The Global Atmospheric Tracer Model TM3

Model Description and User’s Manual
Release 3.8a

Martin Heimann and Stefan Körner

Max-Planck-Institut für Biogeochemie, Jena
Technical Reports – Max-Planck-Institut für Biogeochemie 5, 2003

Editor of this report:

Stefan Körner
Max-Planck-Institut für Biogeochemie
Postfach 10 01 64
07701 Jena
Germany
Tel.: +49 (0)3641 576 352
E-mail: Stefan.Koerner@bgc-jena.mpg.de
http://www.bgc-jena.mpg.de

Following authors from the Max Planck Institute for Biochemistry contributed to this manual release of June 11, 2003:

Martin Heimann (original program author)
Stefan Körner (System Description, User’s Manual, transport diagnostics)
Ina Tegen (many hints and suggestions, proofreading)
Martin Werner (many hints and suggestions, proofreading)

Title picture: William Turner, "Rain, Steam and Speed" (1844)
# Contents

## Introduction

## 1 Model Description

1.1 Meteorological Input Data ............................ 6
1.2 Model Physics ................................... 6
   1.2.1 Continuity Equation ........................... 6
   1.2.2 Advection .................................. 7
   1.2.3 Vertical Convection ........................... 9
   1.2.4 Horizontal Diffusion ........................... 14

## 2 System Description

2.1 Overview ..................................... 15
2.2 Coordinate Systems and Time Steps ...................... 16
2.3 Program Modules and Source Code Files .................... 17
2.4 Global Variables and Constants ........................ 17
2.5 The Model Main Loop ................................ 18
2.6 Time and Calendar Calculations ........................ 18

## 3 User’s Manual

3.1 Building and Running TM3 - Overview .................... 20
   3.1.1 Building a TM3 Executable ........................ 20
   3.1.2 Running TM3 .................................. 21
3.2 Module userconstants ................................ 22
3.3 Module userroutines ............................... 23
   3.3.1 Example 1: Volume Source ........................ 24
   3.3.2 Example 2: Sink Proportional to Tracer Concentration ....... 24
   3.3.3 Example 3: Surface Source ....................... 25
3.4 Program Inputs .................................. 29
   3.4.1 Runtime Parameter File tm3.in ..................... 29
   3.4.2 Meteorological Input Files and their Presumed Organization .... 30
   3.4.3 Other Standard Input Files ........................ 31
3.5 Program Outputs ................................ 33
   3.5.1 Output to the Standard Output Device ............... 33
   3.5.2 Tracer-Related Main Output ........................ 34
   3.5.3 Diagnostic Output ................................ 36
   3.5.4 Other Output .................................. 37
## CONTENTS

3.6 Programming Hints .................................................. 38
   3.6.1 Useful Auxiliary Subroutines ................................. 38
   3.6.2 Multiprocessing with TM3 ....................................... 38
   3.6.3 Job Chains .......................................................... 40
   3.6.4 Jobscript Example ............................................... 41
   3.6.5 Migration Guidelines for Prior TM3 Releases .................. 43
   3.6.6 CPU expense ....................................................... 43

A Meteorological Data Preprocessing ................................. 44
   A.1 Airmass Fluxes ...................................................... 44
      A.1.1 Interpolation/Integration of the Horizontal Mass Fluxes ...... 44
      A.1.2 Ensuring Mass Conservation ..................................... 44
      A.1.3 Airmass Flux processing Programs .............................. 45
   A.2 Sub-Gridscale Transport Information ............................ 45
      A.2.1 Vertical Diffusion Coefficients ................................ 46
      A.2.2 Cloud Transport Parameter Determination ........................ 47
      A.2.3 Programs .......................................................... 50

B Horizontal Grid Layouts .............................................. 51

C Vertical Coordinate Tables ........................................... 61

D Global Program Variables ............................................... 65

E Runtime Parameters ..................................................... 71

F Diagnostic Results ..................................................... 75
   F.1 The Effects of Varying Grid Resolutions ........................... 75
      F.1.1 Surface Point Emissions ........................................ 75
      F.1.2 Radon ............................................................. 109
      F.1.3 Sulfur Hexafluoride SF6 ........................................ 119
   F.2 Regional Transport: The ATMES-II Modelling Exercise ............ 124
      F.2.1 Conclusions ........................................................ 127

G Getting the Code .......................................................... 130
Introduction

The global atmospheric tracer model TM3 is a three-dimensional transport model that solves the continuity equation for an arbitrary number of atmospheric tracers. It uses meteorological output fields from an atmospheric general circulation model or from a weather forecast model.

The tracer advection is calculated using the ”slopes scheme” of Russell and Lerner [Russel and Lerner, 1981]. Vertical transport due to convective clouds is computed using a cloud mass flux scheme by Tiedke [Tiedke, 1989]. Turbulent vertical transport is calculated by stability dependent vertical diffusion according to Louis [Louis, 1979].

The model is coded in Fortran 95 and available as source code. For a quick start see section 3.1 on page 20.
Chapter 1

Model Description

1.1 Meteorological Input Data

TM3 solves the continuity equation based on given time-dependent two- or three-dimensional meteorological fields of the surface pressure, wind velocity, air temperature, specific humidity and geopotential. Additionally, evaporation fluxes are needed in order to calculate the sub-gridscale transport by cumulus clouds (see section 1.2.3 and appendix A.2).

These forcing fields may be obtained either from meteorological analyses or from the output of an atmospheric general circulation model. For a realistic representation of synoptic time scale transport processes the meteorological fields have to be available with a timestep of 12 hours or shorter.

1.2 Model Physics

The model equations of TM3 are formulated in Eulerian coordinates. For their derivation in this section a Cartesian coordinate system \( \{e_x, e_y, e_z\} \) is used for simplicity. The actual model coordinate system is displayed in section 2.2.

1.2.1 Continuity Equation

The atmospheric tracer model TM3 solves the continuity equation for a tracer in flux form

\[
\frac{\partial}{\partial t} \rho \chi + \nabla \cdot (\rho \vec{u} \chi) = Q
\]  

(1.1)

on an Eulerian three-dimensional grid spanning the entire globe. \( \chi \) denotes the tracer mixing ratio (in kg tracer mass per kg air mass), \( \rho \) the air density, \( \vec{u} \) the wind velocity and \( Q \) the volume source/sink of the tracer. Denoting time-space averages over the model grid elements and model timestep by overbars and deviations from these averages with primes, we obtain the equation for the averaged quantities:

\[
\frac{\partial}{\partial t} \bar{\rho} \bar{\chi} + \nabla \cdot (\bar{\rho} \bar{\vec{u}} \bar{\chi}) + \nabla \cdot (\bar{\rho} \bar{\vec{w}} \bar{\chi}') = \bar{Q}
\]  

(1.2)

where we have neglected density variations within the averaging volumes. Separating the horizontal and vertical directions we get

\[
\frac{\partial}{\partial t} \bar{\rho} \bar{\chi} + \nabla_h \cdot (\bar{\rho} \bar{\vec{u}}_h \bar{\chi}) + \frac{\partial}{\partial z} (\bar{\rho} \bar{\vec{w}} \bar{\chi}') + \frac{\partial}{\partial z} (\bar{\rho} \bar{\vec{w}}' \bar{\chi}') = \bar{Q}
\]  

(1.3)
where $w$ is the vertical wind velocity and the subscript $h$ refers to horizontal vectors.

The second and third term on the left hand side of equation (1.3) correspond to transport scales that are resolved on the model grid and are termed “advection” (see section 1.2.2). The fourth and fifth term represent the effect of the unresolved scales on the volume averaged tracer mixing ratio. Assuming that no cross correlation exists between vertical and horizontal deviations from the volume averaged quantities we treat these terms independently. In the troposphere, for which the model is primarily designed, this assumption is justified, but it may not be appropriate in the stratosphere. The fourth term, called “horizontal diffusion”, represents changes induced by correlations between the horizontal deviations of wind and tracer concentration from their grid averages (see section 1.2.4 on page 14). The last term on the left hand side of equation (1.3) represents sub-gridscale transport in the vertical direction and is termed “vertical convection”. It is parameterized in the model by transport induced by cumulus clouds and by turbulent, stability dependent vertical diffusion (see section 1.2.3 on page 9).

The model numerically solves the continuity equation (1.3) in flux form and splits the transport operator in time. Formally, we may view the tracer masses in all the gridboxes as the components of a large vector, $n(t)$. The discrete version of equation (1.3) is then written as

$$n(t + \Delta t) = L_{vconvec} L_{hdiff} L_{advec} (n(t) + Q(t) \Delta t)$$

(1.4)

where the discrete operators $L_{advec}$, $L_{hdiff}$ and $L_{vconvec}$ corresponding to the transport processes as described above, operate sequentially on $n$. Because the advection operator is represented in an explicit form, the length of the base time step, $\Delta t$, is limited by the CFL criterion\(^1\). $Q(t)\Delta t$ is the vector of tracer mass increments resulting from source-sink processes which is added to the tracer vector during $\Delta t$.

### 1.2.2 Advection

Advection, i.e. tracer transport by the three-dimensional airmass fluxes resolved on the model grid, is calculated using the “slopes scheme” developed by Russell and Lerner [Russell and Lerner, 1981]. In this scheme each tracer is represented within the Eulerian gridboxes by the tracer mixing ratio, $\chi$, and the components of the spatial gradient of the tracer mixing ratio within the gridbox, $(\frac{\partial \chi}{\partial x}, \frac{\partial \chi}{\partial y}, \frac{\partial \chi}{\partial z})$.

For program efficiency the primary variables in the model are not the mixing ratios but the tracer masses, $n$, and the “slopes” of the tracer mass $(n_x, n_y, n_z)$ in the gridboxes. Assuming no variations of the air density within a particular gridbox the tracer mass is given by

$$n = \chi m = \chi \rho \Delta x \Delta y \Delta z,$$

(1.5)

where $m$ denotes the airmass in the gridbox. For efficiency of the program the tracer mass “slope” in the x-direction is defined as

$$n_x = m \frac{\partial \chi}{\partial x} \frac{\Delta x}{2}$$

(1.6)

and analogously in the y- and z-direction.

\(^1\)The size of the airmass fluxes determines the maximum permissible value of $\Delta t$: The total amount of air moved out of any gridbox during $\Delta t$ must not exceed the airmass present in the gridbox at the beginning of the timestep (Courant-Friedrichs-Levi criterion).
With the additional information on slopes, the scheme exhibits much less numerical diffusion than e.g. a simple upstream formulation. This advantage is however obtained at the expense of storage and computational resources since four arrays are required for the representation of each simulated tracer.

For certain tracers, negative concentrations are unacceptable. In regions with steep concentration gradients the calculated tracer mass slopes may become so large that negative tracer masses emerge. If this is deemed unacceptable, this behaviour may be inhibited by limiting the absolute size of the tracer mass slopes (logical input variable limits). This, however, effectively implies a larger numerical diffusion of the advection scheme, but in contrast to some other schemes the transport remains linear.

The slopes scheme requires the airmasses in each gridbox as input data together with the airmass fluxes, i.e. the amount of air crossing the borders of the gridboxes during each time step. For a meteorological dataset (meteorological analyses or climate model output), the fields of airmasses and airmass fluxes are computed on a TM3 model grid during a preprocessing stage (see page 44), stored on disk and then read during tracer model runs.

The three-dimensional advection process is subdivided into one-dimensional transport in longitudinal, meridional and vertical direction. Because of the different typical sizes of the mass fluxes in the three directions the one-dimensional advection steps of length $\Delta t$ are further subdivided and performed sequentially in a so-called spatial leap-frog pattern:

1. zonal advection lasting $\frac{1}{4}\Delta t$,
2. meridional advection lasting $\frac{1}{2}\Delta t$,
3. zonal advection lasting $\frac{1}{4}\Delta t$,
4. vertical advection lasting $\Delta t$,
5. zonal advection lasting $\frac{1}{4}\Delta t$,
6. meridional advection lasting $\frac{1}{2}\Delta t$,
7. zonal advection lasting $\frac{1}{4}\Delta t$

With this particular setup the effective advective time step in the longitudinal direction is $\frac{1}{4}$ and in the meridional direction $\frac{1}{2}$ of the base time step $\Delta t$.

Close to the polar regions, where the zonal extent of the gridboxes becomes small, the zonal advection step is further subdivided in order to meet the CFL criterion. The grid-dependent widths of these bands around south and north poles are hardwired as $\text{lat248}$ in a Fortran source file. It has great influence on the achievable length of the integration timesteps.

During a one-dimensional advection time step, the slopes scheme performs the following operations for a particular grid box (the mathematical formulae are given in Russell and Lerner [Russel and Lerner, 1981]):

1. Calculate the new airmass in the gridbox based on the old airmass and the airmass fluxes crossing the borders.
2. Calculate the amount of tracer that crosses the borders of the gridbox, based on the masses and slopes of the tracer in the gridbox and its adjacent neighbours.
3. Calculate the new tracer mass in the gridbox.

4. Calculate the new slopes of the tracer in the gridbox. The new slope in the transport direction under consideration is calculated by a “hardwired” least squares fit to the displaced tracer mass distribution. The new slopes in the two other spatial directions are calculated using an upstream formula.

### 1.2.3 Vertical Convection

Sub-gridscale vertical transport is parameterized in the model by two processes: vertical diffusion and cumulus cloud transport.

The vertical diffusion coefficients are calculated based on the stability of the air using the formulae given by [Louis, 1979] (listed in appendix A.2).

Tracer transport by sub-gridscale cumulus clouds is calculated using the massflux scheme of Tiedke [Tiedke, 1989]. Formally this scheme defines a statistical stationary cloud containing an updraft and a downdraft in each vertical grid column. The magnitude, entrainment and detrainment rates of these up- and downdrafts are computed based on the horizontal below-cloud divergence of moisture, and on the buoyancy of the in-cloud air relative to the air outside of the cloud. Tracer mass is entrained into the up- and downdraft, mixed with the air inside the up- and downdraft and detrained into the surrounding air. Furthermore, the net vertical airmass flux of the up- and downdrafts induces a sub-gridscale subsidence flux of surrounding air outside the cloud. Figure 1.1 shows a schematic sketch of this sub-gridscale transport parametrization.

![Diagram of vertical sub-gridscale processes](image)

Figure 1.1: Vertical subgridscale processes (“vertical convection”), computed in TM3
CHAPTER 1. MODEL DESCRIPTION

In a preprocessing stage, both the vertical diffusion coefficients and the parameters for the cumulus cloud transport are computed from the background meteorological fields (see appendix A.2). In a tracer model simulation those coefficients and parameters are read from disk at each meteorological time step $\Delta t_{\text{met}}$. One consequence of this setup is that the cloud transport processes operate during the entire meteorological time step $\Delta t_{\text{met}}$. Formally, the statistical cloud extends through all grid levels of the model. In practice the cloud extends only through the levels as computed in the cloud model, with the appropriate en- and detrainment fluxes of the other layers set to zero.

Discretization

Since the tracer transport is linear, the effect of the sub-gridscale transport processes on the tracer masses on the different model levels during one base time step $\Delta t$ can be represented simply by the multiplication of the vector of tracer masses in a vertical grid column by a “convection matrix” $\mathbf{C}$. The element $[\mathbf{C}]_{lk}$ of this matrix represents the fraction of the tracer mass of the gridbox at level $k$, which ends up in the gridbox at level $l$ after the “convection process”. The elements of the convection matrix are determined from the discretized representation of the subgrid-scale transport processes. They are recomputed each time the diffusion coefficients and cumulus cloud transport parameters (en- and detrainment rates of the up- and downdraft) are read from disk.

Considering only the vertical convection process alone, we have to discretize and solve the following equation:

$$\frac{\partial}{\partial t} \bar{\rho} \bar{\chi} = - \frac{\partial}{\partial z} \bar{\rho} \chi' w'$$

(1.7)

during the timestep $\Delta t$. The right hand side of equation (1.7) is represented by the divergence of the sub-gridscale tracer fluxes $F_{\text{sub}}$ (defined positive in the upward direction):

$$\frac{\partial}{\partial t} \bar{\rho} \bar{\chi} = - \frac{\partial}{\partial z} F_{\text{sub}} = - \frac{\partial}{\partial z} (F_u + F_d + F_s + F_{\text{vdiff}})$$

(1.8)

where $F_u$ and $F_d$ denote the tracer mass flux in the up- and downdraft, respectively, $F_s$ the tracer mass flux by subsidence and $F_{\text{vdiff}}$ the tracer mass fluxes by turbulent vertical diffusion.

Integrating equation 1.8 over a gridbox at height level $l$ gives

$$\frac{d}{dt} n_l = (F_{\text{sub},l-\frac{1}{2}} - F_{\text{sub},l+\frac{1}{2}})$$

(1.9)

where $n_l$ denotes the tracer mass in the box. The sub-gridscale fluxes $F_{\text{sub},l+\frac{1}{2}}$ at the lower, $(l-\frac{1}{2})$, and upper, $(l+\frac{1}{2})$, border of the box can be represented as a linear function of the grid average tracer mixing ratios throughout the vertical column and hence of the tracer masses in each of the boxes in the vertical grid column:

$$F_{\text{sub},l+\frac{1}{2}} = \sum_{k=1}^{l_m} f_{l+\frac{1}{2},k} n_k.$$  \hspace{1cm} (1.10)

The coefficient $f_{l+\frac{1}{2},k}$ thus represents the fraction of the tracer mass of layer $k$, which crosses the layer boundary $l + \frac{1}{2}$ by means of sub-gridscale vertical transport processes.
1.2. MODEL PHYSICS

per unit time. It is further split into components representing the tracer mass fractions crossing the layer boundary due to each individual sub-gridscale vertical transport process:

\[ f_{l+\frac{1}{2},k} = f_{l+\frac{1}{2},k}^u + f_{l+\frac{1}{2},k}^d + f_{l+\frac{1}{2},k}^s + f_{l+\frac{1}{2},k}^{\text{diff}}. \] (1.11)

Explicit forms of \( f_{l+\frac{1}{2},k}^x \) for each of the different vertical sub-gridscale processes are derived in the following subsections.

Equation (1.9) thus can be written for the entire column in matrix form:

\[ \frac{d}{dt} n = M \cdot n, \] (1.12)

where the elements of matrix \( M \) are given as

\[ [M]_{l,k} = f_{l-\frac{1}{2},k} - f_{l+\frac{1}{2},k}. \] (1.13)

For reasons of stability equation (1.12) is integrated in implicit form:

\[ n(t + \Delta t) = n(t) + \Delta t M \cdot n(t + \Delta t). \] (1.14)

The tracer masses after the convection step are therefore related to the tracer masses before the convection step by

\[ n(t + \Delta t) = (I - \Delta t M)^{-1} \cdot n(t) = C \cdot n(t), \] (1.15)

where \( I \) denotes the identity matrix. \( C \) is the convection matrix.

Sub-gridscale transport also affects the slopes of the tracer mass within the gridboxes. The horizontal tracer mass slopes \( n_x \) and \( n_y \) are transported like the tracer mass, i.e. they are updated by

\[ n_x(t + \Delta t) = C \cdot n_x(t) \] (1.16)

and

\[ n_y(t + \Delta t) = C \cdot n_y(t). \] (1.17)

The vertical slopes are treated differently. It is assumed that the sub-gridscale processes tend to reduce the tracer mass slopes in the vertical direction. In the present version the vertical slopes are reduced to the fraction of tracer mass that remains at a particular level. These fractions are the diagonal elements of the convection matrix, hence for level \( l \) we have

\[ n_{z,l}(t + \Delta t) = [C]_{l,l} \cdot n_{z,l}(t). \] (1.18)

Transport by cumulus clouds

Sub-gridscale transport induced by cumulus clouds is represented by the three components

\[ F_u + F_d + F_s = M_u \chi_u + M_d \chi_d - (M_u + M_d) \bar{\chi} \] (1.19)

corresponding to the tracer flux in the updraft (subscript \( u \)), in the downdraft (subscript \( d \)) and induced by subsidence (subscript \( s \)). \( M_u \) denotes the updraft airmass flux, \( M_d \) the downdraft and \( -(M_u + M_d) \) the mass flux by subsidence, all expressed in kg m\(^{-2}\) s\(^{-1}\).

\(^2\)In this section the symbol \( n \) refers to the vector of tracer masses of a particular grid column only.
CHAPTER 1. MODEL DESCRIPTION

Entrainment and detrainment rates (expressed in kg m\(^{-3}\) s\(^{-1}\)) into the updraft \((E_u(z), D_u(z))\) and into the downdraft \((E_d(z), D_d(z))\) are given as functions of height, \(z\), from the physical cloud model (see appendix A.2). Updraft and downdraft mass fluxes and their in-cloud tracer mixing ratios are obtained by solving the discrete version of the following set of equations for the updraft:

\[
\frac{\partial}{\partial z} M_u = E_u - D_u \tag{1.20}
\]

\[
\frac{\partial}{\partial z} F_u = E_u \bar{\chi} - D_u \chi_u \tag{1.21}
\]

and for the downdraft:

\[
\frac{\partial}{\partial z} M_d = E_d - D_d \tag{1.22}
\]

\[
\frac{\partial}{\partial z} F_d = E_d \bar{\chi} - D_d \chi_d \tag{1.23}
\]

where we have denoted the fluxes of tracer in the up- and downdraft by

\[
F_u = M_u \chi_u \tag{1.24}
\]

\[
F_d = M_d \chi_d \tag{1.25}
\]

The appropriate boundary conditions are

\[
M_u = F_u = 0 \text{ at } z = z_{\text{surf}}
\]

\[
M_d = F_d = 0 \text{ at } z = z_{\text{top}} \tag{1.26}
\]

where \(z_{\text{surf}}\) and \(z_{\text{top}}\) are the geopotential height of the bottom and the top of the grid column, respectively.

The continuity of mass also requires that

\[
\int_{z_{\text{surf}}}^{z_{\text{top}}} (E_u - D_u)dz = 0 \tag{1.27}
\]

\[
\int_{z_{\text{surf}}}^{z_{\text{top}}} (E_d - D_d)dz = 0 \tag{1.28}
\]

Using 1.24 and 1.25, the equations for the updraft and downdraft tracer fluxes (1.21) and (1.23) can be expressed as

\[
\frac{\partial}{\partial z} F_u = E_u \bar{\chi} - D_u \frac{F_u}{M_u} \tag{1.29}
\]

\[
\frac{\partial}{\partial z} F_d = E_d \bar{\chi} - D_d \frac{F_d}{M_d} \tag{1.30}
\]

Updraft

The discretized form of the equation for the updraft airmass flux, (1.20), is obtained by integrating over a gridbox at height level \(l\)

\[
M_{u,l,\frac{1}{2}} - M_{u,l-\frac{1}{2}} = E_{u,l} - D_{u,l} \tag{1.31}
\]
where \( E_u,l \) and \( D_u,l \) denote the airmass entrainment and detrainment, respectively (in units of \( \text{kg s}^{-1} \)), and \( M_{u,l+\frac{1}{2}} \) denotes the updraft airmass flux at the model layer boundary \( z_{l+\frac{1}{2}} \) (in units of \( \text{kg s}^{-1} \)). Similarly the discrete version of the equation for the updraft tracer mass flux, \( (1.29) \), becomes

\[
F_{u,l+\frac{1}{2}} - F_{u,l-\frac{1}{2}} = E_{u,l} \bar{\chi}_l - D_{u,l} F_{u,l+\frac{1}{2}} M_{u,l+\frac{1}{2}} \tag{1.32}
\]

where we have used an implicit formulation on the right hand side, for stability reasons. Solving for \( F_{u,l+\frac{1}{2}} \) and using \( (1.31) \) we obtain

\[
F_{u,l+\frac{1}{2}} = (F_{u,l-\frac{1}{2}} + E_{u,l} \frac{n_l}{m_l})(1 - \frac{D_{u,l}}{M_{u,l-\frac{1}{2}} + E_{u,l}}) \tag{1.33}
\]

where we have substituted the grid averaged tracer mixing ratio \( \bar{\chi} \) by the ratio of tracer mass \( n_l \) divided the airmass \( m_l \) in the gridbox.

Equations \( (1.31) \) and \( (1.33) \) represent recursive relations that may be solved by starting from the surface using the boundary conditions \( (1.26) \) and working upwards. It is easily seen that, using this procedure, \( F_{u,l+\frac{1}{2}} \) at any layer boundary is a linear function of the tracer masses \( n_l, l = 1, \ldots, l_m \) of all the gridboxes in the grid column under consideration, in the form of equation \( (1.10) \).

