP} \times 86400}{\text{DTIME}} \]

where DTIME is the time step (secs). NSTOP, the variable tested for end-of-run conditions is set = NESTEP+1.

3.3 Write-up times

NNUMWT specifies the frequency (in time steps) of history file write-up. If NNUMWT < 0, then NNUMWT is a frequency given in hours and the frequency in time steps at which history files are to be written must be determined.

3.4 Printing and diagnostics

NFSTAT specifies the frequency of diagnostic printouts. If NFSTAT < 0, then NFSTAT is expressed in hours and is converted to number of time steps.

\[ \text{NFSTAT} = \frac{-(\text{NFSTAT} \times 3600)}{\text{DTIME}} \]

(default set in PRESET)
3.7 Physical values

Determine NTAPES, the number of history tape labels input.
Print $NEWRUN parameters.

HISTORY:

7 JUL 80, U. Cubasch, ECMWF.
SUBROUTINE DATCOM
*DECK DATCOM.2

Initializes /COMBAS/ and /COMMAP/. Fills: (1) sigma level dependent arrays, (2) arrays dependent upon NMAX, (3) arrays dependent upon MMAX, and (4) arrays dependent upon Gaussian latitude. The integration matrices of the hydrostatic equation (G) and conversion term (A) are computed.

ALGORITHM:


HISTORY:

8 DEC 78, M. Jarraud, ECMWF.
13 JAN 83, T. Mayer, NCAR.
SUBROUTINE DATINI
*DECK DATIN.2

Initializes variables (not all) in /COMHKP/, /COMBUD/, /COMOUT/, /COMIMP/, /COMFFT/, and /COMIOC/.
The following is done for K=0 and again for K=1 if a restart run.
Calls CONST(K), which initializes /COMPHO/, /COMPH1/, /COMDAD/, /COMPBL/, /COMBUD/, and /COMZER/.
Calls TRUNC which sets /COMTRU/.

ALGORITHM:

1.5 Set internal pointers stored in common /COMISP/

2.2 Fill physical common blocks
   CALL CONST(k) for k=0, or 1

2.25 Fill rest of common blocks /COMBUD/ and /COMOUT/

2.5 Compute truncation parameters
   CALL TRUNC

3.0 Compute semi-implicit time step constants (/COMIMP/)

3.1 Initialize reference temperatures
   TO(J) = 300     J=1,NLEV

3.2 Calculation of the elements of matrix TAU in thermodynamic equation

   TAU(K+J) = \( \frac{1}{2\Delta\sigma_k} \) 
   \[ \{(T_{o_{k+1}} - T_{o_k})[\sigma_{k+1/2} \Delta\sigma_j - \Delta\sigma_j \begin{pmatrix} 0 & (k < j) \\ 0 & (k > j) \end{pmatrix}]
   + \frac{1}{2\Delta\sigma_k} \{(T_{o_k} - T_{o_{k-1}})[\sigma_{k-1/2} \Delta\sigma_j - \Delta\sigma_j \begin{pmatrix} 0 & (k-1 < j) \\ 0 & (k-1 \geq j) \end{pmatrix}]
   + \kappa T_{o_k} A_{G_{kj}} \]
3.3 Compute sum $T_0(J)\Delta\sigma(J)$ and $[T_0(J)^2]\Delta\sigma(J)$

\[
STODS = \sum_{j=1}^{K} T_0 J \Delta\sigma_j
\]

\[
STO2DS = \sum_{j=1}^{K} T_0^2 \Delta\sigma_j
\]

4.0 Compute and invert matrix for solution of helmholtz equation

4.1 Compute matrix $B$, and multiply matrix $G$ by $R$

$B$ is the vertical structure matrix which gives the gravity wave speeds associated with the mean atmosphere used for the semi-implicit time differences.

$G$ is the integration matrix for the hydrostatic equation.

\[
ZB(J) = R \ast (G \ast TAU + T_0 \ast PI)
\]

\[
G(J) = G(J) \ast R
\]

4.3 Compute and invert matrix $A$

\[
A(N) = [RCN \ast I + B \ast DELT \ast 2]
\]

Call to MINV for matrix inversion.

The inverse, stored in $BM1$, is used to solve the semi-implicit equations.

5.0 Compute constants related to Legendre transform

5.2 Compute and reorder $ALP$ and $DALP$ arrays

CALL PHCS and REORD

5.3 Multiply $ALP$ and $DALP$ by $\sqrt{2}$ for proper normalization

($DALP = -1.\ast DALP$ to correct for $SIGN$ in Copenhagen definition.)

6.0 Compute constants for FFT

Call FAX and FFTRIG to set up trigonometric tables

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE DAYTIM
*DECK CLASSU.124

Uses CRAY system utilities DATE and CLOCK to fetch the current date and time of the run and prints values on unit NOUT (=6).

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE DOMADJ(KTOP, KBOT, KLEVTO, KLEVBO, KT, KQ, KQSAT, KL, PEPS)
*DECK MADADJ2.2

Performs moist adiabatic adjustment of $T_k$ and $q_k$ for $KTOP \leq k \leq KBOT$.

ARGUMENTS:

KTOP : [in] index of top of unstable layers
KBOT : [in] index of bottom of unstable layers
KLEVTO: [in] offset for B array for level KTOP
KLEVBO: [in] offset for B array for level KBOT
KT : [in] offset for B array for $T$
KQ : [in] offset for B array for $q$
KQSAT : [in] offset for B array for $q_{s*}$
KL : [in] longitudinal grid point index
PEPS : [in] convergence criteria $\delta$

ALGORITHM:

$\gamma_{W*}$ for the unstable layers initially passed to DOMADJ in COMMON/COMCOL/

1.1 Calculate $T$ and $q$ at level $k=KBOT$

$$\frac{dq_{s*}}{dT}_k = \frac{q_k p_s}{(\sigma_k p_s - \alpha_k e_s)} \frac{L}{R_{H2O} T_k^2} q_{s*}$$

where, as before, $e_s$ is given by a call to ESTABL

$$S_0_k = \sum_{\ell=k}^{KBOT-1} \gamma_{\ell+1/2}^* \gamma_{W*}$$

$$S_2 = \sum_{k=KTOP}^{KBOT} \Delta q_k [C_p + L \frac{dq_{s*}}{dT}_k]$$

$$S_1 = \sum_{k=KTOP}^{KBOT} \Delta q_k [(C_p + L \frac{dq_{s*}}{dT}_k)(T_k + S_0_k) + L(q_{s*} - q_{s*})]$$
1.5 Calculate \( T \) and \( q \) for the other levels for \( KBOT > k \geq KTOP \)

\[
q_{KBOT} = q^*_{KBOT} + \left( \frac{dq_S}{dT} \right)_{KBOT}^* \left( \frac{S_1}{S_2} - T_{KBOT} \right)
\]

\[
T_{KBOT} = \frac{S_1}{S_2}
\]

1.6 Calculate new \( q^*_S \) and \( \gamma^*_w \) values

\( q^*_S \) for \( k=KTOP,KBOT \)

\( \gamma^*_w \) for \( k=KTOP,KBOT-1 \)

are computed using same formulas as in MADADJ

if \( (T_{k+1} - T_k) > \gamma^*_w + \delta \)

The calculations above are repeated until

\( (T_{k+1} - T_k) < \gamma^*_w + \delta \)

at all levels or until 20 iterations have been completed.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE DTRADS
*DECK DTRADS.2

Adds radiative heating rate to other terms in thermodynamic equation and accumulates heating rate for budget calculation.

ALGORITHM:

2. Do radiative adjustment

\[
T_2 = T_2 + Q^a
\]

3. Do budget calculation

\[
DHRAD = DHRAD + 2 \Delta t \frac{C_p}{g} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{i=1}^{I} \frac{p^{n-1}}{s} Q^n \Delta a_k \omega_j /(2I)
\]

HISTORY:

27 OCT 78, U. Cubasch, ECMWF.
SUBROUTINE DYN
*DECK DYNAM.2

For current latitude, calls subroutines GRMULT which computes and
accumulates non-linear terms and calls routines to Fourier transform
values at time n-1 and non-linear quantities at time n, computes
symmetric and antisymmetric parts and calls subroutine LEG to
accumulate sums for Gaussian integration.

ALGORITHM:

2. Do dynamics
2.1 Calculate non-linear part of tendencies

   CALL GRMULT

2.2 FFT of T-1 values of prognostic variables

   via calls to FFT991

2.3 FFT of non-linear quantities

   via calls to FFT991

2.4 Compute symmetric and antisymmetric parts

For the global case the following is done when the current
row is in the Southern Hemisphere (and the corresponding
Northern Hemisphere row is available from the previous call
to DYN).

$\psi S(-\mu_i) = \frac{1}{2} [\psi(\mu_i)+\psi(-\mu_i)]$

$\psi A(-\mu_i) = \frac{1}{2} [-\psi(\mu_i)+\psi(-\mu_i)]$

where $(\mu_i)$ is the current (Southern Hemisphere)
latitude line

CALL LEG
to accumulate sums in the Gaussian quadrature.

HISTORY:

11 JAN 78, A.P.M. Baede, ECMWF.
SUBROUTINE ENDRUN
*DECK DUMMIES.19

Releases restart files through calls to RETURN and PURGE.

HISTORY:

16 JAN 80, U. Cubasch, ECMWF.
SUBROUTINE ESTABL(ES,T)
*DECK ESATUR.2

Determine saturation vapor pressure $e_s$ for a given temperature $T$.

ARGUMENTS:

ES: [out] saturation vapor pressure ($e_s$)
T : [in] temperature ($T$)

ALGORITHM:

Linear interpolation within a table of values specified for every
1° temperature from 173.16 K to 375.16 K. Values for temperatures
outside the table are set equal to the end values.

HISTORY:

1 MAR 81, ECMWF
Modified 7 JAN 83, M. Kuhn, NCAR.
SUBROUTINE ESTABV (KES, KQSLEV, KTEMP1, KTEMP2, N, SIGHK, CRITES)

*DECK ESTABV

Determines a vector of saturation vapor pressures es and a vector of saturation mixing ratios qs from given vectors of temperatures T and surface pressures Ps.

ARGUMENTS:

KES: [in] offset for BUFFER array for output saturation vapor pressure
KQSLEV: [in] offset for BUFFER array for output saturation mixing ratio
KTLEV: [in] offset for BUFFER array for input temperature
KPS: [in] offset for BUFFER array for input surface pressure
KTEMP1: [in] offset for BUFFER array for a temporary vector of length N
KTEMP2: [in] offset for BUFFER array for a temporary vector of length N
N: [in] length of vectors
SIGHK: [in] sigma value used to get pressure from surface pressure
CRITES: [in] critical es factor (α) at which condensation occurs

ALGORITHM:

es from linear interpolation within a table of values specified for every 1° temperature from 173.16 K to 375.16 K. Values for temperatures outside the table are set equal to the end values.

\[
q_s = \varepsilon \frac{\alpha e_s}{\alpha p - \alpha e_s}
\]

HISTORY:

10 JAN 83, M. Kuhn, NCAR.
SUBROUTINE FLDLST(NFLDS, MAXSIZ, MFLDS, IMXLNF, NLEV, NLONP2)

*DECK FLDLST.2

Produces field list information for history tape header and returns values for NFLDS and MAXSIZ.

ARGUMENTS:

NFLDS: [out] number of fields being written on the history volume

MAXSIZ: [out] length of data record on history volume

MFLDS(5,NFLDS): [out]
MFLDS(1,IF): field name (Character)
MFLDS(2,IF): layer location and instantaneous or accumulated field flags (Integer). The one's digit is for the layer location flag, the ten's digit flags whether the field is an instantaneous value or an accumulation since the last file was written. The one's digit of MFLDS(IF,2)
   = 0 for a single level field
   = 1 for a multilevel field at half levels
   = 2 for a multilevel field at full levels

The ten's digit
   = 0 for an instantaneous field
   = 1 for an accumulated field

MFLDS(3,IF): pointer to first value in data record (Int.)
MFLDS(4,IF): data packing flag (Int.)
MFLDS(5,IF): units of field (Char.)

IMXLNF: [in] maximum length allowed for the array MFLDS

NLEV: [in] number of model levels

NLONP2: [in] number of data points on a latitude line

HISTORY:

12 JAN 82, T. Mayer, NCAR
SUBROUTINE GAUAW(A,W,K)
*DECK LGNDRE.2

Calculates abscissa and weights for Gaussian integration.

ARGUMENTS:

A(K): [out] sine of Gaussian latitudes (u = sin\(\phi\)) from N pole to S pole
W(K): [out] corresponding weights
K : [in] number of Gaussian latitudes from pole to pole

ALGORITHM:


HISTORY:

1 MAR 81, ECMWF.
FUNCTION GLBAVG
*DECK GLBAVG.2

Accumulates line by line the global average of Tn-1 and
(un-1)^2 + (vn-1)^2 in /COMVDF/

ALGORITHM:

After completion of last latitude line

\[ TGA_k = \sum_{j=1}^{NOREC} \sum_{i=1}^{NLON} \frac{T_{k}^{n-1}(\lambda_i, \phi_j)}{2*NLON} \]

\[ UVGA_k = \left\{ \sum_{j=1}^{NOREC} \sum_{i=1}^{NLON} \frac{[u_{k}^{n-1}(\lambda_i, \phi_j)]^2 + [v_{k}^{n-1}(\lambda_i, \phi_j)]^2}{w_j/(2*NLON)} \right\}^{1/2} \]

HISTORY:

5 MAY 82, R. Sato, NCAR
SUBROUTINE GRCALC
*DECK GRCALC.2

Computes Fourier coefficients for current latitude line from spherical harmonic coefficients.

ALGORITHM:

0.1 Set pointers to blank COMMON

CALL POINTS

0.2 Fill COMLEG

check for successful completion of BUFFER IN of $P_n^m$ and $H_n^m$
for current line
initiate BUFFER IN of $P_n^m$ and $H_n^m$ at next latitude line

0.3 Compute ALPN and DALPN

create complex vectors of polynomials $P_n^m$ and required derivatives in B array

$$ALP(m,n) = P_n^m$$

$$DALP(m,n) = H_n^m / a$$

$$ALPN(m,n) = P_n^m \times \frac{ma}{n(n+1)}$$

$$DALPN(m,n) = H_n^m \times \frac{a}{n(n+1)}$$

1.1 Zero part of B

zero out the region of B used to accumulate sums

2.0 Computation for NLEV-level variables

The contributions to the Fourier coefficients from the symmetric and antisymmetric polynomials are accumulated separately.
Let $\sum S_n^{m} = N(m)$ denote $\sum S_n^{m}$ including only symmetric polynomials or derivatives of symmetric polynomials.

and $\sum A_n^{m} = N(m)$ denote $\sum A_n^{m}$ including only antisymmetric polynomials or derivatives of antisymmetric polynomials.

The following are computed for each vertical level.

$$TS(m) = \sum S_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$TA(m) = \sum A_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$qS(m) = \sum S_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$qA(m) = \sum A_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$\delta S(m) = \sum S_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$\delta A(m) = \sum A_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$\zeta S(m) = \sum S_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$\zeta A(m) = \sum A_n^{m} * ALP(m,n) \quad \text{for } m=0,M$$
$$US(m) = \sum S_n^{m} * DALPN(m,n) \quad \text{for } m=0,M$$
$$UA(m) = \sum S_n^{m} * DALPN(m,n) \quad \text{for } m=0,M$$
\[ \text{VS}(m) = - \sum_{n} \delta_{n}^{m} \times \text{DALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{VA}(m) = - \sum_{s} \delta_{s}^{m} \times \text{DALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{UIS}(m) = \sum_{n} \delta_{n}^{m} \times \text{ALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{UIA}(m) = \sum_{n} \delta_{n}^{m} \times \text{ALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{VIS}(m) = \sum_{n} \delta_{n}^{m} \times \text{ALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{VIA}(m) = \sum_{n} \delta_{n}^{m} \times \text{ALPN}(m,n) \quad \text{for } m=0,M \]

\[ \text{UA}(m) = \text{UA}(m) - i\text{UIA}(m) \quad \text{for } m=0,M \]

\[ \text{US}(m) = \text{US}(m) - i\text{UIS}(m) \quad \text{for } m=0,M \]

\[ \text{VA}(m) = \text{VA}(m) - i\text{VIA}(m) \quad \text{for } m=0,M \]

\[ \text{VS}(m) = \text{VS}(m) - i\text{VIS}(m) \quad \text{for } m=0,M \]

\[ \text{UA}(0) = \text{UA}(0) - \frac{\Omega}{a \sqrt{.375}} H_{1}^{0} \]

When the vertical index equals IWRITE (=NL/2+1) the BUFFER OUT of the Northern Hemisphere grid values on the previous line which was initiated in SCAN2 is checked for completion. The BUFFER OUT for the Southern Hemisphere grid values on the previous line which were also computed previously in SCAN2 is initiated.

