

→ HWMF



HIRLAM DOCUMENTATION MANUAL

System 2.5

June 1996

Edited by E. Källén

CONTENTS	Page
CHAPTER 1	
Introduction	1
CHAPTER 2	
2. The Forecast Model	2.1
2.1 Adiabatic formulation	2.6
2.1.1 The continuous equations	2.6
2.1.2 Finite difference scheme	2.7
2.1.3 The semi-implicit time scheme	2.12
2.1.4 Horizontal diffusion	2.17
2.1.5 Boundary relaxation scheme	2.20
2.1.6 Statistics	2.21
2.1.7 Extra scalars	2.23
2.1.8 The semi-Lagrangian time scheme	2.24
2.2 Physical parameterization	2.25
2.2.1 Radiation	2.26
2.2.2 Vertical diffusion	2.35
2.2.2 A Theory and general procedure	2.35
2.2.2 B Diffusion coefficients	2.40
2.2.2 C Additional computations at the ground	2.40
2.2.2 D Transfer of information	2.42
2.2.3 Condensation and precipitation	2.44
2.2.3 A General	2.44
2.2.3 B Convective release of latent heat and associated cloud cover	2.44
2.2.3 C Stratiform condensation	2.45
2.2.3 D Parameterization of microphysical processes	2.47

2.2.3 E	Numerical treatment of the governing equations	2.50
2.2.5	Soil Processes	2.52
2.2.6	Diagnostic calculation of near surface parameters	2.57
2.3.1	Dynamic part	2.59
2.3.1	Algorithm and calling tree for the dynamics	2.63
2.3.2	Physical Parameterization	2.74
2.4	Data organization	2.80

CHAPTER 3

3.	Initialization	
3.1	The implicit normal mode initialization scheme	3.1
3.1.1	Introduction	3.1
3.1.2	Linear equation system	3.1
3.1.3	Implicit NMI scheme	3.3
3.1.4	Boundary values	3.5
3.2	Computercode organization	3.7
3.3	Data organization	3.8
3.3.1	Table: The input namelists namrun for initialization	3.9

CHAPTER 4

4.	The Analysis System	
4.1	Analysis of wind, mass and humidity fields	4.1
4.1.1	The ECMWF analysis scheme and HIRLAM modifications - an introduction	4.2
4.1.2	Adaptations to the limited area version	4.4
4.1.3	Adaptations to the HIRLAM file structure	4.5
4.1.4	Setting of file unit numbers	4.7

4.1.6	Adaptations to water vapour saturation pressure calculations	4.8
4.1.7	Preparation of the observation file from the ECMWF AOF	4.9
4.1.7	The Script Prepob	4.9
4.1.8	De-staggering of wind components of the first guess and staggering of wind components of analysis increments	4.10
4.1.9	Creation of HIRLAM analysis coefficient files	4.11
4.2	Analysis of sea surface temperature and ice coverage	4.12
4.2.1	General features	4.13
4.2.2	Input of sea-surface temperature and ice coverage data from the Swedish Meteorological and Hydrological Institute	4.13
4.2.3	Method of computations	4.14
4.2.4	Computer code organization	4.14
4.2.5	Data organization	

CHAPTER 5

5.	Data structures	5.1
5.1	Fields	5.1
5.1.1	General	5.1
5.1.2	Definition of mandatory descriptor record (DDR)	5.1
5.1.3	GRIB format	5.4
5.1.4	Programs for packing and unpacking	5.5
5.1.5	Indicator of Parameter	5.7
5.2	Data Structures - Observations (BUFR)	5.8
5.2.2	BUFR decoding; Program Library BUFR	5.9
5.2.3	BUFR tables	5.11
5.2.4	Script Make BUFRtabs	5.11
5.2.5	Program BUTA	5.12

5.3	Statistics files	5.12
5.3.1	General features of the statistics files	5.12
5.3.2	Structure and contents	5.12
5.3.3	Transmission of statistics files	5.13
5.3.4	Contents of the statistics files	5.13
5.3.5	FORTRAN formats for records in the statistics files	5.18

CHAPTER 6

6.	Pre-processing	6.1
6.1	Horizontal interpolation	6.1
6.1.1	Selection of boundary files	6.1
6.1.2	File format transformation	6.2
6.1.3	Horizontal interpolation	6.2
6.1.4	Bitmaps	6.4
6.2	Vertical interpolation from pressure levels to HIRLAM hybrid levels	6.4
6.2.1	Notations	6.5
6.2.2	Calculation of surface pressure	6.6
6.2.3	Vertical interpolations	6.9
6.2.4	Program organization and data organization	6.10
6.3	Vertical interpolation from one hybrid level system to another hybrid level system	6.12
6.3.1	Notations	6.13
6.3.2	Estimation of a preliminary HIRLAM surface pressure	6.15
6.3.3	Vertical interpolation of temperature, humidity and wind components	6.18
6.3.4	Computation of the final HIRLAM surface pressure	6.19
6.3.6	Program organization and data organization	6.20

CHAPTER 7

7.	Postprocessing	7.1
7.1	Vertical interpolation and extrapolation	7.1
7.2	Mean sea level pressure	7.3
7.3	Time Series Files	7.4
7.3.1	Generating Time Series Files	7.4
7.3.1.1	Initialising - INITRF	7.4
7.3.1.2	Copy data to Time Series buffer - CPDTRF	7.6
7.3.1.3	Writing the Time Series File - WRITRF	7.7
7.3.2	Reading Time Series Files	7.7
7.3.2.1	Get information on Time Series File - GINFSF	7.7
7.3.2.2	Reading Single Level Time Series - RDSESF	7.9
7.3.2.3	Reading Multi Level Time Series - RDMETF/RAMETF	7.10
7.4	Vertical velocity	7.11
7.5	Relative humidity	7.12
7.6	Surface fluxes	7.13
7.7	Computer code organization and namelist parameters	7.14

CHAPTER 8

8.	The surface orography and climate system	8.1
8.1	Introduction	8.1
8.2	Orography, surface roughness and fraction of land	8.3
8.3	Surface climate fields	8.7
8.4	HIRLAM climate data files	8.8

CHAPTER 9

9.3	General verification against observations	9.1
-----	---	-----

CHAPTER 10

10. The HIRLAM system under Unix
 A unified script system

This chapter has a separate subsection and page numbering

I. INTRODUCTION

The HIRLAM forecasting system is a limited area, short range weather forecasting system. Its two main components are the analysis system and the forecasting model. This manual contains a documentation of both parts of the system. Basic scientific aspects of the system as well as technical details are included.