Explicitly, the coefficients \( f_{u,l+\frac{1}{2},k} \) are determined by the recursion formula:

\[
f_{u,l+\frac{1}{2},k}^u = 0 \quad \forall \; k = 1, \ldots, l_m
\]

\[
f_{l+\frac{1}{2},k}^u = (f_{l-\frac{1}{2},k}^u + \delta_{k,l} \frac{E_{u,l}}{m_l})(1 - \frac{D_{u,l}}{M_{u,l-\frac{1}{2}} + E_{u,l}}) \tag{1.34}
\]

where

\[
\delta_{k,l} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases}
\tag{1.35}
\]

**Downdraft**

The derivation of the coefficients \( f_{l+\frac{1}{2},k}^d \) for the downdraft tracer fluxes are derived analogous to the updraft as given in the previous section. We obtain the recursion formula

\[
f_{l+\frac{1}{2},k}^d = 0 \quad \forall \; k = 1, \ldots, l_m
\]

\[
f_{l-\frac{1}{2},k}^d = (f_{l+\frac{1}{2},k}^d - \delta_{k,l} \frac{E_{d,l}}{m_l})(1 + \frac{D_{d,l}}{M_{d,l+\frac{1}{2}} - E_{d,l}}) \tag{1.36}
\]

**Subsidence**

The sub-gridscale flux induced by the subsidence air mass flux \(-(M_u + M_d)\) outside of the cloud is given by

\[
F_s = -(M_u + M_d) \bar{\chi} \tag{1.37}
\]
CHAPTER 1. MODEL DESCRIPTION

Using an upstream formulation this flux is written at the model layer boundaries as

\[ F_{s,l+\frac{1}{2}} = -(M_{u,l+\frac{1}{2}} + M_{d,l+\frac{1}{2}}) \frac{n_{l+1}}{m_{l+1}} \]  

(1.38)

Hence the coefficients \( f_{l+\frac{1}{2},k}^s \) result in

\[ f_{l+\frac{1}{2},k}^s = -\frac{M_{u,l+\frac{1}{2}} + M_{d,l+\frac{1}{2}}}{m_{l+1}} \delta_{l+1,k} \]  

(1.39)

**Vertical diffusion**

The diffusive flux through the upper boundary of a gridbox at height level \( l \) is

\[ F_{\text{vdiff}} = -A\rho K \frac{\partial \bar{\chi}}{\partial z} \]  

(1.40)

where \( A \) denotes the horizontal area of the gridbox, \( \rho \) the air density and \( K \) the diffusion index. Rewritten in discretized form as functions of the tracer and air masses in the adjacent boxes this becomes

\[ F_{\text{vdiff},l+\frac{1}{2}} = -A \frac{\rho_{l+\frac{1}{2}}}{z_{l+1} - z_l} K_{l+\frac{1}{2}} \frac{n_{l+1}}{m_{l+1}} \left( \frac{n_l}{m_l} - \frac{n_{l+1}}{m_{l+1}} \right) \]  

(1.41)

and the coefficients \( f_{l+\frac{1}{2},k}^{\text{vdiff}} \) are thus given by

\[ f_{l+\frac{1}{2},k}^{\text{vdiff}} = -A \frac{\rho_{l+\frac{1}{2}}}{z_{l+1} - z_l} K_{l+\frac{1}{2}} \frac{\delta_{l+1,k}}{m_{l+1}} \left( \frac{\delta_{l,k}}{m_l} - \frac{\delta_{l+1,k}}{m_{l+1}} \right) \]  

(1.42)

**Global tuning of convection parametrization**

For testing purposes the parametrization of vertical convection can be “tuned” by scaling the ent- and detrainment fluxes with a global scaling parameter \( \zeta_{\text{cu}} \) and the vertical diffusion coefficients with a global scaling parameter \( \zeta_K \).

### 1.2.4 Horizontal Diffusion

Simulation experiments with earlier versions of the transport model (TM1, see Heimann and Keeling [Heimann and Keeling, 1989]) indicated that the interhemispheric transport on annual average was too weak, depending on the meteorological fields used to drive the model, compared to that inferred from other tracer studies (notably \( ^{85} \text{Kr} \) and \( ^{11} \text{F} \)). Based on this observation, a horizontal diffusion term was included in the model in a way similar to the parametrization developed by Prather et al. [Prather et al., 1987]. The strength of this additional term was controlled by a global parameter (a length scale) which provided a tool to “fine tune” the model’s interhemispheric exchange.

With the use of meteorological datasets from the analyses after 1985 the interhemispheric transport became stronger (most probably because of more realistic analyzed fields in the tropical regions). In the present version of TM3 the code for the horizontal diffusion has been removed.
Chapter 2

System Description

2.1 Overview

The TM3 model system consists of two components: The preprocessing programs MFLUX, SUBSCALE and PASCHA, which generate the airmass fluxes and sub-gridscale transport information on the tracer model grid from meteorological fields, and TM3, the actual tracer model code itself. In practice the steps MFLUX, SUBSCALE and PASCHA are performed only once for a particular meteorological dataset and transport model geometry configuration. Figure 2.1 shows the TM3 information flow.

![Figure 2.1: TM3 model information flow](image)

The preprocessing modules are described in appendix A.1 and A.2. The following descrip-
CHAPTER 2. SYSTEM DESCRIPTION

Section focuses on the code of the TM3 tracer transport model itself.

2.2 Coordinate Systems and Time Steps

TM3 uses an equidistant latitude-longitude grid, with $i_m$ boxes in the zonal, $j_m$ boxes in the meridional and $l_m$ layers in the vertical dimension. Currently there exist five grid versions:

<table>
<thead>
<tr>
<th>notation</th>
<th>identifier</th>
<th>$i_m$</th>
<th>$j_m$</th>
<th>$l_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>coarse</td>
<td>cg</td>
<td>36</td>
<td>24</td>
<td>9</td>
</tr>
<tr>
<td>fine</td>
<td>fg</td>
<td>72</td>
<td>48</td>
<td>19</td>
</tr>
<tr>
<td>very fine</td>
<td>vfg</td>
<td>192</td>
<td>96</td>
<td>28/31</td>
</tr>
<tr>
<td>extra fine</td>
<td>xfg</td>
<td>320</td>
<td>161</td>
<td>31</td>
</tr>
</tbody>
</table>

In the zonal direction the longitudes $\phi_i$ of the grid box centers are located at

$$\phi_i = -180^\circ + (i - 1)\Delta x \quad i = 1, \ldots, i_m$$

with a grid spacing of $\Delta x = 360^\circ / i_m$.

In the meridional direction the $j_m$ grid boxes have an extent of $\Delta y = 180^\circ / (j_m - 1)$. They are spaced such that the latitudes $\Theta_j$ of the grid box centers are located at

$$\Theta_j = -90^\circ + (j - 1)\Delta y \quad j = 1, \ldots, j_m$$

The first and last (indices $j = 1$ and $j = j_m$) grid box are centered on the poles itself and have a meridional extent of $\Delta y / 2$. These polar boxes are not divided in the zonal direction and are referenced with the zonal index $i = 1$. For indices $j = 1$ and $j = j_m$ the elements with the zonal indices $i > 1$ are undefined.

The horizontal TM3 grid layouts are shown in appendix B on page 51.

In the vertical dimension TM3 uses a hybrid coordinate defined by two real coefficients $a$ and $b$ per level $l$ and the pressure equation

$$p_l = a_l + b_l \cdot p_s , \quad l = 0, \ldots, l_m,$$

where $p_s$ is the surface pressure. Depending on the vertical coordinate system of the underlying meteorological forcing fields, these coefficients can be chosen so that the vertical coordinate is purely $\sigma$-like ($\sigma = p/p_{surf}$) near the ground and purely pressure-like at the top of the atmosphere. This is the case for ERA/ECMWF meteorological fields. NCEP/NCAR fields have pure $\sigma$-coordinates, which are represented in TM3’s hybrid coordinate system as special case with $a_l = 0$.

The vertical box boundaries in hybrid coordinates and in meters above ground are tabulated in appendix C on page 61.

Transport is calculated using a base time step $\Delta t$ as defined in section 1.2.1. The availability of the meteorological data defines a second time step, $\Delta t_{met}$, which must be a multiple of the base time step $\Delta t$. 

16
2.3 Program Modules and Source Code Files

A standard TM3 code archive comes with the following directories and contents:

- **doc**: contains this manual, a change log and a commented template of the control parameter input file `tm3.in`.
- **examples**: contains simple example user files.
- **src**: contains the TM3 Fortran sources.
- **shared**: contains source files shared with the preprocessing programs.

The root directory contains a Makefile, which the user can copy to his work directory and adjust to his needs.

Generally, the files were given the name of the contained Fortran module, if possible. These modules were named, in turn, according to their function. The individual files and their purposes are tabulated in the following:

<table>
<thead>
<tr>
<th>File</th>
<th>Purpose</th>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>advection.f90</code></td>
<td>advection module</td>
<td></td>
</tr>
<tr>
<td><code>aux_flux.f90</code></td>
<td>handling of surface source fluxes</td>
<td></td>
</tr>
<tr>
<td><code>aux_grid.f90</code></td>
<td>auxiliary routines for grid calculations</td>
<td></td>
</tr>
<tr>
<td><code>convection.f90</code></td>
<td>convection module</td>
<td></td>
</tr>
<tr>
<td><code>diagnostics.f90</code></td>
<td>diagnostic routines (air and tracers)</td>
<td></td>
</tr>
<tr>
<td><code>griddefs_&lt;metID&gt;_&lt;gridID&gt;.f90</code></td>
<td>defines constants related to grid and meteorology</td>
<td></td>
</tr>
<tr>
<td><code>mass_conservation.f90</code></td>
<td>mass conservation check</td>
<td></td>
</tr>
<tr>
<td><code>time_and_aktiv.f90</code></td>
<td>date/time/calendar calculations</td>
<td></td>
</tr>
<tr>
<td><code>tm3global.f90</code></td>
<td>main global TM3 constants and variables</td>
<td></td>
</tr>
<tr>
<td><code>tm3io_flux.f90</code></td>
<td>I/O of surface flux files (unformatted)</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3io_flux_netCDF.f90</code></td>
<td>I/O of surface flux files (netCDF)</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3io_mix_netCDF.f90</code></td>
<td>I/O of instantaneous tracer mixing ratios (netCDF)</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3io_mix_unformatted.f90</code></td>
<td>output of instantaneous tracer mixing ratios (unformatted)</td>
<td></td>
</tr>
<tr>
<td><code>tm3io_mixstn_ascii.f90</code></td>
<td>output of mixing ratios at stations</td>
<td></td>
</tr>
<tr>
<td><code>tm3io_mixstn_unformatted.f90</code></td>
<td>output of mixing ratios at stations</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3io_prepro.f90</code></td>
<td>input of preprocessed driver fields</td>
<td></td>
</tr>
<tr>
<td><code>tm3io_schedule.f90</code></td>
<td>output of mean mixing ratios at station locations</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3io_mmass_unformatted.f90</code></td>
<td>output of instantaneous tracer masses</td>
<td></td>
</tr>
<tr>
<td><code>tm3main.f90</code></td>
<td>main loop and misc.</td>
<td>yes</td>
</tr>
<tr>
<td><code>tm3netCDF.f90</code></td>
<td>TM3 netCDF interface</td>
<td>yes</td>
</tr>
</tbody>
</table>

2.4 Global Variables and Constants

The principal physical variables are defined in one single Fortran module `TM3global`, contained in source file `tm3global.f90`. All physical subroutines operate on these variables, as well as the user-specified code for sources, sinks and/or chemistry. For details see the listing in appendix D on page 65, or the documented module `src/tm3global.f90` itself.
CHAPTER 2. SYSTEM DESCRIPTION

The runtime parameters of TM3 are combined in a Fortran namelist inputz, which is also declared in module TM3global. At program start, TM3 tries to read a text file tm3.in from the current directory, where these parameters have to be set according to the Fortran namelist input rules. Defaults apply for variables not specified there. A documented template can be found under doc/tm3.in, which is listed in appendix E on page 71. More details are explained in section 3.4.1 on page 29.

2.5 The Model Main Loop

The file tm3main.f90 contains, among others, the main program and the subroutine main-loop with the principal model loop. The main loop is entered after initialization actions, and after it is exited, shut-down operations are executed. The sequence of performed tasks is listed in Table 2.1.

Each task is controlled by variables contained in the Fortran namelist inputz. These control parameters specify e.g. the time steps for the particular task. Since the model simulation time is divided in increments of the base advection time step ndyn, the values of each of these time steps must be set to a multiple of ndyn. This is checked during program startup.

2.6 Time and Calendar Calculations

The basic time unit of the model is the second. The basic time step is defined by variable ndyn which describes the length of an advection timestep. The counter itau contains the current simulation time in seconds since 0:00:00 UT, January 1, iyear0. Functions are provided to convert simulation time instants expressed in seconds to an easily readable 6-element integer vector (/year, month, day, hour, minute, second/), or to a character string.

Four different calendar options are available, which can be selected by the control parameter icalendo:

1. Permanent 360 day years with 12 months of 30 days each.
2. Calendar with 365/366 day years (including correct leap years).
3. Permanent 365 day year.
4. Permanent 366 day leap year.

The default is 2. The choice of the calendar kind depends on the meteorological data to be used: (1) is appropriate with climate model output which uses 360 days per year, (2) for simulations using multiyear meteorological analyses, (3) and (4) for simulations over several years but by cycling repeatedly through the meteorology of a particular year.
### 2.6. *TIME AND CALENDAR CALCULATIONS*

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Timing by Variable</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>-</td>
<td>TM3 initialization</td>
</tr>
<tr>
<td>trace0</td>
<td>-</td>
<td>user-specific initializations</td>
</tr>
<tr>
<td>trace1</td>
<td>-</td>
<td>user-specific initializations (if istart = 10+i)</td>
</tr>
</tbody>
</table>

**begin of main loop**

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Timing by Variable</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_prepro</td>
<td>nread</td>
<td>reads meteorological fields from storage medium</td>
</tr>
<tr>
<td>init_advection</td>
<td>nread</td>
<td>initializes advection module</td>
</tr>
<tr>
<td>init_convection</td>
<td>nread</td>
<td>initializes convection module</td>
</tr>
<tr>
<td>interp_prepro</td>
<td>ndyn</td>
<td>interpolates some of the input fields in time</td>
</tr>
<tr>
<td>chem1</td>
<td>nchem</td>
<td>updates tracer fields $\chi$ by chemistry process 1</td>
</tr>
<tr>
<td>chem2</td>
<td>nchem</td>
<td>updates tracer fields $\chi$ by chemistry process 2</td>
</tr>
<tr>
<td>source1</td>
<td>nsrce</td>
<td>updates tracer fields $\chi$ by source process 1</td>
</tr>
<tr>
<td>source2</td>
<td>nsrce</td>
<td>updates tracer fields $\chi$ by source process 2</td>
</tr>
<tr>
<td>advection_step</td>
<td>ndyn</td>
<td>calculates horizontal advection</td>
</tr>
<tr>
<td>hdiff</td>
<td>ndiff</td>
<td>calculates horizontal diffusion</td>
</tr>
<tr>
<td>convec_step</td>
<td>nconv</td>
<td>calculates vertical mixing by convection and vertical diffusion</td>
</tr>
<tr>
<td>acc_diag</td>
<td>ndiag</td>
<td>accumulates diagnostics</td>
</tr>
<tr>
<td>inctime</td>
<td>ndyn</td>
<td>increments model time by one basic time step ndyn and updates calendar</td>
</tr>
<tr>
<td>write_cnsrv</td>
<td>ndiagp1</td>
<td>writes out mass conservation check</td>
</tr>
<tr>
<td>write_diag</td>
<td>ndiagp2</td>
<td>writes out mean field diagnostics</td>
</tr>
<tr>
<td>writc</td>
<td>ncheck</td>
<td>writes out tracer mixing ratio $\bar{\chi}$ at checkpoint locations</td>
</tr>
<tr>
<td>write_mix</td>
<td>ninst</td>
<td>writes out instantaneous tracer mixing ratio field $\bar{\chi}$</td>
</tr>
<tr>
<td>write_mixstn</td>
<td>ninststn</td>
<td>writes out instantaneous tracer mixing ratio $\bar{\chi}$ at station locations</td>
</tr>
<tr>
<td>write_tmass</td>
<td>ninsttmass</td>
<td>writes out instantaneous tracer mass field $\bar{\chi} \cdot m$</td>
</tr>
</tbody>
</table>

**end of main loop**

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Timing by Variable</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>tracee</td>
<td>-</td>
<td>user-specific shut-down</td>
</tr>
<tr>
<td>finish</td>
<td>-</td>
<td>TM3 shut-down</td>
</tr>
</tbody>
</table>

Table 2.1: Calling Sequence in TM3
Chapter 3

User’s Manual

The TM3 program code is written with the endeavour to follow the Fortran 95 standard\(^1\) as much as possible. The only known portability issues concern the "make" process using GNU make and the use of the "/" or "\" character as filename separators. The program has been successfully built and run "as is" on numerous UNIX-, MS-Windows- and Apple Macintosh OS-X platforms.

Since release 3.6a, the TM3 program features symmetric multiprocessing, which is especially useful for higher resolutions. This option is selected by a special make target (type make without any target for a complete list) and OpenMP environment variables. When many processors are used, the performance might depend strongly on whether the user’s routines are parallelized efficiently or not. Multiprocessing with TM3 is explained in more detail in sec. 3.6.2 on p. 38.

If preprocessed files are transferred from one platform to another, the user has to take care of matching integer and real data representations (width, IEEE or not, big or little endian), as well as the sequential record structure, which is not part of the Fortran 95 standard.

All setup and run actions can take place in an arbitrary working directory. The link to the TM3 code is made by specifying its path in the Makefile.

3.1 Building and Running TM3 - Overview

For the following, a working Fortran 95 compiler and the GNU make utility are prerequisite.

3.1.1 Building a TM3 Executable

The user needs three files in a working directory: Two user-supplied Fortran files and a Makefile.

Step 1: Setting up the user Fortran source files

For building an executable, TM3 needs two user-specified Fortran source files, user-constants.f90 and userroutines.f90. In the former, the user has to define only the precision of certain TM3 variables and the number of tracers \texttt{ntrace}. In the latter

\(^1\)Language constructs defined in the Fortran 95 Technical Reports (TR) are not used.
reside the user codes for sources/sinks/chemistry, together with user variables. Examples are given in the examples directory. The preparation of these files is described below (sec. 3.2 and 3.3).

**Step 2: Setting up the Makefile**

A Makefile template resides in the root directory of the TM3 tar archive. It must be copied to the working directory. Adjust the commented Configuration Section inside the Makefile copy according to your needs.

**Step 3: Running the GNU make command make or gmake**

After adjusting the Makefile, type ”make <target>” (or ”gmake <target>”) on the command line, where <target> is the kind of executable you want to build. A list of possible targets (optimized/debugging, single processor/multiprocessor etc.) is displayed when typing ”make” (or ”gmake”) without target. Normally, the make process produces an executable named tm3.

### 3.1.2 Running TM3

Standard prerequisites for running TM3 are the executable, the meteorology driver files and a runtime parameter file tm3.in, specifying the runtime parameters.

A template for the runtime parameter file tm3.in can be found in the doc directory. Not all parameters need to be specified there, one can delete lines or leave them commented out, if the defaults can be used. For more details see sec. 3.4.1.

Make sure that all files and directories specified really exist. Then start the executable tm3.
CHAPTER 3. USER’S MANUAL

3.2 Module userconstants

The small module userconstants in file userconstants.f90 defines the numerical precisions of the standard TM3 I/O files (pages 30 and 33), and TM3’s internal precision. By default, everything is set to standard REAL. Furthermore, the number of tracer species is selected here (parameter ntrace).

The user can find examples in the examples directory.

Example

```fortran
!*******************************************************************************
! mandatory user-defined TM3 constants
!*******************************************************************************
MODULE userconstants

!===============================================================================
!
! Description
!
! Only the TM3 parameters below need to be specified.
! User-variables and parameters are better defined in the file
! ‘userroutines.f90’.
!
!===============================================================================
!
! History
!
! 06.05.2002 Initial free format version. Stefan Koerner
!
! $Id: userconstants.f90,v 1.3 2002/08/19 17:21:21 tpobw Exp $
!===============================================================================

IMPLICIT NONE

!--- Fortran kinds of real TM3 variables

INTEGER, PARAMETER :: k4 = KIND(1e0), & ! default REAL
k8 = KIND(1d0), & ! double precision
ki = k4, & ! prepro input fields
kf = k4, & ! flux input fields
kc = k4, & ! calculation
ko = k4 ! mix output fields

!--- number of tracers

INTEGER, PARAMETER :: ntrace=1

END MODULE userconstants
```
3.3 Module **userroutines**

In the module **userroutines** (file **userroutines.f90**) the user can place his source-, sink- and chemistry-process calculations, as well as user-defined variables and parameters. This module can be set up almost entirely according to the user’s needs; the TM3 core modules need access only to ten **PUBLIC** subroutines, to be declared without dummy arguments:

- **trace0** is called during program startup by subroutine **start** to initialize the module, e.g. opening tracer input files, initializing tracer fields,
- **trace1** is called from subroutine **start** after **trace0**, if \( \text{istart} = 10+i \),
- **source1** is called every \( \text{nsrce} \) seconds for calculation of source1,
- **source2** is called every \( \text{nsrce} \) seconds for calculation of source2,
- **chem1** is called every \( \text{nchem} \) seconds for calculation of chemistry1,
- **chem2** is called every \( \text{nchem} \) seconds for calculation of chemistry2,
- **everystep** is called every step (\( \text{ndyn} \)) for auxiliary operations (e.g. reading data),
- **tracee** is called at program end by subroutine **finish** for a user shut-down,
- **save_userstatus** in case of a job continuation: called after main loop completion instead of **tracee** to save the user module status,
- **restore_userstatus** in case of a job continuation: called from subroutine **start** instead of **trace0** (and possibly **trace1**) to restore the status of the user module.

The user can write arbitrary additional subroutines or data into the module (remember to **SAVE** all variables that should retain their values across invocations). The diagnostic routines keep track separately the tracer conservation statistics for each of the four possible source and chemistry processes, but in principle the calculations assigned to **source1**/2 and **chem1**/2 can be chosen arbitrarily.

The important physical fields like tracer mass, slopes, air mass etc. are all declared and documented in the module **TM3global**, residing in file **src/tm3global.f90** and listed in appendix D on page 65. They can get into the current programming scope by a **USE** statement. The tracer masses and slopes must be modified within **userroutines** according to the processes under consideration.

The modification of the tracer slopes by emissions requires information about the emission distribution within a grid box. With a given emission field inside a box, built up by tracer emission during a time interval \( \Delta t \), it is consistent with the advection scheme to perform a least-square of a 3d linear function to the sum of old and emitted tracer distribution. The modification of the slopes is then determined by their definition in eq. 1.6.

Some standard emissions are treated in the following.
3.3.1 Example 1: Volume Source

Let a source strength for grid element \((i, j, l)\) be denoted as \(Q_{ijl}\) and given in kg/s. The tracer mass fields in kg before and after the emission are denoted by \(n_{ijl}^k\) and \(n_{ijl}^{k+1}\). Then the tracer mass field is updated by

\[ n_{ijl}^{k+1} = n_{ijl}^k + Q_{ijl} \Delta t, \]

where \(\Delta t\) is the length of the emission time step.

This update corresponds to the following line in subroutine \texttt{source1}:

\[ \text{rm}(i,j,l,1) = \text{rm}(i,j,l,1) + q(i,j,l)*\text{nsrce} \]

In TM3, \texttt{nsrce} is the identifier for the \texttt{source1} and \texttt{source2} timestep and a runtime parameter.

3.3.2 Example 2: Sink Proportional to Tracer Concentration

In the case of a decay process with mass loss proportional to the mass itself, \(n(t + \Delta t) = n(t) \exp(-\lambda \Delta t)\) in continuous time, like e.g. radioactive decay, the update equation for sufficiently small timesteps \(\Delta t\) reads

\[ n_{ijl}^{k+1} = n_{ijl}^k (1 - \lambda \Delta t). \]

The slopes \(nx, ny\) and \(nz\), defined by eq. 1.6, have to be multiplied by the same factor:

\[ nx_{ijl}^{k+1} = nx_{ijl}^k (1 - \lambda \Delta t), \]
\[ ny_{ijl}^{k+1} = ny_{ijl}^k (1 - \lambda \Delta t), \]
\[ nz_{ijl}^{k+1} = nz_{ijl}^k (1 - \lambda \Delta t). \]

In case of a large decay rate, e.g. if \(\lambda \Delta t > \frac{1}{2}\), an implicit discretization might be required in order to prevent instabilities or/and negative concentrations. For the tracer mass the implicit update equation reads

\[ n_{ijl}^{k+1} = n_{ijl}^k / (1 + \lambda \Delta t), \]

with according equations for the slopes.

The explicit update equations correspond to the following lines in subroutine \texttt{chem1}:

\[ \text{rm}(i,j,l,1) = \text{rm}(i,j,l,1) * (1.-\lambda\text{chem}) \]
\[ \text{rxm}(i,j,l,1) = \text{rxm}(i,j,l,1) * (1.-\lambda\text{chem}) \]
\[ \text{rym}(i,j,l,1) = \text{rym}(i,j,l,1) * (1.-\lambda\text{chem}) \]
\[ \text{rzm}(i,j,l,1) = \text{rzm}(i,j,l,1) * (1.-\lambda\text{chem}) \]

In TM3, \texttt{chem} is the identifier for the \texttt{chem1} and \texttt{chem2} timestep and a runtime parameter.
3.3.3 Example 3: Surface Source

Let the source strength for the surface grid element \((i, j)\) be denoted as \(Q_{ij}\) and given in kg/s. The tracer mass fields in kg before and after the emission are again denoted by \(n_{ijl}^{k}\) and \(n_{ijl}^{k+1}\). Then the tracer mass field is updated by

\[
n_{ijl}^{k+1} = n_{ijl}^{k} + Q_{ij} \Delta t_s,
\]

where \(\Delta t_s\) is the length of the emission time step. Only the surface grid elements are recomputed.