3.0 Computation for 1-level variables

\[ [\&n_{P_{S}}]^{m} \]

\[ \text{PSS}(m) = \sum_{n} \&n_{P_{S}}^{m} \times \text{ALP}(m,n) \quad \text{for } m=0,M \]
PSA(m) = \sum_{n} \alpha_{nP}^{m} \cdot \text{ALP}(m,n) \quad \text{for } m=0,M

\frac{1}{a} [(1-\mu^2) \partial \alpha_{nP} / \partial \mu]^{m}

DPSDMS(m) = \sum_{n} \alpha_{nP}^{m} \cdot \text{DALP}(m,n) \quad \text{for } m=0,M

DPSDMA(m) = \sum_{S} \alpha_{nP}^{m} \cdot \text{DALP}(m,n) \quad \text{for } m=0,M

\frac{1}{a} [\partial \alpha_{nP} / \partial \lambda]^{m}

DPSDLS(m) = i \frac{m}{a} \text{PSS}(m,n) \quad \text{for } m=0,M

DPSDLA(m) = i \frac{m}{a} \text{PSA}(m,n) \quad \text{for } m=0,M

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE GRMULT
*DECK GRMULT.2

Performs nonlinear dynamical computations in grid point space.

ALGORITHM:

2. Zero auxiliary fields in blank common
   \[ B(\text{NKDDS}) = 0. \]
   \[ B(\text{NKVPDS}) = 0. \]
   \[ B(\text{NNDDS}) = 0. \]
   \[ B(\text{NNVPDS}) = 0. \]
   \[ B(\text{NKAD}) = 0. \]
   \[ B(\text{NKAVP}) = 0. \]
   \[ B(\text{NKSDOT}) = 0. \]
   \[ B(\text{NSDTMD}) = 0. \]
   \[ B(\text{NVDKDP}) = 0. \]
   \[ B(\text{NTKMI}) = 0. \]
   \[ B(\text{NQKMI}) = 0. \]
   \[ B(\text{NUKMI}) = 0. \]
   \[ B(\text{NVKMI}) = 0. \]

3. Calculate some auxiliary quantities

   \[ B(\text{NNDDS}) = \sum_{\lambda=1}^{K} \delta_{\lambda} \Delta \sigma_{\lambda} \]

   \[ B(\text{NNVPDS}) = \sum_{\lambda=1}^{K} \left[ \frac{U_{k}}{a} \frac{\Delta \eta_{\lambda} s_{k}}{\lambda} + \frac{V_{k}(1-\mu^{2})}{a} \frac{\Delta \eta_{\mu} s_{k}}{\mu} \right] \frac{\Delta \sigma_{k}}{(1-\mu^{2})} \]

3.20 Virtual temperature not used

   \[ T'_{v_{k}} = B(\text{NTV}) = T'_{k} \]
4. First part of tendencies

The following is done in the order indicated for $k = 1$ to $K$

$$B(NVKDP) = (V \cdot \nabla \eta p_s)_k$$

$$B(NKAD) = \sum_{s=1}^{k} C_{k \ell} \delta_{\ell}$$

$$B(NKAVP) = \sum_{s=1}^{k} C_{k \ell} (V_j \cdot \nabla \eta p_s)_\ell$$

$$B(IGRU2) = U_{k+1}$$

$$B(IGRV2) = V_{k+1}$$

$$B(IGRD2) = \delta_{k+1}$$

4.13 Calculate vertical velocity omega

$$\omega_k = [V_k \cdot \nabla \eta p_s - \sum_{j=1}^{k} C_{kj} \delta_j - \sum_{j=1}^{k} C_{kj} (V_j \cdot \nabla \eta p_s)] \alpha_k p_s$$

4.2 UT, VT, UQ, VQ, KIN ENERGY

Part of rotation terms, temp. and hum. tendencies
Define weights for level k to convert from ECMWF vertical differences to Australian finite differences

\[ w_{1-} = 0 \quad w_{1+} = \frac{2\sigma_1}{\sigma_2 - \sigma_1} \]

\[ wq_{1-} = w_{1-} \quad wq_{1+} = w_{1+} \]

\[ wT_{1-} = wq_{1-} \quad wT_{1+} = wq_{1+} \]

\[ \Delta \sigma_1^* = \sigma_1 \]

\[ w_{K-} = \frac{2(1 - \sigma_K)}{\sigma_K - \sigma_{K-1}} \quad w_{K+} = 0 \]

\[ wq_{K-} = \frac{2(1 - \sigma_K)}{\sigma_K - \sigma_{K-1} \ln(\sigma_K / \sigma_{K-1})} \quad wq_{K+} = w_{K+} \]

\[ wT_{K-} = \frac{2\Delta \sigma_K}{\sigma_K \ln(\sigma_K / \sigma_{K-1})} \quad wT_{K+} = wq_{K+} \]

\[ \Delta \sigma_k^* = \frac{1}{2} (\sigma_k - \sigma_{k-1}) \]

\[ w_{k-} = \frac{\sigma_{k+1} - \sigma_k}{\sigma_k - \sigma_{k-1}} \quad w_{k+} = \frac{\sigma_k - \sigma_{k-1}}{\sigma_{k+1} - \sigma_k} \quad 2 \leq k \leq K-1 \]

\[ wq_{k-} = w_{k-} \quad wq_{k+} = w_{k+} \quad 2 \leq k \leq K-2 \]

\[ wq_{k-} = w_{k-} \quad wq_{k+} = \frac{\sigma_k - \sigma_{k-1}}{\sigma_{k+1} - \sigma_{k+1/2} \ln(\sigma_{k+1} / \sigma_k)} \quad k = K-1 \]

\[ wT_{k-} = wq_{k-} \quad wT_{k+} = wq_{k+} \quad 2 \leq k \leq K-1 \]

\[ \Delta \sigma_k^* = \frac{1}{2} (\sigma_k - \sigma_{k-1}) \quad 2 \leq k \leq K-1 \]
\[ B(\text{NUT})_k = U_k T'_k \]
\[ B(\text{NVT})_k = V_k T'_k \]
\[ B(\text{NUQ})_k = U_k q_k \]
\[ B(\text{NVQ})_k = V_k q_k \]
\[ B(\text{NE})_k = \frac{1}{2} \left( U_k^2 + V_k^2 \right) \]
\[ B(\text{NFU})_k = B(\text{NFU})_k + V_k \xi_k \]
\[ - \frac{w_{k-1}}{2 \Delta q_k} \left[ \dot{a}_k - \frac{1}{2} (U_k + U_{k-1}) \right] - RT_k \frac{\partial n_p}{\partial \alpha} \]
\[ \text{B(NKSDOT)} \quad \text{B(NUKMI)} \]
\[ B(\text{NFV})_k = B(\text{NFV})_k - U_k \xi_k \]
\[ - \frac{w_{k-1}}{2 \Delta q_k} \left[ \dot{a}_k - \frac{1}{2} (V_k - V_{k-1}) \right] - RT_k \frac{(1-\mu^2)}{a} \frac{\partial n_p}{\partial \mu} \]
\[ \text{B(NKSDOT)} \quad \text{B(NVKMI)} \]
\[ B(NT2)_k = B(NT2)_k + \delta_k T'_k \]

\[
- \frac{1}{2\Delta \sigma_k} \left\{ \sum_{\varepsilon=1}^{K} \delta_{\varepsilon} \Delta \sigma_{\varepsilon} + \sum_{\varepsilon=1}^{K} V_{\varepsilon} \cdot \nu \lambda \nu p_s \right\} \]

\[
\frac{k-1}{2} \left( \sum_{\varepsilon=1}^{K} \delta_{\varepsilon} \Delta \sigma_{\varepsilon} + \sum_{\varepsilon=1}^{K} V_{\varepsilon} \cdot \nu \lambda \nu p_s \right) \]

\[
- \left\{ \delta_k + V_k \cdot \nu \lambda \nu p_s \right\} \Delta \sigma_k^* \times \]

\[
\left[ \left( T'_k - T'_{k-1} \right) wT_k^- + \left( T'_{k+1} - T'_k \right) wT_k^+ \right] \]

\[
- \frac{1}{2\Delta \sigma_k} \left\{ \sum_{\varepsilon=1}^{K} \delta_{\varepsilon} \Delta \sigma_{\varepsilon} - \sum_{\varepsilon=1}^{K-1} V_{\varepsilon} \cdot \nu \lambda \nu p_s \right\} \]

\[
- \nu \cdot \nu \lambda \nu p_s \Delta \sigma_k^* \times \]

\[
\left[ \left( T_{\omega_k} - T_{\omega_{k-1}} \right) wT_k^- + \left( T_{\omega_{k+1}} - T_{\omega_k} \right) wT_k^+ \right] \]

\[
- k T_{\omega_k} \sum_{\varepsilon=1}^{K} C_{\varepsilon k} \left( \nu \cdot \nu \lambda \nu p_s \right)_\varepsilon \]

\[
- k T'_k \sum_{\varepsilon=1}^{K} C_{\varepsilon k} \left( \delta + \nu \cdot \nu \lambda \nu p_s \right)_\varepsilon \]

\[
+ k \left( T_{\omega_k} + T'_k \right) \left( \nu \cdot \nu \lambda \nu p_s \right)_k \]
\[ B(NQM1)_k = B(NQM1)_k \]

\[ + 2\Delta t \left[ \delta_k q_k - \frac{wq_k}{2\Delta \sigma_k} q_{k-1/2} \left( q_k - q_{k-1} \right) \right] \]

\[ B(NKSDOT) B(NQKMI) \]

5. Second part of tendencies

The following is not done for \( k = K \)

\[ B(NTKM1) = T_k \]
\[ B(NQKM1) = q_k \]
\[ B(NUKM1) = u_k \]
\[ B(NVKM1) = v_k \]
\[ B(NKDDS) = \sum_{l=1}^{k} \delta_l \Delta \sigma_l \]
\[ B(NKVPDS) = \sum_{l=1}^{k} (V \cdot v)_{kp} \Delta \sigma_l \]
\[ B(NSDTMD) = q_{k+1/2} \sum_{l=1}^{K} (V \cdot v)_{kp} \Delta \sigma_l \]
\[ - \sum_{l=1}^{k} (V \cdot v)_{kp} \Delta \sigma_l \]
\[ B(NVDPDS) \]

\[ \tilde{a}_{k+1/2} = B(NKSDOT) = B(NSDTMD) \]

\[ + q_{k+1/2} \sum_{l=1}^{K} \delta_l \Delta \sigma_l \]

\[ - \sum_{l=1}^{k} \delta_l \Delta \sigma_l \]
5.4 Rest of rotation terms, temperature, and humidity

\[ B(\text{NFU})_k = B(\text{NFU})_k - \frac{w_k^+}{2\Delta\alpha_k} q_{k+1/2} ( U_{k+1} - U_k ) \]
\[ B(\text{NKSDOT}) \cdot B(\text{NUKM1}) \]

\[ B(\text{NFV})_k = B(\text{NFV})_k - \frac{w_k^+}{2\Delta\alpha_k} \cdot q_{k+1/2} ( V_{k+1} - V_k ) \]
\[ B(\text{NKSDOT}) \cdot B(\text{NVKM1}) \]

\[ B(\text{NQM1})_k = B(\text{NQM1})_k - \frac{wq_k^+}{2\Delta\alpha_k} \cdot q_{k+1/2} ( q_{k+1} - q_k ) \]
\[ B(\text{NKSDOT}) \cdot B(\text{NTKM1}) \]

\[ B(\text{NQM1})_k \] is the sum of all undifferentiated terms in the moisture equation

(end of k loop)

5.7 Compute \( TM_1 + TV - T + DT \times T2 \)

\[ \text{BUF}(\text{NTVM1})_k = \frac{1}{2} T^{n-1}_k + T'_V - T'_k = \frac{1}{2} T^{n-1}_k, \quad k=1 \text{ to } K \]

but be careful, the lower part is overwritten later by \( \text{BUF}(\text{NR}) \)

6.0 Calc. part of \( R \) of RHS divergence equation

The following is done in the order indicated for \( k=1 \text{ to } K \)

6.2 Compute first part of \( R \)

\[ \text{B}(\text{NR})_k = \phi_s + \frac{1}{2} R T_{o_k}^{n} \frac{\eta_p}{\eta_s}^{n-1} + E_k + R B_{kk} \left( \frac{1}{2} \bar{T}_k^{n-1} \right) \]
6.25 Compute contribution from nontriangular part of G

\[ B(NR)_K = B(NR)_K + RB_{K,K-1} \left( \frac{1}{2} \frac{n-1}{K-1} \right) \] for \( k=K \) only

6.3 Compute second part of \( R \)

The following is done when \( k < K \)

\[ B(NR)_k = B(NR)_k + \sum_{\ell=k+1}^{K} RB_{\ell,k} \left( \frac{1}{2} \frac{n-1}{\ell} \right) \]

\[ B(NR)_k = \frac{Dv_k}{2 \Delta t} \] is the sum of all the explicit \( v^2 \) terms in the divergence equation.

(end of \( k \) loop)

7. Add tendencies to previous timestep values of prognostic variables \( \ln(p\text{star}), T, \) and \( q \)

7.1 Surface pressure

\[ B(NLPSM1) = \frac{\ln p_s}{n-1} - 2 \Delta t B(NNPDS) \]

\[ B(NLPSM1) = B(NLPSM1) - \Delta t \sum_{k=1}^{K} \Delta q_k \frac{\delta n}{\delta_k} \]

\( B(NLPSM1) \) is the sum of all undifferentiated explicit terms in the surface pressure equation.

7.2 Temperature (humidity was finished in 5.3)

\[ B(NTM1)_k = B(NTM1)_k + 2 \Delta t B(NT2)_k \]

\[ B(NTM1)_k = B(NTM1)_k - \Delta t \sum_{k=1}^{K} \tau_{k,j} \frac{\delta n}{\delta_k} \]

\( B(NTM1) \) is the sum of all undifferentiated explicit terms in the temperature equation.
HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
Converted to Australian finite differences by D. Williamson, NCAR.
SUBROUTINE HORDIF
*DECK HORDIF.2

Computes horizontal diffusion of \( \zeta, \delta, T \) and \( q \) in spectral space using implicit linear \( \nabla^2 \).

ALGORITHM:

1.1 Define scalar constants

1.2 Define diffusion constants

\[
KTQ(n) = \left[ 1 + 2 \Delta t K_H \frac{n(n+1)}{a^2} \right]^{-1} \quad n = 1, K
\]
\[
KZD(n) = \left[ 1 + 2 \Delta t K_H \left( \frac{n(n+1)}{a^2} - \frac{2}{a^2} \right) \right]^{-1} \quad n = 1, K
\]

\( KTQ(1) = 1 \)
\( KZD(1) = 1 \)

2. Horizontal diffusion of \( Z, D, T, Q \)

\[
\begin{align*}
\zeta^n_m &= KZD(n) \cdot \zeta^n_m \\
\delta^n_m &= KZD(n) \cdot \delta^n_m \\
T^n_m &= KTQ(n) \cdot T^n_m \\
q^n_m &= KTQ(n) \cdot q^n_m
\end{align*}
\]

\( n \geq N \) for \( k > 2 \)
\( \text{all } n \) for \( k \leq 2 \)

HISTORY:

17 AUG 78, M. Jarraud, ECMWF, \( \nabla^4 \) form.
22 DEC 81, D. Williamson, NCAR, converted to \( \nabla^2 \) form.
SUBROUTINE HVAR(KNAME,KVALUE)
*DECK CLASSU.73

Writes a label and the value of a Hollerith variable to unit NOUT (=6).

ARGUMENTS:

KNAME : [in] label to be written (up to 8 characters)
KVALUE: [in] Hollerith variable to be written

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
FUNCTION IGTSEQ(DUM)
*DECK IGTSEQ.2

    Returns job sequence number as Hollerith data right justified.

ARGUMENT:

    DUM: dummy argument

HISTORY:

    12 JAN 82, T. Mayer, NCAR
SUBROUTINE INIDAT
*DECK INIDAT.2

Transforms initial grid point data to spherical harmonic coefficients. Recomputes all required grid point fields from coefficients.