The forecasting system has originally been developed as a co-operative project between seven meteorological institutes. The institutes are

- Danish Meteorological Institute, Copenhagen, Denmark (DMI)
- Finnish Meteorlogical Institute, Helsinki, Finland (FMI)
- Icelandic Meteorological Office, Reikjavik, Iceland (IMO)
- Irish Meteorological Service, Dublin, Ireland (IMS)
- Norwegian Meteorological Institute, Oslo, Norway (DNMI)
- Royal Dutch Meteorological Institute, De Bilt, the Netherlands (KNMI)
- Swedish Meteorological and Hydrological Institute, Norrköping Sweden (SMHI).

On a research project basis the French meteorological service, Meteo-France, has joined the HIRLAM project and more recently also the Spanish Meteorological Institute (INM) is taking part in the HIRLAM project.

The first HIRLAM system, HIRLAM level 1, was ready in 1990 and is documented in the first version of this documentaion manual. The HIRLAM project continued into a second phase, HIRLAM 2, and is now in its third phase, HIRLAM 3. The HIRLAM system is continuously being updated with enhancements and extensions. We have adopted a numbering system where the present basic model is HIRLAM 2 with an appended number corresponding to the latest model version. The latest major update took place in October 1995 and the present model version is 2.5. In the model code an additional number is appended which refers to the last minor modification of the system (i.e. HIRLAM 2.5.x).

As the forecasting system is over a limited area, externally prescribed boundary values are required at the lateral boundaries. The HIRLAM system is intended for use with lateral boundary values from a global model such as the ECMWF forecasting system. It is also possibel to run a sequence of HIRLAM systems nested into each other, but there is only a one way interaction at the lateral boundaries. The co-ordinate system used is a rotated latitude-longitude grid horizontally and a hybrid p-sigma system in the vertical. The analysis is of the optimum interpolation type and intended for an intermittent data assimilation system. The documentation of the analysis in the present version of the manual is not at the same level of detail as the documentation of the forecasting model. The main reason is that present development work on a completely new variational data assimilation system is underway and we intend to wait with an update of the analysis documentation until the variational system is ready.

The forecast model documentation contains both a description of the continuous formulation of the governing equations as well as the discretised system. A grid point representation is used together with a semi-implicit, leap frog time stepping scheme. The physical parameterisations are treated in detail and a complete list of variable names in the forecast model code is included. To define the properties of the earth's surface a special, so called climate file generation system is

included in HIRLAM. This part of the documentation manual is not up to date at present. The HIRLAM system also contains a verification package where the forecasted fields can be checked against observations. A documentation of the verification package is lacking at the present time.

To run the forecasting system on a computer a UNIX based operating system is required. This manual contains a description of the UNIX scripts which are included in HIRLAM. The FORTRAN code is as standardized as possible, at present the forecasting system has been run mainly on Cray and Convex supercomputers. The forecasting model runs on a much wider range of computers including some workstations. The main coding limitations are present in the analysis system. Pre- and post-processing software is available, once again the documentation is not entirely up to date. To visualize the forecast results a graphics package is required, this is not included in HIRLAM.

This manual is a substantially revised version of the original HIRLAM 1 forecast system manual published in 1990. A large number of people from the meteorological institutes participating in HIRLAM have contributed to this manual. The first version of the HIRLAM manual was compiled and written by Nils Gustafsson and Per Källberg at SMHI. The present revised manual has been built on the first version, but due to technical problems with word processing and extensive revisions in the HIRLAM system a large part of the manual has been re-written. The following persons have contributed with substantial text revisions:

Gerard Cats, KNMI
Stefan Gollvik, SMHI
Bent Hansen Sass, DMI
Jan Erik Haugen, DNMI
Toon Moene, KNMI
Per-Olof Olofsson, SMHI
Laura Rontu, FMI
Ben Wijchers Schreur, KNMI
Dance Zurovac-Jevtic, SMHI (and Department of Meteorology, Stockholm University, Sweden)

Secretarial assistance has been given by Åsa Nilsson and Gun Sigurdson at SMHI and Anne-Grethe Svensen at DNMI.

Norrköping in June 1996.

Erland Källén, HIRLAM project leader

2. THE FORECAST MODEL

The HIRLAM Level 2 forecast model has been developed from earlier limited area versions of the first ECMWF gridpoint model. The present model has undergone major redesigns and revisions, the most important are the following:

- (1) The model is formulated in hybrid coordinates in the vertical and prepared for a general conformal horizontal coordinate system or map-projection. At present the model employs hybrid coordinates in the vertical and rotated spherical coordinates in the horizontal.
- (2) The physics parameterization is inspired in many aspects from the ECMWF model, while some other parts are developed entirely by the HIRLAM group.
- (3) The model has been rewritten in order to utilize the potential of vector-computers with large internal memories. Special emphasis has been put on preparing the model for alternative numerical methods, e.g. spectral and/or semi-Lagrangian techniques. The physics package is made more independent of the dynamical part and operates on subareas which consist of a number of grid-columns.