If one can assume a homogeneous emission inside the surface elements, the slopes are not changed by this process. But if one approximates the vertical emission density to be delta-distributed on the surface, \(\rho(z) = \delta(z)Q\Delta t_s\), with \(\int_{-\infty}^{\infty} \delta(z)dz = 1\), the new slopes are computed by a least-square fit to the sum of old tracer distribution and the emitted profile. This scheme is consistent with the scheme used in the advection algorithm by Russel and Lerner [Russel and Lerner, 1981]. As result the vertical slope \(n_z\) is updated by

\[
n_{izj1}^{k+1} = n_{izj1}^{k} - Q_{ij} \Delta t_s.
\]

Depending on the total tracer mass in the box \(n_{ij1}\), this update might generate negative mixing ratios (the mixing ratio varies in the vertical between \((n_{ij1} - |n_{zij1}|)/m_{ij1}\) and \((n_{ij1} + |n_{zij1}|)/m_{ij1}\), if zero \(n_z\) and \(n_y\) slopes are assumed; \(m_{ij1}\) is the air mass). There are two solutions: Either one uses the update formula above and corrects for negative mixing ratios, e.g. by setting the runtime parameter limits to .TRUE., or one uses an update that is the best achievable fit to the delta emission under the side condition not to generate new negative mixing ratios. The latter yields

\[
n_{izj1}^{k+1} = n_{izj1}^{k} - Q_{ij} \Delta t_s.
\]

A Commented Code for Example 3

```fortran
!*******************************************************************************
! user-supplied subroutines and user-variables
!*******************************************************************************

MODULE userroutines

!-------------------------------------------------------------------------------
!
! Purpose
! -------
!
! This module reads at model startup a temporally constant source field <flux>
! from file 'foss2_fg.d', which is supposed to contain a surface emission flux
! field in kg/year. In subroutine source1, this source emits the tracer into
! the lowest model layer.
!
!-------------------------------------------------------------------------------
!
! Description
! -----------
!
! The following module contains all user-supplied subroutines associated
```
CHAPTER 3. USER’S MANUAL

! with sources and sinks (due to chemical reactions) and possibly
! with the initial tracer fields.
!
! trace0 called from subroutine 'start' to initialize the user
! module (e.g. opening tracer input files, possibly
! initializing tracer fields)
! trace1 called from subroutine 'start' after 'trace0', if
! <istart> = 10+i
! source1 called every <nsrce> seconds for calculation of source1
! source2 called every <nsrce> seconds for calculation of source2
! chem1 called every <nchem> seconds for calculation of
! chemistry1
! chem2 called every <nchem> seconds for calculation of
! chemistry2
! everystep called in every step for auxiliary operations (e.g.
! reading of user data)
! tracee called after completion of the main loop for possible
! final operations
! save_userstatus in case of a job continuation: called after main loop
! completion instead of 'tracee' to save the user module
! status
! restore_userstatus in case of a job continuation: called from subroutine
! 'start' instead of 'trace0' (and possibly 'trace1') to
! restore the status of the user module

!-------------------------------------------------------------------------------
! $Id: userroutines.f90,v 1.6 2003/06/11 11:24:15 tpobw Exp $
!-------------------------------------------------------------------------------

USE userconstants, ONLY: kc, & ! computational precision
ntrace
USE TM3global, ONLY: im, jm, lm, & ! grid dimensions
rm, rzm, rym, rxm, & ! tracer field and slopes
nsrce, & ! time interval for source calc.
peth_in_s & ! path to input source data

IMPLICIT NONE

PRIVATE

!---- subroutines called by the TM3 core
PUBLIC :: trace0, trace1, &
source1, source2, &
chem1, chem2, &
everystep, tracee, &
save_userstatus, &
restore_userstatus

!---- private user variables

REAL(kc), SAVE :: flux(im,jm) ! see also the module 'aux_flux'
! for useful flux routines

!---- define user parameters to be read together with the runtime
!---- parameter list (tm3.in)

!NAMESLIST /userparams/ ...

CONTAINS

! initialize source module
3.3. MODULE USERROUTINES

! Example: read in stationary source strength field <flux>
! units: kg/year
!----------------------------------------------------------------------------
SUBROUTINE trace0

USE TM3io_prepro, ONLY: get_free_lun ! utility subroutine
IMPLICIT NONE

INTEGER :: my_unit

!----------------------------------------------------------------------------
my_unit = get_free_lun() ! get a free unit number
OPEN (my_unit,FILE=TRIM(path_in_s)//'foss2_fg.d', STATUS='OLD')
READ (my_unit,*+* flux
CLOSE (my_unit)

flux = flux/(365.*24.*3600.) ! transform to kg/s

END SUBROUTINE trace0

!----------------------------------------------------------------------------

! this subroutine is called at program startup after 'trace0', if
! <istart> = 10+i
!----------------------------------------------------------------------------
SUBROUTINE trace1
IMPLICIT NONE

END SUBROUTINE trace1

!----------------------------------------------------------------------------

! this subroutine changes the tracer mass and its slopes
! source process 1
!
! Example: add tracer to first layer
! source given in kg/gridbox/year
!----------------------------------------------------------------------------
SUBROUTINE source1
IMPLICIT NONE

INTEGER :: i, j, n
REAL(kc) :: x

!----------------------------------------------------------------------------
DO n=1,ntrace
  DO j=1,jm
    DO i=1,im
      x = flux(i,j)*nsrce ! nsrce is the emission timestep
      rm (i,j,1,n) = rm (i,j,1,n) + x
      rzm(i,j,1,n) = rzm(i,j,1,n) - x ! surface emission
    ENDDO
  ENDDO
ENDO

END SUBROUTINE source1

!----------------------------------------------------------------------------

! this subroutine changes the tracer mass and its slopes
! source process 2
!----------------------------------------------------------------------------
CHAPTER 3. USER’S MANUAL

SUBROUTINE source2
  IMPLICIT NONE
END SUBROUTINE source2

!----------------------------------------------------------------------------
! this subroutine changes the tracer mass and its slopes by chemistry
! process 1
!----------------------------------------------------------------------------
SUBROUTINE chem1
  IMPLICIT NONE
END SUBROUTINE chem1

!----------------------------------------------------------------------------
! this subroutine changes the tracer mass and its slopes by chemistry
! process 2
!----------------------------------------------------------------------------
SUBROUTINE chem2
  IMPLICIT NONE
END SUBROUTINE chem2

!----------------------------------------------------------------------------
! procedure called after every time step
!----------------------------------------------------------------------------
SUBROUTINE everystep
  IMPLICIT NONE
END SUBROUTINE everystep

!----------------------------------------------------------------------------
! procedure called after completion of main loop
!----------------------------------------------------------------------------
SUBROUTINE tracee
  IMPLICIT NONE
END SUBROUTINE tracee

!----------------------------------------------------------------------------
! in case of a job continuation: called after main loop completion
! instead of ‘tracee’ to save the user module status
!----------------------------------------------------------------------------
SUBROUTINE save_userstatus
  IMPLICIT NONE
END SUBROUTINE save_userstatus

!----------------------------------------------------------------------------
! in case of a job continuation: called from subroutine ‘start’
! instead of ‘trace0’ (and possibly ‘trace1’) to restore the status of
! the user module
!----------------------------------------------------------------------------
SUBROUTINE restore_userstatus
  IMPLICIT NONE
END SUBROUTINE restore_userstatus

! The only user variable to be restored if a continuation run starts
! is flux. It’s done by reading the flux file again.

CALL trace0
3.4 Program Inputs

3.4.1 Runtime Parameter File tm3.in

In the formatted sequential (ASCII) file tm3.in the user has to specify TM3’s runtime parameters. tm3.in is being read from the working directory during program start. It must contain specifications for items of TM3’s namelist inputz, which compiles all TM3 runtime parameters. Defaults apply for variables not quoted there, hence only those items need to be specified for which the defaults can not be used. The runtime parameters of namelist inputz are defined in the module for global variables TM3global.

The user can append his own namelist with runtime parameters for his user-defined routines to tm3.in.

A documented example with the complete list of runtime parameters, their meaning, and default values can be found in the doc directory. It is listed in appendix E on page 71. The following paragraph describes one of the most important runtime parameters.

The Start Mode

The integer runtime parameter istart defines the startup behaviour of TM3. It can be set to the following values:

1 Cold start
   Initial tracer fields are set to 0. They can afterwards be partially or totally overwritten in subroutine trace0. No other status fields are read.

2 Cold start with initial tracer fields
   Initial tracer fields are read from file 
   
   <path.out>\<jobid>\'modelstatus.b'. No other status fields are read.

3 Job chain mode
   The complete model status, including tracer fields, but excluding the runtime parameters of namelist inputz, is read at program start from file
   
   <path.out>\<jobid>\'modelstatus.b'. Useful for automatic job chains in time limited queues (see also sec. 3.6.3).

10+i As above, with i = 1, 2 or 3. In addition the user-routine trace1 will be called after trace0.
3.4.2 Meteorological Input Files and their Presumed Organization

TM3 is an off-line model that reads the necessary meteorological fields for tracer transport from a storage medium. Normally those fields are produced by a preprocessing step, described in appendixes A.1 and A.2.

For these input files as well as for the output files, TM3 applies a naming convention, with ".b" as filename suffix for unformatted files and ".d" for formatted (ASCII) files.

The unformatted ".b"-files read by TM3 have to be of Fortran type "sequential unformatted". Each record is of "TM3 type", i.e. it is written by statements like

```fortran
INTEGER :: itau, idate(6)
REAL (ki):: array(dim1,dim2[,dim3])
...
WRITE (lun) itau, idate, array
```

`itau` and `idate` represent two reference times for the 2- or 3d array, `itau` in seconds since midnight January 1, `<iyear0>` of the preprocessing run, and `idate` as 6-dimensional integer vector (/year, month, day, hour, minute, second/) (see section 2.6). Positioning the read pointer inside TM3 to the requested reference time is done using `idate` only.

Assuming 4 bytes per `REAL` variable, TM3 reads in particular

<table>
<thead>
<tr>
<th>Field</th>
<th>Default filenames for grid res. fg</th>
<th>Dim</th>
<th>Size/yr in MB for 6-hourly meteorology</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>entrainment and detrainment rates for the convection module</td>
<td>eu_{fg}.b, ed_{fg}.b, du_{fg}.b, dd_{fg}.b</td>
<td>3</td>
<td>4×366</td>
<td>kg/m³s</td>
</tr>
<tr>
<td>vertical diffusion coefficient</td>
<td>k_{fg}.b</td>
<td>3</td>
<td>385</td>
<td>m²/s</td>
</tr>
<tr>
<td>geopotential height</td>
<td>z_{fg}</td>
<td>3</td>
<td>385</td>
<td>m</td>
</tr>
<tr>
<td>time-staggered surface pressure and mass flux rates in x- and y-direction (cyclic meteorology)</td>
<td>stag_{ps}<em>{fg}.b, stag</em>{pu}<em>{fg}.b, stag</em>{pv}_{fg}.b</td>
<td>2, 3, 3</td>
<td>19+2×366</td>
<td>Pa, kg/s</td>
</tr>
<tr>
<td>time-staggered surface pressure and mass flux rates in x- and y-direction (continuous meteorology)</td>
<td>stag_{ps}<em>{fg}.b, stag</em>{pu}<em>{fg}.b, stag</em>{pv}_{fg}.b</td>
<td>2, 3, 3</td>
<td>19+2×366</td>
<td>do.</td>
</tr>
<tr>
<td>massflux information for adaptive integration stepwidth mode (optional)</td>
<td>maxndyn_{fg}.d, maxndync_{fg}.d</td>
<td>1</td>
<td>0.06</td>
<td>s</td>
</tr>
<tr>
<td>temperature, spec. humidity (both optional)</td>
<td>t_{fg}.b, q_{fg}.b</td>
<td>3</td>
<td>2×366</td>
<td>K, kg/kg</td>
</tr>
</tbody>
</table>

The files are organized per year for cg and fg, and per month for higher resolutions, to avoid file sizes beyond 2 GB. They must reside in a single directory with the reference year YYYY as directory name for grid resolutions cg and fg, and with reference year and month YYYYMM as name for higher grid resolutions.

For a "cyclic mode", using the meteorology of a particular year for simulations over multiple years, airmass flux fields are read in that differ from those for the continuous mode by the record of the last reference time of the year. Their file names are `stag_*_.b` for cyclic meteorology instead of `stagc_*_.b` for the continuous.
3.4. PROGRAM INPUTS

Example

```bash
$ pwd
/Net/Groups/AG_Heimann/tm3/prepro/fg/NCEP
$ ls
$ ls 2001
@job0.log eu_fg.b q_fg.b stagc_pu_fg.b
@job1_fg.log k_fg.b stag_ps_fg.b stagc_pv_fg.b
dd_fg.b maxdyn_fg.d stag_pu_fg.b t_fg.b
du_fg.b maxdynmc_fg.d stag_pv_fg.b z_fg.b
ed_fg.b p_fg.b stagc_ps_fg.b
```

The path of these annually or monthly input directories must be specified in the runtime parameter file `tm3.in`, namelist variable `path_in_p`, without trailing slash. The path parts for resolution and meteorology are appended by the program. Hence, for the preceding example the path specification in `tm3.in` reads

```
path_in_p = '/Net/Groups/AG_Heimann/tm3/prepro'
```

3.4.3 Other Standard Input Files

`plandf.<gridID>.d` Fraction of land in each grid cell (optional)

This file is not needed by the model code itself, but this information might be useful in the tracer specific subroutines.

Control parameters in namelist `inputz`: `read_plandf`, `name_plandf`

Default: `read_plandf=.FALSE. (do not read <name_plandf>)`

`modelstatus.b` Complete model status or initial tracer fields only

The complete form is always written at a normal program end. It is needed for a model cold start with initial tracer fields (selected with `istart=2`), or a continuation run (selected with `istart=3`).

For a user-defined startup tracer distribution (`istart=2`), the user needs an initial file `modelstatus.b` in the directory `<path_out>` with the following shape (TM3 record structure):

```plaintext
INTEGER :: itaur, idater(6)
REAL(kc) :: rm(im,jm,lm,ntrace), rxm(im,jm,lm,ntrace), &
            rym(im,jm,lm,ntrace), rzm(im,jm,lm,ntrace)
WRITE (lun) itaur, idater, rm, rxm, rym, rzm
```

The date `idater` is checked at program startup and a warning is issued, if it does not match the program start date `idatei` of namelist `inputz`. `itaur` is not checked.

For an automatic job chain (`istart=3`) see sec. 3.6.3 or the explanations in `doc/tm3.in`.

Control parameters in namelist `inputz`: `istart`, `nsave`, `jobid`, `path_out`

Default: Do not read `modelstatus.b` at startup (`istart=1`, cold start)

`modelstatus.d` Restart information for job chains (optional)

This formatted file contains the next model start date `idatei` of the form

```plaintext
INTEGER :: idatei(6)
```

as well as user service data. See sec. 3.6.3 on page 40 or the file `doc/tm3.in` for more
information.

Control parameters in namelist inputz: istart, nsave, jobid, path_out.

stationlist.d Station coordinates list (optional)

For output of tracer mixing ratio timeseries at station locations. It must be a text file with one line per station of the form

    name lat lon height

Here, name is a character string with up to 12 characters, lat and lon give the horizontal coordinates in degrees north and east, and height is counted in meters above sea level. Alternatively, height can be set to a negative integer; then its absolute value is being taken as the model layer index l (useful to force the station to be in the lowest layer by setting height to -1).

Lines with the token # or ! as first non-blank character are interpreted as comments. Control parameters in namelist inputz: name_stationlist
3.5 Program Outputs

The standard output files are written to an output directory specified by the runtime parameter `path_out` in namelist `inputz`. All file names quoted here are relative to that path.

For most output files there exists a corresponding runtime parameter in `inputz`, where a different name can be specified. This section refers to the default names. They get the value of the runtime parameter string `jobid` as prefix, with an underscore appended. The default is `jobid = ""` (nothing will be prefixed).

The naming conventions and record structure of the formatted and unformatted files, described in sec. 3.4.2, also apply for the output files.

*TM3 has a convention for those runtime parameters in `inputz` that control output time intervals. If such a parameter is set to an integer greater than 0, this means "interval in seconds since 0:00:00 UT, January 1, iyear0". 0 means "no output", -1 "daily at 0:00", -2 "monthly at 1st 0:00", and -3 "yearly, January 1 0:00".*

3.5.1 Output to the Standard Output Device

During runtime, TM3 prints a description of the performed actions to standard output.

Example

Reading model control parameters from "tm3.in".. ok

```
------------
Global Atmospheric Tracer Model TM3, version 3.8
Max Planck Institute for Biogeochemistry, Jena
Fine grid (fg, 560 km), NCEP meteorology
Starting job: ""

-----------------------------
1998-Jan-01 00:00:00 opening output files:
   "output/mix.b"
1998-Jan-01 00:00:00 opening met input files
1998-Jan-01 00:00:00 opening mass conservation output files:
   "output/consrv_tracer.d"
   "output/consrv_air.d"
1998-Jan-01 00:00:00 opening diagnostic output files:
   "output/zonal_avrg.d"
   "output/mmix.b"
1998-Jan-01 00:00:00 *** starting main model loop
1998-Jan-01 00:00:00 *** starting main model loop
1998-Jan-01 00:00:00 maxc: 0.461 0.242 0.515
1998-Jan-03 00:00:00 maxc: 0.396 0.216 0.483
1998-Jan-04 00:00:00 maxc: 0.350 0.207 0.481
1998-Jan-05 00:00:00 maxc: 0.425 0.209 0.424
1998-Jan-06 00:00:00 maxc: 0.471 0.242 0.582
1998-Jan-07 00:00:00 maxc: 0.592 0.471 1.233, ncflv: 0 0 12
1998-Jan-08 00:00:00 maxc: 0.608 0.268 0.500
1998-Jan-08 00:00:00 *** main model loop finished
1998-Jan-08 00:00:00 status arrays saved to "output/modelstatus.b"

-----------------------------
```
CHAPTER 3. USER’S MANUAL

---

--

---

Job "" completed normally.

End condition: finish date has been reached
Number of timesteps: 252
Cumul. number of CFL violations: 12
CPU time: 0 h 1 min 4 s
CPU per step: .25 s
CPU per tracer and year: 0 h 55 min

---

---

In this example the runtime parameter ndiap1 was set to -1, hence the daily maximum Courant numbers\(^2\) maxc for the three space dimensions are printed. In case CFL criterion violations happened during the last ndiap1 time interval, then counters for Courant numbers > 1 are appended for the three space dimensions (ncflv). After finishing the main loop the CPU statistics is printed, which helps in scheduling larger jobs.

3.5.2 Tracer-Related Main Output

Mixing Ratios $\bar{\chi}$

By default, the model writes the simulated instantaneous tracer mixing ratio fields in the file mix.b. The output interval in seconds is given by the runtime parameter ninst of namelist inputz. The runtime parameter idate defines the date after which instantaneous output shall begin; this allows for economically writing instantaneous output after some model spin-up period.

Tracer mixing ratios written to mix.b have the unit “kg tracer/kg air”, and are multiplied by the scaling factor fscale(i) (runtime parameter) for tracer i. If unformatted output is selected in the Makefile, the record structure is "TM3 type" (sec. 3.4.2), i.e. it can be read as:

```fortran
REAL :: mix(im,jm,lm,ntrace) ! spatial grid dim. and tracer number
INTEGER :: tau, date(6)
...
OPEN (UNIT=10, FILE=..., FORM='UNFORMATTED', STATUS='OLD')
READ (10) tau, date, mix
```

If netCDF output is selected in the Makefile instead, instantaneous mixing ratio fields are written to a netCDF file mix.nc. This file contains

- longitudes, latitudes, level numbers, tau (TM3 reference time),
- the vertical hybrid coordinate coefficients a and b,
- the land fraction array,

\(^2\)See footnote on p. 7.
3.5. PROGRAM OUTPUTS

- the grid cell area array,
- the instantaneous mixing ratios,
- the instantaneous surface source fluxes, if any,
- the instantaneous air mass, and
- the instantaneous surface pressure.

For a detailed description see the comment header of src/tm3io_mix.netCDF.f90.

Station Output

If TM3 finds a station list at program startup (see sec. 3.4.3), it writes time series of tracer mixing ratios at all locations specified in the station list. Values are linearly interpolated in 3 dimensions between the 8 surrounding grid boxes next to the station location. The units are the same as for the 3d field output, described in the two preceding sections.

In the default case (unformatted sequential output to file mixstn.b), a matrix with the values for all tracers for all stations is written for each reference time. It can be read as:

```fortran
REAL :: mix(stations_n,ntrace) ! number of stations, tracer number
INTEGER :: itau, idate(6)
...
OPEN (UNIT=10, FILE=..., FORM='UNFORMATTED', STATUS='OLD')
READ (10) itau, idate, mix
```

If the Makefile is set up to produce formatted (ASCII) output instead, instantaneous mixing ratios are written to several files mix_stationname.d, one file per station. Each reference time corresponds to one line, with itau at column 1, the reference date idate at columns 2–7 and the mixing ratios of all tracers starting from column 8.

Diagnostic mixing ratios for some gridpoints with known indices (i,j,l) can be written directly in file check.d (see sec. 3.5.3).

Tracer Mass Fields

The model can also write the simulated instantaneous tracer mass fields in the file tmass.b (default name). The output interval in seconds is given by the runtime parameter nintendo in namelist inputz. The runtime parameter idatet defines the date after which instantaneous output begins. As for the mixing ratios, this allows for economically writing instantaneous output after some model spin-up period.

Tracer mixing mass written to tmass.b has the unit “kg tracer”. The record structure is ”TM3 type” as for the mixing ratios, and can be read as:

```fortran
REAL :: tmass(im,jm,lm,ntrace) ! spatial grid dim. and tracer number
INTEGER :: itau, idate(6)
...
OPEN (UNIT=10, FILE=..., FORM='UNFORMATTED', STATUS='OLD')
READ (10) itau, idate, tmass
```
CHAPTER 3. USER’S MANUAL

3.5.3 Diagnostic Output

The tracer model writes several ASCII and binary files with diagnostic information:

- **mmix.b** Time-averaged tracer mixing ratio 3d fields (switched on by default),
- **zonal_avg.d** Tables with zonally and temporally averaged tracer mixing ratios (switched on by default),
- **consrv_tracer.d** Tables with conservation statistics of the tracers, and zonally averaged tracer mixing ratios (switched on by default),
- **consrv_air.d** Tables with conservation statistics of the air mass (switched on by default),
- **check.d** Mixing ratio timeseries at a set of check locations (switched off by default), and
- **debug.d** Detailed protocol of the model run (switched off by default).

**Time-Averaged Mixing Ratios**

The model computes time averaged tracer mixing ratio fields by sampling the tracer masses every $n_{diag}$ seconds (runtime parameter), starting at $idatem$ (runtime parameter). These averaged fields are written in a file **mmix.b** by default. The time averaging interval can be selected by the runtime parameter $n_{diagp2}$.

As with the instantaneous mixing ratios, **mmix.b** contains tracer fields with unit “kg tracer/kg air”, multiplied by the scaling factor $f_{scale}(i)$ (runtime parameter) for tracer $i$.

This file has the same structure as the instantaneous mixing ratio file - see above. Note that in this case the time/date stamp recorded with each field is the time/date when the field is written, i.e. the first time instant after the averaging time interval. E.g. in the case of monthly averaged fields, the recorded time/date stamp shows the beginning of the month immediately after the averaging period.

**Zonally and Temporally Averaged Tracer Mixing Ratios**

The zonally averaged tracer mixing ratios are written in a set of ASCII tables with default filename **zonal_avg.d**. The output interval is controlled by the value of runtime parameter $n_{diagp2}$.

**Tracer Conservation Statistics**

The file **consrv_tracer.d** contains ASCII tables of various conservation statistics for each tracer. The instantaneous tracer masses and their changes due to the individual transport and source/sink processes are listed here, for each latitude interval of the model grid, for each hemisphere and for the entire globe. Furthermore, for each tracer a simple latitude-height table of the zonally averaged tracer mixing ratio is written, with units “kg tracer/kg air” and multiplied by the factor $f_{scale}(i)$ (runtime parameter) for tracer $i$. The runtime parameter $n_{diagp1}$ selects the output and averaging time intervals for these tables.
3.5. PROGRAM OUTPUTS

Air Mass Conservation Statistics

The conservation statistics of the air mass is written in a set of ASCII tables similar to those for the individual tracers. The default filename is consrvair.d. The output interval is also controlled by the value of runtime parameter ndiap1. These tables are determined directly from the input airmass flux fields. They can be inspected to check the conservation of air mass.

Mixing Ratio Timeseries at Check Locations

Up to 10 check locations can be defined with the runtime parameter array indc(3,10), which is included in namelist inputz. The triplet indc(:,k) denotes the (i, j, l) grid indices of check location k, where k = 1, noindc (runtime parameter). Depending on the value of the runtime parameter ncheck, time series of the mixing ratios of all tracers are written in file check.d. The mixing ratios have the unit “kg tracer/kg air”, multiplied by the factor fscale(i) (runtime parameter) for tracer i.