ALGORITHM:

5.11 Compute \( U = U \sqrt{1 - \mu \mu} \), \( V = V \sqrt{1 - \mu \mu} \), etc.

\[
U = u \cos \phi \\
V = v \cos \phi \\
\ln P_n = \ln(P/10^5)
\]

\( T' = T - To \)

5.12 Compute \((U, V, \ln(P*), T, Q, \text{AND PHI}*)M\)

Calls FFT991 to create Fourier coefficients of fields indicated. Computes symmetric and antisymmetric components of each.

\[
\psi_i^m = \frac{1}{2} (\psi_i^m + \psi_{-i}^m) \quad i=1, I
\]
\[
\psi_i^m = \frac{1}{2} (\psi_i^m - \psi_{-i}^m)
\]

5.13 Read ALP, DALP and compute VZMN, DMN and \( \ln(P*)MN \) and also PHI*MN, TMN, and QMN

\[
\phi_i^m = \sum_{i=1}^{I} 2w_i \phi_s^m(\mu_i)P_n^m(\mu_i)
\]

where \( \phi_s^m(\mu_i) \) is taken to be \( \phi_s^m \) for \( m-n \) even and \( \phi_s^m \) for \( m-n \) odd.
The sum runs from pole to equator. Similarly in the following sums

\[
\ln P_{sn}^m = \sum_{i=1}^{I} 2w_i \ln P_{sn}^m(\mu_i)P_n^m(\mu_i)
\]
\[
\varsigma_n^m = \sum_{i=1}^{I} 2w_i \left[ u_i^m(u_i)H_n^m(u_i) + i m u_i^m(u_i)P_n^m(u_i) \right]/\left[ a(1-\mu_i^2) \right]
\]

\[
\phi_n^m = \sum_{i=1}^{I} 2w_i \left[ -v_i^m(u_i)H_n^m(u_i) + i m v_i^m(u_i)P_n^m(u_i) \right]/\left[ a(1-\mu_i^2) \right]
\]

\[
T_n^m = \sum_{i=1}^{I} 2w_i T_i^m(u_i)P_n^m(u_i)
\]

\[
q_n^m = \sum_{i=1}^{I} 2w_i q_i^m(u_i)P_n^m(u_i)
\]

Note: In the \( \varsigma \) and \( \phi \) equations the parity of the coefficients of the terms involving the derivatives \( H \) is opposite that of the undifferentiated terms.

5.19 Calculate semi-implicit matrices for first time step

CALL SETTAU

5.2 Compute grid point values of \( \text{PHI*}, U, V, \text{LN(P*)}, T, Q, VZ, D, \) and \( \text{GRAD(LN(P*)}) \)

The following are done for the symmetric and antisymmetric parts of each field by summing over symmetric or antisymmetric polynomials.

\[
\phi_s^m = \sum_{n=m}^{N(m)} \phi_s^m p_n^m
\]

\[
\lambda p_s^m = \sum_{n=m}^{N(m)} \lambda p_s^m p_n^m
\]

\[
[\partial \lambda p_s^m / \partial \lambda]^m/a = m \sum_{n=m}^{N(m)} \frac{1}{a} \lambda p_s^m p_n^m
\]

\[
[(1-\mu^2) \partial \lambda p_s^m / \partial \mu]^m/a = \sum_{n=m}^{N(m)} \lambda p_s^m H_n^m
\]

\[
U^m = \sum_{n=m}^{N(m)} \left[ \varsigma_n^m H_n^m - i m \phi_n^m p_n^m \right]/n(n+1)
\]
\[ \psi^m = \sum_{n=m}^{N(m)} \left[ -im \zeta_n^m \Phi_n^m - \delta_n^m H_n^m \right] \frac{a}{n(n+1)} \]

\[ \zeta^m = \sum_{n=m}^{N(m)} \zeta_n^m \Phi_n^m \]

\[ T^m = \sum_{n=m}^{N(m)} T_n^m \Phi_n^m \]

\[ q^m = \sum_{n=m}^{N(m)} q_n^m \Phi_n^m \]

\[ \delta^m = \sum_{n=m}^{N(m)} \delta_n^m \Phi_n^m \]

Again, note that the parity of the terms involving the derivatives \( H \) is opposite that of the undifferentiated terms.

5.23 Correction to get the absolute vorticity

\[ \zeta^0 = \zeta^0 + \frac{\Omega}{\sqrt{0.375}} P_1^0 \]

5.24 Recompute real fields from symmetric and antisymmetric parts

\[ \psi_i^m = \psi_S_i^m + \psi_A_i^m \quad , \quad i > 0 \quad \text{Northern hemisphere} \]

\[ \psi_i^m = \psi_S_i^m - \psi_A_i^m \quad , \quad i < 0 \quad \text{Southern hemisphere} \]

5.25 Compute grid point values

calls FFT991 to produce grid point fields from Fourier coefficients

HISTORY:

11 DEC 78, M. Jarraud, ECMWF.
SUBROUTINE INITIAL
*DECK INITAL.2

Controls initialization of a model run. Calls various other routines to accomplish specific initialization functions.

ALGORITHM:

1.3 Attaches dataset for initial data values
   CALL ATTACH

2.0 Assigns units for restart and Legendre polynomial datasets
   CALL REQUEST

2.5 Initialize /COMBAS/ and /COMMAP/ and read /COMHKP/ initial values
   CALL RDHDR
   CALL DATCOM

2.7 Initialize model control variables and physical constants
   CALL DATINI
   CALL CON
   CALL SETUP interface to NCAR radiation package
   CALL IMPT1

   (CON, SETUP, and IMPT1 are associated with the radiation program library and will be documented in the future.)

3.1 Initialize blank common buffer to zero

5.0 Input initial state of model variables
   CALL INIDAT

HISTORY:

8 DEC 78, M. Jarraud, ECMWF.
1 MAY 81, R.K. Sato, NCAR, modified for NCAR operations.
SUBROUTINE IVAR(KNAME, KVALUE)
*DECK CLASSU.57

    Writes a label and the value of an integer variable to unit NOUT
    (= 6).

ARGUMENTS:
    KNAME: [in] label to be written (up to eight characters)
    KVALUE: [in] integer variable to be written

HISTORY:
    19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE JOBTIM(PTIME)
*DECK CLASSU.200

Retrieves user supplied CPU time limit in seconds by call to subroutine TIMCPU.

ARGUMENTS:

PTIME: [out] The CPU time limit in seconds as obtained from TIMCPU.

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE LABRUN
*DECK C1S1.2

    Writes on unit NOUT (=6) labeling stored in LABEL1-LABEL8.
    Also, writes descriptor LABEL1 to unit NDIARY (=7).

HISTORY:

    19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE LEG
*DECK LEG.2

Accumulates latitudinal sums of Fourier coefficients involved in
the grid to spectral Legendre transformations.

ALGORITHM:

2. Contribution to spectral fields

\[ P^m_n = \sum_{i=1}^{I} P^m_n(u_i) P^m_n(u_i) 2\omega_i \]

the sum is performed over the even and odd polynomials

2.3 Other fields

the following are computed for each wavenumber for
symmetric and antisymmetric components

\[ I^* = I - 2\Delta t \frac{im}{a(1-u^2)} (UT) \]

\[ Q^* = Q - 2\Delta t \frac{im}{a(1-u^2)} (UQ) \]

\[ D^* = D + 2\Delta t \frac{im}{a(1-u^2)} (FU) \]

\[ V^* = V + 2\Delta t \frac{im}{a(1-u^2)} (FV) \]

the following sums are performed over the hemisphere for
the even and odd polynomials

\[ I^m_n = \sum_{i=1}^{I} \left[ R^m_n(u_i) P^m_n(u_i) + (VT)^m \frac{2\Delta t}{a(1-u^2)} H^m_n(u_i) \right] 2\omega_i \]

\[ Q^m_n = \sum_{i=1}^{I} \left[ G^m_n(u_i) P^m_n(u_i) + (VQ)^m \frac{2\Delta t}{a(1-u^2)} H^m_n(u_i) \right] 2\omega_i \]
\[ \frac{p^m}{P_n} = \sum_{i=1}^{I} \left[ \left( \mathcal{O}^m(u_i) + 2 \Delta t \frac{n(n+1)}{a^2} \frac{\partial}{\partial y} \right) p^m_n(u_i) \right. \\
\left. \quad - (FV)^m \frac{2 \Delta t}{a(1-\mu^2)} H^m_n(u_i) \right] 2w_i \]

\[ \frac{v^m}{V_n} = \sum_{i=1}^{I} \left[ (\mathcal{V}^m(u_i)p^m_n(u_i) + (\mathcal{FU})^m \frac{2 \Delta t}{a(1-\mu^2)} H^m_n(u_i) \right] 2w_i \]

\text{HISTORY:}

17 AUG 78, M. Jarraud, ECMWF.
Controls calculations at line NROW by calling appropriate subroutines.

ALGORITHM:

0.1 Set pointers to blank COMMON
   CALL POINTS

0.2 Fill COMLEG
   Sets some variables in COMLEG to appropriate values for line NROW.

0.5 Determine physical constants after first forward timestep
   If NROW=1 and it is the first time step after restarting, variables involving 2Δt are multiplied by 2 to change time differences to centered after the first forward time step.

1.17 Compute $P_s$
   $P_s = \exp(\ln P_s) \times 10^5$

1.5 Convective adjustment
   CALL CONVAD

1.52 Time filter (Part 2)
   The second half of the time filter is applied to appropriate values by call to TIMESM (, , 1, ε, )

2. Write-up time
   If it is a history tape write time, current line values are written via a call to WSHIST.

2.7 Update start data set
   CALL SDS to update pointers for next write-up time.
   CALL CONCAT to concatenate restart datasets.
   CALL SAVDISP to dispose history tape and restart dataset.
   Reposition unit NDATA and set up for next history tape.
3. Time filter (Part 1)
   The first half of the time filter is applied to appropriate values by call to TIMESM ( , , , 1-2e, e, )

4. Physics
   CALL PHYS

5. Dynamics
   CALL DYN

HISTORY:

16 JAN 80, U. Cubasch, ECMWF.
17 DEC 82, Modified by L. Bath, NCAR.
FUNCTION LUNDN(IUNIT)
*DECK SAVDISP.61

Returns a CRAY dataset name (alias) based on logical unit number IUNIT.

ARGUMENTS:

IUNIT: [in] one(n) or two(jj) digit integer
   n < 10 , then tape FTON
   jj ≥ 10 , then tape FTjj

LUNDN: [out] name of dataset created from IUNIT input

HISTORY:

1 MAY 81, R. Wolski, NCAR.
SUBROUTINE MADADJ
*DECK MADADJ1.2

Determines top and bottom indices of moist adiabatically unstable layers and calls DOMADJ to perform the adjustment.

ALGORITHM:

1. Displacements of variables set to adjust temperature and moisture at time level n

2. Calculate QSAT(T) and specify grid points with supersaturated layers

The saturation vapor pressure $e_{sk}$ and saturation mixing ratio $q^*_{sk}$ are determined by a call to ESTABV

$LVDRY = \begin{cases} 
1 & \text{if } q_k/q^*_{sk} > 0.9999 \text{ and } q_{k+1}/q^*_{sk+1} > 0.9999 \\
0 & \text{otherwise}
\end{cases}$

5. Moist adiabatic adjustment

If $LVDRY \neq 0$, at least one pair of adjacent levels is supersaturated and the following computation is done from top to bottom a maximum of 20 times for each value of $\delta$. $\delta = 0.01$ initially

$KSAT(k+1) = \begin{cases} 
1 & \text{if } q_k/q^*_{sk} > 0.9999 \text{ and } q_{k+1}/q^*_{sk+1} > 0.9999 \\
0 & \text{otherwise}
\end{cases}$

if $KSAT(k+1) \neq 0$, both levels $k$ and $k+1$ are supersaturated and the following computations are performed

$e_{sk+1/2}$ is computed for temperature $(T_k + T_{k+1})/2$

by a call to ESTABL

by a call to ESTABL
$e_{s^+}$ and $e_{s^-}$ are computed for temperatures $(T_k + T_{k+1})/2 + 0.5$ and $(T_k + T_{k+1})/2 - 0.5$, respectively, via calls to ESTABL:

$\gamma_{W_{k+1/2}}^* = c(k+1)(T_k + T_{k+1})[p_{k+1/2}^+ c_2 e_{s_{k+1/2}}^+/(T_k + T_{k+1})]/$

$[p_{k+1/2}^+ c_1 (e_s^+ - e_s^-) L/C_p]$

where $p_{k+1/2} = p_s \sigma_{k+1/2}$

$c(k+1) = \frac{R}{2c \sigma_{k+1/2} p_{k+1/2}} \Delta \sigma_{k+1/2}$

$c_1 = \alpha_k \epsilon$

$c_2 = 2\alpha_k \delta L$

$\alpha_k = 0.8$

$KSAT(k+1) = \begin{cases} 2 & \text{if } T_{k+1} - T_k \geq \gamma_{W_{k+1/2}}^* + \delta \\ \text{unchanged otherwise} & \end{cases}$

5.4 Find unstable + saturated layer and do adjustment there

$KB = 0$
$KT = K+1$

The column is searched from $k=1$ to $K-1$

$KTOP = $ one plus first value of $k$ for which $KSAT(k+1) = 2$
$KBOT = $ first value of $k$ after $KTOP$ for which $KSAT(k+1) \neq 2$
$KB = \max(KB, KBOT)$
$KT = \min(KT, KTOP)$
Tk and qk are adjusted simultaneously for KTOP ≤ k ≤ KBOT by a call to DOMADJ.

The column search and correction described above are repeated from \( k = KBOT + 1 \) to \( K-1 \) until all contiguous groups of unstable layers are found and stabilized.

If convergence is not reached after the 20th pass, the convergence criterion \( \delta \) is doubled, a message is printed to unit NOUT(=6), and the process is repeated. If \( \delta \) becomes > .1 and the column still has at least one unstable layer, the model stops.

6. Diagnostics

\[
QCD = \sum_{k=1}^{K} (\hat{q}_k - q_k) \cdot c(k) \cdot p_s
\]

where the \( \hat{q} \) denotes the value before adjustment and \( q \) the value after

\[
c(k) = \frac{\Delta \sigma_k}{p_{H2O} \cdot g}
\]

\[
QC_k = (\hat{q}_k - q_k) \cdot p_s \cdot \frac{\Delta \sigma_k}{g} / 2 \Delta t
\]

6.1 Calculate precipitation

\[
PRC = PRC + QCD/2
\]

HISTORY:

1 MAR 81, ECMWF.
15 JAN 83, Modified by D. Williamson, NCAR.
SUBROUTINE MASTER
*DECK COSO.2

Retrieves user-supplied CPU time limit in seconds. Retrieves system-supplied cumulative CPU time. Calls subroutine BASIC which initializes the labeled commons COMBAS and COMDDP.

The date and time of the current job are fetched and printed.

Calls subroutine COTROL (which controls the run).

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF
SUBROUTINE MESAGE(KMESS)
*DECK  CLASSU.2

    Writes a message of up to 40 characters on unit NOUT (=6).

ARGUMENTS:

    KMESS: [in] variable of dimension (=5) containing the desired message

HISTORY:

    19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE MINV(A,N,D,L,M)
*DECK MINV.2

Inverts a general matrix.

ARGUMENTS:

A : [in] input matrix
    [out] inverse of input matrix A
N : [in] order of matrix A
D : [out] resultant determinant
L : [in] work vector of length N
M : [in] work vector of length N

ALGORITHM:

Standard Gauss--Jordan method. The determinant is also computed.
A determinant of zero indicates that the matrix is singular.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE MODIFY
*DECK C0S2.2

Reads input data in NAMELIST $EXPDEF from unit NREAD and writes values to print unit NOUT. Attaches restart dataset if NSREST=1, indicating normal restart. If restart, calls SPLITF to split out the save files from the restart dataset. Sets logical variables NLRES and NLHST. Calls subroutine SDS to initialize /COMSDS/.

Stops run with printed message if errors occur in the data acquisition.

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
17 DEC 82, Modified by L. Bath, NCAR.
FUNCTION NEWROW(N)
*DECK NEWROW.2

Computes a hemispheric row number (NEWROW) for use with Legendre polynomials from a global or hemispheric grid row number (N).

ARGUMENTS:

N : [in] global or hemispheric row number
NEWROW: [out] hemispheric row number

HISTORY:

22 NOV 78, M. Jarraud, ECMWF.
FUNCTION NORSOU(N)
*DECK NORSOU.2

Returns the Gaussian grid line number minus one corresponding to
the Nth line that the model processes. The Gaussian grid lines are
numbered consecutively from 1 at the northernmost line to NOREC at
the southernmost line.