Table of constants and symbols

Symbol	Meaning	Value	Unit
a	radius of earth	$6.371 \cdot 10^6$	m
$A_{k+1/2}$	constant defining vertical coordinate		Pa
$B_{k+1/2}$	constant defining vertical coordinate		
c^2	eigenvalues for the vertical normal modes in the semi-implicit and normal mode scheme		
c_{pd}	specific heat for dry air at constant pressure	1004.64	J/kg/K
c_{pv}	specific heat for moist air at constant pressure	1869.46	J/kg/K
d	divergence		1/s
E	kinetic energy in the momentum equation <u>and</u> eigenvector matrix in the semi-implicit and the nonlinear normal mode scheme		m^2/s^2
F	mean value of Coriolis parameter		1/s
f	Coriolis parameter		1/s
g	acceleration of gravity	9.8065	m/s^2
G	vertical coupling matrix in semi-implicit scheme and in non-linear normal mode initialization scheme		
H	equivalent depths for the vertical normal modes in the semi-implicit and normal mode schemes		
h_x	metric coefficient along x-coordinate		
H_x	mean value of metric coefficient along x-coordinate		
h_y	metric coefficient along y-coordinate		
H_y	mean value of metric coefficient along y-coordinate		
k	index for numbering model layers		

K	horizontal diffusion coefficient	
K_x	tendency of variable X due to horizontal diffusion	[X]/s
n	index for numbering timesteps	
N_x	nonlinear term for variable X in nonlinear normal mode initialization	[X]/s
NLEV	number of model levels	
P	auxiliary potential in semi-implicit scheme and in non-linear normal mode initialization scheme	
p	pressure	Pa
p_o	reference pressure for vertical coordinate	Pa
p_r	reference pressure in semi-implicit scheme and in non-linear normal mode initialization scheme	Pa
p_s	surface pressure	Pa
P_x	tendency for variable X due to physical parameterization	(X)/s
q_l	cloud liquid water	kg/kg
R_d	gas constant for dry air	287.04 J/kg/K
R_v	gas constant for water vapour	461.51 J/kg/K
R_x	total explicit tendency for variable X	(X)/s
x	"zonal" coordinate in model coordinate system	degrees
X	prognostic variable	
\bar{X}_i^x	average operator for variable X in x-direction	[X]
	$\bar{X}_i^x = (X_{i+1/2} + X_{i-1/2})/2 \text{ where } x = i\Delta x$	
δx	"zonal" distance on earth	m

$\delta_x X_i$	finite difference operator for variable X in x-direction	[X]/[x]
	$\delta_x X_i = (X_{i+1/2} - X_{i-1/2})/\Delta x \text{ where } x=i\Delta x$	
$\delta_x^2 X_i$	finite difference operator for the second order derivative of variable X in x-direction	[X]/[x] ²
	$\delta_x^2 X_i = (X_{i-1} - 2X_i + X_{i+1})/(\Delta x)^2 \text{ where } x=i\Delta x$	
y	"meridional" coordinate in model coordinate system	degrees
δy	"meridional" distance on earth	m
q	specific humidity	kg/kg
t	time	s
Δt	timestep	s
T	temperature	K
T_r	reference temperature in semi-implicit scheme and in nonlinear normal mode initialization scheme	K
T_v	$= (1+(1/\epsilon-1)q)T$; virtual temperature	K
u	wind component along x-coordinate	m/s
U	auxiliary wind component $U = u \overline{\Delta p}^x$	Pa m/s
v	wind component along y-coordinate	m/s
\mathbf{v}_k	$= (u, v)$: horizontal wind vector	m/s
V	auxiliary wind component $V = v \overline{\Delta p}^y$	Pa m/s
\mathbf{V}	$= (U, V)$: auxiliary wind vector	Pa m/s
Z	auxiliary absolute vorticity	1/(Pa s)
α_k	weight function between model levels	
α_b	boundary relaxation function	

β_k	weight function between model layers	
δ	$= c_{pv}/c_{pd}$	
ε	$= R_d/R_v$	
ε_f	coefficient for time-filter	0.05
η	$= A/p_0 + B$: generalized vertical coordinate	
$\dot{\eta}$	$= d\eta/dt$: η -coordinate vertical velocity	1/s
λ	longitude in rotated spherical coordinates	
θ	latitude in rotated spherical coordinates	
κ	$= R_d/c_{pd}$	0.286
ξ	relative vorticity	1/s
ω	pressure vertical velocity	Pa/s
ϕ	geopotential	m^2/s^2
ϕ_s	surface geopotential	m^2/s^2
τ	constant matrix in semi-implicit scheme and in non-linear normal mode initialization scheme	
ν	constant matrix in semi-implicit scheme and in non-linear normal mode initialization scheme	
γ	constant matrix in semi-implicit scheme and in non-linear normal mode initialization scheme	
ΔX_k	vertical difference $X_{k+1/2} - X_{k-1/2}$	

2.1 Adiabatic formulation.

2.1.1 The continuous equations

The equations will be written down for a general pressure based and terrain following vertical coordinate $\eta(p_s, p_e)$ where

$$\eta(0, p_s) = 0 \quad \wedge \quad \eta(p_s, p_e) = 1 \quad (2.1.1.1)$$

The formulation corresponds to the hybrid system used at ECMWF (Simmons and Strüfing, 1981), with some modifications because the continuity equation is integrated upwards in the HIRLAM model.