Debugging Output

If the runtime parameter debug is set to .TRUE., a detailed protocol of all tasks performed by the model is written in file debug.d. This option is intended primarily for the debugging of the transport model code itself and of little use during normal operation. The amount of information written in debug.d is very large. Therefore this switch should be used only for short test runs of length less than a few days.

3.5.4 Other Output

tm3.in_complete At startup the complete list of all runtime parameters is written in a file tm3.in_complete, including those not specified in tm3.in. If such a file already exists it will be overwritten.

ps.b An instantaneous surface pressure field for possible vertical interpolation of the mixing ratio output to pressure or sigma levels in a postprocessing stage. This output is triggered at the same time instants as the output of the instantaneous mixing ratio fields, and consistent with the instantaneous mass and mixing ratio fields. The output is controlled by the runtime parameter cwrtps and switched off by default.

modelstatus.b When finishing a run normally, TM3 writes its complete status, excluding the runtime parameters, in a file modelstatus.b. The tracer fields and slopes after the last integration time step are also contained therein. If a file of that name exists, it will be overwritten. In conjunction with the job chain mode (istart=3, described in sec. 3.6.3), this file can be used as input for continuation runs, or, in its abridged form, for “cold starts with initial tracer distributions (istart=2)”. The file structure is described in sec. 3.4.3.

modelstatus.d If the user has selected the job chain mode with istart=3, TM3 writes an output file modelstatus.d, after the runtime parameter time nsave has passed by without reaching the model end date idatee. Its existence indicates the necessity
CHAPTER 3. USER’S MANUAL

of a continuation run. After reaching the end date idatee this file is deleted. See sec. 3.4.3 and sec. 3.6.3 for more details.

3.6 Programming Hints

3.6.1 Useful Auxiliary Subroutines

TM3 comes with several subroutines helping the user in fulfilling his tasks:

<table>
<thead>
<tr>
<th>Module</th>
<th>File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aux</td>
<td>flux aux.f90              subroutines for handling surface source fluxes</td>
</tr>
<tr>
<td>aux</td>
<td>grid aux.f90              auxiliary routines for grid calculations</td>
</tr>
<tr>
<td>tm3io</td>
<td>schedule tm3io            subroutines for calculation/output of mean mixing ratios measured at station locations according to a given time schedule</td>
</tr>
<tr>
<td>time</td>
<td>and date                 routines for date/time/calendar calculations</td>
</tr>
</tbody>
</table>

In addition, the function get_free_lun() in module tm3io_prepro provides a free logical I/O unit number at run time.

For details see the description in the source files.

3.6.2 Multiprocessing with TM3

Since release 3.6R2, TM3 features symmetric multiprocessing with OpenMP 1.0 (see www.openmp.org), especially useful for higher resolutions.

Remark 1 With TM3 release 3.8, the computational work is divided among the tracers. Hence the speed gain is largest if either the number of tracers n is a multiple of the number of processors p, or the fraction n/p is large compared to 1.

Remark 2 If running TM3 in parallel, don’t change parallelized TM3 code parts unless having understood how the related OpenMP constructs work. The same applies for all data involved.

Setting up a Multiprocessing Executable

The OpenMP programming interface includes source code constructs in the shape of special Fortran comments, library calls and environment variables. Commonly, OpenMP parallelized source code is written in a manner that retains the capability of compiling and running the code on a single CPU, without any additional precautions.

Setting up a multiprocessing executable is platform dependent. Usually it is done by quoting special compiler options, or by calling a special compiler. For the platforms supported by the Makefile, this is ready-for-use. Type ”make” without targets to see what is implemented.

Running a parallel TM3

Generally, at run time an OpenMP-instructed executable accesses environment variables that influence the behaviour of its parallel regions. The user needs to set only one,
3.6. PROGRAMMING HINTS

OMP\_NUM\_THREADS, which specifies the number of threads (CPUs) to be used. Under a UNIX shell, the command `export OMP\_NUM\_THREADS=4`, e.g., instructs the TM3 executable to use 4 processors. The default is platform dependent.

At start-up with OMP\_NUM\_THREADS=4, TM3 gives a note about multiprocessing alike:

---------------------------------------------------------------------
Global Atmospheric Tracer Model TM3, release 3.8  
Max Planck Institute for Biogeochemistry, Jena  
Grid version: fine grid, NCEP meteorology  
Multiprocessing with 4 threads  
...  
---------------------------------------------------------------------

TM3 presumes to have the specified number of CPUs entirely for itself. If this is not the case, speed degradations are likely (the loop schedules are set to "static"). The achievable speedup depends on platform, compiler, TM3 setup parameters and the user code. Test results with empty user-routines are shown in the figure below.

The Influence of Serial Program Parts

Running TM3 on more than one CPU has effect only on those program parts that have been parallelized. With more processors, the parts that are executed serially become a performance bottleneck.

Example  In serial mode, a program executes over 20 seconds in serial program regions and 80 seconds in program regions that can also be run in parallel. I.e., 20 % of the program’s execution time is not parallelized. When run on 4 processors the total execution time is 40 s, assuming that the parallel parts are accelerated by a factor of 4. But 50 %
CHAPTER 3. USER’S MANUAL

of the execution time is now spent in the small part that is not parallelized. 3 CPUs are idle at half of the total execution time.

To achieve a reasonable speedup even with few processors, it was necessary to parallelize large parts of TM3. If one has serially programmed, runtime expensive user routines, then the total speedup will be seriously restricted. The only way to achieve higher model speedup is parallelizing these parts as well.

3.6.3 Job Chains

Job chains, i.e. a (repeated) cycle of computation and finishing with status saving, are explicitly supported by TM3, making the work with time-limited job queueing systems easier. The controlling runtime parameters are istart and nsave. istart needs to be set to 3 for job chains.

In TM3, the existence of a file <path_out>///jobid//modelstatus.d is used as indicator if this is a continuation run within a job chain or the start of it. The user can employ this as end condition in a self-calling job script.

IF this file does NOT EXIST, then the start date is that of tm3.in. Initial tracer fields are read from the file <path_out>///jobid//modelstatus.b, if existent, or otherwise set to 0.

IF the file EXISTS, then the namelist inputz is first read from file tm3.in. Afterwards the start date idatei is overwritten by the date stored in <path_out>///jobid//modelstatus.d, saved by a previous model run at time intervals of <nsave> seconds, counted from 01-Jan-<iyear0> 00:00:00.

Initial tracer fields are read from file <path_out>///jobid//modelstatus.b, if existent. That file will always be written after a model run, independently of what istart is set to.

The continuation parameter file <path_out>///jobid//modelstatus.d contains the start date in the form YYYY MM DD hh mm ss as first line. The last line contains all prepro directory identifiers needed by the next continuation cycle, e.g. 1999 2000 2001 or 199911 199912 200001. This is useful for getting prepro files out of an archive by shell script commands, like shown in the jobscript example of sec. 3.6.4. After reaching the end date idatee, TM3 deletes this file.

Furthermore, two TM3global variables support the initialization of user-defined subroutines with istart=3:

jobcont is set to .TRUE. at program startup if a job continuation is present, i.e., if modelstatus.d exists, and to .FALSE. otherwise,

newsrun is set to .TRUE. at program startup and to .FALSE. after the first timestep.

For saving and restoring user data for the next job cycle, the subroutines save_userstatus and restore_userstatus in module userroutines can be utilized. They are called instead of trace0, trace1 and tracee in case TM3 detects a job continuation. The user can do the saving in save_userstatus by contracts like

SUBROUTINE save_userstatus
IMPLICIT NONE
INTEGER :: lun
!
--- job chain: save user fields
3.6. PROGRAMMING HINTS

lun = get_free_lun() ! get a free unit number
OPEN (UNIT=lun, FILE=TRIM(path_out)//'userfields.b', FORM='UNFORMATTED', &
STATUS='REPLACE')
WRITE (lun) <the user’s data>
CLOSE (lun)

END SUBROUTINE save_userstatus
END IF

and read the fields accordingly in subroutine restore_userstatus.

3.6.4 Jobscript Example

The following example shows work-alikes of the TM3 features

- parallel processing with 4 processors,
- automatic job chaining with istart=3,
- using the service information of modelstatus.d to copy only the required meteo files out of an archive.

Platform-dependent are the Korn shell syntax and the usage of a NQS job queueing environment.

```bash
#@$-s /bin/ksh
#@$-c 4 # number of cpus, default is 1
#@$-lM 4GB # memory limit per request
#@$-lT 3:45:00 # time limit per request
#@$-r mylabel # job label
#@$-eo # send stderr to stdout

# cd "$QSUB_WORKDIR"

#--- the system provides a very fast file system,
#--- where later the prepro files are copied to
ln -s $TMPDIR input

#--- set OpenMP environment
export OMP_NUM_THREADS=4

#--- get the proper prepro files from an archive
if [ ! -f output/modelstatus.d ]; then
  # initial run
dirlist="199901 199902 199903"
else
  # continuation run: get the necessary directory names
  # from the last line of the service file ‘modelstatus.d’
dirlist='tail -1 output/modelstatus.d'
fi

for dir in $dirlist; do
  for ppfile in eu ed du dd stag_ps ... ; do
    cp [mf]cp my_storagedir/$dir/$ppfile_xfg.b input
  done
done
```

41
#--- Set up the TM3 runtime parameter list.
#--- For parameters not specified the defaults are ok.
#--- Fortran namelist I/O syntax is required here.

```bash
# inputz

! 1st Most frequently user-changed parameters
! -------------------------------------------

  path_in_p = 'input/' ! path to input prepro (met.) data
  path_in_s = 'input/' ! path to input source data
  path_out = 'output/' ! path to output
  idatei = 1999 1 1 0 0 0 ! for start of model run
  idatee = 2000 12 31 18 0 0 ! for end of model run

...  

! 2nd Other Adjustable Parameters
! -------------------------------

  istart = 3 ! do job chaining
  nsave = 7257600 ! 3x28 days, in seconds

...  

#--- Start TM3.
#--- Output is already redirected by NQS.

./tm3

#--- some cleanup

rm -r input
tm tm3.in

#--- If the job hasn't come to an end, send this
#--- script again to the queueing system.
#--- Leave an opportunity to kill this job gracefully,
#--- in case something goes wrong.

if [ -f output/modelstatus.d ]; then
cat < /dev/null > kill_me_now
sleep 20
rm kill_me_now
qsub <the_name_of_this_script>
else
  (send a mail, e.g.)
fi
```
3.6. PROGRAMMING HINTS

3.6.5 Migration Guidelines for Prior TM3 Releases

TM3 3.7 and older

1. Since release 3.8, the TM3 source code is written in Fortran 95 free format. The Makefile accesses the files named userconstants.f90 and userroutines.f90, but for fixed format code this could be changed in the Makefile to the suffix "f". Programming in free format is a lot more convenient, hence the recommendation is to convert fixed format user code to free format. There exist many converters around. In case of doubt the user can contact the TM3 support (see appendix G).

2. Compared to earlier releases, some files and modules are new or have been renamed. E.g., the Fortran module containing the most important physical fields like tracer masses etc. is now tm3global, rather than variables or tm3variables. Source code files have the name of the module contained within, wherever possible.

3. The module encapsulation has been advanced. Previously global program entities may now require an USE statement for the module defining that entity.

4. A log of changes can be found in doc/CHANGES.

TM3 3.8

To get the sources compiled, three empty subroutines trace1, save_userstatus and restore_userstatus need to be added to the module userroutines. Furthermore, the format of some outputs was slightly changed - see the file doc/CHANGES.

3.6.6 CPU expense

The CPU times of TM3 runs depend largely on platform, setup parameters (e.g. number of tracers, grid resolution, number of used processors, the user-routines, compiler options) and runtime parameters (e.g. time steps, whether limits is set or not, usage of variable timesteps). TM3 version 3.8a is optimized to run efficiently on the vector computer NEC SX-6 at the Deutsches Klimarechenzentrum Hamburg, and effectivity tests yield the result that an effective application on superscalar architectures requires code adjustments, in particular in the advection module.

The following table shows CPU times $T$ for one year integration period with a conservative setup of TM3 release 3.8a for one tracer, ERA-15 meteorology and empty user-routines, using 1 CPU on the NEC SX-6 at the Deutsches Klimarechenzentrum Hamburg and on a standard PC:

<table>
<thead>
<tr>
<th>platform</th>
<th>grid res.</th>
<th>fg</th>
<th>vfg</th>
<th>xfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEC SX-6</td>
<td>9 min</td>
<td>2 h 50 min</td>
<td>12 h</td>
<td></td>
</tr>
<tr>
<td>PC AMD Athlon 1.3 GHz</td>
<td>1h 30 min</td>
<td>68 h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix A

Meteorological Data Preprocessing

A.1 Airmass Fluxes

TM3 requires as input fields the horizontal airmass fluxes crossing the gridbox boundaries during each meteorological time step. These fluxes are calculated from fields of geopotential heights, horizontal wind velocities and surface pressure. The data processing is performed in three steps described in the following subsections.

A.1.1 Interpolation/Integration of the Horizontal Mass Fluxes over the Grid Box Sides

For each instant of time of the original meteorological analyses the horizontal mass fluxes are obtained by vertical-meridional, respectively vertical-zonal integration of $\rho u_h$ using trapezoidal integration formulae (program MFLUX).

A.1.2 Ensuring Mass Conservation

After interpolating the horizontal mass flux fields and surface pressure the resulting fields are not exactly mass conserving any more. To restore this fundamental property a correction procedure is applied as follows.

The conservation of mass requires that the divergence of the airmass flux density, vertically integrated, equals the surface pressure tendency:

$$\frac{\partial p_s}{\partial t} = \frac{g}{\rho} \int_{z_{surf}}^{z_{top}} \nabla \cdot (\rho u_h) dz$$  \hspace{1cm} (A.1)

This is derived from combining the continuity equation with the hydrostatic approximation. Denoting by $\vec{F}$ the vertically integrated airmass flux density, equation (A.1) is reexpressed as

$$\frac{1}{g} \frac{\partial p_s}{\partial t} = \nabla \cdot \vec{F}.$$  \hspace{1cm} (A.2)

We split

$$\vec{F} = \vec{F}_{obs} + \vec{F}_{corr}$$  \hspace{1cm} (A.3)

where $\vec{F}_{obs}$ are the airmass flux densities as determined from the analyses and $\vec{F}_{corr}$ represents a correction to be determined in order to fulfil equation (A.2).
For convenience, we consider a time interval \( t \in [0, \Delta t] \). To compute a correction for \( \vec{F} \) only, we have to assume \( \partial p_s / \partial t \) as known, in fact we define

\[
\frac{\partial p_s}{\partial t} \bigg|_{t=\Delta t/2} = \frac{p_s(\Delta t) - p_s(0)}{\Delta t}.
\]

To be consistent, the airmass flux densities \( \vec{F} \) are also considered from now at \( t = \Delta t/2 \) by \( \vec{F}_{\text{obs}}(\Delta t/2) = \left( \vec{F}_{\text{obs}}(\Delta t) + \vec{F}_{\text{obs}}(0) \right) / 2 \) (time staggering).

Hence the correction flux equation becomes

\[
\nabla \cdot \vec{F}_{\text{corr}} = \frac{1}{g} \frac{p_s(\Delta t) - p_s(0)}{\Delta t} - \nabla \cdot \vec{F}_{\text{obs}}.
\]

This is only the divergence of \( \vec{F}_{\text{corr}} \). The rotational part of \( \vec{F}_{\text{corr}} \) is not defined by this and set to zero in order to obtain minimal corrections. We make a potential ansatz

\[\vec{F}_{\text{corr}} = \nabla \Theta, \quad (A.4)\]

and obtain a Poisson equation for the 2d field \( \Theta \):

\[
\Delta \Theta = \frac{1}{g} \frac{p_s(\Delta t) - p_s(0)}{\Delta t} - \nabla \cdot \vec{F}_{\text{obs}}, \quad (A.5)
\]

which has to be solved for on a horizontal grid spanning the entire globe. The boundary conditions are that the meridional component of the gradient of \( \Theta \) vanish at the poles and that \( \Theta \) be periodic in the zonal direction.

The discrete version of equation (A.5) is solved efficiently using two-dimensional Fourier transforms. This yields the vertically summed correction airmass flux field which is subsequently distributed in vertical direction proportionally to the absolute values of the x- and y-components of \( \vec{F}_{\text{obs}} \).

### A.1.3 Airmass Flux processing Programs

The three preprocessing steps described above are performed using the following programs:

- **MFLUX** Interpolation/integration of airmass fluxes
- **PASCHA** Time staggering and adjustment for conservation of mass
- **SUBSCALE** Interpolation of non-flux fields

Note that, for historical reasons, in SUBSCALE the direction of the vertical coordinate grid is reversed compared to TM3, i.e. the vertical levels start at the top of the atmosphere \((l = 1)\) and end at the surface \((l = l_{\text{surf}})\).

### A.2 Sub-Gridscloud Transport Information

TM3 requires as input global fields the vertical diffusion coefficient and the parameters for the cumulus cloud transport (entrainment and detrainment rates into up- and downdraft). These are calculated in the preprocessing stage from meteorological analyses of geopotential, surface pressure, horizontal wind, temperature and specific humidity. Furthermore surface fluxes of latent heat are needed.

All sub-gridscale transport parameters are calculated in every grid column of the original grid of the meteorological analyses. Subsequently the parameters are averaged onto the coarser grid of TM3.
APPENDIX A. METEOROLOGICAL DATA PREPROCESSING

A.2.1 Vertical Diffusion Coefficients

The stability dependent vertical diffusion coefficients are calculated using the formulae from [Louis, 1979] as implemented in the ECMWF operational model. The diffusion coefficient is given by

\[ K_v = l_h^2 \left| \frac{\partial u}{\partial z} \right| f_h(Ri) \]  \hspace{1cm} (A.6)

where \( Ri \) is the Richardson number. It is calculated at the layer boundary \( k + 1/2 \) from the analysis variables on the adjacent level centers

\[ Ri_{k+1/2} = g(z_k - z_{k+1}) \frac{c_{pd}(T_k - T_{k+1}) + g(z_k - z_{k+1})}{|\vec{u}|_{k+1/2}^2 c_{pd} T_{k+1/2}} \]  \hspace{1cm} (A.7)

\( l_h \) is a mixing length which is calculated as a function of height \( z \) above ground from

\[ l_h = \frac{kz}{1 + \frac{A_z}{\lambda_h}} \]  \hspace{1cm} (A.8)

The stability function \( f_h(Ri) \) is computed according to the sign of \( Ri \). In the stable case (\( Ri > 0 \)):

\[ f_h(Ri) = \frac{1}{1 + 3bRi\sqrt{1 + dRi}} \]  \hspace{1cm} (A.9)

In the unstable case (\( Ri <= 0 \)):

\[ f_h(Ri) = 1 - \frac{3bRi}{1 + G(Ri)} \]  \hspace{1cm} (A.10)

and the function \( G(Ri) \) is given by

\[ G(Ri) = 3bCt_h^2 \sqrt{\frac{-Ri}{z}} \left\{ \frac{1}{A_z^2} \left[ \left( 1 + \frac{A_z}{z} \right)^{1/3} - 1 \right] \right\}^3 \]  \hspace{1cm} (A.11)

\( \Delta z \) is the distance between the adjacent model layer centers.

Numerical values for the constants appearing in equations (A.8)-(A.11) are given in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>5.</td>
</tr>
<tr>
<td>( C )</td>
<td>5.</td>
</tr>
<tr>
<td>( d )</td>
<td>5.</td>
</tr>
<tr>
<td>( k )</td>
<td>0.4</td>
</tr>
<tr>
<td>( c_{pd} )</td>
<td>1005.46 J kg(^{-1}) K(^{-1})</td>
</tr>
<tr>
<td>( g )</td>
<td>9.80665 m s(^{-2})</td>
</tr>
<tr>
<td>( \lambda_h )</td>
<td>438.18 m</td>
</tr>
</tbody>
</table>
A.2.2 Cloud Transport Parameter Determination

The cloud transport parameters (ent- and detrainment rates into up- resp. downdraft) are calculated according to the scheme described in [Tiedke, 1989]. This scheme calculates the entrainment and detrainment rates in the sequence of steps described in the following subsections.

The scheme detects and parameterizes three types of convective clouds: deep or penetrative, shallow and mid-level convective clouds. Their presence in a model grid column is diagnosed by different conditions, and each has different cloud parameters (see table A.2).

The physical constants employed are listed in table A.1. All air properties of the environment are defined on the model level boundaries.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas constant of dry air</td>
<td>$R_d$</td>
<td>287.05</td>
<td>J kg$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td>Gas constant of water vapour</td>
<td>$R_v$</td>
<td>461.51</td>
<td>J kg$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\epsilon$</td>
<td>$R_d/R_v$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$C_p$</td>
<td>1005.46</td>
<td>J kg$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$L_v$</td>
<td>2.5008 $10^6$</td>
<td>J kg$^{-1}$</td>
</tr>
</tbody>
</table>

Table A.1: Physical constants used for the cloud calculations

<table>
<thead>
<tr>
<th>Cloudtype</th>
<th>D + ML</th>
<th>shallow</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_u$</td>
<td>$10^{-4}$</td>
<td>$3 \times 10^{-4}$</td>
<td>m$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$\delta_u$</td>
<td>$10^{-4}$</td>
<td>$3 \times 10^{-4}$</td>
<td>m$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$\epsilon_d$</td>
<td>$2 \times 10^{-4}$</td>
<td></td>
<td>m$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$\delta_d$</td>
<td>$2 \times 10^{-4}$</td>
<td></td>
<td>m$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.0</td>
<td>0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_p$</td>
<td>$2 \times 10^{-3}$</td>
<td></td>
<td>s$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$z_{min}$</td>
<td>1500</td>
<td></td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: Values of the parameters in the cloud parametrization, see text. D + ML: “deep and mid level”.

Detection of cloud base height

The base of a potential convective cloud is determined by lifting surface air dry adiabatically until saturation occurs. “Surface air” thereby is defined as an air parcel with the physical properties of the surface layer except for a tunable humidity, which can be selected between grid average humidity (parameter $\alpha = 0$) and 100% saturation (parameter $\alpha = 1$). The operational value is $\alpha = 0.2$, which was determined empirically by comparisons with cloud cover and height extent of the ERA-15 ECMWF analyses.
APPENDIX A. METEOROLOGICAL DATA PREPROCESSING

During the ascent, the temperature of the air parcel is modified from one layer boundary to the next:
\[ T_l = T_{l+1} - g(z_l - z_{l+1})/C_p \]  
(A.12)
while the moisture remains unchanged:
\[ q_l = q_{l+1} \]  
(A.13)

As soon as saturation is detected \( q > q_{sat} \) condensation is computed by solving the implicit equation for the temperature \( T_s \) of the air parcel at saturation:
\[ C_p(T - T_s) = L(q - q_{sat}(T_s, p)) \]  
(A.14)
where \( T, q \) are temperature and moisture of the air parcel prior to condensation. The solution of (A.14) is obtained iteratively by Newton’s method.

The buoyancy of the air parcel with respect to the environment is checked in terms of their virtual temperature, defined as:
\[ T_v = T(1 + q \frac{1 - \epsilon}{\epsilon}) \]  
(A.15)
If the parcel is buoyant, a cloud is detected, else there is no convective or shallow cloud in that particular grid column.

**Determination of the updraft air flux at the cloud base**

If a cloud base exists, the vertically integrated horizontal water vapour convergence below the cloud is computed:
\[ \text{CONV} = - \int_{p=p_s}^{p=p_{cb}} \nabla \cdot \rho v q dp \]  
(A.16)
where \( p_{cb} \) denotes the pressure of the cloud base. If this convergence is positive, a penetrative or a midlevel convective cloud exists, else a shallow convection cloud exists.

If a cloud is detected, the water vapour flux at the base of the cloud is determined by adding to the large scale water vapour convergence below the cloud the evaporation from the surface. Since also the downdraft (if existing) delivers moisture to the below cloud air volume, this contribution is included in an iterative way by computing firstly the cloud properties without the contribution from the downdraft. In a second iteration the moisture contribution from the downdraft is included. The airmass flux at the base of the cloud is then determined as the ratio between the water vapour flux at the base of the cloud and the saturation specific humidity at the temperature of the base of the cloud.

Finally, detrainment rates into the updraft below the cloud base are set to zero and entrainment rates below the cloud are set proportional to the below cloud layer thicknesses, such that the sum of the below cloud entrainment rates equal the massflux at the base of the cloud.