ARGUMENTS:

N : [in] model latitude index
NORSOU: [out] corresponding sequential row index running from 0 at
the north pole to NOREC-1 at the south pole

ALGORITHM:

\[
\text{NORSOU} = \begin{cases} 
\frac{(N-1)}{2} & \text{if } N \text{ odd} \\
\text{NOREC} - \frac{N}{2} & \text{if } N \text{ even}
\end{cases}
\]

where NOREC is the total number of Gaussian latitude
lines.

HISTORY:

22 NOV 78, M. Jarraud, ECMWF.
FUNCTION NUDATE(NODT, NOSC, INC, NCDT, NCSC)
*DECK NUDATE.2

Determines current date and second.

ARGUMENTS:

NODT: [in] beginning date of this model run encoded as a 6 digit integer, stored as YR/MO/DY. E.g., NODT=791117 for 17 November 1979.

NOSC: [in] the no. of seconds into the day this forecast started, e.g., NOSC=43200 for 12Z.

INC: [in] increment in seconds between initial time and the current time.

NCDT: [out] the current date encoded in the same manner as NODT.

NCSC: [out] the number of seconds into the date specified in NCDT.

ALGORITHM:

1.0 Decode original date information account for leap years.
1.5 Find day increment and compute new value for NCSC.
2.0 Compute new date, month, and year increment.
2.3 Compute new year and new value for NCDT.

HISTORY:

12 JAN 82, T. Mayer, NCAR.
SUBROUTINE OUTPUT(J)
*DECK OUTPUT.2

    Writes on unit NOUT (=6) the number of completed timesteps if J=2.

ARGUMENT:

    J: [in] = 2, then number of completed timesteps (NSTEP) is printed
        ≠ 2, then return

HISTORY:

    1 MAR 81, ECMWF.
SUBROUTINE PAGE
*DECK CLASSU.23

Advances to new page on unit NOUT (=6).

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE PB2GFD(KROW)
*DECK GFDLPB2.2

Solves implicit equation \( f(T_s) = 0 \) for surface temperature \( T_s \) over non-ocean points using Newton-Raphson iterative procedure.

\[
T_s^{k+1} = T_s^k - \frac{f(T_s^k)}{f'(T_s^k)}
\]

Iterations repeated until \( |T_s^{k+1} - T_s^k| \leq \varepsilon \).

Initially \( \varepsilon = 0.02 \). A maximum of 40 iterations are performed for each value of \( \varepsilon \). If no convergence after 40 iterations, \( \varepsilon \) is doubled and the process repeated. If \( \varepsilon > 0.1 \) and convergence is not reached, the model stops.

Computes vertical diffusion of sensible heat using new \( T_s^{k+1} \).

ARGUMENT:

KROW: [in] latitude line index

ALGORITHM:

2. Specify and store some expressions

\[
|V| = \begin{cases} 
\text{max} \ (\sqrt{u^2 + v^2}, 1) \text{ over non-ocean points} \\
\sqrt{u^2 + v^2} \text{ over ocean points}
\end{cases}
\]

\[
b = \rho NLEV C D C P |V|
\]

\[
d = \rho NLEV D W C D L |V|
\]

\[
e = K/I
\]

\[
c = -b \frac{T_{NLEV}}{\phi^{NLEV}} - d q_{NLEV} - e T_{SW} + L z M - (1-A) S_0 - F_0
\]

\[
a = [1 - f_g (1 - E_g)] \phi
\]

3.2 Land, normal conditions, Newton-Raphson procedure

\[
q_s^k = \varepsilon \frac{e^k}{P_s - e_s^k}
\]
where \( e_s^k \) is determined by a call to ESTABL with temperature \( T_s^k \).

\[
T_{s}^{k+1} = T_s^k - \frac{a(T_s^k)^4 + b T_s^k + d q_s^k + e T_s^k + c}{4a(T_s^k)^3 + b + \frac{p_s}{p_s-e_s^k} \frac{d L_1 q_s^k}{R_{H2O}(T_s^k)} + e}
\]

Continue iteration until \(|T_{s}^{k+1} - T_s^k| < \epsilon\).

Over snow \( T_s^{k+1} = \min(T_s^k, T_{MLT}) \).

3.3 Calculate surface fluxes

\[
WFLUX = 2\Delta t \frac{q_s}{\Delta \alpha_K} \frac{p_s}{RT_K} \frac{n-1}{n-1} \frac{C_D}{D_W} \left| \frac{v}{n-1} \right| \left[ q_s(T_s^{n-1}) - \overline{q}_K^{n-1} \right]
\]

\[
\overline{q}_K^{n-1} = q_s^{n-1} + WFLUX
\]

\[
HFLM = 2\Delta t \frac{q_s}{\Delta \alpha_K} \frac{p_s}{RT_K} \frac{n-1}{n-1} \frac{C_D}{D_W} \left| \frac{v}{n-1} \right| \left( a_K T_s^{n+1} - \overline{T}_K^{n-1} \right)
\]

\[
T_2 = T_2 + HFLM/2\Delta t
\]

\[
HFS = HFS + \frac{C_p}{2g} \Delta \alpha_K \frac{p_s}{RT_K} \frac{n-1}{n-1} HFLM
\]

\[
NFU = NFU - \frac{q_s}{\Delta \alpha_K} \frac{p_s}{RT_K} \frac{n-1}{n-1} \frac{C_D}{D_W} \left| \frac{v}{n-1} \right| \left( u_k^{n-1} \right)
\]
\[ NFV = \text{NFV} - \frac{g}{\Delta a_k} \frac{\alpha_k P s_{n-1}}{\text{RT}_k} C_D | \sqrt{V_{n-1}} | v_{K_{n-1}} \]

3.5 Do budget calculations

\[ ZDWF = \sum_{i=1}^{NLON} \text{WFLUX} \times P s_{n-1} \]

\[ ZSUMH = \sum_{i=1}^{NLON} \text{HFLM} \times P s_{n-1} \]

\[ DWFLX = \frac{1}{NLON} \sum_{j=1}^{NOREC} L g \Delta a_k ZDFLUX \frac{w_j}{2} \times 10^{-3} \]

\[ DHFLX = \frac{1}{NLON} \sum_{j=1}^{NOREC} \frac{C}{g} \Delta a_k ZSUMH \frac{w_j}{2} \times 10^{-3} \]

HISTORY:

15 AUG 78, U. Cubasch, ECMWF.

10 DEC 82, Modified by D. Williamson, NCAR.
SUBROUTINE PHCS(PMN,HMN,MAX,JMAX,X1)
*DECK LGNDRE.90

Computes associated Legendre polynomials $p_n^m(\mu)$ and derivatives

$$H_n^m(\mu) = -(1-\mu^2) \frac{d p_n^m(\mu)}{d \mu}$$

for one value of $\mu$.

ARGUMENTS:

PMN: [out] associated Legendre polynomials stored by columns

$$p_n^m = PMN(1 + (n-m) + m*(N+1)); m=0..M, n=m,N+m$$

HMN: [out] derivatives stored by columns

MAX: [in] M + 1

JMAX: [in] N + 1

X1: [in] $\mu = \sin \phi$, where $\phi$ is desired latitude

ALGORITHM:

$P_n^m$; S.L. Belousov, Tables of Normalized Associated Legendre
Polynomials. $p_n^m$ are normalized such that

$$\int_{-1}^{1} (P_n^m)^2 d\mu = \frac{1}{2}.$$  

They must be multiplied by $\sqrt{2}$ to match tables in Belousov.

$$H_n^m(x) = nxP_n^m(x) - \left[\frac{(n^2 - m^2)(2n + 1)}{2n - 1}\right]^{1/2} P_{n-1}^m(x)$$

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE PHYS
*DECK PHYS.2

Calls subroutines which calculate the various physical parameterizations.

ALGORITHM:

1.1 Calculate temperature
\[ T_{k}^{n-1} = T_{k}^{n-1} + T_{0,k} \]
    via CALL TCALC(1)

1.2 Calculate U and V
\[ u^{n-1} = u^{n-1}/\cos \phi \]
\[ v^{n-1} = v^{n-1}/\cos \phi \]

1.3 Calculate surface pressure
\[ p_{s}^{n-1} = \exp(\frac{z_{m}p_{s}^{n-1}}{n}) \times 10^{5} \]

2.5 Do radiation calculation
    Call radiation subroutine

2.7 Add radiation to temperature tendency
    CALL DTRADS

3. Do vertical diffusion calculations

4. Calculate surface values
    CALL PB2GFD(NROW)
5.1 Calculate the temperature deviation

\[ T_k^{n-1} = T_k^{n-1} - T_{0k} \]

via CALL TCALC(-1)

5.2 Calculate the logarithm of the surface pressure

\[ \ln P_s^{n-1} = \text{ALOG} \left( \frac{P_s^{n-1}}{10^5} \right) \]

5.3 Multiply \( u, v \), and the equivalent tendencies by \( \cos(\phi) \)

\[ \bar{u}^{n-1} = \bar{u}^{n-1} \times \cos\phi \]
\[ \bar{v}^{n-1} = \bar{v}^{n-1} \times \cos\phi \]
\[ F_uG = F_uG \times \cos\phi \]
\[ F_vG = F_vG \times \cos\phi \]

HISTORY:

27 APR 78, U. Cubasch, ECMWF.
SUBROUTINE POINTS
*DECK POINTS.2

Defines offsets to BUFFER array in /COMGRD/ for variables involved in the model calculation.

ALGORITHM:

See USERS GUIDE for details of the offset definitions.

HISTORY:

27 APR 78, U. Cubasch.
SUBROUTINE POSUN(IU,NR)
*DECK UTIL.100

Rewinds unit IU and skips NR+1 records on unit IU.

ARGUMENTS:

IU: [in] Unit number upon which records are to be skipped.
NR: [in] Number of records minus one to be skipped.

ALGORITHM:

Rewinds unit=IU. Successively reads and checks records via calls to READBF and CHKBF. Any error is flagged and program execution is stopped.

HISTORY:

1 FEB 81, R.K. Sato, NCAR.
SUBROUTINE PRESET
*DECK PRESESP.2

Sets default parameters for the model.

ALGORITHM:

1.1 I/O device number

If (NLRES = false), then
   IFIL = 0
   NXTAP = 1
If (NLRES = true), then use values read in from tape.

1.2 Output Tape Default Values
   MFILT = 30
   MFILS = 10
   Blank all LTAPES

1.3 Diagnostic parameters
   NLSTAT = .TRUE.
   NFSTAT = -12
   NMAXVE = (NLONP2-4)/4

1.4 Numerical Scheme default values
   NSPHER = 1
   EPS = 0.06
   DIF = 2.5E5
   NESTEP = -10
   NNUMWT = -12
   NWTIME(I) = 0, I=1,50
   NWPTR = 1 if (.NOT.NLRES)
   DTIME = 1080
   IRAD = 24
1.5 Parameterization scheme default values

NLPHYS = .T.
WSSAT = 0.15

1.6 Header defaults
MAXHDL = 1200
MFTEM = 1
NDVUTM = -1 (no packing is done)

HISTORY:

7 JUL 80, U. Cubasch, ECMWF.
17 DEC 82, L. Bath, NCAR, eliminate/modify values.
13 JAN 83, T. Mayer, NCAR, header defaults added.
SUBROUTINE PRICOM
*DECK PRICOM.2

Writes on unit NOUT(=6) common blocks containing the constants for
the physical parameterization subroutines:

/COMPHO/, /COMPH1/, /COMPBL/, /COMBUD/, /COMSDS/.

HISTORY:

14 DEC 78, U. Cubasch, ECMWF.
SUBROUTINE PRNTHD(INTHD, RLHD)
*DECK PRNTHD.2

Prints the header to unit NOUT(=6).

ARGUMENTS:

INTHD: [in] header array (integer)
RLHD: [in] same header array but considered real

HISTORY:

12 JAN 83, T. Mayer, NCAR.
SUBROUTINE PRSCHD(INTHD,RLHD)
*DECK PRSCHD.2

Prints the scalar portion (first 31 values) of the history tape header to unit NOUT(=6).

ARGUMENTS:

INTHD: [in] header array (integer)
RLHD: [in] same header array but considered real

HISTORY:

12 JAN 83, T. Mayer, NCAR.
SUBROUTINE PURGE(IF,IU,IPDN,ICY,ITXT)
*DECK UTIL.77

    Deletes and releases permanent dataset IPDN. Currently returns immediately (no effect).

ARGUMENTS:

    IF : [in] error flag (not used)
    IU : [in] unit number
    IPDN: [in] dataset name
    ICY : [in] cycle number (not used)
    ITXT: [in] text field (not used)

HISTORY:

    1 DEC 81, D. Sato, NCAR.
SUBROUTINE QNEGAT(KQ)
*DECK UTIL.137

Attempts to eliminate negative mixing ratios by taking moisture from the four adjacent point in height and longitude. If the total moisture at these four points is not enough to make the center point zero, nothing is done.

ALGORITHM:

Pointers set to work on time level n

\[ DQ = -q_{kj} \Delta \sigma_k \gamma_j \]

\[ AQ = \sum_{(kj),'} \max (0, q_{(kj)}, \Delta \sigma_k', \gamma_{j,'}) + \sum_{(kj),''} q_{(kj),''} \Delta \sigma_k'' \gamma_{j,``} \]

if AQ < DQ, nothing is done

if AQ > DQ

\[ q_{kj} = 0 \]

\[ \beta = 1 - DQ/AQ \]

\[ q_{(kj),'} = \min (q_{(kj),'}, \beta q_{(kj),'}) \]

\[ q_{(kj),''} = \beta q_{(kj),''} \]

where (kj)' ranges over the points (k+1,j), and (k,j+1) and (kj)'' ranges over (k-1,j) and (k,j-1) but not including (k+1,j) when k=K or (k-1,j) when k=1.

HISTORY:

22 DEC 81, R. Sato, NCAR.
SUBROUTINE QREIG(A,I,J,K,B,C)
*DECK   QREIG.2

    Creates a complex matrix with real part =A, and imaginary part =0.
    Calls COMHES and COMLR to compute the eigenvalues of the complex
    matrix.
    Takes the real part of the eigenvalues.

ARGUMENTS:

A:  [in] = general real input matrix A.
I:  [in] = order and dimension of A 
J:  [in] = not used 
K:  [in] = not used 
B:  [out] the real part of the eigenvalues 
C:  not used 

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE RARRAY(KNAME, PA, KDIM)
*DECK CLASSU.88

Prints label and values of a real array on unit NOUT (=6).

ARGUMENTS:

KNAME: [in] label
PA  : [in] location of first value to be printed
KDIM : [in] number of values printed

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE RDHDR
*DECK RDHDR.2
  Reads the header into /COMHKP/ and checks certain model parameters,
data card values, and variables set in PRESET with their counterparts on the header (/COMHKP/)

HISTORY:

  12 JAN 82, T. Mayer, NCAR.
SUBROUTINE READBF(KUNIT, PA, KLEN)
*DECK READBF.2

    Initiates BUFFER IN operation.

ARGUMENTS:

    KUNIT:  [in] unit number of read
    PA:     [in] read data into area starting at PA(1)
    KLEN:   [in] number of words to be read

HISTORY:

    24 FEB 77, Jan Haseler, ECMWF.
SUBROUTINE READR(KDV, PA, KLEN, KREC)
*DECK READR.2

    Initiates BUFFER IN operation possibly preceded by REWIND.

ARGUMENTS:

    KDV:  [in] unit number of read
    PA:   [in] read data into area starting at PA(1)
    KLEN: [in] number of words read
    KREC: [in] if KREC = 1, REWIND KDV prior to BUFFER IN

HISTORY:

    8 MAY 78, A.P.M. Baede, ECMWF.
SUBROUTINE REORD(ALP,ZALP,DALP,ZDALP)
*DECK LGNDRE.169

Reorders associated Legendre polynomials from column rhomboidal storage to diagonal pentagonal storage.

ARGUMENTS:

ALP: [out] $P_n^m$'s at given latitude with diagonal pentagonal storage

ZALP: [in] $P_n^m$'s at given latitude with column rhomboidal storage

DALP: [out] $H_n^m$'s diagonal storage

ZDALP: [in] $H_n^m$'s column storage

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE REPTHD(KCLASS,KSUB,KPOINT)
*DECK CLASSU.108
Prints the three integer arguments on unit NOUT(=6) to serve as a heading for a diagnostic report.

ARGUMENTS:

KCLASS: [in] calling subroutine CLASS
KSUB: [in] calling subroutine
KPOINT: [in] point in calling subroutine

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE REQUEST(IFAIL,IU,IBSZ)
*DECK UTIL.37

Assigns unit IU.