The model is programmed for a spherical coordinate system (λ, θ) , but in the formulation and in the code two metric coefficients (h_x, h_y) have been introduced. This is done to prepare the model for any orthogonal coordinate system or map projection with axes (x, y) . However, this is not fully implemented in all parts of the dynamics. For a distance $(\delta X, \delta Y)$ on the earth, it yields that

$$\delta X = h_x \delta x \quad \wedge \quad \delta Y = h_y \delta y \quad (2.1.1.2)$$

In case of spherical rotated coordinates we have that

$$\delta X = a \cos \theta \delta \lambda \quad \wedge \quad \delta Y = a \delta \theta \quad (2.1.1.3)$$

The momentum, thermodynamic and moisture equations are

$$\frac{\partial u}{\partial t} = (f + \xi)v - \dot{\eta} \frac{\partial u}{\partial \eta} - \frac{1}{h_x} [R_d T_v \frac{\partial \ln p}{\partial x} + \frac{\partial}{\partial x}(\varphi + E)] + P_u + K_u \quad (2.1.1.4)$$

$$\frac{\partial v}{\partial t} = -(f + \xi)u - \dot{\eta} \frac{\partial v}{\partial \eta} - \frac{1}{h_y} [R_d T_v \frac{\partial \ln p}{\partial y} + \frac{\partial}{\partial y}(\varphi + E)] + P_v + K_v \quad (2.1.1.5)$$

where

$$\xi = \frac{1}{h_x h_y} [\frac{\partial}{\partial x}(h_y v) - \frac{\partial}{\partial y}(h_x u)] \quad (2.1.1.6)$$

$$E = \frac{1}{2} (u^2 + v^2) \quad (2.1.1.7)$$

$$\frac{\partial T}{\partial t} = - \frac{u}{h_x} \frac{\partial T}{\partial x} - \frac{v}{h_y} \frac{\partial T}{\partial y} - \dot{\eta} \frac{\partial T}{\partial \eta} + \frac{\kappa T_v \omega}{(1 + (\delta - 1)q)p} + P_T + K_T \quad (2.1.1.8)$$

$$\frac{\partial q}{\partial t} = - \frac{u}{h_x} \frac{\partial q}{\partial x} - \frac{v}{h_y} \frac{\partial q}{\partial y} - \dot{\eta} \frac{\partial q}{\partial \eta} + P_q + K_q \quad (2.1.1.9)$$

The terms P_x and K_x represent tendencies from the physical parameterization and horizontal diffusion, respectively

The hydrostatic equation takes the form

$$\frac{\partial \phi}{\partial \eta} = -\frac{R_d T_v}{p} \frac{\partial p}{\partial \eta} \quad (2.1.1.10)$$

and the continuity equation is

$$\frac{\partial}{\partial \eta} \frac{\partial p}{\partial t} + \nabla \cdot (\vec{v}_h \frac{\partial p}{\partial \eta}) + \frac{\partial}{\partial \eta} (\dot{\eta} \frac{\partial p}{\partial \eta}) = 0 \quad (2.1.1.11)$$

The definition of the divergence operator is

$$\nabla \cdot \vec{v}_h = \frac{1}{h_x h_y} [\frac{\partial}{\partial x} (h_y u) + \frac{\partial}{\partial y} (h_x v)] \quad (2.1.1.12)$$

By integrating the continuity equation, using the boundary conditions $\dot{\eta}=0$ at $\eta=0$ and $\eta=1$, we obtain the equation for the surface pressure tendency

$$\frac{\partial p_s}{\partial t} = - \int_0^1 \nabla \cdot (\vec{v}_h \frac{\partial p}{\partial \eta}) d\eta \quad (2.1.1.13)$$

the equation for pressure vertical velocity

$$\omega = \frac{\partial p_s}{\partial t} + \int_{\eta}^1 \nabla \cdot (\vec{v}_h \frac{\partial p}{\partial \eta}) d\eta + \vec{v}_h \cdot \nabla p \quad (2.1.1.14)$$

and the equation for $\dot{\eta}$

$$\dot{\eta} \frac{\partial p}{\partial \eta} = (1 - \frac{\partial p}{\partial p_s}) \frac{\partial p_s}{\partial t} + \int_{\eta}^1 \nabla \cdot (\vec{v}_h \frac{\partial p}{\partial \eta}) d\eta \quad (2.1.1.15)$$

An additional equation for cloud liquid water is included, which takes the following form.

$$\frac{\partial q_l}{\partial t} = -\frac{u}{h_x} \frac{\partial q_l}{\partial x} - \frac{v}{h_y} \frac{\partial q_l}{\partial y} + P_{q_l} \quad (2.1.1.16)$$

In the adiabatic form ($P=0$) there is no vertical advection term or horizontal diffusion of q_l in contrast to the equation for specific humidity q (eq. 2.1.1.9). Furthermore the horizontal advection term is formulated as an upstream scheme which is described in section 2.1.2.

2.1.2 Finite difference scheme

From the general expression

$$p_{k+1/2} = A_{k+1/2}(\eta) + B_{k+1/2}(\eta) \cdot p_s(x,y) \quad \text{for } k=0, \dots, NLEV \quad (2.1.2.1)$$

which defines the vertical surfaces, pure pressure surfaces are obtained for B=0 and pure sigma surfaces for A=0.

The model is formulated for Arakawa C-grid, and for the conservation properties of the horizontal scheme the following auxilliary velocities are defined:

$$\begin{aligned} U_k &= \overline{\Delta p_k}^x u_k \\ V_k &= \overline{\Delta p_k}^y v_k \end{aligned} \quad (2.1.2.2)$$

The first step is to calculate the surface pressure tendency

$$\begin{aligned} \frac{\partial p_s}{\partial t} &= - \sum_{j=1}^{NLEV} \frac{1}{h_x h_y} [\delta_x(h_y U_j) + \delta_y(h_x V_j)] \\ &= - \frac{1}{h_x h_y} [\delta_x(h_y \sum_{j=1}^{NLEV} U_j) + \delta_y(h_x \sum_{j=1}^{NLEV} V_j)] \end{aligned} \quad (2.1.2.3)$$