**Computation of cloud properties in up- and downdraft**

Cloud temperature, moisture and liquid water content are initialized with values at the base of the cloud. Subsequently the air parcel ascent is followed by solving the equations
A.2. SUB-GRIDSCALE TRANSPORT INFORMATION

(8) in [Tiedke, 1989] for the updraft mass flux \( F_u \), dry static energy \( s_u = C_p T_u + g z \), moisture \( q_u \) and liquid water content \( l_u \):

\[
\begin{align*}
\frac{\partial F_u}{\partial z} &= E_u - D_u \\
\frac{\partial F_u s_u}{\partial z} &= E_u \bar{s} - D_u s_u + L \bar{\rho} c_u \\
\frac{\partial F_u q_u}{\partial z} &= E_u \bar{q} - D_u q_u - \bar{\rho} c_u \\
\frac{\partial F_u l_u}{\partial z} &= -D_u l_u + \bar{\rho} c_u - \bar{\rho} G_p
\end{align*}
\]

(A.17) (A.18) (A.19) (A.20)

where \( c_u \) is the net condensation produced in the updraft. Overbars indicate environmental values. Inserting equation (A.17) into the equations for temperature, moisture and liquid water content yields:

\[
\begin{align*}
F_u \frac{\partial s_u}{\partial z} &= E_u (\bar{s} - s_u) + L \bar{\rho} c_u \\
F_u \frac{\partial q_u}{\partial z} &= E_u (\bar{q} - q_u) - \bar{\rho} c_u \\
F_u \frac{\partial l_u}{\partial z} &= -E_u l_u + \bar{\rho} c_u - \bar{\rho} G_p
\end{align*}
\]

(A.21) (A.22) (A.23)

which are solved in discretized form going from one layer boundary to the next above. Similar equations are used for the downdraft computation (equations 17 in [Tiedke, 1989]). In each cloud layer the following sequence is computed:

1. Air entrainment and detrainment rates are set proportional to the updraft mass flux with constants \( \epsilon_u \) and \( \delta_u \) as given in table A.2:

\[
E_u = \epsilon_u F_u \Delta z \\
D_u = \delta_u F_u \Delta z
\]

(A.24) (A.25)

where \( F_u \) denotes the updraft mass flux at the lower boundary of the layer and \( \Delta z \) the layer thickness (in m).

2. Cloud air parcel temperature, moisture and liquid water content are first adiabatically lifted and then adjusted by the entrainment of environmental air. This results in preliminary values:

\[
T_l = T_{l+1} - g \Delta z / C_p + \epsilon_u \Delta z (\bar{T}_l - T_{l+1}) \\
q_l = q_{l+1} + \epsilon_u \Delta z (\bar{q}_l - q_{l+1}) \\
l_l = l_{l+1} + \epsilon_u \Delta z (-q_{l+1})
\]

(A.26) (A.27) (A.28)

3. Condensation is computed and the parcel temperature increased to force the cloud water vapour at saturation. This is performed with the same iteration procedure as described above (equation (A.14)). The condensate, \( q - q_{sat}(T_s, p) \), is added to the liquid water content of the cloud air parcel.
4. If the cloud layer is higher than \( z_m \) above the cloud base, precipitation rate out of the layer is computed from

\[
G_p = k_p l F_u
\]  

(A.29)

and the liquid water content adjusted using an implicit formulation in order to prevent negative liquid water values (superscript +, resp. - indicate values after, resp. before adjustment):

\[
l^+ = l^- - l^- \frac{\bar{\rho} G_p}{1 + \bar{\rho} G_p}
\]  

(A.30)

5. Check for level of free sinking (LFS) where the cumulus downdraft starts. The LFS is assumed to be the highest model level where a mixture of equal parts of cloud air and environmental air (at wet bulb temperature) becomes negative buoyant with respect to environmental air.

6. Compute downdraft if an LFS is detected and and only if there is precipitation (i.e. above \( z_{min} \)). This is performed similar to the calculations described above with an initial downdraft mass flux at the LFS of \( \gamma F_{u,cb} \). Ent- and detrainment rates are defined by the constants \( \epsilon_d \) and \( \delta_d \). The downdraft air parcel is forced to remain at saturation by evaporation of liquid water. Below the cloud base, the downdraft air mass flux is detrained in all layers proportional to the layer thicknesses.

7. The ascent of the updraft cloud parcel is continued above the LFS if the air parcel is still buoyant.

If the cloud parcel is no longer buoyant then the remaining airmass flux is detrained in the two layers above the cloud with a fraction of \( \beta \) in the first and \( (1 - \beta) \) in the second layer to simulate the effect of cloud overshoots.

A.2.3 Programs

The parameters of the sub-gridscale transport are calculated in the following programs and subroutines:

- **SUBSCALE**: main program
- **louis**: calculates vertical diffusion coefficients
- **cloud**: calculates cloud ent- and detrainment rates

Note that, for historical reasons, within this program the direction of the vertical coordinate grid is reversed compared to TM3, i.e. the vertical levels start at the top of the atmosphere \( (l = 1) \) and end at the surface \( (l = l_{surf}) \)!
Appendix B

Horizontal Grid Layouts

Inside the plots, the indices in x- and y-direction are for those grid boxes that have the next left/lowest bold reference line as left/lower boundary. See the next pages.

The auxiliary subroutines coords2indices and indices2coords in the Fortran module aux_grid can be used for converting TM3 grid indices to real-world coordinates and vice versa.
APPENDIX B. HORIZONTAL GRID LAYOUTS

Grid layout for resolution cg
Table B.1: Western/southern grid box boundaries for the TM3 cg grid (36x24L9).

<table>
<thead>
<tr>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>175.00</td>
<td>(-90.00)</td>
<td>10</td>
<td>-95.00</td>
<td>-23.48</td>
<td>19</td>
<td>-5.00</td>
<td>46.96</td>
<td>28</td>
<td>85.00</td>
</tr>
<tr>
<td>2</td>
<td>-175.00</td>
<td>-86.09</td>
<td>11</td>
<td>-85.00</td>
<td>-15.65</td>
<td>20</td>
<td>5.00</td>
<td>54.78</td>
<td>29</td>
<td>95.00</td>
</tr>
<tr>
<td>3</td>
<td>-165.00</td>
<td>-78.26</td>
<td>12</td>
<td>-75.00</td>
<td>-7.83</td>
<td>21</td>
<td>15.00</td>
<td>62.61</td>
<td>30</td>
<td>105.00</td>
</tr>
<tr>
<td>4</td>
<td>-155.00</td>
<td>-70.43</td>
<td>13</td>
<td>-65.00</td>
<td>0.00</td>
<td>22</td>
<td>25.00</td>
<td>70.43</td>
<td>31</td>
<td>115.00</td>
</tr>
<tr>
<td>5</td>
<td>-145.00</td>
<td>-62.61</td>
<td>14</td>
<td>-55.00</td>
<td>7.83</td>
<td>23</td>
<td>35.00</td>
<td>78.26</td>
<td>32</td>
<td>125.00</td>
</tr>
<tr>
<td>6</td>
<td>-135.00</td>
<td>-54.78</td>
<td>15</td>
<td>-45.00</td>
<td>15.65</td>
<td>24</td>
<td>45.00</td>
<td>86.09</td>
<td>33</td>
<td>135.00</td>
</tr>
<tr>
<td>7</td>
<td>-125.00</td>
<td>-46.96</td>
<td>16</td>
<td>-35.00</td>
<td>23.48</td>
<td>25</td>
<td>55.00</td>
<td>94.00</td>
<td>34</td>
<td>145.00</td>
</tr>
<tr>
<td>8</td>
<td>-115.00</td>
<td>-39.13</td>
<td>17</td>
<td>-25.00</td>
<td>31.30</td>
<td>26</td>
<td>65.00</td>
<td>105.00</td>
<td>35</td>
<td>155.00</td>
</tr>
<tr>
<td>9</td>
<td>-105.00</td>
<td>-31.30</td>
<td>18</td>
<td>-15.00</td>
<td>39.13</td>
<td>27</td>
<td>75.00</td>
<td>115.00</td>
<td>36</td>
<td>165.00</td>
</tr>
</tbody>
</table>
APPENDIX B. HORIZONTAL GRID LAYOUTS

Grid layout for resolution fg
<table>
<thead>
<tr>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>177.50</td>
<td>(-90.00)</td>
<td>19</td>
<td>-92.50</td>
<td>-22.98</td>
<td>37</td>
<td>-2.50</td>
<td>45.96</td>
<td>55</td>
<td>87.50</td>
</tr>
<tr>
<td>2</td>
<td>-177.50</td>
<td>-88.09</td>
<td>20</td>
<td>-87.50</td>
<td>-19.15</td>
<td>38</td>
<td>2.50</td>
<td>49.79</td>
<td>56</td>
<td>92.50</td>
</tr>
<tr>
<td>3</td>
<td>-172.50</td>
<td>-84.26</td>
<td>21</td>
<td>-82.50</td>
<td>-15.32</td>
<td>39</td>
<td>7.50</td>
<td>53.62</td>
<td>57</td>
<td>97.50</td>
</tr>
<tr>
<td>4</td>
<td>-167.50</td>
<td>-80.43</td>
<td>22</td>
<td>-77.50</td>
<td>-11.49</td>
<td>40</td>
<td>12.50</td>
<td>57.45</td>
<td>58</td>
<td>102.50</td>
</tr>
<tr>
<td>5</td>
<td>-162.50</td>
<td>-76.60</td>
<td>23</td>
<td>-72.50</td>
<td>-7.66</td>
<td>41</td>
<td>17.50</td>
<td>61.28</td>
<td>59</td>
<td>107.50</td>
</tr>
<tr>
<td>6</td>
<td>-157.50</td>
<td>-72.77</td>
<td>24</td>
<td>-67.50</td>
<td>-3.83</td>
<td>42</td>
<td>22.50</td>
<td>65.11</td>
<td>60</td>
<td>112.50</td>
</tr>
<tr>
<td>7</td>
<td>-152.50</td>
<td>-68.94</td>
<td>25</td>
<td>-62.50</td>
<td>0.00</td>
<td>43</td>
<td>27.50</td>
<td>68.94</td>
<td>61</td>
<td>117.50</td>
</tr>
<tr>
<td>8</td>
<td>-147.50</td>
<td>-65.11</td>
<td>26</td>
<td>-57.50</td>
<td>3.83</td>
<td>44</td>
<td>32.50</td>
<td>72.77</td>
<td>62</td>
<td>122.50</td>
</tr>
<tr>
<td>9</td>
<td>-142.50</td>
<td>-61.28</td>
<td>27</td>
<td>-52.50</td>
<td>7.66</td>
<td>45</td>
<td>37.50</td>
<td>76.60</td>
<td>63</td>
<td>127.50</td>
</tr>
<tr>
<td>10</td>
<td>-137.50</td>
<td>-57.45</td>
<td>28</td>
<td>-47.50</td>
<td>11.49</td>
<td>46</td>
<td>42.50</td>
<td>80.43</td>
<td>64</td>
<td>132.50</td>
</tr>
<tr>
<td>11</td>
<td>-132.50</td>
<td>-53.62</td>
<td>29</td>
<td>-42.50</td>
<td>15.32</td>
<td>47</td>
<td>47.50</td>
<td>84.26</td>
<td>65</td>
<td>137.50</td>
</tr>
<tr>
<td>12</td>
<td>-127.50</td>
<td>-49.79</td>
<td>30</td>
<td>-37.50</td>
<td>19.15</td>
<td>48</td>
<td>52.50</td>
<td>88.09</td>
<td>66</td>
<td>142.50</td>
</tr>
<tr>
<td>13</td>
<td>-122.50</td>
<td>-45.96</td>
<td>31</td>
<td>-32.50</td>
<td>22.98</td>
<td>49</td>
<td>57.50</td>
<td>92.81</td>
<td>67</td>
<td>147.50</td>
</tr>
<tr>
<td>14</td>
<td>-117.50</td>
<td>-42.13</td>
<td>32</td>
<td>-27.50</td>
<td>26.81</td>
<td>50</td>
<td>62.50</td>
<td>96.62</td>
<td>68</td>
<td>152.50</td>
</tr>
<tr>
<td>15</td>
<td>-112.50</td>
<td>-38.30</td>
<td>33</td>
<td>-22.50</td>
<td>30.64</td>
<td>51</td>
<td>67.50</td>
<td>100.43</td>
<td>69</td>
<td>157.50</td>
</tr>
<tr>
<td>16</td>
<td>-107.50</td>
<td>-34.47</td>
<td>34</td>
<td>-17.50</td>
<td>34.47</td>
<td>52</td>
<td>72.50</td>
<td>104.24</td>
<td>70</td>
<td>162.50</td>
</tr>
<tr>
<td>17</td>
<td>-102.50</td>
<td>-30.64</td>
<td>35</td>
<td>-12.50</td>
<td>38.30</td>
<td>53</td>
<td>77.50</td>
<td>108.04</td>
<td>71</td>
<td>167.50</td>
</tr>
<tr>
<td>18</td>
<td>-97.50</td>
<td>-26.81</td>
<td>36</td>
<td>-7.50</td>
<td>42.13</td>
<td>54</td>
<td>82.50</td>
<td>111.84</td>
<td>72</td>
<td>172.50</td>
</tr>
</tbody>
</table>

Table B.2: Western/southern grid box boundaries for the TM3 fg grid (72x48L19).
APPENDIX B. HORIZONTAL GRID LAYOUTS

Grid layout for resolution vfg
<table>
<thead>
<tr>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>179.06</td>
<td>-90.00</td>
<td>2</td>
<td>179.06</td>
<td>-89.05</td>
<td>3</td>
<td>177.19</td>
<td>-87.16</td>
<td>4</td>
<td>175.31</td>
</tr>
<tr>
<td>5</td>
<td>173.44</td>
<td>-83.37</td>
<td>6</td>
<td>171.56</td>
<td>-81.47</td>
<td>7</td>
<td>169.69</td>
<td>-79.58</td>
<td>8</td>
<td>167.81</td>
</tr>
<tr>
<td>11</td>
<td>165.94</td>
<td>-77.68</td>
<td>12</td>
<td>164.06</td>
<td>-75.79</td>
<td>13</td>
<td>162.19</td>
<td>-73.89</td>
<td>14</td>
<td>160.31</td>
</tr>
<tr>
<td>17</td>
<td>158.44</td>
<td>-72.00</td>
<td>18</td>
<td>156.56</td>
<td>-70.11</td>
<td>19</td>
<td>154.69</td>
<td>-68.21</td>
<td>20</td>
<td>152.81</td>
</tr>
<tr>
<td>25</td>
<td>146.06</td>
<td>-64.42</td>
<td>26</td>
<td>143.44</td>
<td>-62.53</td>
<td>27</td>
<td>141.56</td>
<td>-60.63</td>
<td>28</td>
<td>139.69</td>
</tr>
<tr>
<td>33</td>
<td>137.81</td>
<td>-58.74</td>
<td>34</td>
<td>135.94</td>
<td>-56.83</td>
<td>35</td>
<td>134.06</td>
<td>-54.95</td>
<td>36</td>
<td>132.19</td>
</tr>
<tr>
<td>41</td>
<td>129.31</td>
<td>-52.05</td>
<td>42</td>
<td>127.44</td>
<td>-50.16</td>
<td>43</td>
<td>125.56</td>
<td>-48.26</td>
<td>44</td>
<td>123.69</td>
</tr>
<tr>
<td>49</td>
<td>121.79</td>
<td>-46.37</td>
<td>50</td>
<td>119.91</td>
<td>-44.47</td>
<td>51</td>
<td>118.03</td>
<td>-42.58</td>
<td>52</td>
<td>116.14</td>
</tr>
<tr>
<td>57</td>
<td>114.25</td>
<td>-40.69</td>
<td>58</td>
<td>112.37</td>
<td>-38.80</td>
<td>59</td>
<td>110.48</td>
<td>-37.02</td>
<td>60</td>
<td>108.60</td>
</tr>
<tr>
<td>65</td>
<td>106.71</td>
<td>-35.23</td>
<td>66</td>
<td>104.83</td>
<td>-33.45</td>
<td>67</td>
<td>102.94</td>
<td>-31.67</td>
<td>68</td>
<td>101.05</td>
</tr>
<tr>
<td>73</td>
<td>99.16</td>
<td>-29.89</td>
<td>74</td>
<td>97.27</td>
<td>-28.11</td>
<td>75</td>
<td>95.38</td>
<td>-26.33</td>
<td>76</td>
<td>93.49</td>
</tr>
<tr>
<td>81</td>
<td>91.60</td>
<td>-24.55</td>
<td>82</td>
<td>89.71</td>
<td>-22.77</td>
<td>83</td>
<td>87.82</td>
<td>-20.99</td>
<td>84</td>
<td>85.93</td>
</tr>
<tr>
<td>89</td>
<td>84.04</td>
<td>-19.21</td>
<td>90</td>
<td>82.15</td>
<td>-17.43</td>
<td>91</td>
<td>80.26</td>
<td>-15.65</td>
<td>92</td>
<td>78.37</td>
</tr>
<tr>
<td>97</td>
<td>76.48</td>
<td>-13.87</td>
<td>98</td>
<td>74.59</td>
<td>-12.09</td>
<td>99</td>
<td>72.70</td>
<td>-10.31</td>
<td>100</td>
<td>70.81</td>
</tr>
<tr>
<td>105</td>
<td>68.92</td>
<td>-8.53</td>
<td>106</td>
<td>67.03</td>
<td>-6.75</td>
<td>107</td>
<td>65.14</td>
<td>-4.97</td>
<td>108</td>
<td>63.25</td>
</tr>
<tr>
<td>113</td>
<td>61.36</td>
<td>-3.19</td>
<td>114</td>
<td>59.47</td>
<td>-1.41</td>
<td>115</td>
<td>57.58</td>
<td>-0.63</td>
<td>116</td>
<td>55.69</td>
</tr>
<tr>
<td>121</td>
<td>53.80</td>
<td>0.21</td>
<td>122</td>
<td>51.91</td>
<td>1.93</td>
<td>123</td>
<td>49.02</td>
<td>3.15</td>
<td>124</td>
<td>47.13</td>
</tr>
<tr>
<td>129</td>
<td>45.24</td>
<td>4.37</td>
<td>130</td>
<td>43.35</td>
<td>5.59</td>
<td>131</td>
<td>41.46</td>
<td>6.81</td>
<td>132</td>
<td>39.57</td>
</tr>
<tr>
<td>137</td>
<td>37.68</td>
<td>8.03</td>
<td>138</td>
<td>35.79</td>
<td>9.25</td>
<td>139</td>
<td>33.90</td>
<td>10.47</td>
<td>140</td>
<td>32.01</td>
</tr>
<tr>
<td>145</td>
<td>29.32</td>
<td>11.69</td>
<td>146</td>
<td>27.43</td>
<td>12.91</td>
<td>147</td>
<td>25.54</td>
<td>14.13</td>
<td>148</td>
<td>23.65</td>
</tr>
</tbody>
</table>

Table B.3: Western/southern grid box boundaries for the TM3 vfg grid (192x96L31/L28).
<table>
<thead>
<tr>
<th>i/j lon</th>
<th>lat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 179.44 (-90.00)</td>
<td>81 -90.56 -0.56</td>
</tr>
<tr>
<td>2 -179.44 -89.44</td>
<td>82 -89.44 0.56</td>
</tr>
<tr>
<td>3 -178.31 -88.31</td>
<td>83 -88.31 1.69</td>
</tr>
<tr>
<td>4 -177.19 -87.19</td>
<td>84 -87.19 2.81</td>
</tr>
<tr>
<td>5 -176.06 -86.06</td>
<td>85 -86.06 3.94</td>
</tr>
<tr>
<td>6 -174.94 -84.94</td>
<td>86 -84.94 5.06</td>
</tr>
<tr>
<td>7 -173.81 -83.81</td>
<td>87 -83.81 6.19</td>
</tr>
<tr>
<td>8 -172.69 -82.69</td>
<td>88 -82.69 7.31</td>
</tr>
<tr>
<td>9 -171.56 -81.56</td>
<td>89 -81.56 8.44</td>
</tr>
<tr>
<td>10 -170.44 -80.44</td>
<td>90 -80.44 9.56</td>
</tr>
<tr>
<td>11 -169.31 -79.31</td>
<td>91 -79.31 10.69</td>
</tr>
<tr>
<td>12 -168.19 -78.19</td>
<td>92 -78.19 11.81</td>
</tr>
<tr>
<td>13 -167.06 -77.06</td>
<td>93 -77.06 12.94</td>
</tr>
<tr>
<td>14 -165.94 -75.94</td>
<td>94 -75.94 14.06</td>
</tr>
<tr>
<td>15 -164.81 -74.81</td>
<td>95 -74.81 15.19</td>
</tr>
<tr>
<td>16 -163.69 -73.69</td>
<td>96 -73.69 16.31</td>
</tr>
<tr>
<td>17 -162.57 -72.56</td>
<td>97 -72.56 17.44</td>
</tr>
<tr>
<td>18 -161.44 -71.44</td>
<td>98 -71.44 18.56</td>
</tr>
<tr>
<td>19 -160.31 -70.31</td>
<td>99 -70.31 19.69</td>
</tr>
<tr>
<td>20 -159.19 -69.19</td>
<td>100 -69.19 20.81</td>
</tr>
<tr>
<td>21 -158.06 -68.06</td>
<td>101 -68.06 21.94</td>
</tr>
<tr>
<td>22 -156.94 -66.94</td>
<td>102 -66.94 23.06</td>
</tr>
<tr>
<td>23 -155.81 -65.81</td>
<td>103 -65.81 24.19</td>
</tr>
<tr>
<td>24 -154.69 -64.69</td>
<td>104 -64.69 25.31</td>
</tr>
<tr>
<td>25 -153.56 -63.56</td>
<td>105 -63.56 26.44</td>
</tr>
<tr>
<td>26 -152.44 -62.44</td>
<td>106 -62.44 27.56</td>
</tr>
<tr>
<td>27 -151.31 -61.31</td>
<td>107 -61.31 28.68</td>
</tr>
<tr>
<td>28 -150.19 -60.19</td>
<td>108 -60.19 29.81</td>
</tr>
<tr>
<td>29 -149.06 -59.06</td>
<td>109 -59.06 30.94</td>
</tr>
<tr>
<td>30 -147.94 -57.94</td>
<td>110 -57.94 32.06</td>
</tr>
<tr>
<td>31 -146.81 -56.81</td>
<td>111 -56.81 33.19</td>
</tr>
<tr>
<td>32 -145.69 -55.69</td>
<td>112 -55.69 34.31</td>
</tr>
<tr>
<td>33 -144.56 -54.56</td>
<td>113 -54.56 35.44</td>
</tr>
<tr>
<td>34 -143.44 -53.44</td>
<td>114 -53.44 36.56</td>
</tr>
<tr>
<td>35 -142.31 -52.31</td>
<td>115 -52.31 37.69</td>
</tr>
<tr>
<td>36 -141.19 -51.19</td>
<td>116 -51.19 38.81</td>
</tr>
<tr>
<td>37 -140.06 -50.06</td>
<td>117 -50.06 39.94</td>
</tr>
<tr>
<td>38 -138.94 -49.84</td>
<td>118 -48.94 41.06</td>
</tr>
<tr>
<td>39 -137.81 -48.71</td>
<td>119 -47.81 42.19</td>
</tr>
<tr>
<td>40 -136.69 -47.69</td>
<td>120 -46.69 43.31</td>
</tr>
<tr>
<td>41 -135.56 -46.56</td>
<td>121 -45.56 44.44</td>
</tr>
<tr>
<td>42 -134.44 -45.44</td>
<td>122 -44.44 45.56</td>
</tr>
<tr>
<td>43 -133.31 -44.31</td>
<td>123 -43.31 46.69</td>
</tr>
<tr>
<td>44 -132.19 -43.21</td>
<td>124 -42.19 47.81</td>
</tr>
<tr>
<td>45 -131.06 -42.19</td>
<td>125 -41.06 48.94</td>
</tr>
<tr>
<td>46 -129.94 -41.06</td>
<td>126 -40.94 50.06</td>
</tr>
<tr>
<td>47 -128.81 -40.06</td>
<td>127 -39.81 51.19</td>
</tr>
<tr>
<td>48 -127.69 -39.06</td>
<td>128 -38.81 52.31</td>
</tr>
<tr>
<td>49 -126.56 -38.06</td>
<td>129 -37.69 53.44</td>
</tr>
<tr>
<td>50 -125.44 -37.06</td>
<td>130 -36.56 54.56</td>
</tr>
<tr>
<td>51 -124.31 -36.06</td>
<td>131 -35.14 55.69</td>
</tr>
<tr>
<td>52 -123.19 -35.06</td>
<td>132 -34.19 56.81</td>
</tr>
<tr>
<td>53 -122.06 -34.06</td>
<td>133 -33.06 57.94</td>
</tr>
<tr>
<td>54 -120.94 -33.06</td>
<td>134 -32.06 59.06</td>
</tr>
<tr>
<td>55 -119.81 -32.06</td>
<td>135 -31.06 60.19</td>
</tr>
<tr>
<td>56 -118.69 -31.06</td>
<td>136 -30.06 61.31</td>
</tr>
<tr>
<td>57 -117.56 -30.06</td>
<td>137 -29.06 62.44</td>
</tr>
<tr>
<td>58 -116.44 -29.06</td>
<td>138 -28.06 63.56</td>
</tr>
<tr>
<td>59 -115.31 -28.06</td>
<td>139 -27.06 64.69</td>
</tr>
<tr>
<td>60 -114.19 -27.06</td>
<td>140 -26.06 65.81</td>
</tr>
</tbody>
</table>
### APPENDIX B. HORIZONTAL GRID LAYOUTS