ARGUMENTS:

IFAIL: [in] a check variable for failure (not used)
IU : [in] unit assigned via a CALL ASSIGN to the CRAY
IBSZ : not currently used

HISTORY:

8 SEP 81, R.K. Sato, NCAR.
SUBROUTINE RESETH(KA,KDIM,KVALUE)
*DECK CLASSU.181

Resets the specified Hollerith array elements to a specified
Hollerith expression.

ARGUMENTS:

KA: [in] array whose elements are to be reset
KDIM: [in] the number of elements (8 characters/word) to be
reset
KVALUE: [in] new expression for each element of KA

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE RESETI(KA,KDIM,KVALUE)
*DECK CLASSU.162
Resets a specified integer array to a specified integer value.

ARGUMENTS:

KA:   [in] integer array whose elements are to be reset
KDIM: [in] the number of elements to be reset
KVALUE: [in] the value for each element of KA

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE RESETL(KLA,KDIM,KLVAL)
*DECK CLASSU.216

Resets the specified logical array elements to a specified logical value.

ARGUMENTS:

KLA:   [in] logical array whose elements are to be reset
KDIM:  [in] the number of elements to be reset
KLVAL: [in] the logical value to which each element of KLA is set

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE RESETR(PA,KDUM,PVALUE)

Resets the specified real array elements to a new specified value.

ARGUMENTS:

PA: [in] real array value whose elements are to be reset
KDUM: [in] the number of elements to be reset
PVALUE: [in] the real value for each element of PA

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE RESUME
*DECK RESUME.2

Initializes COMMON variables from a previous record for resumption of a run (restart).

ALGORITHM:

1.0 Preserve /COMBAS/ variables.
2.4 Acquire and position history tape.
   CALL ACQHIS
2.5 Read /COMHKP/ and compute /COMBAS/ and /COMMAP/.
   CALL RDHDR
   CALL DATCOM
2.6 Restore /COMBAS/ variables.
2.7 Initialize constants.
   CALL DATINI
2.8 Initialize NCAR radiation.
   CALL CON
   CALL SETUP
   CALL IMPT1
3.0 Initialize work files
   Calls RESETR which zeros blank COMMON
   Assigns save units for model output.
   If last history tape is full, rewind NDATA.

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
1 MAY 81, R.K. Sato, NCAR, modified for NCAR version.
SUBROUTINE RETURN(IF,IU)
*DECK UTIL.65

Releases unit IU.

ARGUMENTS:

IF: [in] error flag
   = 0 stop execution if error incurred when releasing unit
   ≠ 0 return to calling program in all cases

IU: [in] unit number to be released

HISTORY:

4 SEP 81, D. Sato, NCAR.
SUBROUTINE RVAR(KNAME,PVALUE)
*DECK  RVAR.2

    Writes a label and the value of a real variable to unit NOUT (=6).

ARGUMENTS:

    KNAME : [in] label to be written
    PVALUE: [in] real variable to be written

HISTORY:

    1 MAR 81, ECMWF.
SUBROUTINE SAVDISP(IUNIT,LTAPE,IPASWD,IMVN)
*DECK SAVDISP.2

Saves (on the CRAY disk) and disposes (to the TBM) a copy of the logical unit IUNIT, using the default or user set write password. Optionally disposes (to the TBM) on a dedicated mass store reel.

ARGUMENTS:

IUNIT: [in] logical unit to be saved on CRAY disk and disposed to the TBM
LTAPE: [in] name used for CRAY pdn and TBM sdn
IPASWD: [in] write password (Default in WRITE)
IMVN: [in] Dedicated MSD reel name (Default is blank)

HISTORY:

1 MAY 81, R.K. Sato, NCAR.
SUBROUTINE SCAN1
*DECK SCAN1.2

Controls north-south grid scan containing the physics and dynamics calculations.

ALGORITHM:

1.1 Swap buffer pointers
1.2 Start to write NROW-1 to work file B
   Return if NROW = MAXROW + 1
1.4 Start to read NROW+1 work files A and B
1.5 Perform calculation on row NROW
   CALL LINEMS
   NROW = NROW + 1
   go to beginning

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE SCAN2
*DECK SCAN2.2

Computes grid point values line by line from spherical harmonic coefficients.

ALGORITHM:

1.1 Calculate grid point values

CALL GRCALC to obtain Fourier coefficients of symmetric and antisymmetric components of fields at current northern hemisphere latitude line, and except when NROW=1, buffer out southern hemisphere grid values on previous line.

1.105 Recompute real fields from symmetric and antisymmetric parts

For the hemispheric case, variables are complete as returned from GRCALC. For the global case

\[ \psi^m (\omega) = \psi^S (\omega) + \psi^A (\omega) \]  
(Northern Hemisphere)

\[ \psi^m (-\omega) = \psi^S (\omega) - \psi^A (\omega) \]  
(Southern Hemisphere)

where \( \psi \) represents the vector of all variables transformed to grid points.

CALL FFT991 to transform Northern Hemisphere coefficients to grid values.

1.30 Begin to write first row

Check for completion of previous BUFFER OUT.
Initiate BUFFER OUT of northern hemisphere grid values.

For the global case, CALL FFT991 to transform southern hemisphere coefficients to grid values. (Note: the BUFFER OUT of these southern hemisphere grid values is initiated in the next call to GRCALC.)
1.4 Test for southern boundary

If not at southern boundary (first line north of equator), increment row count (NROW), swap buffer pointers, and go back to the beginning (1.1).

1.5 Wait for completion of last read

If at the southern boundary, check for completion of previous BUFFER OUT. For the global case, initiate BUFFER OUT of Southern Hemisphere grid values and check for completion of this BUFFER operation.

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE SDS(KSDS,KERR,KCALL)
*DECK SDS.2

Reads or writes restart data set, containing variables in /COMSDS/, /COMTAP/ and /COMBUD/ or updates pointers for next write time.

ARGUMENTS:

KSDS : [in] unit number for read or write of restart data set
KERR : [out] error flag returned to calling routine
   = 1, no error
   = 2, no record found
   = 3, I/O error
KCALL: [in] flag to indicate read or write
   = 1, read a record from unit KSDS
   = 2, write a restart data set record
   = 3, update pointers for next write time

NOTES.

Checks for major I/O errors and prints appropriate diagnostics.

HISTORY:

18 Aug 78, M. Jarraud, ECMWF.
SUBROUTINE SETTAU(ZDT)
*DECK SETTAU.2

Sets matrices dependent on To associated with semi-implicit time step

ARGUMENTS:

ZDT: [in] \Delta t of following semi-implicit time step

ALGORITHM:

2. Reset mean temperature To from spectral coefficients

\[ To^{n+1} = To^n + \frac{0}{To/\sqrt{2}} \]

3.2 Calculation of the elements of matrix TAU in thermodynamic equation

\[ TAU(K+J) = \frac{1}{2\Delta\sigma_k} \left\{ (To_{k+1} - To_k) [\sigma_{k+1/2} \Delta\sigma_j] - \begin{bmatrix} 0 & (k<j) \\ \Delta\sigma_j & (k\geq j) \end{bmatrix} \right\} \]

\[ + \frac{1}{2\Delta\sigma_k} \left\{ (To_k - To_{k-1}) [\sigma_{k-1/2} \Delta\sigma_j] - \begin{bmatrix} 0 & (k-1 < j) \\ \Delta\sigma_j & (k-1 \geq j) \end{bmatrix} \right\} \]

\[ + \kappa To_k \Delta\sigma_{kj} \]

4.0 Compute and invert matrix for solution of Helmholtz equation

4.1 Compute matrix B = G*TAU + R*T*PI (assumes G has been multiplied by R in DATINI)

B is the vertical structure matrix which gives the gravity wave speeds associated with the mean atmosphere used for the semi-implicit time differences

\[ ZB(J) = G*TAU + R*To*PI \]

4.3 Compute and invert matrix A(N)=(I+SQ*B*DELT**2)
Call MINV for matrix inversion.
This inverse, stored in BM1, is used to solve the semi-implicit equations.

HISTORY

26 OCT 82, D. Williamson, NCAR.
SIGMA

FUNCTION SIGMA(N,A,IA,B,IB)
*DECK UTIL.117

Computes the sum of N elements of A if there are three arguments or the scalar dot product of A and B over N elements if there are 5 arguments.

ARGUMENT LIST:

N : [in] number of elements in sum
A : [in] location of first point in the sum
IA: [in] increment between elements of A
B : [in] location of first point in the sum
IB: [in] increment between elements of B

ALGORITHM:

for three arguments

\[ \text{SIGMA} = \sum_{I=1}^{N} A(1+(I-1)*IA) \]

for five arguments

\[ \text{SIGMA} = \sum_{I=1}^{N} A(1+(I-1)*IA) \times B(1+(I-1)*IB) \]

HISTORY:

1 DEC 80, R.K. Sato, NCAR.
PROGRAM SPECTR
*DECK MAIN.2

Calls MASTER.

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE SPLITF
*DECK SPLITF.2

Splits the restart dataset on VSN=NSVSN2 into the three component datasets NRA1, NRB1 and NSDS.

HISTORY:

10 DEC 1982, L. Bath, NCAR.
SUBROUTINE STARTN  
*DECK STRTN.2

Starts the north-south grid scan for the physics and dynamics calculations.

ALGORITHM:

1.0 Initialize displacements of work buffers
1.5 Initialize Legendre polynomials buffer
2.0 Start read/calculate cycle
2.1 Read northern row
   NROW = 1
2.2 Begin read for second row
2.3 Calculation for northern row
   CALL LINEMS
   NROW = NROW + 1
   RETURN

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE STATS
*DECK STATIS.2

Computes and prints selected global average quantities.

ALGORITHMS:

global average quantities (variables without subscripts are functions of longitude(i), latitude(j), and sigma(k), and recall $\sum_{j=1}^{J} w_j = 2$)

$$\text{RMSZ} = \left\{ \frac{\left[ \sum_k \left[ \sum_j \sum_i (z^n)^2 w_j \right] \Delta q_k \right]}{(2I)} \right\}^{1/2}$$

$$\text{RMSD} = \left\{ \frac{\left[ \sum_k \left[ \sum_j \sum_i (d^n)^2 w_j \right] \Delta q_k \right]}{(2I)} \right\}^{1/2}$$

$$\text{RMST} = \left\{ \frac{\left[ \sum_k \left[ \sum_j \sum_i (T_0^n 2 I + 2 T_{0k} I \sum T^n + \sum T^n^2) w_j \right] \Delta q_k \right]}{(2I)} \right\}^{1/2}$$

$$\text{STPS} = \sum_j \left[ \sum_i \left( P^n_{sw_j} \right) \right] / (2I)$$

$$\text{STQ} = \left[ \sum_k \left[ \sum_j \sum_i \left( p^n_{qw_j} \right) \right] \Delta q_k \right] / (2gI)$$

$$\text{STPE} = \sum_k \left[ \sum_j \sum_i \left( Cp(T_0 + T^n) p^n_{sw_j} \right) \Delta q_k \right] / (2gI)$$

$$+ \sum_j \left[ \sum_i \left( p^n_{qsw_j} \right) \right] / (2gI)$$

$$\text{STKE} = \sum_k \left[ \sum_j \sum_i \left( p^n_{sw_j} \right) \Delta q_k \right] / (2gI)$$

where $E^n = \left[ \left( u^n \right)^2 + \left( v^n \right)^2 \right] / 2$
STTE = STPE + STKE

Horizontal average quantities

\[ VZ_{2k} = \left\{ \left[ \sum_{j=1}^{J} \left( \sum_{i=1}^{I} (e_{k}^{j})^2 w_j \right) \right]/(2I) \right\}^{1/2} \]

\[ D_{2k} = \left\{ \left[ \sum_{j=1}^{J} \left( \sum_{i=1}^{I} (e_{k}^{j})^2 w_j \right) \right]/(2I) \right\}^{1/2} \]

\[ T_{2k} = \left\{ \sum_{j=1}^{J} \left[ (T_{0k})^2 I + 2T_{0k} \sum_{i=1}^{I} (T_{k}^i)^n + \sum_{i=1}^{I} (T_{k}^i)^n \right]/(2I) \right\}^{1/2} \]

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE STEPON
*DECK STPN.2

Controls the time step loop.

ALGORITHM.

1. First Gaussian scan
1.2 Start timestep at first row
   CALL STARTN
1.3 First scan from north to south
   CALL CHECKR(NRB2)
   REWIND NRB2
   CALL SCAN1
   CALL CHECKR(NRB2)
   REWIND NRB2
   if end of run RETURN

2. Perform time extrapolation
   CALL TSTEP
2.15 Vertical diffusion in spectral space
   CALL VIRDIF
2.2 Linear horizontal diffusion
   CALL HORDIF
2.5 Recalculate semi-implicit matrices for next time step
   CALL SETTAU

3. Second Gaussian scan
   CALL CHECKR(NRA2)
   REWIND NRA2

3.1 Second scan from north to south
   NROW = 1
   CALL SCAN2
   CALL CHECKR(NRA2)
   REWIND NRA2
   REWIND NTPLEG
4.2 Increment timestep and swap work files B

NSTEP = NSTEP + 1
swap A and B disk pointers
NRA1 = NRA2
NRB1 = NRB2

4.3 Update start data set on unit NSDS for restart

Go to beginning of subroutine

HISTORY:

18 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE TCALC(ISCHAL)
*DECK   TCALC.2

    Adds or subtracts the average temperature $T_0^{-1}$ to or from the
temperature field in the BUFFER array at offset NTM1.

ARGUMENT:

    ISCHAL:  [in] = +1 add
              = -1 subtract

HISTORY:

    10 MAY 78, U. Cubasch, ECMWF.
SUBROUTINE TESEND
*DECK DUMMIES.2

Controls the termination of the run. Tests for normal termination.

ALGORITHM:

If NSTEP = NRUN, set NLEND = .TRUE.

HISTORY:

19 JAN 76, D.M. Burridge, ECMWF.
SUBROUTINE TIMCPU(LIMIT,NSES,MSECS)
*DECK DUMMIES.82

Returns default time limit values.

ARGUMENTS:

LIMIT: [out] = 999999
NSES : [out] = 999999
MSECS: [out] = 999999

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE TIMESM(KDISA,KDISB,KDISC,PA,PB,KNO)
*DECK TIMESM.2
Performs either step of the time filter by the general operation
\[ C = PA*A + PB*B \]
ARGUMENTS:

KDISA: [in] displacement in BLANK COMMON of first element of A
KDISB: [in] displacement in BLANK COMMON of first element of B
KDISC: [in] displacement in BLANK COMMON of first element of C
PA: [in] coefficient multiplying BUF(KDISA + J)
PB: [in] coefficient multiplying BUF(KDISB + J)
KNO: [in] number of elements to be operated on

HISTORY:

1 MAR 81, ECMWF.
SUBROUTINE TRUNC
*DECK DATIN.435

Computes offsets carried in /COMHKP/ and /COMTRU/ for spectral
coefficient arrays for either global or hemispheric case. (See
User's Guide for details.)

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE TSTEP
*DECK TSTEP.2

Solves the Helmholtz equation arising from the semi-implicit time differences and completes the time step.

ALGORITHM:

2. Complete RHS of Helmholtz equation

2.15 Diagonal terms

\[(DS_n^m)_k = (DS_n^m)_k + \Delta t \frac{n(n+1)}{2} \left[ RB_{kk} (TS_n^m)_k + RTO_k (PS_n^m) \right] \]

2.34 Nontriangular terms of G

\[(DS_n^m)_K = (DS_n^m)_K + \Delta t \frac{n(n+1)}{2} RB_{KK-1} (TS_n^m)_K \]

2.36 Triangular terms of G

\[(DS_n^m)_k = (DS_n^m)_k + \Delta t \frac{n(n+1)}{2} \sum_{l=k+1}^{K} RB_{kl} (TS_n^m)_l \]

3. Solution of Helmholtz equation

\[ZR_n^m = (A_n)^{-1} (DS_n^m) \]

3.7 Store solution in field RH

\[\delta_n^m = ZR_n^m \]

4.2 Add semi-implicit part to surface pressure

\[(z_np_s)_n^m = (PS)_n^m - \Delta t \sum_{k=1}^{K} (\delta_n^m)_{n,k} \Delta q_k \]

4.3 Add semi-implicit part to temperature

\[\bar{T}_n^m = TS_n^m - \Delta t \bar{\delta}_n^m \]

HISTORY:

17 AUG 78, M. Jarraud, ECMWF.
SUBROUTINE VIRDIF
*DECK VIRDIF.2

Computes linear vertical diffusion of $\zeta$, $\delta$, $T$ and $q$ in spectral space.