The remaining tendencies can now be found layer by layer, by integrating the continuity equation and the hydrostatic equation upwards. The hydrostatic equation is discretized as

$$\Psi_{k+1/2} - \Psi_{k-1/2} = -R_d(T_v)_k \Delta ln p_k \quad (2.1.2.4)$$

To obtain full level values of geopotential the computations are divided into two parts for each layer:

$$\begin{aligned} \Psi_k &= \Psi_{k+1/2} + \alpha_k R_d(T_v)_k \\ \Psi_{k-1/2} &= \Psi_k + \beta_k R_d(T_v)_k \end{aligned} \quad (2.1.2.5)$$

where

$$\begin{aligned} \alpha_k &= 1 - \frac{P_{k-1/2}}{\Delta p_k} \Delta ln p_k \quad \text{for } k=2, \dots, NLEV \\ \alpha_1 &= \ln(2) \\ \beta_k &= \Delta ln p_k - \alpha_k \quad \text{for } k=2, \dots, NLEV \end{aligned} \quad (2.1.2.6)$$

$$\Psi_{NLEV+1/2} = \Psi_s$$

From the continuity equation in finite difference form

$$(\dot{\eta} \frac{\partial p}{\partial \eta})_{k+1/2} = (1 - B_{k+1/2}) \frac{\partial p_s}{\partial t} + \sum_{j=k+1}^{NLEV} \nabla \cdot \vec{V}_j \quad (2.1.2.7)$$

the following recurrence formula is used

$$(\dot{\eta} \frac{\partial p}{\partial \eta})_{k-1/2} = (\dot{\eta} \frac{\partial p}{\partial \eta})_{k+1/2} + \Delta B_k \frac{\partial p_s}{\partial t} + \nabla \cdot \vec{V}_k \quad (2.1.2.8)$$

with

$$(\dot{\eta} \frac{\partial p}{\partial \eta})_{1/2} = (\dot{\eta} \frac{\partial p}{\partial \eta})_{NLEV+1/2} \equiv 0 \quad (2.1.2.9)$$

The vertical advection term is given by

$$(\dot{\eta} \frac{\partial X}{\partial \eta})_k = \frac{1}{2\Delta p_k} [(\dot{\eta} \frac{\partial p}{\partial \eta})_{k+1/2} (X_{k+1} - X_k) + (\dot{\eta} \frac{\partial p}{\partial \eta})_{k-1/2} (X_k - X_{k-1})] \quad (2.1.2.10)$$

Since the computations are done layer by layer, the term is calculated in two parts, so that only one layer of $\dot{\eta} \partial p / \partial \eta$ has to be stored.

The energy conversion term is written as

$$\begin{aligned} \left(\frac{\kappa T_v \omega}{1 + (\delta - 1)q} \right)_k &= \frac{\kappa}{[1 + (\delta - 1)q_k] \Delta p_k} [\{ \Delta \ln p_k \left(\frac{\partial p_s}{\partial t} + \sum_{j=k+1}^{NLEV} \nabla \cdot \vec{V}_j \right) + \beta_k \nabla \cdot \vec{V}_k \} (T_v)_k \\ &\quad + \frac{1}{h_x h_y} \{ \overline{U_k (T_v)_k^x} h_y \delta_x \ln p_k^x + \overline{V_k (T_v)_k^y} h_x \delta_y \ln p_k^y \}] \end{aligned} \quad (2.1.2.11)$$

where $\ln p_k$ is computed as

$$\ln p_k = \frac{1}{\Delta p_k} (p_{k+1/2} \ln p_{k+1/2} - p_{k-1/2} \ln p_{k-1/2}) \quad (2.1.2.12)$$

and $\Delta \ln p_k = 0$ for $k=1$. Bearing in mind these expressions, the equations for the momentum, temperature and humidity with the remaining terms become:

$$\begin{aligned} \left(\frac{\partial u}{\partial t} \right)_k &= \frac{1}{h_x^x} \overline{Z_k^y V_k h_x^{xy}} - \frac{1}{h_x^x} [\delta_x (\varphi + E) + R_d \overline{(T_v)^x} \delta_x \ln p]_k \\ &\quad - \frac{1}{\Delta p_k^x} \overline{[(\dot{\eta} \frac{\partial p}{\partial \eta}) \Delta_\eta u]}_k + P_u + K_u \end{aligned} \quad (2.1.2.13)$$

$$\begin{aligned} \frac{\partial v}{\partial t}_k = & -\frac{1}{h_y^y} \overline{Z}_k^x \overline{U_k h_y^{xy}} - \frac{1}{h_y^y} [\delta_y (\varphi + E) + R_d \overline{(T_v)^y} \delta_y \ln p]_k \\ & - \frac{1}{\Delta p_k^y} \overline{[(\dot{\eta} \frac{\partial p}{\partial \eta}) \Delta_\eta v]}_k + P_v + K_v \end{aligned} \quad (2.1.2.14)$$

where

$$E_k = \frac{1}{2} \left[\frac{1}{h_y} \overline{u^2 h_y^x} + \frac{1}{h_x} \overline{v^2 h_x^y} \right] \quad (2.1.2.15)$$

$$Z_k = \frac{1}{h_x h_y \Delta p_k^{xy}} [\overline{f h_x h_y^{xy}} + \delta_x (\overline{h_y^y} v) - \delta_y (\overline{h_x^x} u)]_k \quad (2.1.2.16)$$

$$\begin{aligned} \left(\frac{\partial T}{\partial t} \right)_k = & -\frac{1}{\Delta p_k} \left\{ \frac{1}{h_x h_y} [\overline{U h_y \delta_x T^x} + \overline{V h_x \delta_y T^y}]_k + \overline{[(\dot{\eta} \frac{\partial p}{\partial \eta}) \Delta_\eta T]}_k \right\} \\ & + \left[\frac{\kappa T_v \omega}{(1 + (\delta - 1)q)p} \right]_k + P_T + K_T \end{aligned} \quad (2.1.2.17)$$

$$\begin{aligned} \left(\frac{\partial q}{\partial t} \right)_k = & -\frac{1}{\Delta p_k} \left\{ \frac{1}{h_x h_y} [\overline{U h_y \delta_x q^x} + \overline{V h_x \delta_y q^y}]_k + \overline{[(\dot{\eta} \frac{\partial p}{\partial \eta}) \Delta_\eta q]}_k \right\} \\ & + P_q + K_q \end{aligned} \quad (2.1.2.18)$$