<table>
<thead>
<tr>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
<th>lat [°]</th>
<th>i/j</th>
<th>lon [°]</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>-113.06</td>
<td>-23.06</td>
<td>141</td>
<td>-23.06</td>
<td>66.94</td>
<td>221</td>
<td>66.94</td>
<td></td>
<td>301</td>
<td>156.94</td>
</tr>
<tr>
<td>62</td>
<td>-111.94</td>
<td>-21.94</td>
<td>142</td>
<td>-21.94</td>
<td>68.06</td>
<td>222</td>
<td>68.06</td>
<td></td>
<td>302</td>
<td>158.06</td>
</tr>
<tr>
<td>63</td>
<td>-110.81</td>
<td>-20.81</td>
<td>143</td>
<td>-20.81</td>
<td>69.19</td>
<td>223</td>
<td>69.19</td>
<td></td>
<td>303</td>
<td>159.19</td>
</tr>
<tr>
<td>64</td>
<td>-109.69</td>
<td>-19.69</td>
<td>144</td>
<td>-19.69</td>
<td>70.31</td>
<td>224</td>
<td>70.31</td>
<td></td>
<td>304</td>
<td>160.31</td>
</tr>
<tr>
<td>65</td>
<td>-108.56</td>
<td>-18.56</td>
<td>145</td>
<td>-18.56</td>
<td>71.44</td>
<td>225</td>
<td>71.44</td>
<td></td>
<td>305</td>
<td>161.44</td>
</tr>
<tr>
<td>66</td>
<td>-107.44</td>
<td>-17.44</td>
<td>146</td>
<td>-17.44</td>
<td>72.56</td>
<td>226</td>
<td>72.56</td>
<td></td>
<td>306</td>
<td>162.56</td>
</tr>
<tr>
<td>67</td>
<td>-106.31</td>
<td>-16.31</td>
<td>147</td>
<td>-16.31</td>
<td>73.69</td>
<td>227</td>
<td>73.69</td>
<td></td>
<td>307</td>
<td>163.69</td>
</tr>
<tr>
<td>68</td>
<td>-105.19</td>
<td>-15.19</td>
<td>148</td>
<td>-15.19</td>
<td>74.81</td>
<td>228</td>
<td>74.81</td>
<td></td>
<td>308</td>
<td>164.81</td>
</tr>
<tr>
<td>69</td>
<td>-104.06</td>
<td>-14.06</td>
<td>149</td>
<td>-14.06</td>
<td>75.94</td>
<td>229</td>
<td>75.94</td>
<td></td>
<td>309</td>
<td>165.94</td>
</tr>
<tr>
<td>70</td>
<td>-102.94</td>
<td>-12.94</td>
<td>150</td>
<td>-12.94</td>
<td>77.06</td>
<td>230</td>
<td>77.06</td>
<td></td>
<td>310</td>
<td>167.06</td>
</tr>
<tr>
<td>71</td>
<td>-101.81</td>
<td>-11.81</td>
<td>151</td>
<td>-11.81</td>
<td>78.19</td>
<td>231</td>
<td>78.19</td>
<td></td>
<td>311</td>
<td>168.19</td>
</tr>
<tr>
<td>72</td>
<td>-100.69</td>
<td>-10.69</td>
<td>152</td>
<td>-10.69</td>
<td>79.31</td>
<td>232</td>
<td>79.31</td>
<td></td>
<td>312</td>
<td>169.31</td>
</tr>
<tr>
<td>73</td>
<td>-99.56</td>
<td>-9.56</td>
<td>153</td>
<td>-9.56</td>
<td>80.44</td>
<td>233</td>
<td>80.44</td>
<td></td>
<td>313</td>
<td>170.44</td>
</tr>
<tr>
<td>74</td>
<td>-98.44</td>
<td>-8.44</td>
<td>154</td>
<td>-8.44</td>
<td>81.56</td>
<td>234</td>
<td>81.56</td>
<td></td>
<td>314</td>
<td>171.56</td>
</tr>
<tr>
<td>75</td>
<td>-97.31</td>
<td>-7.31</td>
<td>155</td>
<td>-7.31</td>
<td>82.69</td>
<td>235</td>
<td>82.69</td>
<td></td>
<td>315</td>
<td>172.69</td>
</tr>
<tr>
<td>76</td>
<td>-96.19</td>
<td>-6.19</td>
<td>156</td>
<td>-6.19</td>
<td>83.81</td>
<td>236</td>
<td>83.81</td>
<td></td>
<td>316</td>
<td>173.81</td>
</tr>
<tr>
<td>77</td>
<td>-95.06</td>
<td>-5.06</td>
<td>157</td>
<td>-5.06</td>
<td>84.94</td>
<td>237</td>
<td>84.94</td>
<td></td>
<td>317</td>
<td>174.94</td>
</tr>
<tr>
<td>78</td>
<td>-93.94</td>
<td>-3.94</td>
<td>158</td>
<td>-3.94</td>
<td>86.06</td>
<td>238</td>
<td>86.06</td>
<td></td>
<td>318</td>
<td>176.06</td>
</tr>
<tr>
<td>79</td>
<td>-92.81</td>
<td>-2.81</td>
<td>159</td>
<td>-2.81</td>
<td>87.19</td>
<td>239</td>
<td>87.19</td>
<td></td>
<td>319</td>
<td>177.19</td>
</tr>
<tr>
<td>80</td>
<td>-91.69</td>
<td>-1.69</td>
<td>160</td>
<td>-1.69</td>
<td>88.31</td>
<td>240</td>
<td>88.31</td>
<td></td>
<td>320</td>
<td>178.31</td>
</tr>
</tbody>
</table>

Table B.4: Western/southern grid box boundaries for the TM3 xfg grid (320x161L31).
Appendix C

Vertical Coordinate Tables

The following tables contain for each TM3 grid resolution the yearly and globally averaged pressures and heights for the TM3 model layer midpoints and layer boundaries. \( l \) denotes the TM3 layer number (— for a boundary), \( \eta \) the vertical hybrid coordinate \( \eta = a/p_0 + b \) (reference pressure \( p_0 = 985 \) hPa), and \( p \) and \( h \) the respective time- and space-averaged pressures and heights.

cg grid, met. fields: NCEP/NCAR reanalysis 1998

<table>
<thead>
<tr>
<th>( l )</th>
<th>( \eta )</th>
<th>( p/\text{hPa} )</th>
<th>( h/\text{m} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>0.000</td>
<td>0</td>
<td>81057.5</td>
</tr>
<tr>
<td>9</td>
<td>0.030</td>
<td>30</td>
<td>49686.0</td>
</tr>
<tr>
<td>—</td>
<td>0.061</td>
<td>60</td>
<td>18388.9</td>
</tr>
<tr>
<td>8</td>
<td>0.103</td>
<td>102</td>
<td>15914.1</td>
</tr>
<tr>
<td>—</td>
<td>0.146</td>
<td>143</td>
<td>13485.9</td>
</tr>
<tr>
<td>7</td>
<td>0.200</td>
<td>197</td>
<td>12044.5</td>
</tr>
<tr>
<td>—</td>
<td>0.255</td>
<td>251</td>
<td>10651.1</td>
</tr>
<tr>
<td>6</td>
<td>0.325</td>
<td>320</td>
<td>9357.8</td>
</tr>
<tr>
<td>—</td>
<td>0.396</td>
<td>390</td>
<td>8107.8</td>
</tr>
<tr>
<td>5</td>
<td>0.478</td>
<td>471</td>
<td>6483.0</td>
</tr>
<tr>
<td>—</td>
<td>0.560</td>
<td>552</td>
<td>4890.2</td>
</tr>
<tr>
<td>4</td>
<td>0.650</td>
<td>641</td>
<td>3726.8</td>
</tr>
<tr>
<td>—</td>
<td>0.740</td>
<td>729</td>
<td>2582.3</td>
</tr>
<tr>
<td>3</td>
<td>0.806</td>
<td>794</td>
<td>1866.2</td>
</tr>
<tr>
<td>—</td>
<td>0.871</td>
<td>858</td>
<td>1159.9</td>
</tr>
<tr>
<td>2</td>
<td>0.908</td>
<td>895</td>
<td>766.8</td>
</tr>
<tr>
<td>—</td>
<td>0.945</td>
<td>931</td>
<td>378.0</td>
</tr>
<tr>
<td>1</td>
<td>0.972</td>
<td>958</td>
<td>188.4</td>
</tr>
<tr>
<td>—</td>
<td>1.000</td>
<td>985</td>
<td>0.0</td>
</tr>
</tbody>
</table>
APPENDIX C. VERTICAL COORDINATE TABLES

fg grid, met. fields: NCEP/NCAR reanalysis 1998

<table>
<thead>
<tr>
<th>l</th>
<th>η</th>
<th>p/hPa</th>
<th>h/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>0.000</td>
<td>0</td>
<td>81058</td>
</tr>
<tr>
<td>19</td>
<td>0.010</td>
<td>10</td>
<td>53062</td>
</tr>
<tr>
<td>—</td>
<td>0.020</td>
<td>20</td>
<td>25094</td>
</tr>
<tr>
<td>18</td>
<td>0.030</td>
<td>30</td>
<td>22698</td>
</tr>
<tr>
<td>—</td>
<td>0.041</td>
<td>40</td>
<td>20317</td>
</tr>
<tr>
<td>17</td>
<td>0.051</td>
<td>50</td>
<td>19345</td>
</tr>
<tr>
<td>—</td>
<td>0.061</td>
<td>60</td>
<td>18389</td>
</tr>
<tr>
<td>16</td>
<td>0.081</td>
<td>80</td>
<td>17507</td>
</tr>
<tr>
<td>—</td>
<td>0.102</td>
<td>100</td>
<td>16641</td>
</tr>
<tr>
<td>15</td>
<td>0.112</td>
<td>111</td>
<td>15822</td>
</tr>
<tr>
<td>—</td>
<td>0.123</td>
<td>121</td>
<td>15020</td>
</tr>
<tr>
<td>14</td>
<td>0.146</td>
<td>144</td>
<td>14244</td>
</tr>
<tr>
<td>—</td>
<td>0.170</td>
<td>167</td>
<td>13486</td>
</tr>
<tr>
<td>13</td>
<td>0.197</td>
<td>194</td>
<td>12751</td>
</tr>
<tr>
<td>—</td>
<td>0.225</td>
<td>221</td>
<td>12032</td>
</tr>
<tr>
<td>12</td>
<td>0.256</td>
<td>252</td>
<td>11333</td>
</tr>
<tr>
<td>—</td>
<td>0.287</td>
<td>283</td>
<td>10651</td>
</tr>
<tr>
<td>11</td>
<td>0.342</td>
<td>336</td>
<td>9989</td>
</tr>
<tr>
<td>—</td>
<td>0.396</td>
<td>390</td>
<td>9343</td>
</tr>
<tr>
<td>10</td>
<td>0.436</td>
<td>429</td>
<td>8137</td>
</tr>
<tr>
<td>—</td>
<td>0.475</td>
<td>468</td>
<td>6944</td>
</tr>
<tr>
<td>9</td>
<td>0.518</td>
<td>510</td>
<td>5912</td>
</tr>
<tr>
<td>—</td>
<td>0.560</td>
<td>552</td>
<td>4890</td>
</tr>
<tr>
<td>8</td>
<td>0.605</td>
<td>596</td>
<td>4064</td>
</tr>
<tr>
<td>—</td>
<td>0.649</td>
<td>640</td>
<td>3246</td>
</tr>
<tr>
<td>7</td>
<td>0.695</td>
<td>684</td>
<td>2630</td>
</tr>
<tr>
<td>—</td>
<td>0.740</td>
<td>720</td>
<td>2019</td>
</tr>
<tr>
<td>6</td>
<td>0.785</td>
<td>773</td>
<td>1588</td>
</tr>
<tr>
<td>—</td>
<td>0.829</td>
<td>817</td>
<td>1160</td>
</tr>
<tr>
<td>5</td>
<td>0.870</td>
<td>857</td>
<td>872</td>
</tr>
<tr>
<td>—</td>
<td>0.910</td>
<td>897</td>
<td>585</td>
</tr>
<tr>
<td>4</td>
<td>0.928</td>
<td>914</td>
<td>481</td>
</tr>
<tr>
<td>—</td>
<td>0.945</td>
<td>931</td>
<td>378</td>
</tr>
<tr>
<td>3</td>
<td>0.959</td>
<td>945</td>
<td>295</td>
</tr>
<tr>
<td>—</td>
<td>0.973</td>
<td>958</td>
<td>213</td>
</tr>
<tr>
<td>2</td>
<td>0.983</td>
<td>968</td>
<td>147</td>
</tr>
<tr>
<td>—</td>
<td>0.992</td>
<td>977</td>
<td>82</td>
</tr>
<tr>
<td>1</td>
<td>0.996</td>
<td>981</td>
<td>41</td>
</tr>
<tr>
<td>—</td>
<td>1.000</td>
<td>985</td>
<td>0</td>
</tr>
</tbody>
</table>
vfg NCEP grid, met. fields: NCEP/NCAR reanalysis 1998

<table>
<thead>
<tr>
<th>η</th>
<th>p/hPa</th>
<th>h/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0</td>
<td>81058</td>
</tr>
<tr>
<td>0.003</td>
<td>3</td>
<td>57239</td>
</tr>
<tr>
<td>0.007</td>
<td>6</td>
<td>33421</td>
</tr>
<tr>
<td>0.010</td>
<td>10</td>
<td>30913</td>
</tr>
<tr>
<td>0.014</td>
<td>14</td>
<td>28404</td>
</tr>
<tr>
<td>0.018</td>
<td>18</td>
<td>26749</td>
</tr>
<tr>
<td>0.023</td>
<td>23</td>
<td>25094</td>
</tr>
<tr>
<td>0.029</td>
<td>28</td>
<td>23799</td>
</tr>
<tr>
<td>0.035</td>
<td>34</td>
<td>22504</td>
</tr>
<tr>
<td>0.042</td>
<td>41</td>
<td>21411</td>
</tr>
<tr>
<td>0.049</td>
<td>48</td>
<td>20317</td>
</tr>
<tr>
<td>0.058</td>
<td>57</td>
<td>19353</td>
</tr>
<tr>
<td>0.067</td>
<td>66</td>
<td>18389</td>
</tr>
<tr>
<td>0.078</td>
<td>77</td>
<td>17515</td>
</tr>
<tr>
<td>0.089</td>
<td>88</td>
<td>16641</td>
</tr>
<tr>
<td>0.103</td>
<td>101</td>
<td>15830</td>
</tr>
<tr>
<td>0.117</td>
<td>115</td>
<td>15020</td>
</tr>
<tr>
<td>0.133</td>
<td>131</td>
<td>14253</td>
</tr>
<tr>
<td>0.149</td>
<td>147</td>
<td>13486</td>
</tr>
<tr>
<td>0.168</td>
<td>166</td>
<td>12759</td>
</tr>
<tr>
<td>0.188</td>
<td>185</td>
<td>12032</td>
</tr>
<tr>
<td>0.210</td>
<td>207</td>
<td>11342</td>
</tr>
<tr>
<td>0.233</td>
<td>229</td>
<td>10651</td>
</tr>
<tr>
<td>0.259</td>
<td>255</td>
<td>9997</td>
</tr>
<tr>
<td>0.284</td>
<td>280</td>
<td>9343</td>
</tr>
<tr>
<td>0.313</td>
<td>308</td>
<td>8725</td>
</tr>
<tr>
<td>0.341</td>
<td>336</td>
<td>8108</td>
</tr>
<tr>
<td>0.372</td>
<td>367</td>
<td>7526</td>
</tr>
<tr>
<td>0.403</td>
<td>397</td>
<td>6944</td>
</tr>
<tr>
<td>0.436</td>
<td>429</td>
<td>6406</td>
</tr>
<tr>
<td>0.469</td>
<td>462</td>
<td>5869</td>
</tr>
<tr>
<td>0.502</td>
<td>494</td>
<td>5380</td>
</tr>
<tr>
<td>0.535</td>
<td>527</td>
<td>4890</td>
</tr>
<tr>
<td>0.568</td>
<td>560</td>
<td>4452</td>
</tr>
<tr>
<td>0.601</td>
<td>592</td>
<td>4014</td>
</tr>
<tr>
<td>0.633</td>
<td>624</td>
<td>3629</td>
</tr>
<tr>
<td>0.665</td>
<td>655</td>
<td>3246</td>
</tr>
<tr>
<td>0.694</td>
<td>684</td>
<td>2914</td>
</tr>
<tr>
<td>0.724</td>
<td>713</td>
<td>2582</td>
</tr>
<tr>
<td>0.751</td>
<td>740</td>
<td>2301</td>
</tr>
<tr>
<td>0.778</td>
<td>766</td>
<td>2019</td>
</tr>
<tr>
<td>0.801</td>
<td>789</td>
<td>1784</td>
</tr>
<tr>
<td>0.825</td>
<td>813</td>
<td>1549</td>
</tr>
<tr>
<td>0.846</td>
<td>833</td>
<td>1354</td>
</tr>
<tr>
<td>0.866</td>
<td>853</td>
<td>1160</td>
</tr>
<tr>
<td>0.884</td>
<td>871</td>
<td>1001</td>
</tr>
<tr>
<td>0.901</td>
<td>888</td>
<td>842</td>
</tr>
<tr>
<td>0.916</td>
<td>902</td>
<td>714</td>
</tr>
<tr>
<td>0.931</td>
<td>917</td>
<td>585</td>
</tr>
<tr>
<td>0.943</td>
<td>928</td>
<td>481</td>
</tr>
<tr>
<td>0.955</td>
<td>940</td>
<td>378</td>
</tr>
<tr>
<td>0.964</td>
<td>950</td>
<td>295</td>
</tr>
<tr>
<td>0.974</td>
<td>960</td>
<td>213</td>
</tr>
<tr>
<td>0.982</td>
<td>967</td>
<td>147</td>
</tr>
<tr>
<td>0.990</td>
<td>975</td>
<td>82</td>
</tr>
<tr>
<td>0.995</td>
<td>980</td>
<td>41</td>
</tr>
<tr>
<td>1.000</td>
<td>985</td>
<td>0</td>
</tr>
</tbody>
</table>
### APPENDIX C. VERTICAL COORDINATE TABLES

vfg and xfg ERA grid, met. fields: ERA-15 reanalysis 1993

<table>
<thead>
<tr>
<th>l</th>
<th>m</th>
<th>p/hPa</th>
<th>l/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>31</td>
<td>0.010</td>
<td>10</td>
<td>50717.5</td>
</tr>
<tr>
<td>30</td>
<td>0.020</td>
<td>20</td>
<td>25674.2</td>
</tr>
<tr>
<td>29</td>
<td>0.030</td>
<td>30</td>
<td>23481.8</td>
</tr>
<tr>
<td>28</td>
<td>0.041</td>
<td>40</td>
<td>21289.4</td>
</tr>
<tr>
<td>27</td>
<td>0.051</td>
<td>50</td>
<td>20034.3</td>
</tr>
<tr>
<td>26</td>
<td>0.061</td>
<td>60</td>
<td>18759.3</td>
</tr>
<tr>
<td>25</td>
<td>0.071</td>
<td>70</td>
<td>17876.5</td>
</tr>
<tr>
<td>24</td>
<td>0.081</td>
<td>80</td>
<td>16993.7</td>
</tr>
<tr>
<td>23</td>
<td>0.091</td>
<td>90</td>
<td>16309.1</td>
</tr>
<tr>
<td>22</td>
<td>0.102</td>
<td>100</td>
<td>15624.6</td>
</tr>
<tr>
<td>21</td>
<td>0.112</td>
<td>111</td>
<td>15045.7</td>
</tr>
<tr>
<td>20</td>
<td>0.123</td>
<td>121</td>
<td>14466.8</td>
</tr>
<tr>
<td>19</td>
<td>0.134</td>
<td>132</td>
<td>13948.0</td>
</tr>
<tr>
<td>18</td>
<td>0.146</td>
<td>143</td>
<td>13429.2</td>
</tr>
<tr>
<td>17</td>
<td>0.158</td>
<td>155</td>
<td>12949.6</td>
</tr>
<tr>
<td>16</td>
<td>0.170</td>
<td>167</td>
<td>12469.9</td>
</tr>
<tr>
<td>15</td>
<td>0.183</td>
<td>180</td>
<td>12019.4</td>
</tr>
<tr>
<td>14</td>
<td>0.196</td>
<td>193</td>
<td>11568.9</td>
</tr>
<tr>
<td>13</td>
<td>0.210</td>
<td>207</td>
<td>11142.7</td>
</tr>
<tr>
<td>12</td>
<td>0.225</td>
<td>221</td>
<td>10716.5</td>
</tr>
<tr>
<td>11</td>
<td>0.241</td>
<td>236</td>
<td>10311.5</td>
</tr>
<tr>
<td>10</td>
<td>0.255</td>
<td>251</td>
<td>9906.5</td>
</tr>
<tr>
<td>9</td>
<td>0.271</td>
<td>267</td>
<td>9520.1</td>
</tr>
<tr>
<td>8</td>
<td>0.287</td>
<td>283</td>
<td>9133.6</td>
</tr>
<tr>
<td>7</td>
<td>0.305</td>
<td>300</td>
<td>8763.2</td>
</tr>
<tr>
<td>6</td>
<td>0.322</td>
<td>317</td>
<td>8392.8</td>
</tr>
<tr>
<td>5</td>
<td>0.340</td>
<td>335</td>
<td>8036.9</td>
</tr>
<tr>
<td>4</td>
<td>0.358</td>
<td>352</td>
<td>7681.0</td>
</tr>
<tr>
<td>3</td>
<td>0.377</td>
<td>371</td>
<td>7338.7</td>
</tr>
<tr>
<td>2</td>
<td>0.396</td>
<td>390</td>
<td>6996.3</td>
</tr>
<tr>
<td>1</td>
<td>0.415</td>
<td>409</td>
<td>6657.5</td>
</tr>
<tr>
<td>1.1</td>
<td>0.435</td>
<td>428</td>
<td>6337.6</td>
</tr>
<tr>
<td>1.2</td>
<td>0.455</td>
<td>448</td>
<td>6020.7</td>
</tr>
<tr>
<td>1.3</td>
<td>0.475</td>
<td>468</td>
<td>5703.9</td>
</tr>
<tr>
<td>1.4</td>
<td>0.496</td>
<td>489</td>
<td>5399.0</td>
</tr>
<tr>
<td>1.5</td>
<td>0.517</td>
<td>510</td>
<td>5094.2</td>
</tr>
<tr>
<td>1.6</td>
<td>0.539</td>
<td>531</td>
<td>4801.1</td>
</tr>
<tr>
<td>1.7</td>
<td>0.560</td>
<td>552</td>
<td>4508.0</td>
</tr>
<tr>
<td>1.8</td>
<td>0.583</td>
<td>574</td>
<td>4226.4</td>
</tr>
<tr>
<td>1.9</td>
<td>0.605</td>
<td>595</td>
<td>3944.9</td>
</tr>
<tr>
<td>2.1</td>
<td>0.627</td>
<td>615</td>
<td>3674.9</td>
</tr>
<tr>
<td>2.2</td>
<td>0.649</td>
<td>634</td>
<td>3405.0</td>
</tr>
<tr>
<td>2.3</td>
<td>0.672</td>
<td>654</td>
<td>3146.9</td>
</tr>
<tr>
<td>2.4</td>
<td>0.695</td>
<td>674</td>
<td>2888.9</td>
</tr>
<tr>
<td>2.5</td>
<td>0.717</td>
<td>694</td>
<td>2643.6</td>
</tr>
<tr>
<td>2.6</td>
<td>0.740</td>
<td>715</td>
<td>2398.3</td>
</tr>
<tr>
<td>2.7</td>
<td>0.763</td>
<td>737</td>
<td>2161.7</td>
</tr>
<tr>
<td>2.8</td>
<td>0.785</td>
<td>759</td>
<td>1935.9</td>
</tr>
<tr>
<td>2.9</td>
<td>0.807</td>
<td>781</td>
<td>1720.7</td>
</tr>
<tr>
<td>3.0</td>
<td>0.829</td>
<td>804</td>
<td>1505.4</td>
</tr>
<tr>
<td>3.1</td>
<td>0.850</td>
<td>827</td>
<td>1300.8</td>
</tr>
<tr>
<td>3.2</td>
<td>0.873</td>
<td>850</td>
<td>1111.4</td>
</tr>
<tr>
<td>3.3</td>
<td>0.895</td>
<td>874</td>
<td>935.7</td>
</tr>
<tr>
<td>3.4</td>
<td>0.917</td>
<td>900</td>
<td>760.1</td>
</tr>
<tr>
<td>3.5</td>
<td>0.940</td>
<td>923</td>
<td>610.1</td>
</tr>
<tr>
<td>3.6</td>
<td>0.955</td>
<td>947</td>
<td>460.2</td>
</tr>
<tr>
<td>3.7</td>
<td>0.973</td>
<td>973</td>
<td>341.5</td>
</tr>
<tr>
<td>3.8</td>
<td>0.992</td>
<td>997</td>
<td>222.8</td>
</tr>
<tr>
<td>3.9</td>
<td>1.000</td>
<td>1020</td>
<td>142.8</td>
</tr>
<tr>
<td>4.0</td>
<td>0.990</td>
<td>984</td>
<td>62.8</td>
</tr>
<tr>
<td>4.1</td>
<td>0.966</td>
<td>981</td>
<td>31.4</td>
</tr>
<tr>
<td>4.2</td>
<td>0.900</td>
<td>985</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Appendix D