ALGORITHM:

1. Define diffusion coefficients

$$C(K+1) = \frac{|UVGA(K+1) - UVGA(K)|}{(TGA(K+1) - TGA(K))^3} \times C2(K+1)$$

$$A(K) = \frac{|UVGA(K+1) - UVGA(K)|}{(TGA(K+1) - TGA(K))^3} \times C1(K)$$

where

$$TGA(K) = \left[ \frac{n-1}{n} \right]$$

$$UVGA(K) = \text{SQRT} \left\{ \left[ \left( \frac{n-1}{n} \right)^2 + \left( \frac{n-1}{n} \right)^2 \right] \right\}$$

and the brackets $[ ]$ denote a global average.

$$C1(k) = \frac{2 \Delta t}{\Delta \sigma_k} \times g \left( \sigma_{k+1/2} \right)^2 \left( \frac{2 \sigma_{k+1/2}}{R} \right)^3 \times \left( \frac{1}{\sigma_{k+1} - \sigma_k} \right)^2$$

$$C2(k) = \frac{2 \Delta t}{\Delta \sigma_k} \times g \left( \sigma_{k-1/2} \right)^2 \left( \frac{2 \sigma_{k-1/2}}{R} \right)^3 \times \left( \frac{1}{\sigma_k - \sigma_{k-1}} \right)^2$$

$$B(KTOP) = 1 + A(KTOP)$$

$$B(K) = 1 + A(K) + C(K) \quad \text{KTOP} + 1 \leq K \leq \text{KBOT} - 1$$

$$B(KBOT) = 1 + C(KBOT)$$
1.5 Add mean to temperature

\[ T_{k_0}^0 = T_{k_0}^0 + \bar{T}_{n-1} \]

2.0 Vertical diffusion of Z, D, T, Q.

Following is done for each wavenumber.

\[
\begin{align*}
E(K_{TOP}) &= \frac{A(K)}{B(K) - C(K) \cdot E(K-1)} \\
ET(K) &= \frac{A(K) \cdot \sigma(K) / \sigma(K+1)^{\kappa}}{B(K) - C(K) \cdot \sigma(K) / \sigma(K+1)^{\kappa} \cdot ET(K-1)} \\
FD(K) &= \frac{\delta(K) + C(K) \cdot FD(K-1)}{B(K) - C(K) \cdot ET(K-1)} \\
FT(K) &= \frac{T(K) + C(K) \cdot \sigma(K) / \sigma(K-1)^{\kappa} \cdot FT(K-1)}{B(K) - C(K) \cdot \sigma(K) / \sigma(K-1)^{\kappa} \cdot ET(K-1)} \\
FQ(K) &= \frac{q(K) + C(K) \cdot FQ(K-1)}{B(K) - C(K) \cdot E(K-1)} \\
FZ(K) &= \frac{\zeta(K) + C(K) \cdot FZ(K-1)}{B(K) - C(K) \cdot E(K-1)}
\end{align*}
\]
\[ \delta(K) = E(K) \times \delta(K+1) + FD(K) \]
\[ T(K) = ET(K) \times T(K+1) + FT(K) \]
\[ q(K) = E(K) \times q(K+1) + FQ(K) \]
\[ c(K) = E(K) \times c(K+1) + FZ(K) \]

3.0 Subtract mean from temperature

\[ T_{k0}^o = T_{k0}^o - \bar{T}_{k-1}^n \]

HISTORY:

11 MAY 82, D. Williamson, NCAR.
SUBROUTINE WECOEF(PCOEF,KMBIG,KNBIG)
*DECK WECOEF.2

  Writes the spectral coefficients of an even field for $m \leq KMBIG$ and 
  $n \leq KNBIG$ for vertical levels 1 to NLEV.

ARGUMENTS:

  PCOEF: [in] even spectral coefficient array
  KMBIG: [in] upper limit for longitudinal wavenumber $m$
  KNBIG: [in] upper limit for the degree of the associated Legendre
         polynomial $n$

HISTORY:

  23 JAN 83, D. Williamson, NCAR.
SUBROUTINE WOCOEF(PCOEF,KMBIG,KNBIG)
*DECK WOCOEF.2

  Writes the spectral coefficients of an odd field for \( m \leq KMBIG \) and \( n \leq KNBIG \) for vertical levels 1 to NLEV.

ARGUMENTS:

  PCOEF: [in] odd spectral coefficient array
  KMBIG: [in] upper limit for longitudinal wavenumber \( m \)
  KNBIG: [in] upper limit for the degree of the associated Legendre polynomial \( n \)

HISTORY:

  23 JAN 83, D. Williamson, NCAR.
SUBROUTINE WRITBF(KUNIT,PA,KLEN)
*DECK WRITBF.2

    Initiates BUFFER OUT operation.

ARGUMENTS:

    KUNIT: [in] unit number of write
    PA:    [in] array to be written
    KLEN:  [in] number of words written

HISTORY:

    24 FEB 77, Jan Haseler, ECMWF.
SUBROUTINE WRITER(KDV, PA, KLEN, KREC)
*DECK WRITER.2

Initiates BUFFER OUT operation possibly preceded by REWIND.

ARGUMENTS:

KDV: [in] unit number of write
PA: [in] array to be written out
KLEN: [in] number of words written
KLEC: [in] if KREC = 1, REWIND KDV prior to BUFFER OUT

HISTORY:

8 MAY 78, A.P.M. Baede, ECMWF.
SUBROUTINE WRTHDR
*DECK WRTHDR.2

Updates and writes out on unit NDATA the header /COMHKP/. There are 4 time levels for the updating of header information, first, when the case begins, second when a job (start or restart) is submitted, third, at the start of a new history tape volume, and fourth, upon any entry to WRTHDR. WRITBF is used to write out the data.

Note: All variables being updated are in /COMHKP/.

ALGORITHM:

1.0 Store temporary variables used to save certain header fields.

2.0 This block executed when IFLGC = 1.
   Call FLDLST to set up header field list and return values for MAXSIZ and NFLDH. Then the following variables are updated.
   
   NFLDH
   MAXSIZ
   LENHD
   MFILTH
   NDAVU
   NSPHER
   MDT
   MHISF
   MCASE
   MCSTIT
   LNHSHTF
   LDHSTF
   LTHSTF
   LSHSTF

3.0 This block executed when IFLGR = 1 (the first pass through WRTHDR). Note: IFLGR is set = 1 in a data statement and set = 0 upon exit of WRTHDR.

   NSTPRH
   MFSTRT
   LSHSTC
4.0 This block executed when IFLGT = 1
These variables are updated for the first header on a new volume.

LNHSTC
LNHSTP
LDHSTP
LTHSTP
LSHSTP

5.0 These variables are changed for each header. NUDATE is called to update date information.

MFILLH
NSTEPH
NITSLF
NCDATA
NTDATA
NCDT
NCSC
LDHSTC
LTHSTC

6.0 Write out header.

HISTORY:

12 JAN 82, T. Mayer, NCAR.
SUBROUTINE WSHIST
*DECK WSHIST.2

    Writes data for current latitude line at time n-1 to history tape, using partial record BUFFER OUTs.

ALGORITHM.

    see USERS GUIDE for details of fields written to tape

HISTORY:

    21 DEC 78, U. Cubasch, ECMWF.
COMMON

*COMDECK BLANCOM1.2

Consists of blank common used to store grid point data. See Section III.C of the "User's Guide" for details of its structure. There are several versions of blank common given by different UPDATE COMDECKs.

VARIABLES:

ABCD(5)
BUF(PLNBUF)

EQUIVALENCE:

(BUFFER(1),B(1),BUF(1))

Note: The 5-word dummy array preceding the model buffer array BUF is used to reference elements in the model buffer array BUF indirectly (e.g., ABCDE(5+N)=BUF(N)) and was introduced as a means of circumventing the inhibition of vectorization due to possible vector dependencies.

HISTORY:

4 JULY 78, U. Cubasch, ECMWF.
COMMON
*COMDECK BLANCOM2.2

Consists of blank common used to store grid point data. See Section III.C of the "User's Guide" for details of its structure. There are several versions of blank common given by different UPDATE COMDECKs.

VARIABLES:

A(1)
B(1)
C(1)
D(1)
E(1)
BUF(PLNBUF)

Note: The five one word dummy arrays preceding the model buffer array BUF are used to reference elements in BUF indirectly (e.g., C(3+N)=BUF(N)) and were introduced as a means of circumventing the inhibition of vectorization due to possible vector dependencies.

HISTORY:

11 MAY 78, U. Cubasch, ECMWF.
COMMON
*COMDECK BLANCOM4.2

Consists of blank common used to store grid point data. See Section III.C of the "User's Guide" for details of its structure. There are several versions of blank common given by different UPDATE COMDECKs.

VARIABLES:

VA(1)  
VB(1)  
VC(1)  
VD(1)  
VE(1)  
BUFFER(PLNBUF)

EQUIVALENCE:

(BUFFER(1),B(1),BUF(1))

Note: The five one word dummy arrays preceding the model buffer array BUFFER are used to reference elements in BUFFER indirectly (e.g., VB(4+N)=BUFFER(N)) and were introduced as a means of circumventing the inhibition of vectorization due to possible vector dependencies.

HISTORY:

21 SEPT 78, M. Jarraud, ECMWF.
COMMON /COMBAS/
*COMDECK COMBAS.2

Contains constants related to the running of the model, and some unit numbers. These are set in BASIC, DATCOM, MODIFY and PRESET.

VARIABLES:

The following are set in BASIC:

ALTIME = 0.0
STIME = 0.0
CPTIME = 0.0
NSDS = 30
NDATA = 20
NSRE = 31
NRA1 = 11
NRB1 = 21
NRA2 = 12
NRB2 = 22
NLEND = .F.
NONLIN = 1
NPRINT = 6
NOUT = NPRINT
NREAD = 5
NREC = 1
NSTEP = 0
NPUNCH = 7
NDIARY = NPUNCH
NIN = NREAD
NRun = 1

LABEL1(J) - LABEL8(J): Set to BLANK. (Dimensioned as five words each).

In MODIFY:

NLRES = .T. if a restart or regeneration run is requested via NSREST
NLHST = .T. if a regeneration run is requested.

In PRESET:

NLPHYS = .T.
NWTIME = 0 (dimensioned 50)
NESTEP = -10

HISTORY:

1 MAR 81, ECMWF.
COMMON /COMBUD/
*COMDECK COMBUD.2

Contains selected global average quantities, accumulated conversion
terms, and associated constants.

VARIABLES:

The following are filled in DATINI.

NBUDGE = 2, number of iterations between calls to BUDGET. After 1 day NBUDGE is reset in LINMS
NCBUD (not used)
MODBUD (not used)
NTWO flag set in LINMS to control BUDGET call
KCUO (not used)
CPOT (not used)

The following are set in CONSTS.

CHAIR = \frac{C_p}{g*NLO} \times 10^{-3}
CKIN = \frac{1/2}{g*NLO} \times 10^{-3}
CWAVA = \frac{L}{g*NLO} \times 10^{-3}
CWSOIL = \rho_{H2O} \times \frac{L}{NLO} \times 10^{-3} (not used)
CSNOW = CWSOIL (not used)
CHSOIL = 0 (if .NOT. NLPHEC) (not used) (also set in CSURF)

The following are initialized in BUDGET (KCALL=1) and accumulated in the routine indicated.

EPOTO (not used)
EPOT (not used)
DPCVPH (not actively used)
DPCVPK (not actively used)
DPFILT (not actively used)
DPDEF (not actively used)
HAIRO (calculated in BUDGET)
HAIR (calculated in BUDGET)
DHCVHP (not actively used)
DHRAD (calculated in DTRAD)
DHCDST (calculated in COND)
DHCDCU (calculated in MADADJ)
DHFLUX (calculated in GFPBMQ)
DHFILT (not actively used)
DHDIFF (not actively used)
DHDEF (calculated in BUDGET)
KINO (real) (calculated in BUDGET)
KIN (real) (calculated in BUDGET)
DKCVKP (not actively used)
DKDISS (calculated in GFPBMQ)
DWDIFF (not actively used)
DKFILTER (not actively used)
DKDEF (calculated in BUDGET)
KINZ (real) (calculated in BUDGET)
KINZ (real) (calculated in BUDGET)
HWAVAO (calculated in BUDGET)
HWAVA (calculated in BUDGET)
DWCDST (calculated in COND)
DWCDCU (calculated in MADADJ)
DWFLUX (calculated in GFPBMQ)
DWDIFF (not actively used)
DWFILTER (not actively used)
DWMIXC (not actively used)
DWVVCU (not actively used)
DWFMCU (not actively used)
DWDDEF (calculated in BUDGET)
WSOIL (calculated in BUDGET)
WDAWIL (calculated in BUDGET)
DWSPPRE (calculated in COND, MADADJ)
DWSVEM (calculated in GFPBMQ)
DWRUNO (calculated in COND, MADADJ)
DWMELT (calculated in PB2GFD)
SNOWO (calculated in BUDGET)
SNOW  (calculated in BUDGET)
DSPREC (calculated in COND, MADADJ)
DSMELT (calculated in PB2GFD)
DSEVAP (calculated in GFPBMQ)
HSOilo (calculated in BUDGET)
HSOIL  (calculated in BUDGET)
DHSRAD (not actively used)
DHSFLA (not actively used)
DHSMEL (not actively used)
DHSFLG (not actively used)
DWFLXG (not actively used)
HSTO  (calculated in BUDGET)
HST   (calculated in BUDGET)
DHSTCD (calculated in COND)
DHSTAC (not actively used)
DHSTCC (not actively used)
DHSTEC (not actively used)
DHSTER (calculated in COND)
DHCUCD (calculated in MADADJ)
DHCUER (not actively used)

The following is filled in DATINI

COSBUD(J) = w(J)/2 , J=1,NOREC

HISTORY:

1 MAR 81, ECMWF.
COMMON /COMCOL/
*COMDECK  COMCOL.2

VARIABLES:

   Filled in MADADJ

   PK : $1/(akPs)$ , k=1,K
   PI : $ak+1/2Ps$ , k=1,K
   TK :
   TI :

   GAMMA: used to pass moist adiabatic lapse rate $\gamma_w$ for
   supersaturated layers between MADADJ and DOMADJ

HISTORY:

   1 MAR 81, ECMWF.
VARIABLES:
Filled in SUBROUTINE CONST.

\[
\begin{align*}
C_{1\text{MAD}}(J) & = \frac{\kappa \Delta \sigma(J-1/2)}{\sigma(J-1/2)} , \ JK=2,NLEV \\
C_{1\text{DAD}}(J) & = \frac{\kappa \Delta \sigma(J-1/2)}{2 \sigma(J-1/2)} , \ JK=2,NLEV \\
C_{2\text{DAD}}(J) & = \frac{1. - \kappa \Delta \sigma(J-1/2)}{\sigma(J-1/2)} , \ JK=2,NLEV \\
C_{3\text{DAD}}(J) & = \frac{\Delta \sigma(J)}{\Delta \sigma(J) \cdot C_{2\text{DAD}}(J+1) + \Delta \sigma(J+1)} , \ JK=1,NLEV-1 \\
C_{4\text{DAD}}(J) & = \frac{\Delta \sigma(J)}{\Delta \sigma(J-1) \cdot C_{2\text{DAD}}(J) + \Delta \sigma(J)} , \ JK=2,NLEV \\
\end{align*}
\]

HISTORY:
1 MAR 81, ECMWF.
COMMON /COMDDP/
*COMDECK COMDDP.2

Contains development and diagnostic parameters.
All are set in BASIC.

VARIABLES:

MAXDUM   = 20
MXDUMP   = 20
NADUMP(J) = 0, J=1,20
NCLASS   = 0
NPDUMP(J) = 0, J=1,20
NPOINT   = 0
NSUB     = 0
NVDUMP(J) = 0, J=1,20
NLCHED    = .F.
NLHEAD(J) = .F., J=1,9
NLOMT1(J) = .F., J=1,50
NLOMT2(J) = .F., J=1,100
NLOMT3(J) = .F., J=1,50
NLREPT    = .F.

HISTORY:

1 MAR 81, ECMWF
COMMON /COMEST/
*COMDECK  COMEST.2

Contains table of saturation vapor pressures for temperature at 1° intervals from 173.16°K to 375.16°K. Set in subroutine CONST.

VARIABLES:

TABLE = Table of values
TMIN  = 173.16°K
TMAX  = 375.16°K

HISTORY:

1 MAR 81, ECMWF
Converted to COMDECK with units changed 28 OCT 82, M. Kuhn, NCAR.
COMMON /COMEXP/
*COMDECK COMEXP.2

Contains information on switch settings, tape numbers and file numbers for current job (and for a possible restart or continuation job).