The finite difference form of the equation for cloud water is

$$\begin{aligned} \left(\frac{\partial q_l}{\partial t} \right)_k = & -\frac{1}{\Delta p_k h_x h_y} [\overline{U h_y^x \delta_x q_l^x} - |\overline{U h_y^x}| \frac{\Delta x}{2} \delta_x^2 q_l \\ & + \overline{V h_x^y \delta_y q_l^y} - |\overline{V h_x^y}| \frac{\Delta y}{2} \delta_y^2 q_l]_k + P_{ql} \end{aligned} \quad (2.1.2.19)$$

This is the usual upstream advection scheme written in a conservative form similar to the equation for specific humidity. In order to see the behavior of this equation, assume for the moment that $v = P = 0$. Then

$$\left(\frac{\partial q_l}{\partial t} \right)_k = \begin{cases} -\frac{1}{\Delta p_k h_x h_y} \overline{U_k h_y^x} [(q_l)_i - (q_l)_{i-1}] / \Delta x & \text{if } u_k > 0 \\ +\frac{1}{\Delta p_k h_x h_y} \overline{U_k h_y^x} [(q_l)_i - (q_l)_{i+1}] / \Delta x & \text{if } u_k < 0 \end{cases} \quad (2.1.2.20)$$

Figure 2.1. Vertical structure and horizontal grid of the HIRLAM Level 2 limited area model.

2.1.3 The semi-implicit time scheme

The Eulerian semi-implicit scheme is the default time stepping scheme in the HIRLAM model. This algorithm treats the main terms responsible for gravity waves in an unconditionally stable manner, and the time step can be increased compared with an explicit scheme until the CFL criterion for advection is met. The scheme is formulated so that the implicit adjustment is given as additional terms to the explicit tendency:

$$\begin{aligned}\overline{\delta_t u}^t &= R_u - \frac{1}{h_x} \delta_x (\frac{1}{2} \Delta_{tt} P) \\ \overline{\delta_t v}^t &= R_v - \frac{1}{h_y} \delta_y (\frac{1}{2} \Delta_{tt} P) \\ \overline{\delta_t T}^t &= R_T - \tau \frac{1}{2} \Delta_{tt} d \\ \overline{\delta_t \ln p_s}^t &= R_{\ln p_s} - v \frac{\tau}{2} \Delta_{tt} d\end{aligned}\tag{2.1.3.1}$$

where the terms R_x represent the explicit dynamic tendency including horizontal diffusion terms K_x . The tendencies due to physical parameterization, P_x , are added after the semi-implicit adjustment. An auxilliary variable P is introduced which is defined:

$$P = \gamma T + R_d T_r \ln p_s\tag{2.1.3.2}$$

and the horizontal divergence is

$$d = \nabla \cdot \vec{V}_k = \frac{1}{h_x h_y} [\frac{\partial}{\partial x} (h_y u) + \frac{\partial}{\partial y} (h_x v)]$$

The temporal operators are defined as

$$\overline{\delta_t X}^t = (X^{n+1} - X^{n-1}) / 2 \Delta t\tag{2.1.3.3}$$

and

$$\Delta_{tt} X = X^{n+1} + X^{n-1} - 2X^n\tag{2.1.3.4}$$

where n is the time step counter, $t = n \Delta t$.

The matrices γ , τ and v include constants in the vertical discretization. The linearization is done around a constant reference temperature T_r and surface pressure p_r , where p_r automatically disappears if all levels are sigma-surfaces. Their values are

$$T_r = 300 \text{ K} \quad \text{and} \quad p_r = 80000 \text{ Pa}$$

The matrices are defined as

$$\gamma T_k = \alpha_k^r R_d T_k + \sum_{j=k+1}^{NLBV} R_d T_j \Delta ln p_j^r \quad (2.1.3.5)$$

$$(\tau d)_k = \kappa T_r \left(\frac{\Delta ln p_k^r}{\Delta p_k^r} \sum_{j=1}^{k-1} d_j \Delta p_j^r + \alpha_k^r d_k \right) \quad (2.1.3.6)$$

$$v^T d = \frac{1}{p_r} \sum_{j=1}^{NLBV} d_j \Delta p_j^r \quad (2.1.3.7)$$

where the subscript r refers to the values at the reference pressure $p=p_r$.

The semi-implicit system is solved in three main parts, by first from the explicit forecasted values form the right hand side of a Helmholtz equation for $\Delta_{tt} d$. When the solution of the Helmholtz equation is found, we obtain the semi-implicit corrections as shown below:

If we let the subscript e denote the explicit forecasted value,

$$F_e^{n+1} = F^{n-1} + 2\Delta t \cdot R(n-1, n)$$

we have that

$$T^{n+1} = T_e^{n+1} - \Delta t \tau \Delta_{tt} d \quad (2.1.3.8)$$

$$ln p_s^{n+1} = ln p_{se}^{n+1} - \Delta t v^T \Delta_{tt} d \quad (2.1.3.9)$$

and

$$\Delta_{tt} P = P_e^{n+1} + P^{n-1} - 2 \cdot P^n \quad (2.1.3.10)$$

In the momentum equation, we define

$$\vec{v}_{prel}^{n+1} = \vec{v}_e^{n+1} - \Delta t \nabla (\Delta_{tt} P_e) \quad (2.1.3.11)$$

The final velocity is then given by

$$\vec{v}^{n+1} = \vec{v}_{prel}^{n+1} - \Delta t \nabla (P^{n+1} - P_e^{n+1}) \quad (2.1.3.12)$$

An expression for $P^{n+1} - P_e^{n+1}$ can be found from the prognostic equation for P, which gives

$$P^{n+1} - P_e^{n+1} = - \Delta t G \Delta_{tt} d \quad (2.1.3.13)$$

where $G = \gamma \tau + R_d T_r v^T$ is the vertical structure matrix.