Global Program Variables

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Module declarations</td>
</tr>
<tr>
<td>==============================================================</td>
</tr>
<tr>
<td>USE userconstants, ONLY: kc, ki, kf, ko, &amp; ntrace</td>
</tr>
<tr>
<td>USE griddefs, ONLY: im, jm, lm, TM_res_ID, met_ID, ndyn_default</td>
</tr>
<tr>
<td>IMPLICIT NONE</td>
</tr>
<tr>
<td>SAVE</td>
</tr>
<tr>
<td>==============================================================</td>
</tr>
<tr>
<td>Module body</td>
</tr>
<tr>
<td>==============================================================</td>
</tr>
<tr>
<td>!(Main global TM3 constants and variables.)</td>
</tr>
<tr>
<td>! $Id: tm3global.f90,v 1.15 2003/06/11 11:25:44 tpohw Exp $</td>
</tr>
<tr>
<td>!*******************************************************************************</td>
</tr>
<tr>
<td>! Main global TM3 constants and variables.</td>
</tr>
<tr>
<td>! $Id: tm3global.f90,v 1.15 2003/06/11 11:25:44 tpohw Exp $</td>
</tr>
<tr>
<td>!*******************************************************************************</td>
</tr>
<tr>
<td>MODULE TM3global</td>
</tr>
<tr>
<td>USE userconstants, ONLY: kc, ki, kf, ko, &amp; ntrace</td>
</tr>
<tr>
<td>USE griddefs, ONLY: im, jm, lm, TM_res_ID, met_ID, ndyn_default</td>
</tr>
<tr>
<td>IMPLICIT NONE</td>
</tr>
<tr>
<td>SAVE</td>
</tr>
<tr>
<td>==============================================================</td>
</tr>
<tr>
<td>! runtime parameters, collected in namelist 'inputz'</td>
</tr>
<tr>
<td>! Convention for output time intervals:</td>
</tr>
<tr>
<td>! &gt;0 means 'interval in seconds',</td>
</tr>
<tr>
<td>! 0 no output,</td>
</tr>
<tr>
<td>! -1 &quot;daily at 00:00&quot;,</td>
</tr>
<tr>
<td>! -2 &quot;monthly at 1st 00:00&quot;,</td>
</tr>
<tr>
<td>! -3 &quot;yearly, January 1 0:00&quot;</td>
</tr>
<tr>
<td>!===============================================================================</td>
</tr>
<tr>
<td>! 1st Most frequently user-changed parameters</td>
</tr>
<tr>
<td>!===============================================================================</td>
</tr>
<tr>
<td>CHARACTER(32) :: jobid = ' '</td>
</tr>
<tr>
<td>! label: job ID (prefixing output file names)</td>
</tr>
<tr>
<td>CHARACTER(100) :: xlabel = ' '</td>
</tr>
<tr>
<td>! label of model run</td>
</tr>
<tr>
<td>CHARACTER(100) :: path_in_p = 'prepro'</td>
</tr>
<tr>
<td>! path to input prepro (met.) data</td>
</tr>
<tr>
<td>CHARACTER(100) :: path_in_s = 'input'</td>
</tr>
<tr>
<td>! path to input source data</td>
</tr>
<tr>
<td>REAL(kc) :: fscale(ntrace) = 1.0_kc</td>
</tr>
<tr>
<td>! scale factor for output of mixing ratios:</td>
</tr>
<tr>
<td>! converts AmountTracer/kgAir into application units</td>
</tr>
<tr>
<td>INTEGER :: idatei(6) = (/1988, 1, 1, 0, 0, 0/)</td>
</tr>
<tr>
<td>! date (yr,month,day,hour,min,sec)</td>
</tr>
<tr>
<td>INTEGER :: idatee(6) = (/1988, 1, 3, 0, 0, 0/)</td>
</tr>
<tr>
<td>! date (yr,month,day,hour,min,sec)</td>
</tr>
<tr>
<td>INTEGER :: idatet(6) = (/0, 1, 1, 0, 0, 0/)</td>
</tr>
<tr>
<td>! after which instantaneous output to &lt;name_mix&gt; is written;</td>
</tr>
<tr>
<td>! set year to 0 to write from</td>
</tr>
<tr>
<td>! the beginning of run</td>
</tr>
<tr>
<td>INTEGER :: ndyn = ndyn_default</td>
</tr>
<tr>
<td>! length of full advection step</td>
</tr>
<tr>
<td>INTEGER :: nconv = ndyn_default</td>
</tr>
<tr>
<td>! interval for convection calculation</td>
</tr>
<tr>
<td>INTEGER :: nsrce = ndyn_default</td>
</tr>
<tr>
<td>! interval for source calculations (SUBR 'source1','source2')</td>
</tr>
<tr>
<td>INTEGER :: nchem = ndyn_default</td>
</tr>
<tr>
<td>! interval for chemistry calculations (SUBR 'chem1', 'chem2')</td>
</tr>
<tr>
<td>INTEGER :: ninst = 21600</td>
</tr>
<tr>
<td>! interval for output of instantaneous tracer mixing ratio</td>
</tr>
<tr>
<td>! on file &lt;name_mix&gt;</td>
</tr>
<tr>
<td>INTEGER :: ninststat = 21600</td>
</tr>
<tr>
<td>! interval for output of instantaneous tracer mixing ratio</td>
</tr>
<tr>
<td>! at station locations</td>
</tr>
<tr>
<td>==============================================================</td>
</tr>
</tbody>
</table>


APPENDIX D. GLOBAL PROGRAM VARIABLES

```
INTEGER :: ninsttmass = 0 ! interval for output of instantaneous tracer

! 2nd Other Adjustable Parameters
!
!! mass flux input file names (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/)
CHARACTER(20) :: name_pu = 'stag_pu_'//TM_res_ID//'.b' ! for repeated met. year (if iyearm/=0)
CHARACTER(20) :: name_pv = 'stag_pv_'//TM_res_ID//'.b' ! 
CHARACTER(20) :: name_ps = 'stag_ps_'//TM_res_ID//'.b' ! 
CHARACTER(20) :: name_maxndyn = 'maxndyn_'//TM_res_ID//'.d' ! max. timesteps 'ndyn'
CHARACTER(20) :: name_pu_c = 'stagc_pu_'//TM_res_ID//'.b' ! for consecut. met. years (if iyearm==0)
CHARACTER(20) :: name_pv_c = 'stagc_pv_'//TM_res_ID//'.b' ! 
CHARACTER(20) :: name_ps_c = 'stagc_ps_'//TM_res_ID//'.b' ! 
CHARACTER(20) :: name_maxndyn_c = 'maxndync_'//TM_res_ID//'.d' ! max. timesteps 'ndyn'

!! convection info input file names (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/)
CHARACTER(20) :: name_eu = 'eu_'//TM_res_ID//'.b'
CHARACTER(20) :: name_du = 'du_'//TM_res_ID//'.b'
CHARACTER(20) :: name_ed = 'ed_'//TM_res_ID//'.b'
CHARACTER(20) :: name_dd = 'dd_'//TM_res_ID//'.b'
CHARACTER(20) :: name_vdiff = 'k_'//TM_res_ID//'.b'
CHARACTER(20) :: name_height = 'z_'//TM_res_ID//'.b'

!! additional (optional) files (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/ for t and q)
CHARACTER(20) :: name_t = 't_'//TM_res_ID//'.b' ! temperature
CHARACTER(20) :: name_q = 'q_'//TM_res_ID//'.b' ! humidity
LOGICAL :: creadt = .FALSE. ! read temperature field?
LOGICAL :: creadq = .FALSE. ! read humidity field?
CHARACTER(100) :: name_plandf = 'plandf_'//TM_res_ID//'.b' ! landfraction (with complete path)
LOGICAL :: read_plandf = .FALSE. ! read in landfraction file?
CHARACTER(100) :: name_stationlist = 'input/stationlist.d' ! table of station coordinates

!! output file names (relative to <path_out>/)
CHARACTER(20) :: name_mix = 'mix.b' ! instantaneous mixing ratios
CHARACTER(20) :: name_mixstn = 'mixstn.b' ! instantaneous mixing ratios at stations
CHARACTER(20) :: name_tmass = 'tmass.b' ! instantaneous tracer mass field
CHARACTER(20) :: name_psout = 'ps.b' ! instantaneous surface pressure field
LOGICAL :: cwrtps = .FALSE. ! write instantaneous surface pressure field?
CHARACTER(100) :: name_stationlist = 'input/stationlist.d' ! table of station coordinates

!! start/restart option
INTEGER :: istart = 1 ! 1 Cold start
    ! Initial tracer fields are set to 0.
    ! They can afterwards be partially or
totally overwritten in subroutine
'\trace0'.
    ! So other status fields are read.
2 Cold start with initial tracer fields
    ! Initial tracer fields are read from
file
'\path_out//<jobid>//'modelstatus.b'.
3 Job chain mode
    ! By TM3, the existence of a file
'\path_out//<jobid>//'modelstatus.d'
is used as indicator if this is a
continuation run within a job chain or
the start of it.
The user can employ it as end
condition in a self-calling job
script.
    ! If this file does NOT EXIST, then
the start date is that of 'tm3.in'.
    ! Initial tracer fields are read from
the file
'\path_out//<jobid>//'modelstatus.b',
if existent, or otherwise set to 0.
    ! If the file EXISTS, then the
namelist 'inputz' is first read from
file 'tm3.in'. Afterwards the start
date <idatei> is overwritten by the
date in
'\path_out//<jobid>//'modelstatus.d',
saved by a previous model run after
time interval <nsave>.
    ! Initial tracer fields are read from
the file
'\path_out//<jobid>//'modelstatus.b',
if existent.
    ! That file will always be written after
a model run, independent of what
istart is set to.
The continuation parameter file
'\path_out//<jobid>//'modelstatus.d'
contains the start date in the form
YYYY MM DD hh mm ss as first line.
The last line contains all prepro
directory identifiers needed by the
next continuation cycle, e.g.
```
'1999 2000 2001' or '199911 199912 200001' (without quotes).
This is useful for getting prepro files out of an archive by shell script commands (see the manual for an example).

10+i As above, with i = 1, 2 or 3. In addition the user routine 'trace1' will be called after 'trace0'.

INTEGER :: nsave = 0 ! Only with istart=3: Time interval in seconds since <idatei> for stopping TM3 and writing the restart files <path_out>//<jobid>//'modelstatus.d' (if the end date 'idatee' has not been reached) and <path_out>//<jobid>//'modelstatus.b'. If the end date <idatee> has been reached, the program finishes having <path_out>//<jobid>//'modelstatus.d' deleted; thus that file can taken as indicator for the necessity of a new job job cycle in a recursive job script.

!--- calendar
INTEGER :: iyear0 = 0 ! base year for calendar calculations (time is internally represented as seconds since 01-Jan-iyear0 00:00:00). NOTE: because of overflow problems on a 32 bit machine, only years in the range iyear0 ... <iyear0>+65 can be represented.
iyear0=0: use year of begin of run (<idatei>(1)) as base year

INTEGER :: iyearm = 0 ! year whose meteorology shall be used iyearm=0: use true year else: repeatedly use <iyearm>

INTEGER :: icalendo = 2 ! calendar type
1 permanent 360 day year
2 real calendar
3 permanent 365 day year
4 permanent 366 day year

!--- diagnostics
CHARACTER(20) :: name_mmix = 'mmix.b' ! mean mixing ratios
CHARACTER(20) :: name_tables1 = 'consrv_tracer.d' ! diagnostic table 1: tracer conservation
CHARACTER(20) :: name_tables2 = 'consrv_air.d' ! diagnostic table 2: air conservation
CHARACTER(20) :: name_tables3 = 'zonal_avrg.d' ! diagnostic table 3: mean mixing ratios
LOGICAL :: cdebug = .FALSE. ! write debug info (to <name_debug>)?
INTEGER :: idatem(6) = (/0, 1, 1, 0, 0, 0/) ! time for starting the computation of time averaged fields (files <name_mmix.b> and <name_tables3>); set year to 0 to start from the beginning of run

INTEGER :: ndiag = 21600 ! sampling interval for mean quantities
INTEGER :: ndiagp1 = -1 ! interval for output of conservation statistics on files <name_table1> and <name_table2>, and of the Courant statistics to <name>
INTEGER :: ndiagp2 = -1 ! interval for output of time averaged fields on files <name_mmix.b>, <name_tables3>

!--- checkpoints
CHARACTER(20) :: name_check = 'check.d' ! output filename of mixing ratios at checkpoints (relative to <path_out>)
INTEGER :: ncheck = 0 ! interval for output of tracer mix ratios at checkpoints
INTEGER :: noindc = 0 ! actual number of checkpoints (max. 10)
INTEGER :: indc(3,10) = 0 ! indices of checkpoints (i,j,l)

!--- computational parameters
LOGICAL :: limits = .FALSE. ! prevent negative tracer masses?
REAL(kc) :: couranttarg = -1.0_kc ! If set to something between 0 and 1, adaptive integration stepwidth is switched on, which 'tries' not to exceed 'couranttarg' (preliminary implementation).
REAL(kc) :: czeta = 1.0_kc ! scaling factor for convection
REAL(kc) :: czetak = 1.0_kc ! scaling factor for vertical diffusion
INTEGER :: ndiff = 0 ! interval for horizontal diffusion (currently not implemented)

NAMELIST /inputz/
  !--- 1st part

&
APPENDIX D. GLOBAL PROGRAM VARIABLES

jobid, xlabel, 
path_in_p, path_in_s, path_out, fscale, 
idatei, idatee, idatet, 
nsym, nconv, nsrc, nchem, 
ninst, ninststn, ninsttmass, 

name_pu, name_pv, name_ps, name_maxndyn, 
name_pu_c, name_pv_c, name_ps_c, name_maxndyn_c, 
name_eu, name_du, name_ed, name_dd, name_vdiff, name_height, 

name_plandf, read_plandf, name_stationlist, 
name_mix, name_mixstn, name_tmass, name_psout, cwrtps, 

iyear0, iyearm, icalendo, 

name_mmix, name_tables1, name_tables2, name_tables3, name_debug, 
cdebug, idatem, ndiag, ndiagp1, ndiagp2, 

name_check, ncheck, noindc, indc, 

limits, couranttarget, czeta, czetak, ndiff 

--- total: 66 variables 

!------------------------------------------------------------------------------
! end of 'inputz' runtime parameters
!------------------------------------------------------------------------------

!------------------------------------------------------------------------------
! geometrical constants
!------------------------------------------------------------------------------

REAL(kc), PARAMETER :: pi = 3.141592653589793238462643_kc, &
twopi = 2.0_kc*pi, &
dlon = twopi/im, &
dlat = .5_kc*twopi/(jm-1)

!-------------------------------------------------------------------------------
! I/O related constants
!-------------------------------------------------------------------------------

!--- I/O units for standard TM3 routines

INTEGER, PARAMETER :: kmain = 6 ! main control output
INTEGER, PARAMETER :: kdebug = 10 ! debug output

INTEGER, PARAMETER :: kwind1 = 11, & ! fmu
kwind2 = 12, & ! fmv
kwind3 = 13, & ! ps
kmxndyn = 14 ! max. timesteps 'ndyn'

INTEGER, PARAMETER :: kconv1 = 15, & ! entrainment updraft
kconv2 = 16, & ! downdraft
kconv3 = 17, & ! detrainment updraft
kconv4 = 18, & ! downdiff
kconv5 = 19, & ! vertical diffusion coefficient
kconv6 = 20 ! height of level boundaries

INTEGER, PARAMETER :: kaddt = 21 ! temperature
INTEGER, PARAMETER :: kaddq = 22 ! humidity

INTEGER, PARAMETER :: kcheck = 31 ! mix-ratios at check locations
INTEGER, PARAMETER :: kdiag1 = 32 ! diagnostic tables 1 (tracer info)
INTEGER, PARAMETER :: kdiag2 = 33 ! diagnostic tables 2 (air mass)
INTEGER, PARAMETER :: kdiag3 = 34 ! diagnostic tables 3 (mean field)
INTEGER, PARAMETER :: kmix = 35 ! instantaneous mixing ratio fields
INTEGER, PARAMETER :: kmeanx = 36 ! averaged mixing ratio fields
INTEGER, PARAMETER :: ktsmass = 37 ! mixing ratios at stations
INTEGER, PARAMETER :: ktmass = 38 ! instantaneous tracer mass fields
INTEGER, PARAMETER :: kps = 39 ! instantaneous surface pressure field

!--- I/O units for additional packages and/or user-supplied routines

INTEGER, PARAMETER :: kflux = 40
INTEGER, PARAMETER :: kuser = kflux+1 ! reserved for flux I/O

CHARACTER(*), PARAMETER :: pathsep = '/' ! file path separator character
! internal control variables
! (initialized during SUBROUTINE start)

!-------------------------------------------------------------------------------
! times (in seconds since 1st-jan-year0, 00:00:00)
!-------------------------------------------------------------------------------
INTEGER :: itau ! current model time
INTEGER :: idate(6) ! date corresponding to itau
INTEGER :: itaui ! time corresponding to idatei
INTEGER :: itaut ! time corresponding to idatet
INTEGER :: itaum ! time corresponding to idatem
LOGICAL :: newyr ! .true. if at beginning of a new year and at TM3 start
LOGICAL :: newmonth ! .true. if at beginning of a new month and at TM3 start
LOGICAL :: newday ! .true. if at beginning of a new day and at TM3 start
! (i.e. at 00Z)
LOGICAL :: newsrun ! .true. if at beginning of a new run
INTEGER :: nstep ! advection step counter
LOGICAL :: adapt_ndyn ! flag whether adaptive integration stepwidth is
! in use (set by range testing of <courant_limit>)
INTEGER :: maxndyn ! ndyn value for which the max. Courant number
! is 1 (for adaptive integration stepwidth)
INTEGER :: nistep ! number of integration steps per meteo. step
INTEGER :: istep ! current integration step within meteo. step
LOGICAL :: jobcont ! .true. if a job continuation is present
! (istart=3 and saved parameter file
! 'modelstatus.d' exists)

!-------------------------------------------------------------------------------
! main fields
!-------------------------------------------------------------------------------

! geometry fields
! surface of the globe
! thickness of model layers in sigma units
! (positive numbers!)
REAL(kc) :: areag ! area of grid cells (in m**2)
REAL(kc) :: dsig(lm) ! thickness of model layers in sigma units
! (positive numbers!)
REAL(kc) :: dxyp(jm) ! areas of grid cells (in m**2)
REAL(kc) :: sin_lon(im) ! sin(longitude)
REAL(kc) :: cos_lon(im) ! cos(longitude)
REAL(kc) :: sin_lat(jm) ! sin(latitude)
REAL(kc) :: cos_lat(jm) ! cos(latitude)
REAL(kc) :: lambda_lb(im) ! longitude of western grid cell boundaries in radian
REAL(kc) :: mu_lb(jm) ! sine of southern grid cell boundaries
REAL(kc) :: pland(im,jm) ! land fraction in each grid cell

! prepro input fields
! N-S mass flux
REAL(ki) :: pu(im,jm,lm) ! W-E mass flux
REAL(ki) :: pv(im,jm,lm) ! N-S mass flux
REAL(ki) :: suj(im,jm,lm), & ! surface pressure
                  eul(im,jm,lm) ! entrainment updraft [previous]
REAL(ki) :: dui(im,jm,lm), & ! detrainment updraft [previous]
                  dul(im,jm,lm) ! detrainment updraft [previous]
REAL(ki) :: ed0(im,jm,lm), & ! entrainment downdraft [previous]
                  ed1(im,jm,lm) ! entrainment downdraft [previous]
REAL(ki) :: dd0(im,jm,lm), & ! detrainment downdraft [previous]
                  dd1(im,jm,lm) ! detrainment downdraft [previous]
REAL(ki) :: ak0(im,jm,0:lm), & ! diffusion coefficient [previous]
                  ak1(im,jm,0:lm) ! diffusion coefficient [previous]
REAL(ki) :: gph0(im,jm,0:lm) ! geopotential height [current]
REAL(ki) :: gph1(im,jm,0:lm) ! geopotential height [current]
REAL(ki) :: t(im,jm,lm) ! temperature (optional) [current]
REAL(ki) :: q(im,jm,lm) ! humidity (optional) [current]
REAL(ki) :: q0(im,jm,lm) ! humidity (optional) [current]
REAL(ki) :: q1(im,jm,lm) ! humidity (optional) [current]

! air mass field (kg)
REAL(ki) :: m0(im,jm,lm) ! previous
REAL(ki) :: m(im,jm,lm) ! current
REAL(ki) :: m1(im,jm,lm) ! next

! pressure values (Pascal) at hybrid level boundaries
! current
REAL(ki) :: phlb(im,jm,lm+1), & ! pressure values (Pascal) at hybrid level boundaries
                  phlb1(im,jm,lm+1) ! next timestep

! main model fields
! tracer mass (kg)
REAL(ki) :: rm0(im,jm,lm,trace) ! tracer mass (kg)
REAL(ki) :: rm(im,jm,lm,trace) ! tracer mass (kg)
REAL(ki) :: rmax(im,jm,lm,trace) ! max. tracer mass
REAL(ki) :: y1(m,di/dz)+delta_z/2 (kg)
REAL(ki) :: y2(m,di/dz)+delta_z/2 (kg)
REAL(ki) :: y3(m,di/dz)+delta_z/2 (kg)
REAL(ki) :: y4(m,di/dz)+delta_z/2 (kg)
APPENDIX D. GLOBAL PROGRAM VARIABLES

### Mixing Ratios

```plaintext
REAL(ko) :: mix(im,jm,lm,ntrace) ! instantaneous mixing ratios
! updated by call to 'get_mix'
! mix = fscale * rm/m
! despite the special grid structure
! within the fields m and rm,
! all entries of the array mix
! are well-defined
```

### Temporal Interpolation

```plaintext
REAL(kc) :: w0, w1 ! current weights
```

### Station Data

```plaintext
CHARACTER(12), ALLOCATABLE :: stations_name(:) ! name
REAL(kc), ALLOCATABLE :: stations_lat(:) ! latitude (degrees N)
REAL(kc), ALLOCATABLE :: stations_lon(:) ! longitude (degrees E)
REAL(kc), ALLOCATABLE :: stations_h(:) ! height (masl)
INTEGER :: stations_n ! # of stations
INTEGER, ALLOCATABLE :: stations_inw(:), stations_ine(:), stations_isw(:), stations_ise(:), stations_jnw(:), stations_jne(:), stations_jsw(:), stations_jse(:)
REAL(kc), ALLOCATABLE :: stations_di(:), stations_dj(:) ! fractional distance from SW grid cell

REAL(ko), ALLOCATABLE :: mixstn(:,:) ! mixing ratios
```