Completely initialized in subroutine MODIFY, by a NAMELIST READ from unit 5.

VARIABLES:

NSVSN (A6) : restart dataset name
NSREST (I10): 0 = initial run
               1 = restart run
               2 = regeneration run
NSBUFS (A2) : system buffer size for the work files and for the Legendre polynomials file
NSWRPS (A6) : write password (default is WRITE)
NSMVN (A6) : MSD dedicated reel name (default is blank)

HISTORY:

16 JAN 80, U. Cubash, ECMWF.
1 MAY 82, L. Bath, NCAR, modified for NCAR version.
COMMON /COMFFT/
*COMDECK  COMFFT.2

Contains trig tables, work space, and constants for FFT.

VARIABLES:

TRIG() = filled with call to FFTRIG
IFAX(10) = filled with call to FAX
WORK() = work space for FFT

following are set in DATINI

NTRIA = (4*NLEV+1)/NCRAY
NRSTIA = (4*NLEV+1)-NTRIA*NCRAY
NTRIB = (7*NLEV)/NCRAY
NRSTIB = (7*NLEV)-NTRIB*NCRAY
NTR2 = (6*NLEV+3)/NCRAY
NRST2 = (6*NLEV+3)-NTR2*NCRAY
NCRAY = PCRAY (=64)
NSYM = MAX(NSYM1+NSYM2,NSYM3)
NSYM1 = (4*NLEV+1)*NLONP2
NSYM2 = (7*NLEV)*NLONP2
NSYM3 = (6*NLEV+3)*NLONP2

HISTORY:

21 SEP 78, M. Jarraud, ECMWF
COMMON /COMGRD/
*COMDECK COMGRD.2

Contains pointers to the BUF array. The values of the pointers are defined in POINTS and illustrated in a figure in the "Users' Guide." The contents of BUF array associated with these pointers vary during the course of a time step and are not indicated here. The contents at various points in the integration cycle are specified via figures in the "Users' Guide."

HISTORY:

27 July 78, U. Cabasch, ECMWF.
10 July 82, R. Sato, NCAR.
COMMON /COMHD1/
*COMDECK COMHD1.2

Contains variables needed by WRTHDR to set header.

VARIABLES:

The following are set in DATA

MTITL
MCSTEM

The following are set in PRESET

MSPHTM
NDVUTM
MAXHDL
MFMTEM

HISTORY:

12 JAN 82, T. Mayer, NCAR.
COMMON /COMHKP/
*COMDECK COMHKP.2

Contains history tape header.

VARIABLES:
All are filled in WRTHDR except those provided on the initial tape header indicated by an * after the variable name.

<table>
<thead>
<tr>
<th>Var. Name</th>
<th>Word No.</th>
<th>Description</th>
<th>Var. Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENHD</td>
<td>1</td>
<td>Header length</td>
<td>I</td>
</tr>
<tr>
<td>MFTYP</td>
<td>2</td>
<td>Format code for this history tape</td>
<td>I</td>
</tr>
<tr>
<td>MFILH</td>
<td>3</td>
<td>&quot;Logical&quot; file no. relative to first file on this volume</td>
<td>I</td>
</tr>
<tr>
<td>MFILTH *</td>
<td>4</td>
<td>Maximum no. of files on a history tape volume</td>
<td>I</td>
</tr>
<tr>
<td>NRB *</td>
<td>5</td>
<td>No. of records before data records</td>
<td>I</td>
</tr>
<tr>
<td>MAXSIZ</td>
<td>6</td>
<td>Length of data record for this volume</td>
<td>I</td>
</tr>
<tr>
<td>NDAVU</td>
<td>7</td>
<td>No. of unpacked data values per record</td>
<td>I</td>
</tr>
<tr>
<td>NSPHER</td>
<td>8</td>
<td>Hor. domain flag (1) for global (2) for hemis.</td>
<td>I</td>
</tr>
<tr>
<td>NOLONP2 *</td>
<td>9</td>
<td>No. of long. data points including possible wraparound points</td>
<td>I</td>
</tr>
<tr>
<td>NOLON</td>
<td>10</td>
<td>No. of long. points less wraparound points</td>
<td>I</td>
</tr>
<tr>
<td>NOLONW *</td>
<td>11</td>
<td>No. of long. data values written on the history tape</td>
<td>I</td>
</tr>
<tr>
<td>NOREC</td>
<td>12</td>
<td>No. of lat. lines and no. of data records</td>
<td>I</td>
</tr>
<tr>
<td>NLEV *</td>
<td>13</td>
<td>No. of vertical levels</td>
<td>I</td>
</tr>
<tr>
<td>NTRM *</td>
<td>14</td>
<td>M spectral truncation parameter</td>
<td>I</td>
</tr>
<tr>
<td>NTRN *</td>
<td>15</td>
<td>N spectral truncation parameter</td>
<td>I</td>
</tr>
<tr>
<td>NTRK *</td>
<td>16</td>
<td>K spectral truncation parameter</td>
<td>I</td>
</tr>
<tr>
<td>NFLDH</td>
<td>17</td>
<td>No. of fields on the header</td>
<td>I</td>
</tr>
<tr>
<td>NSTEPH</td>
<td>18</td>
<td>Iteration no. of file being written out (NSTEP-1)</td>
<td>I</td>
</tr>
<tr>
<td>NSTPRH</td>
<td>19</td>
<td>Iteration no. for the start of this run</td>
<td>I</td>
</tr>
<tr>
<td>NITSLF</td>
<td>20</td>
<td>No. of iterations since last file was written</td>
<td>I</td>
</tr>
<tr>
<td>NCBASE *</td>
<td>21</td>
<td>Base day no. for this case</td>
<td>I</td>
</tr>
<tr>
<td>NTBASE *</td>
<td>22</td>
<td>Base no. of seconds for this case</td>
<td>I</td>
</tr>
<tr>
<td>NCDATA</td>
<td>23</td>
<td>Current day corresponding to NSTEPH</td>
<td>I</td>
</tr>
<tr>
<td>NTDATA</td>
<td>24</td>
<td>Current seconds corresponding to NSTEPH</td>
<td>I</td>
</tr>
<tr>
<td>NODT *</td>
<td>25</td>
<td>Base date yr mo day as 6 digit integer</td>
<td>I</td>
</tr>
<tr>
<td>NOSC *</td>
<td>26</td>
<td>Base date seconds to complete NODT date (Note: may be different than NTBASE)</td>
<td>I</td>
</tr>
<tr>
<td>NCDT</td>
<td>27</td>
<td>Current date yr mo day as 6 digit integer corresponding to NSTEPH</td>
<td>I</td>
</tr>
<tr>
<td>NCSC</td>
<td>28</td>
<td>Current seconds for date NCDT using NSTEPH</td>
<td>I</td>
</tr>
<tr>
<td>MDT</td>
<td>29</td>
<td>Model timestep</td>
<td>I</td>
</tr>
<tr>
<td>Var. Name</td>
<td>Word No.</td>
<td>Description</td>
<td>Var. Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>MHISF</td>
<td>30</td>
<td>Hist. volume output frequency in time steps.</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to -1 if NWTIME(1).NE.0</td>
<td></td>
</tr>
<tr>
<td>MFSTRT</td>
<td>31</td>
<td>Flag to indicate type of start</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0 for initial run</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 1 for restart</td>
<td></td>
</tr>
<tr>
<td>WEST</td>
<td>32 *</td>
<td>Initial long. point</td>
<td>R</td>
</tr>
<tr>
<td>MCASE</td>
<td>33</td>
<td>Case I.D.</td>
<td>C</td>
</tr>
<tr>
<td>MCSTIT</td>
<td>34-43</td>
<td>Case title</td>
<td>C</td>
</tr>
<tr>
<td>LNHSTC</td>
<td>44</td>
<td>VSN volume name this file writes to</td>
<td>C</td>
</tr>
<tr>
<td>LDHSTC</td>
<td>45</td>
<td>Date this header rec. was written MM/DD/YY</td>
<td>C</td>
</tr>
<tr>
<td>LTHSTC</td>
<td>46</td>
<td>Time this header rec. was written HH:MM:SS</td>
<td>C</td>
</tr>
<tr>
<td>LSHSTC</td>
<td>47</td>
<td>Sequence no. of run producing this file</td>
<td>C</td>
</tr>
<tr>
<td>LNHSTP</td>
<td>48</td>
<td>VSN volume name of previous hist. volume</td>
<td>C</td>
</tr>
<tr>
<td>LDHSTP</td>
<td>49</td>
<td>Date of last header record written on previous hist. vol. MM/DD/YY</td>
<td>C</td>
</tr>
<tr>
<td>LTHSTP</td>
<td>50</td>
<td>Time of last header record written on previous hist. vol. HH:MM:SS</td>
<td>C</td>
</tr>
<tr>
<td>LSHSTP</td>
<td>51</td>
<td>Sequence no. of run writing last hist. volume</td>
<td>C</td>
</tr>
<tr>
<td>LNHSTF</td>
<td>52</td>
<td>Vol. name of first hist. vol. for this case</td>
<td>C</td>
</tr>
<tr>
<td>LDHSTF</td>
<td>53</td>
<td>Date this case started MM/DD/YY</td>
<td>C</td>
</tr>
<tr>
<td>LTHSTF</td>
<td>54</td>
<td>Time this case started HH:MM:SS</td>
<td>C</td>
</tr>
<tr>
<td>LSHSTF</td>
<td>55</td>
<td>Sequence no. of run that started this case</td>
<td>C</td>
</tr>
<tr>
<td>LNHST1 *</td>
<td>56</td>
<td>VSN vol. name of initialized data</td>
<td>C</td>
</tr>
<tr>
<td>LDHST1 *</td>
<td>57</td>
<td>Date initialized data vol. was created MM/DD/YY</td>
<td>C</td>
</tr>
<tr>
<td>LTHST1 *</td>
<td>58</td>
<td>Time initialized data vol. created HH:MM:SS</td>
<td>C</td>
</tr>
<tr>
<td>LSHST1 *</td>
<td>59</td>
<td>Sequence no. of run creating initialized data</td>
<td>C</td>
</tr>
<tr>
<td>LNHSTA</td>
<td>60</td>
<td>VSN vol. name of analyzed data</td>
<td>C</td>
</tr>
<tr>
<td>LDHSTA</td>
<td>61</td>
<td>Date analyzed data vol. was created MM/DD/YY</td>
<td>C</td>
</tr>
<tr>
<td>LTHSTA</td>
<td>62</td>
<td>Time analyzed data vol. created HH:MM:SS</td>
<td>C</td>
</tr>
<tr>
<td>LSHSTA</td>
<td>63</td>
<td>Sequence no. of run creating analyzed data</td>
<td>C</td>
</tr>
<tr>
<td>MPSIG</td>
<td>64</td>
<td>Pointer to header list of sigma values</td>
<td>I</td>
</tr>
<tr>
<td>MPLAT</td>
<td>65</td>
<td>Pointer to header list of latitude lines</td>
<td>I</td>
</tr>
<tr>
<td>MPWTS</td>
<td>66</td>
<td>Pointer to list of Gaussian WTS</td>
<td>I</td>
</tr>
<tr>
<td>MPFLDS</td>
<td>67</td>
<td>Pointer to header field information list</td>
<td>I</td>
</tr>
</tbody>
</table>

The following lists start at the location indicated by the pointers just defined.

- **MPSIG**: List of model sigma values ordered from top to bottom (2*NLEV+1 values) ranging from 0-1
- **MPLAT**: List of model latitude lines
- **MPWTS**: Gaussian weights corresponding to Gaussian latitude
MPFLDS Field information list described by the following 2-dimensional array MFLDS(5,NFLDH)

-----

MFLDS(1,IF) - List of field names
MFLDS(2,IF) - Layer location flag and instantaneous or accumulated field flag. The one's digit is for the layer location flag, the ten's states whether the field is an instantaneous value or represents an accumulation since the last file was written. If the one's digit of MFLDS(IF,2) = 0 it is a single level field = 1 it is a multilevel field at half levels = 2 it is a multilevel field at full levels

If the ten's digit = 0 this is an instantaneous field = 1 this is an accumulated field

MFLDS(3,IF) - Pointer to first field value in data rec.
MFLDS(4,IF) - Data packing flag (=1 if not packed)
MFLDS(5,IF) - Units of this field (SI units)

See explanation below

Definition of SI units

The SI units are encoded in the following manner

A) Representation of each unit;

Each unit is defined by 1 to 3 capital letters with exponents indicated by the number following the letter, e.g.,

M  - meter
KG - kilogram
S  - second
K  - degree Kelvin
MDL - mole

N  - newton
PA - pascal
J  - Joule
W  - watt
M2 - meter squared
S2 - second squared

Dimensionless fields are 'FRACTION', 'FLAG' or 'INDEX'
B) Order of the units

Combination units come first. Within the standard units the ordering is M,KG,S,K,MOL.

C) Separation of numerator and denominator

All units in the numerator come first, followed by a slash, followed by all units in the denominator, e.g., M/S is meters per second.

HISTORY:

12 JAN 82, T. Mayer, NCAR.
COMMON /COMIMP/
*COMDECK  COMIMP.2

Contains constants associated with the semi-implicit time differences.

VARIABLES:

The following are initially set in INIDAT and reset in SETTAU

TO : To^0_k, k=1,NLEV
TOM : To^{n-1}_k, k=1,NLEV
BM1 : matrix (A_n)^{-1} used to solve coupled equation in semi-implicit time step.
TAU : matrix \tau in thermodynamic equation
AQ : (\Delta t)^2 (RB \tau + RTo'\pi)

STODS : not used
ST02DS: not used
DELTAT: not used

HISTORY:

10 JAN 79, M. Jarraud, ECMWF.
COMMON /COMIOC/
*COMDECK COMIOC.2

Contains trig tables, work space, and constants for FFT.

VARIABLES:

NLINE1(2) 
NLINE2(2) 
NLINE3(2) 
NLINE4(2)

 pointers to BUFFER array,
 see "Users' Guide" for details

NROW = current row index
NLNBUF = maximum size of BUFFER array (set in DATINI)
MAXROW = NOREC, number of Gaussian latitudes (set in TRUNC)
NORS = 2 (set in STRTN)
NSTART = first iteration number
NWRITE = iteration number for next history tape write
NTPLEG = unit number containing associated Legendre polynomials
 P^m = ALP and their derivatives H^m = DALP line by line
 fPom pole to equator (see Users' Guide)

NLEG(2) = BUFFER offset for Legendre polynomials set initially in
 STRTN

 NLEG(1) = 0
 NLEG(2) = 2*NSPT

then repeatedly flipped in GRCALC and LEG after the
 BUFFER IN of the polynomials is initiated

 NLEG(1) \rightarrow NLEG(2)

HISTORY:

12 SEP 78, M. Jarraud, ECMWF.
COMMON /COMISP/
*COMDECK COMISP.2

Contains pointers related to the internal data structures.

VARIABLES:

NPHYS = No. of 2-dimensional fields in PHYS1 + PHYS2 (29+6*NLEV)
LREC = (6*NLEV+2) * NLONP2
NBGRID = 2*NLONP2
NBPHYS = (3*NLEV+23) * NLONP2
NBFLNA = LREC + NBGRID
NBFLNB = LREC + NBPHYS
LDATA = LREC + NBGRID + NBPHYS
NBAUXL = (10*NLEV+1)*NLONP2 + 13*NLON

HISTORY:

12 JAN 82, T. Mayer, NCAR.
COMMON /COMLEG/
*DECK COMLEG.2

Contains Legendre polynomials and Gaussian constants for current Gaussian latitude.

VARIABLES:

\[
ALP = p^m_n
\]

\[
DALP = H^m_n
\]

\[
ADALP = p^m_n \text{ and } H^m_n
\]

In GRCALC and LEG either ALP and DALP or ADALP contains the current line values and the other is being filled with the next line values. Reference is always made to ALP and DALP using displacements NLEG(1) and NLEG(2). (See Users' Guide.)

Following are set in GRCALC and LINEMS

\[
CSJ = \cos^2 \phi_j; \text{ for current } \phi_j;
\]

\[
RCSJ = 1/\cos^2 \phi_j; \text{ for current } \phi_j;
\]

WEIGHT = Gaussian weight at \(\phi_j\); (\(\times 2\) for hemispheric case) set in LINEMS

HISTORY:

5 JUN 78, U. Cubasch, ECMWF.
COMMON /COMMAP/
*COMDECK COMMAP.2

Contains various physical constants, a values, constants associated with the Gaussian grid, and integration matrices for the hydrostatic equation and conversion term. The values are initialized in DATCOM, DATIN, GAUAW and CONST.