Substituted back into equation (2.1.3.12), we get

$$\vec{v}^{n+1} = \vec{v}_{prel}^{n+1} + (\Delta t)^2 G \nabla (\Delta_{tt} d) \quad (2.1.3.14)$$

This gives the three-dimensional Helmholtz equation for $\Delta_{tt} d$

$$\Delta_{tt} d - (\Delta t)^2 G \nabla^2 (\Delta_{tt} d) = \Delta_{tt}^{prel} d \quad (2.1.3.15)$$

where

$$\Delta_{tt}^{prel} d = \nabla \cdot (\vec{v}_{prel}^{n+1} + \vec{v}^{n-1} - 2 \cdot \vec{v}^n) \quad (2.1.3.16)$$

The solution of the three-dimensional Helmholtz equation (2.1.3.15) for the divergence is found by transformation into NLEV shallow water equations, where NLEV is the number of model levels carrying the divergence d . This equation can be vertically decoupled if it is premultiplied with the matrix E^{-1} , where E consists of the eigenvectors of the G matrix. Then $E^{-1}GE$ is a diagonal matrix with the corresponding eigenvalues $c^2 = gH$ as elements, where H is the equivalent depth for each vertical mode. In terms of vertical modes the equation becomes

$$\Delta_{tt} \tilde{d} - (\Delta t)^2 c^2 \nabla^2 (\Delta_{tt} \tilde{d}) = \Delta_{tt}^{prel} \tilde{d} \quad (2.1.3.17)$$

where the tilde denotes a projection of a field onto the vertical eigenmodes. These equations, with the lateral boundary condition equal to zero, are solved by use of Fourier sine-transformation in the west-east direction followed by Gaussian elimination in the south-north direction. After vertical coupling of the solution (premultiplication with the E -matrix) the final semi-implicitly corrected values are obtained by use of equations (2.1.3.8), (2.1.3.9) and (2.1.3.14).

After completion of the semi-implicit adjustment, boundary relaxation and physical parameterization, the values at timestep n are time-filtered (Asselin, 1972) according to the equation.

$$X_f^n = X^n + \epsilon_f (X_f^{n-1} + X^{n+1} - 2 \cdot X^n) \quad (2.1.3.18)$$

where the subscript f means a time filtered value and $\epsilon_f = 0.05$ is used as a default value.

Finite difference formulation of the Helmholtz equation

The model is in general prepared for any orthogonal map projection, but the Helmholtz solver is coded for (rotated) spherical coordinates (λ, θ) which is the default map projection. The Laplacian operator formulated for a general projection is

$$\nabla_{xy}^2 D = \frac{1}{h_x h_y} \left[\frac{\partial}{\partial x} \left(\frac{h_y}{h_x} \frac{\partial D}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{h_x}{h_y} \frac{\partial D}{\partial y} \right) \right] \quad (2.1.3.19)$$

where $D = \Delta_{tt} \tilde{d}$. In terms of spherical coordinates, it becomes

$$\nabla_{\lambda, \theta}^2 D = \frac{1}{a^2} \left[\frac{1}{\cos^2 \theta} \frac{\partial^2 D}{\partial \lambda^2} + \frac{1}{\cos \theta} \frac{\partial}{\partial \theta} \left(\cos \theta \frac{\partial D}{\partial \theta} \right) \right] \quad (2.1.3.20)$$

The finite difference formula for this operator is

$$\begin{aligned}\nabla_{ij}^2 D &= \frac{1}{a^2} \left\{ \left[\frac{1}{\cos^2 \theta_j} \delta_\lambda^2 - \frac{\cos \theta_{j-1/2} + \cos \theta_{j+1/2}}{\cos \theta_j (\Delta \theta)^2} \right] D_j \right. \\ &\quad \left. + \frac{\cos \theta_{j-1/2}}{\cos \theta_j (\Delta \theta)^2} D_{j-1} + \frac{\cos \theta_{j+1/2}}{\cos \theta_j (\Delta \theta)^2} D_{j+1} \right\} \end{aligned} \quad (2.1.3.21)$$

where $\theta = j \Delta \theta$. By use of Fourier sine transformation in the λ -direction, where the Laplacian in the λ -direction is a linear operator, this equation has been separated into a coupled system in the θ -direction for each Fourier component D_l . In order to have a consistent finite difference scheme in both directions, δ_λ^2 is evaluated in a way which is equivalent to the second order finite difference operator

$$\begin{aligned}\delta_\lambda^2 D &= \sum_l D_l [e^{il(\lambda-\Delta\lambda)} - 2e^{il\lambda} + e^{il(\lambda+\Delta\lambda)}]/(\Delta\lambda)^2 \\ &= -\left(\frac{2}{\Delta\lambda}\right)^2 \sum_l D_l \sin^2\left(\frac{l\Delta\lambda}{2}\right)\end{aligned} \quad (2.1.3.22)$$

After a Fourier transform the Helmholtz equation for each wavenumber l becomes

$$\begin{aligned}D_j + (\Delta t)^2 \frac{c^2}{a^2} \left\{ \left[\frac{1}{\cos^2 \theta_j} \left(\frac{2}{\Delta \lambda} \right)^2 \sin^2\left(\frac{l\Delta\lambda}{2}\right) + \frac{\cos \theta_{j-1/2} + \cos \theta_{j+1/2}}{\cos \theta_j (\Delta \theta)^2} \right] D_j \right. \\ \left. - \frac{\cos \theta_{j-1/2}}{\cos \theta_j (\Delta \theta)^2} D_{j-1} - \frac{\cos \theta_{j+1/2}}{\cos \theta_j (\Delta \theta)^2} D_{j+1} \right\} = \Delta_{tt}^{prel} \tilde{d}_j\end{aligned} \quad (2.1.3.23)$$