END MODULE TM3global
Appendix E

Runtime Parameters

The following commented runtime parameter template can be found under doc/tm3.in.

```plaintext
TH3 runtime parameter template

Uncomment and edit the parameters which need adjustment.

Every parameter is listed with the initialization expression it gets in the Fortran source code. If uncommented, this has to be changed to NAMELIST I/O syntax where necessary.

Convention for output time intervals:
>0 means "interval in seconds",
0 no output,
-1 "daily at 0:00",
-2 "monthly at 1st 0:00",
-3 "yearly, January 1 0:00".

TM3 runtime parameter template

!===============================================================================
!
! TM3 runtime parameter template
!
! Uncomment and edit the parameters which need adjustment.
! Fortran source code. If uncommented, this has to be changed to NAMELIST I/O syntax where necessary.
!
! Convention for output time intervals:
! >0 means "interval in seconds",
! 0 no output,
! -1 "daily at 0:00",
! -2 "monthly at 1st 0:00",
! -3 "yearly, January 1 0:00".
!
!===============================================================================

! 1st Most frequently user-changed parameters
! -------------------------------------------
!
! jobid = '' ! label: job ID (prefixing output file names)
! xlabel = '' ! label of model run
! path_in_p = 'prepro' ! path to input prepro (met.) data
! path_in_s = 'input' ! path to input source data
! path_out = 'output' ! path to output
! fscale = 1.0_kc ! scale factor for output of mixing ratios:
! ! converts AmountTracer/kgAir into
! ! application units
! !--- date (yr,month,day,hour,min,sec)
! idatei = 1988, 1, 1, 0, 0, 0 ! for start of model run
! idatee = 1988, 1, 3, 0, 0, 0 ! for end of model run
! idatet = 0, 1, 1, 0, 0, 0 ! after which instantaneous
! ! output to <name_mix> is written;
! ! set year to 0 to write from
! ! the beginning of run
!
! !--- computational time intervals in seconds (0=switched off)
! ndyn = ndyn_default ! length of full advection step
! ! (basic time step of integration)
! nconv = ndyn_default ! interval for convection calculations (SUBR)
! ! ("source", "source2")
! nsrce = ndyn_default ! interval for source calculations (SUBR)
! ! ("chem1", "chem2")
!
! !--- output time intervals in seconds (convention applies)
! ninst = 21600 ! interval for output of instantaneous tracer
! ! mixing ratio on file <name_mix>
! ninstatn = 21600 ! interval for output of instantaneous tracer
! ! into file <name_mixatn>
! ninstmass = 0 ! interval for output of instantaneous tracer
! ! mass on file <name_tmass>
!
!
! 2nd Other Adjustable Parameters
! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```

71
APPENDIX E. RUNTIME PARAMETERS

!--- mass flux input file names (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/)
! name_pu = 'stag_pu_'//TM_res_ID//'.b' ! for repeated met. year (if iyearm/=0)
! name_pv = 'stag_pv_'//TM_res_ID//'.b' !
! name_ps = 'stag_ps_'//TM_res_ID//'.b' !
! name_maxndyn = 'maxndyn_'//TM_res_ID//'.d' ! max. timesteps 'ndyn'
! name_pu_c = 'stagc_pu_'//TM_res_ID//'.b' ! for consecut. met. years (if iyearm==0)
! name_pv_c = 'stagc_pv_'//TM_res_ID//'.b' !
! name_ps_c = 'stagc_ps_'//TM_res_ID//'.b' !
! name_maxndyn_c = 'maxndync_'//TM_res_ID//'.d'! max. timesteps 'ndyn'

! --- convection info input file names (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/)
! name_eu = 'eu_'//TM_res_ID//'.b'
! name_du = 'du_'//TM_res_ID//'.b'
! name_ed = 'ed_'//TM_res_ID//'.b'
! name_dd = 'dd_'//TM_res_ID//'.b'
! name_vdiff = 'k_'//TM_res_ID//'.b'
! name_height = 'z_'//TM_res_ID//'.b'

!--- additional (optional) files (relative to <path_in_p>/<TM_res_ID>/<met_ID>/<year>/ for t and q)
! name_t = 't_'//TM_res_ID//'.b' ! temperature
! name_q = 'q_'//TM_res_ID//'.b' ! humidity
! creadt = .FALSE. ! read temperature field?
! creadq = .FALSE. ! read humidity field?
! name_planlist = 'input/stationlist.d' ! table of station coordinates

! --- output file names (relative to <path_out>/)
! name_mix = 'mix.b' ! instantaneous mixing ratios
! name_mixstn = 'mixstn.b' ! instantaneous mixing ratios at stations
! name_tmass = 'tmass.b' ! instantaneous tracer mass field
! name_plandf = 'plandf_'//TM_res_ID//'.b' ! landfraction (with complete path)
! name_stationlist = 'input/stationlist.d' ! write instantaneous surface pressure field?

! --- start/restart option
! istart = 1 ! 1 Cold start
! Initial tracer fields are set to 0.
! They can afterwards be partially or
totally overwritten in subroutine
! 'trace0'.
! No other status fields are read.
! 2 Cold start with initial tracer fields
! Initial tracer fields are read from
! file
! <path_out>//<jobid>//'modelstatus.b'.
! 3 Job chain mode
! By TCL, the existence of a file
! <path_out>//<jobid>//'modelstatus.d'
! is used as indicator if this is a
! continuation run within a job chain or
! the start of it.
! The user can employ it as end
! condition in a self-calling job
! script.
! IF this file does NOT EXIST, then
! Initial tracer fields are read from
! the file
! <path_out>//<jobid>//'modelstatus.b',
! if existent, or otherwise set to 0.
! IF the file EXISTS, then the
! namelist 'inputs' is first read from
! file 'tm3.in'. Afterwards the start
date <idatei> is overwritten by the
date in
! <path_out>//<jobid>//'modelstatus.d',
! saved by a previous model run after
! time interval <nsave>.
! Initial tracer fields are read from
! file
! <path_out>//<jobid>//'modelstatus.b',
! if existent.
! That file will always be written after
! a model run, independent of what
! istart is set to.
! The continuation parameter file
! <path_out>//<jobid>//'modelstatus.d'
! contains the start date in the form
! YYYY HH MM DD as as first line.
! The last line contains all prepro
! directory identifiers needed by the
! next continuation cycle, e.g.
! '1999 2000 2001' or
! '199911 199912 200001' (without
! quotes).
! This is useful for getting prepro
! files out of an archive by shell
! script commands (see the manual for an
! example).
\texttt{nsave = 0} \quad \text{Only with istart=3: Time interval in} \quad \text{seconds since <idatei> for stopping TM3} \quad \text{and writing the restart} \\
\text{files} \\
\text{<path_out>>/<jobid>>//<modelstatus.d> (if} \quad \text{the end date 'idatee' has not been} \\
\text{reached) and} \\
\text{<path_out>>/<jobid>>//<modelstatus.b>.} \\
\text{If the end date <idatee> has been} \quad \text{reached,} \\
\text{the program finishes having} \\
\text{<path_out>>/<jobid>>//<modelstatus.d>} \\
deleted; thus that file can taken as} \\
\text{indicator for the necessity of a new job} \\
\text{job cycle in a recursive job script.} \\

\textbf{--- calendar} \\
\text{iyear0 = 0} \quad \text{base year for calendar} \\
\text{calculations (time is internally} \quad \text{represented as seconds since} \\
\text{01-Jan-<iyear0> 00:00:00).} \\
\text{NOTE: because of overflow problems} \\
on a 32 bit machine, only years} \quad \text{in the range <iyear0> ... <iyear0>+65} \\
can be represented. \\
iyear0=0: use year of begin of} \quad \text{run (<idatei(l)> as base year} \\
iyearm = 0 \quad \text{year whose meteorology shall be used} \\
iyearm=0: use true year \\
\text{else: repeatedly use <iyearm>}\quad \text{calendar type} \\
iscalendo = 2 \\
1 \quad \text{permanent 360 day year} \\
2 \quad \text{real calendar} \\
3 \quad \text{permanent 365 day year} \\
4 \quad \text{permanent 366 day year} \\

\textbf{--- diagnostics} \\
\text{name_mmix = 'mmix.b'} \quad \text{mean mixing ratios} \\
\text{name_tables1 = 'consrv_tracer.d'} \quad \text{diagnostic table 1: tracer conservation} \\
\text{name_tables2 = 'consrv_air.d'} \quad \text{diagnostic table 2: air conservation} \\
\text{name_tables3 = 'zonal_avg.b'} \quad \text{diagnostic table 3: mean mixing ratios} \\
\text{name_debug = 'debug.d'} \quad \text{debugging info} \\
\text{cdebug = .FALSE.} \quad \text{write debug info (to <name_debug>)?}\quad \text{for starting the computation of time} \\
\text{idatem = 0, 1, 0, 0, 0, 0} \quad \text{averaged fields (files <name_mmix.b> and} \\
\text{<name_tables3>); set year to 0 to start} \\
\text{from the beginning of run} \\
\text{ndiag = 21600} \quad \text{sampling interval for mean quantities} \\
\text{ndiagp1 = -1} \quad \text{interval for output of conservation} \\
\text{statistics on files <name_tables1> and} \\
\text{<name_tables2>, and of the Courant} \\
\text{statistics to <kmain>}\quad \text{interval for output of time averaged} \\
\text{fields on files <name_mmix.b>,} \\
\text{<name_tables3>} \\

\textbf{--- checkpoints} \\
\text{name_check = 'check.d'} \quad \text{output filenames of mixing ratios at} \\
\text{checkpoints (relative to <path_out>/)} \quad \text{intervals for output of tracer mix ratios} \\
\text{at checkpoints} \\
\text{ncheck = 0} \quad \text{actual number of checkpoints (max. 10)} \\
\text{noindc = 0} \quad \text{indices of checkpoints (i,j,l)} \\

\textbf{--- computational parameters} \\
\text{limits = .FALSE.} \quad \text{prevent negative tracer masses?} \\
\text{couranttarget = -1.0_kc} \quad \text{If set to something between 0 and 1,} \\
\text{adaptive integration stepwidth is} \\
\text{switched on, which <true> not to} \\
\text{exceed <couranttarget> (preliminary} \\
\text{implementation).} \\
\text{czeta = 1.0_kc} \quad \text{scaling factor for convection} \\
\text{czetak = 1.0_kc} \quad \text{scaling factor for vertical diffusion} \\
\text{ndiff = 0} \quad \text{interval for horizontal diffusion} \\
\text{(currently not implemented)
Appendix F

Diagnostic Results

F.1 The Effects of Varying Grid Resolutions

F.1.1 Surface Point Emissions

The setup of this synthetic simulation comprised a stationary tracer emission of 1 kg/s into a single surface pixel over a period of a month, starting at 1st January or July 1993. The emission sites were selected according to different mean atmospheric environments: Paris, Manaus (Brazil), Bombay, Los Angeles and Zotino (Siberia). TM3 was run in version 3.8a and fed with ERA-15 reanalysis data of all available resolutions, i.e. cg, fg, vfg and xfg.

Surface Layer Maps of Monthly Means

The following pages 76–95 show surface-layer maps of monthly mixing ratio means $\bar{\chi}$ on the left doublepage, respectively, and the relative deviations compared to xfg results $(\bar{\chi} - \bar{\chi}_{xfg})/\bar{\chi}_{xfg}$ on the corresponding right doublepage. The resolutions are cg (top left), fg (top right) and vfg (bottom left) versus xfg (bottom right), the finest available. In the deviation plots the monthly mean for xfg itself is displayed, and grid cells with a mixing ratio in the corresponding xfg cell of less than $10^{-4}$ times the global xfg maximum were masked out. Furthermore, relative deviations greater than 2.33 were reset to this value, or with other words, the last class in the deviation color bar is open to high values.

Surface-Layer Maps of Instantaneous Mixing Ratios

The relative deviations of instantaneous mixing ratios after the simulated month were in the same order of magnitude (not shown). Map panels of instantaneous surface-layer mixing ratios for the different resolutions are shown on pages 96–105.

Meridional Vertical Cut in Pressure Coordinates

For the emission location Manaus, vertical meridional cuts of instantaneous tracer mixing ratios are shown on pages 106–107.
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS

[Four maps showing diagnostic results with color-coded data distributions.]

96
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS
F.1.2 Radon

This simulation setup comprised a stationary emission of $^{222}$Rn with 1 atom/cm$^2$ over land for the simulation intervals July and December 1993. One month spin-up preceded, respectively. The ice-covered areas Antarctica, Greenland, Iceland, Baffin Island, Svalbard and Novaja Semlia were assumed to have zero emission. TM3 was run in version 3.8 and fed with the ERA-15 reanalysis data of all available resolutions, i.e. cg, fg, vfg and xfg.

Following results are shown on the subsequent pages:

- surface maps of the monthly mean activity, vertically integrated over the thickness of the lowest cg layer for better comparison (pp 110–111),
- zonal averages of the monthly mean activities (pp 112–113),
- vertical cuts at the latitudes $50^\circ$N and $0^\circ$ (pp 114–117), and
- monthly mean budgets of $^{222}$Rn for eight different atmospheric pools: Eurasia, North America, North Atlantic and North Pacific, with each area subdivided in the vertical into boundary layer and free troposphere (page 118).

Within the panels the different resolution maps are placed as follows:

\[
\begin{array}{cccc}
\text{cg} & \text{fg} \\
\text{vfg} & \text{xfg}
\end{array}
\]
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS

[Graphs and charts showing diagnostic results]
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS

![Graphs showing effects of varying grid resolutions.](image-url)
APPENDIX F. DIAGNOSTIC RESULTS

[Images of diagnostic results graphs]
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
APPENDIX F. DIAGNOSTIC RESULTS

Figure F.1: Monthly mean budgets of $^{222}\text{Rn}$ in kg for eight different pools: Eurasia, North America, North Atlantic and North Pacific, with each area subdivided in the vertical into boundary layer and free troposphere. The boundary layer has been approximated by the collectivity of model layers with a monthly mean pressure greater than 700 hPa.
F.1.3 Sulfur Hexafluoride SF$_6$

The setup for this simulation followed the TransCom 2 protocol [Denning et al., 1999], with the exception of using corrected SF$_6$ emissions and initial mixing ratios from start instead of applying post-adjustments to the simulation results. The year under consideration was 1993 with four years spin-up, starting from a uniform SF$_6$ mixing ratio of 1.93 pptv. TM3 was run in version 3.8 and fed with ERA-15 reanalysis data of all available resolutions, i.e. cg, fg, vfg and xfg.

Surface Maps and Zonal Averages

The following two pages show maps of SF$_6$ annual mean surface mixing ratios and zonal averages of the vertical SF$_6$ distribution in pressure coordinates, for the four experimental resolutions, arranged in the picture panels according to resolutions like

\begin{array}{cc}
\text{cg} & \text{fg} \\
\text{vfg} & \text{xfg} \\
\end{array}
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS
Atlantic Transect

High-volume surface air spot samples were collected during an atlantic cruise of the research ship Polarstern in November 1993 from 40ºN to 56ºS [Maiss et al., 1996]. Their locations are shown in the following picture:

These measurements are compared in figure F.2 with November mean values, computed by TM3 at the same coordinates. They show remarkably little resolution dependency. Offsets of ∼ -1.1 ppt, estimated by weighted least-square fits to the observations, have been subtracted from the model values. The vertical bars show the standard deviations of the respective TM3 November time series.

Figure F.2: Comparison of Polarstern transect measurements with TM3 November means
F.1. THE EFFECTS OF VARYING GRID RESOLUTIONS

Time-Series at Background Stations

In the following panel, SF$_6$ mixing ratio time-series for three sampling sites are depicted and compared with modelling results, for Alert (82.45°N 62.51°W), Izaña (28.3°N 16.48°W) and Neumayer (70.66°S 8.25°W) (from [Maiss et al., 1996]). Offsets of $\sim$ 0.1 pptv were added to the modelling results, and a quadratic trend has been subtracted from modelled and observed mixing ratios.

Furthermore, the annual mean mixing ratio difference between the northern (arctic) background station Alert and the southern (antarctic) background station Neumayer are given below:

<table>
<thead>
<tr>
<th></th>
<th>obs</th>
<th>cg</th>
<th>fg</th>
<th>vfg</th>
<th>xfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta \chi$ [pptv]</td>
<td>0.396</td>
<td>0.381</td>
<td>0.377</td>
<td>0.362</td>
<td>0.359</td>
</tr>
</tbody>
</table>

Interhemispheric Transport

Interhemispheric Exchange Times $\tau_{ex}$ This quantity serves as measure for the resistance of the tropical circulation system (including the intertropical convergence zone ITCZ) to interhemispheric air mass exchange. It is derived from a linear ansatz between the north-south tracer mass flux and the hemispheric tracer mass difference:

$$\bar{F}_{NS} = \frac{1}{\tau_{ex}} (\bar{M}_N - \bar{M}_S).$$
Here, $F_{NS}$ denotes the temporally averaged mass flux from the northern to the southern hemisphere and $M_{N/S}$ the temporally averaged hemispheric tracer masses.

For the present computational experiment, the averaging period stretches across the year 1993. A "two boxes" approximation was used:

$$\tau_{ex} = 2 \frac{\bar{M}_N - \bar{M}_S}{\bar{E}_N - \bar{E}_S - \bar{E}_N - \bar{E}_S - \left( \frac{d(M_N(t) - M_S(t))}{dt} \right)}$$

where $\bar{E}_N$ and $\bar{E}_S$ denote the emission rate in the northern resp. southern hemisphere. The anually averaged time derivative in the denominator was approximated by linear regression of the time series $M_{N/S}(t)$, with typical regression coefficients $r^2 \approx 0.7$. The results show a resolution dependency:

<table>
<thead>
<tr>
<th>resolution</th>
<th>cg</th>
<th>fg</th>
<th>vfg</th>
<th>xfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{ex}$ [y]</td>
<td>1.18</td>
<td>0.951</td>
<td>0.878</td>
<td>0.865</td>
</tr>
</tbody>
</table>

Pole-to-Pole Surface Mixing Ratio Difference This quantity was computed as annual average for 1993:

<table>
<thead>
<tr>
<th>resolution</th>
<th>cg</th>
<th>fg</th>
<th>vfg</th>
<th>xfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{\chi}<em>{NP} - \bar{\chi}</em>{SP}$ [pptv]</td>
<td>0.3897</td>
<td>0.3865</td>
<td>0.3692</td>
<td>0.3668</td>
</tr>
</tbody>
</table>

F.2 Regional Transport: The ATMES-II Modelling Exercise

Tracer release experiments represent an outstanding opportunity for the verification of atmospheric transport models because of two reasons. Firstly, by controlled release of an easily detectable agent the main disadvantage of other verification methods, the lack of exact knowledge about flux strength and location of the tracer source, is removed. Secondly, by organizing a dense sampling network prior to the release, the real atmospheric tracer distribution at the surface or in 3d may be rather well known.

ETEX, the European Tracer EXperiment, included two releases of cyclic perfluorocarbons at Monterfil in the North-West of France (48°03'30"N, 2°00'30"W), with westerly flows over western and central Europe during the experiment. The first release started at 16:00 UTC on October 23, 1994, and lasted 11 hours and 50 minutes. 340 kg of PMCH (perfluoromethylcyclohexane) were released. 168 ground-level sampling stations in western and eastern Europe established the sampling network with an average spacing between two sampling stations of about 80 km (figure F.3). The sampling timestep was 3 hours on average. Airborne campaigns with three aircrafts complemented the ground network.

About two years after the ETEX releases, the ATMES-II (Atmospheric Transport Model Evaluation Study II) modelling exercise was launched. Participants were required to calculate the concentration fields of the first ETEX tracer experiment using ECMWF analyzed meteorological data as input of their own atmospheric transport model. The requirement of using ECMWF data should point out the differences of the participating models in respect of transport calculation and usage of the meteorological data. But this requirement was not compulsory.
F.2. REGIONAL TRANSPORT: THE ATMES-II MODELLING EXERCISE

The TM3 simulation setup for the ATMES-II modelling exercise comprised a model run for the concerning time interval with the highest available input data resolution for that time, the vfg resolution of the NCEP/NCAR reanalysis data \((1.875^\circ \times 1.875^\circ, 28\) layers) with TM3 3.8a. A three-hourly concentration output at all 168 ground-level station locations, synchronous to the reference times of the observations, constituted the data base for the analyses in the subsequent paragraphs. The observational dataset (version etex1_v1.1.960505) was obtained from [ETEX homepage].

Analysis of the Model Results

The following definitions of statistical indicators follow those of the ATMES-II report [Mosca et al., 1998], in order to be comparable with other modelling results. The analysis is subdivided into global analysis, where time- and spacial information is disregarded, spacial analysis, where the time instants are fixed, and temporal analysis, where the locations are fixed.

As one can observe in figure F.4 on p. 128, the point tracer source creates a plume with sharp gradients on small scales. This situation makes high demands on the model’s advection package. Furthermore, to resolve spatial structures comparable in scale with the station mesh-width of about 80 km, a model resolution of at least 40 km would be required\(^1\). With the available horizontal resolution of about 135 km x 210 km one cannot expect too much.

Definitions

In the following list of computed indicators, \(\chi\) denotes the tracer concentration, \(P_i\) the prediction (model value) at time/location \(i\), and \(M_i\) the measurement at time/location \(i\).

\(^1\)With a standard finite differences transport scheme. The situation with TM3 is slightly better, because it uses additional sub-scale information of the tracer distribution within the grid cells.
APPENDIX F. DIAGNOSTIC RESULTS

Bias, the average difference between paired predictions and measurements, $B = \frac{1}{N} \sum_i (P_i - M_i)$, is an indicator for over- or underestimation.

Factor of exceedance. Let $N_{P>M}$ be the number of $(P_i, M_i)$-pairs where the predictions exceed the measurements. The factor of exceedance is the fraction of predictions that are too large, normalized to give zero in case of an equal number of over- and underpredictions: $F_{ex} = \frac{N_{P>M}}{N} - \frac{1}{2}$. It can take values between $-\frac{1}{2}$ (everything is underpredicted) and $+\frac{1}{2}$ (everything is overpredicted).

The $FA_k$ coefficient is the fraction of predictions within a factor $k$ from the measurements, $FA_k = \text{prob}(P \leq k \cdot M \land P \geq M/k)$, for non-zero measurement values. $FA_k$ can take values between 0 and 1.

The normalized mean square error $NMSE$ is an index for the magnitude of the deviations between predictions and measurements: $NMSE = \frac{1}{N_{PM}} \sum_i (P_i - M_i)^2$.

Mean $\rho$ Pearson’s linear correlation coefficient, $-1 \leq \rho \leq 1$.

Arrival time of the tracer cloud at a fixed location. Earliest time instant with $\chi(t) \geq \chi_{max}/e$.

Duration time of the tracer cloud at a fixed location. Time interval with $\chi(t) \geq \chi_{max}/e$.

Global Analysis

<table>
<thead>
<tr>
<th>$FA_2$</th>
<th>$F_{ex}$</th>
<th>$P$</th>
<th>$M$</th>
<th>$\rho$</th>
<th>$B$</th>
<th>$NMSE$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.315</td>
<td>0.35</td>
<td>0.15</td>
<td>0.095</td>
<td>0.43</td>
<td>0.057</td>
<td>16.8</td>
</tr>
</tbody>
</table>

Numerical diffusion, probably induced by the comparatively coarse model resolution, leads to overpredictions at many stations. The factor of exceedance $F_{ex}$ indicates 85 % overpredictions. A positive bias $B$ of 0.057 ng/m$^3$ indicates this trend also. That the bias $B$ does not equal the difference of the means originates from missing values in the observational dataset. Only 31.5 % of the predictions are within a factor of 2 around the measurements. The correlation coefficient of $\rho^2 = 0.18$ is small compared to other ATMES-II models.

Spatial Analysis

This part of the analysis is restricted to a figure of concentration contours at different times after the release start (fig. F.4, p. 128). Probably numerical diffusion, induced by the coarse model resolution, is the reason for comparatively small maximum tracer concentrations on the modelling side, as well as for oversized horizontal dimensions of the tracer cloud. The splitting into a north-western and south-eastern part, as visible in the observations from 36 hours after release start, could not be reproduced by the model at all.
Temporal Analysis

The temporal analysis was carried out for 11 stations, forming approximately two arcs perpendicular to the main cloud trajectory, the first arc at a distance of approximately 600 km, the second at a distance of about 1200-1400 km from the source (figure F.5, p. 129).

<table>
<thead>
<tr>
<th>Station-ID</th>
<th>$\Delta \chi_{\text{max}}$</th>
<th>$\Delta t_a$</th>
<th>$\Delta t_d$</th>
<th>$\varrho$</th>
<th>$B$</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ng/m$^3$]</td>
<td>[h]</td>
<td>[h]</td>
<td>[ng/m$^3$]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NL05</td>
<td>-1.35</td>
<td>-18</td>
<td>15</td>
<td>-0.05</td>
<td>0.02</td>
<td>6.8</td>
</tr>
<tr>
<td>B05</td>
<td>0.56</td>
<td>-15</td>
<td>15</td>
<td>0.29</td>
<td>0.24</td>
<td>16</td>
</tr>
<tr>
<td>NL01</td>
<td>-0.43</td>
<td>-12</td>
<td>9</td>
<td>0.31</td>
<td>0.14</td>
<td>5.8</td>
</tr>
<tr>
<td>D44</td>
<td>0.66</td>
<td>0</td>
<td>0</td>
<td>0.98</td>
<td>0.16</td>
<td>2.7</td>
</tr>
<tr>
<td>DK05</td>
<td>-1.70</td>
<td>-6</td>
<td>12</td>
<td>0.77</td>
<td>-0.19</td>
<td>12</td>
</tr>
<tr>
<td>DK02</td>
<td>-0.27</td>
<td>-6</td>
<td>9</td>
<td>0.78</td>
<td>0.02</td>
<td>1.2</td>
</tr>
<tr>
<td>D42</td>
<td>-1.01</td>
<td>-6</td>
<td>3</td>
<td>0.72</td>
<td>-0.11</td>
<td>2</td>
</tr>
<tr>
<td>D05</td>
<td>-0.18</td>
<td>0</td>
<td>3</td>
<td>0.94</td>
<td>0.02</td>
<td>0.27</td>
</tr>
<tr>
<td>PL03</td>
<td>-0.32</td>
<td>-3</td>
<td>3</td>
<td>0.84</td>
<td>-0.03</td>
<td>1.2</td>
</tr>
<tr>
<td>CR03</td>
<td>-0.05</td>
<td>0</td>
<td>3</td>
<td>0.92</td>
<td>0.03</td>
<td>0.64</td>
</tr>
<tr>
<td>H02</td>
<td>-0.36</td>
<td>0</td>
<td>12</td>
<td>0.76</td>
<td>-0.03</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Underpredictions of the peak height at all stations but two characterize the scene here (negative $\Delta \chi_{\text{max}}$). The tracer front arrives at the first arc far too early, what might be attributed again to numerical diffusion, as well as the high positive bias values especially for the stations of the 1st arc. The duration of the front is too long, partially much too long.

F.2.1 Conclusions

Underpredictions of the peak concentrations but mainly overpredictions in the "global" average, tracer fronts that arrive too early and overlong front durations indicate that the used model resolution is too coarse for this kind of experiment, leading to numerical diffusion as prominent error source. One main feature of the large-scale transport, the splitting into two sub-clouds over the Netherlands, is not reproduced at all.

This experiment will be repeated when more highly-resolved preprocessing data from the ECMWF ERA-40 reanalysis are available.
Figure F.4: PMCH surface concentrations, as observed (left column) and modeled by TM3 (right column, different scale!), for 24, 36, 48 and 60 hours after release start. Contouring method was triangular interpolation.
Figure F.5: Location of the stations selected for temporal analysis (figure taken from [ETEX homepage]).
Appendix G

Getting the Code

To get the most recent TM3 sources, preprocessed meteorological input files and the up-to-date manual please contact

Martin Heimann  
Max Planck Institute for Biochemistry  
P.O. Box 10 01 64  
07701 Jena  
GERMANY  
martin.heimann@bgc-jena.mpg.de

There exists a TM3 mailing list to swap experiences and make suggestions. To sign in, please contact Martin Heimann.