VARIABLES:

The following are set in DATCOM.

\[ AE = a \ (6.37 \times 10^6) \]
\[ RA = \frac{1}{a} \]
\[ GA = g \ (9.80665) \]
\[ WW = \Omega \ (7.292 \times 10^{-5}) \]
\[ EZ = \Omega / \sqrt{375} \]
\[ CP = C_p \ (1004.6) \]
\[ R = R \ (287.04) \]
\[ AKAP = \kappa = \frac{R}{C_p} \]
\[ RGA = \frac{1}{g} \]
\[ SIG(K) = \frac{\sigma K + 0.5}{k = 1, \text{NLEV in DATCOM}} \]
\[ SIGKPH(K) = \frac{\sigma K - 0.5}{k = 1, \text{NLEV+1, after CONST is called}} \]
\[ DSIGMA(K) = \Delta \sigma K \]
\[ R2DSIG(K) = \frac{1}{2 \Delta \sigma K} \]
\[ SQ(1) = 0. \]
\[ SQ(J) = \frac{J(J-1)}{a^2}, \ J=2,\text{NMAX} \]
\[ RSQ(1) = 0. \]
\[ RSQ(J) = \frac{1}{SQ(J)}, \ J=2,\text{NMAX} \]
\[ XM(J) = J - 1, \ J=1,\text{NMAX} \]
The following are set in GAUAW, defined from pole to equator.

\[
\text{SIT}(J) = \text{abscissa of Gaussian grid}
\]
\[
= \mu_J = \sin \phi_J, \quad J = 1, \text{NOREC}/2
\]

\[
\text{W}(J) = \text{weights for Gaussian grid}, \quad J = 1, \text{NOREC}/2
\]

\[
\text{CS}(J) = 1 - \mu_J^2 = \cos^2 \phi_J, \quad J = 1, \text{NOREC}/2
\]

\[
\text{RCS}(J) = \frac{1}{1 - \mu_J^2} = \frac{1}{\cos^2 \phi_J}, \quad J = 1, \text{NOREC}/2
\]

The following is set in DATCOM and multiplied by R in DATIN.

\[
\text{G} = \text{integration matrix for hydrostatic equation} = \text{RB}
\]

The following is set in DATCOM.

\[
\text{AG} = \text{integration matrix for conversion term} = \text{C}
\]

The following is set in CONST.

\[
\text{SIGK}(K) = \sigma_K, \quad K = 1, \text{NLEV}
\]

\[
\text{DIF} = 2.5E5, \text{set in PRESET (horizontal diffusion coefficient)}
\]

HISTORY:

12 SEP 78, M. Jarraud, ECMWF.
COMMON /COMPBL/
*COMDECK COMPBL.2

VARIABLES:

The following are filled in CONSTS.

\[
C1PB1(NLEV) = \frac{2 \Delta \text{tg}}{\Delta \alpha_{NLEV}} \times C_D \times \frac{\sigma_{NLEV}}{R}
\]

\[
C1PB1(k) = \frac{2 \Delta \text{tg}}{\Delta \alpha_k} \times \left( \frac{g \varphi_{k+1}}{\Delta \alpha_{k+1/2}} \right)^2 \times \left( \frac{2 \sigma_{k+1/2}}{R} \right)^3 \quad \text{for } k = \text{TOPFL}, NLEV-1
\]

\[
C2PB1(k) = \frac{2 \Delta \text{tg}}{\Delta \alpha_k} \times \left( \frac{g \varphi_{k}}{\Delta \alpha_{k-1/2}} \right)^2 \times \left( \frac{2 \sigma_{k-1/2}}{R} \right)^3 \quad \text{for } k = \text{TOPFL}, NLEV
\]

\[
\text{NTOPFL} = \text{First index above 500 mb (upper limit for vertical diffusion)}
\]

\[
\text{PBZKST( ) = ?}
\]

\[
\text{PBZIST}(k) = \text{height (Z) of level assuming const. lapse rate atmosphere (ICOA)}
\]

\[
\text{ZLE}(k) = 30.0 \quad k = \text{TOPFL+1}, NLEV
\]

(WSSAT is input in $NEWRUN$ with default value .15)

\[
\text{QWSSAT} = \frac{1}{(.75 \times \text{WSSAT})}
\]

\[
\text{ZDC} = C_D = 2.E-3 \text{ (drag coef)}
\]

\[
C1PB2 = \frac{\sigma_{NLEV} \times C_p \times \text{ZDC}}{R}
\]

\[
C2PB2 = \frac{\sigma_{NLEV} \times L \times \text{ZDC}}{R}
\]
C3PB2 = \frac{1}{(\sigma_k^NLEV)}

C4PB2 = \frac{2 \Delta t g (C_p \Delta \sigma^NLEV)}{Cp AoNLEV}

CONDUC = (conductivity of sea ice, not used)

HISTORY:
1 MAR 81, ECMWF.
COMMON /COMPHO/
*COMDECK COMPHO.2

All variables filled in CONST except as indicated.

\[ DSIGI(JK) = \Delta \sigma(JK-1/2) = \sigma(JK) - \sigma(JK-1) \quad ; \quad JK = 2,\text{NLEV} \]
\[ DSIGI(1) = \sigma(1) \]
\[ DSIGI(NLEV+1) = 1.-\sigma(NLEV) \]

\[ SIGKPK(JK) = \sigma(JK)^* ; \quad JK = 1,\text{NLEV} \]
\[ SIGKPK(NLEV+1) = 1. \]
\[ SIGKPK(1) = 0. \]

\[ SIGIPK(JK) = \sigma(JK-1/2)^* ; \quad JK = 1,\text{NLEV} \]

\[ DSIGDG(JK) = \Delta \sigma(JK)/g = \frac{\sigma(JK+1/2) - \sigma(JK-1/2)}{g} \quad ; \quad JK = 1,\text{NLEV} \]

\[ CDNLO = 2./\text{NLON} \]
\[ RTCDO = \text{not used} \]
\[ NMAXUE = (\text{NLONP2}-4)/4 \quad (\text{set in PRESET and DATINI}) \]

\[ C1SIG(JK) = \frac{1.-\sigma(JK+1/2)}{1.-\sigma(JK-1/2)} \quad ; \quad JK = 1,\text{NLEV} \]

\[ C2SIG(JK) = \frac{\Delta \sigma(JK)}{1.-\sigma(JK-1/2)} = \frac{\sigma(JK+1/2) - \sigma(JK-1/2)}{1.-\sigma(JK-1/2)} \quad ; \quad JK = 1,\text{NLEV} \]

**HISTORY:**

1 MAR 81, ECMWF.
COMMON /COMPH1/
*COMDECK COMPH1.2

VARIABLES:

Filled in SUBROUTINE CONST.

REARTH = 6.371E6
OMEGA = 7.292E-5
TILTAX = 23. + 40./60.
DAYLEN = 86400.
GRAVIT = 9.80665
SOLAR = 1.353E3
RAIR = 2.8704E2
RH20 = 4.6151E2
CALUNI = 4.1868
CPAIR = 1.0046E3
CH20 = 4.1868E3
RHOH20 = 1.3
EPSIL0 = 0.622
CAPP = RAIR/CPAIR
TMELT = 273.16
LATVAP = 2.5003E6
LATURE = 3.336E5
STEBOL = 5.67E-8
PZERO = 1.65
CARMAN = 0.35
CLDCP = LATVAP/CPAIR
CLUDR = LATVAP/RAIR
CLUDPH = not used
QCRIT = 1.
CLRH20 = LATVAP/RH20
QLATVA = 1./LATVAP
C1MELT = TWODT/(LATVAP*RH20)
ESMELT = 379.90516
C1ES = 379.90516
C2ES = 17.2693882
C3ES = 273.16
C4ES = 35.86
C5ES = 379.90516*QCRIT
C6ES = C5ES*CLUDR*2.0
C7ES = C5ES*C9ES*4.
C8ES = C5ES*C2ES*(C3ES-C4ES)
C9ES = CLDP+C2ES*(C3ES-C4ES)
C1ESS = QCRIT*ε
C2ESS = 2*C1ESS*L/R
\[ C3ESS = \frac{4 \cdot C2ESS \cdot L^2}{(R_{H2O} \cdot C_p)} \]

\[ CIPREC(JK) = \frac{DSIGK(JK)}{RHOH2O \cdot \text{GRAVIT} \cdot \rho_{H2O}} = \frac{\Delta \sigma(JK)}{\rho_{H2O}} \quad JK=1,NLEV \]

\[ ESCRIT(JK) = .8, \ JK=1, \ NLEV \]

HISTORY:

1 MAR 81, ECMWF.
COMMON /COMRTS/
*COMDECK    COMPRTS.2

Contains variables controlling printouts and other aspects of the model, as indicated below:

VARIABLES:

   NFSTAT  Frequency of statistics printout, read in as part of NAMELIST $NEWRUN.
   NLSTAT  Logical variable controlling whether or not Subr. STATS is executed.
   NLPRCO  Logical variables controlling printout of Budget variables from Subr. LINEMS.
   NFPRRA  Not used.
   NNUMWT  Frequency of history tape write-up. In NAMELIST $NEWRUN.

HISTORY:

   13 JAN 83, L. Bath, NCAR.
COMMON /COMSDS/
*COMDECK 'COMSDS.2

Contains variables needed for restart.

VARIABLES:

NSTEPR  Time step number used to begin a restart run, set in STEPON to new NSTEP.

TWODT  Two times the time step in seconds, set in DATA from DTIME.

EPS  Time filter coefficient, set in PRESET to .06, optionally read in through NAMELIST $NEWRUN.

NSTOP  Time step number to stop model. Computed from NESTEP in NAMELIST $NEWRUN.

NWPTN  Pointer into the NWTIME array which controls history tape write times in manual mode. Read in through NAMELIST $NEWRUN.

HISTORY:

13 JAN 83, L. Bath, NCAR.
COMMON /COMSPE/
*COMDECK COMSPE.2

Contains spectral coefficients of \( \zeta, \delta, T, q, \ln P_s \).

VARIABLES:

- **VZ(PSPOL):** vorticity \( (\zeta) \) coefficients
- **D(PSPEL):** divergence \( (\delta) \) coefficients
- **T(PSPEL):** temperature \( (T) \) coefficients
- **Q(PSPEL):** mixing ratio \( (q) \) coefficients
- **ALPS(PSPE):** \( \ln \) surface pressure \( (\ln P_s) \) coefficients
- **RH:** (unused)

See Users' Guide for a description of the ordering of the coefficients.

HISTORY:

1 MAR 81, ECMWF.
COMMON /COMSTA/
*COMDECK COMSTA.2

        Contains statistics computed in subroutine STATS

VARIABLES:

        See subroutine STATS writeup for description of variables.
COMMON /COMTAP/
*COMDECK COMTAP.2

VARIABLES:

LABDAT        Dataset name (VSN) for initial dataset.
LTAPES(100)   Array of VSNs for history tapes.
NTAPES        Set in DATA, number of LTAPES read in.
NXTAP         Index to current history tape VSN in LTAPES, set in
               PRESET to 1.
MFILT          Set in PRESET to 30, max number of files on hist.
               tape.
MFILS          Set in PRESET to 10, number of files to write to
               hist. tape between DISPOSES.
IFIL           Set in PRESET to 0, current hist. tape file number.

LABDAT, LTAPES, MFILT, MFILS and IFIL are read in Subroutine
DATA via NAMELIST $NEWRUN.

HISTORY:

6 JUN 82, R. Sato, NCAR.
13 JAN 83, L. Bath, NCAR.
*COMMON COMTRU.

*COMDECK COMTRU.2

Contains parameters associated with the linear arrays of associated Legendre polynomials at a given latitude line and with the linear arrays of spectral coefficients. The use of these parameters is described in the Users' Guide.

VARIABLES:

The following are defined in DATCOM

NMAX = K+1
MMAX = M+1

The following are defined in TRUNC

\[
NEMAX = \begin{cases} 
N + 1 & \text{global} \\
\frac{N}{2} + 1 & \text{hemispheric}
\end{cases}
\]

\[
NEMAX = \begin{cases} 
N + 1 & \text{global} \\
\frac{N}{2} + 1 & \text{hemispheric}
\end{cases}
\]

NOMAX = number of diagonals for odd fields

\[
NOMAX = \begin{cases} 
N + 1 & \text{global} \\
\frac{N}{2} + 1 & \text{hemispheric}
\end{cases}
\]

NSPE = number of complex coefficients for even fields

\[
NSPE = \text{NEVEN}(NEMAX + 1) - 1 \quad \text{(see below)}
\]

NSPO = number of complex coefficients for odd fields

\[
NSPO = \text{NODD}(NOMAX + 1) - 1 \quad \text{(see below)}
\]

NSPT = number of associated Legendre polynomials

\[
NSPT = \begin{cases} 
\text{NSPE} & \text{global} \\
\text{NSPE} + \text{NSPO} & \text{hemispheric}
\end{cases}
\]

NEVEN(PEMAXP) = index of first point of Ith diagonal for even fields

\[
\begin{align*}
\text{NEVEN}(1) &= 1 \\
\text{NEVEN}(I+1) &= \text{NEVEN}(I) + \begin{cases} 
\min(M+1, K+2-I) & \text{global} \\
\min(M+1, K+3-2I) & \text{hemispheric}
\end{cases} \quad I=1, NEMAX
\end{align*}
\]
NODD(POMAX) = index of first point on Ith diagonal for odd fields

NODD(1) = 1

\[ \text{NODD}(I+1) = \text{NODD}(I) + \begin{cases} 
\min(M+1, K+2-I) & \text{global} \\
\min(M+1, K+2-2I) & \text{hemispheric} 
\end{cases} \quad I=1,\text{NOMAX} \]

NALPE(PEMAX) = pointer to first point before the diagonal for associated Legendre polynomials associated with the Ith diagonal of even fields

\[ \text{NALPE}(I) = \begin{cases} 
\text{NEVEN}(I) - 1 & \text{global} \\
\text{NEVEN}(I) + \text{NODD}(I) - 2 & \text{hemispheric} 
\end{cases} \quad I=1,\text{NEMAX} \]

NALPO(POMAX) = pointer to first point before the diagonal for associated Legendre polynomials associated with Ith diagonal of odd fields

\[ \text{NALPO}(I) = \begin{cases} 
\text{NODD}(I) - 1 & \text{global} \\
\text{NEVEN}(I+1) + \text{NODD}(I) - 2 & \text{hemispheric} 
\end{cases} \quad I=1,\text{NOMAX} \]

NMEV(PEMAX) = length of the Ith diagonal for even fields

\[ \text{NMEV}(I) = \text{NEVEN}(I+1) - \text{NEVEN}(I), \quad I=1,\text{NEMAX} \]

NMOD(POMAX) = length of the Ith diagonal for odd fields

\[ \text{NMOD}(I) = \text{NODD}(I+1) - \text{NODD}(I), \quad I=1,\text{NOMAX} \]

NEV2(PEMAX)

\[ \text{NEV2}(I) = \text{NEVEN}(I) \times 2, \quad I=1,\text{NEMAX} \]

NOD2(POMAX)

\[ \text{NOD2}(I) = \text{NODD}(I) \times 2, \quad I=1,\text{NOMAX} \]

HISTORY:

18 AUG 78, A. Jarraud, ECMWF.
COMMON /COMVDF/
*COMDECK COMVDF.2

Contains global averages accumulated by GLBAVG.

VARIABLES:

After completion of last latitude line

\[ TGA(K) = \sum_{j=1}^{NOREC} \sum_{j=1}^{NLON} T_{n-1}^{-1}(\lambda_i, \phi_j) w_j / (2*NLON) \]

\[ UVGA(K) = \left\{ \sum_{j=1}^{NOREC} \sum_{j=1}^{NLON} \left[ (u_{K,n-1}(\lambda_i, \phi_j))^2 + (v_{K,n-1}(\lambda_i, \phi_j))^2 \right] w_j / (2*NLON) \right\} \]

HISTORY:

5 MAY 82, R. Sato, NCAR.
COMMON /COMZER/
*COMDECK  COMZERO.2

VARIABLES:
Following are set in CONSTS
CZERO(JL) : =0 for JL=1,NLON
CONE(JL) : =1 for JL=1,NLON

Following are initialized when used
TEmplN(PLONP2): a temporary vector of length NLONP2
NL : a temporary logical variable
NLOGIC(PLOGIC): a temporary logical vector

HISTORY:
20 OCT 77, J.K. Gibson, ECMWF.