After rearranging terms the equation becomes

$$A_j D_{j-1} + B_j D_j + C_j D_{j+1} = F_j \Delta_{tt}^{prel} \tilde{d}_j \quad (2.1.3.24)$$

where

$$\begin{aligned}A_j &= -\cos \theta_{j-1/2} \\ B_j &= C_2 \cos \theta_j + \left[\frac{2}{\Delta \lambda} \sin\left(\frac{l\Delta\lambda}{2}\right) \Delta \theta \right]^2 / \cos \theta_j + \cos \theta_{j-1/2} + \cos \theta_{j+1/2} \\ C_j &= -\cos \theta_{j+1/2} \\ F_j &= C_2 \cos \theta_j\end{aligned} \quad (2.1.3.25)$$

$$C_2 = \frac{a^2(\Delta\theta)^2}{(\Delta t)^2 c^2}$$

Equation (2.1.3.24) forms a tridiagonal system of equations for each wavenumber l , which is solved by Gaussian elimination assuming the boundary condition,

$$D = 0 \quad (2.1.3.26)$$

Finally, an inverse Fourier transform is performed in the λ -direction in order to obtain gridpoint values of the divergence.

The lateral boundary condition for the divergence is consistent with the linear time-interpolation of boundary values between two boundary data sets.

2.1.4 Horizontal diffusion

Two schemes for horizontal diffusion are included in the program libraries. Selection among the two schemes is done by NAMELIST parameters (see section 2.4).

An implicit 2., 4. or 6. order scheme has been implemented as an optional scheme. The 6. order scheme is used in the semi-Lagrangian time stepping algorithm. See section 2.1.8 for documentation.

A fourth order explicit linear diffusion scheme is implemented as default in combination with the eulerian time stepping. It is defined in the following way using the notation in section 2.1.1:

$$K_X = -K(\eta) \nabla^4 X^{n-1} \quad (2.1.4.1)$$

for $X = u, v, T$ and q

A modified scheme is used for temperature and specific humidity in order to avoid an unrealistic heating and moisturing in the vicinity of step mountain. A formula which performs the diffusion approximately along pressure surfaces is found by use of coordinate transformation between η and p .

$$\left(\frac{\partial T}{\partial x} \right)_p = \left(\frac{\partial T}{\partial x} \right)_\eta - \left(\frac{\partial T}{\partial p} \right) \left(\frac{\partial p}{\partial x} \right)_\eta = \left(\frac{\partial T}{\partial x} \right)_\eta - \left(p_s \frac{\partial p}{\partial p_s} \frac{\partial T}{\partial p} \right) \frac{\partial \ln p_s}{\partial x} \quad (2.1.4.2)$$

Equation (2.1.4.1) becomes

$$K_T = -K (\nabla^4 T^{n-1} - T_c \nabla^4 \ln p_s^{n-1}) \quad (2.1.4.3)$$

where

$$T_c = \left(p_s \frac{\partial p}{\partial p_s} \frac{\partial T}{\partial p} \right)_{ref} \quad (2.1.4.4)$$

A similar formula is used for the specific humidity

$$K_q = -K (\nabla^4 q^{n-1} - q_c \nabla^4 \ln p_s^{n-1}) \quad (2.1.4.5)$$

where q_c is computed from T_c and q^{n-1}

$$q_c = c \cdot T_c \cdot q^{n-1} \quad (2.1.4.6)$$

The tuning constant $c = 5.5 \cdot 10^{-2}$.

The level dependent reference values have been taken from Research Manual 2, ECMWF forecast model, adiabatic part.

$$p_s \frac{\partial p}{\partial p_s} \frac{\partial T}{\partial p} \Big|_{ref}]_k = \beta_k \alpha T_{rc} p_{rs}/p_{rk} \quad \text{for } T_{rc} > T_{rt} \\ = 0 \quad \text{for } T_{rc} \leq T_{rt} \quad (2.1.4.7)$$

where

$$T_{rc} = T_{rs} (p_{rk}/p_{rs})^\alpha \\ p_{rk} = A_k + B_k p_{rs} \\ p_{rs} = 101320 \text{ Pa} \\ T_{rs} = 288 \text{ K} \\ \alpha = 1./5.256 \\ T_{rt} = 216.5 \text{ K} \quad (2.1.4.8)$$

The fourth order differential operator is computed as $\nabla^4 X = \nabla^2(\nabla^2 X)$, where the finite difference form of the Laplacian is

$$\nabla^2 X = \frac{1}{h_x h_y} [\delta_x (\frac{h_y}{h_x} \delta_x X) + \delta_y (\frac{h_x}{h_y} \delta_y X)] \quad (2.1.4.9)$$

The explicit time-stepping for the diffusion equation becomes

$$X^{n+1} = X^{n-1} - 2\Delta t K \nabla^4 X^{n-1} \quad (2.1.4.10)$$

Numerical stability

In this section the response function $R_{k,l}$ for each wavenumber k and l in x - and y -direction is computed and the numerical stability is studied together with the formulas for e-folding time T_e as a function of time step Δt , grid distances Δx , Δy and diffusion coefficient K . Assuming $h_x = h_y$ and a wave solution

$$X(x,y,n) = \sum_k \sum_l X_{k,l}^n e^{i(kx+ly)} \quad (2.1.4.11)$$

the finite difference formula for the fourth order operator becomes

$$\nabla^4 X = \frac{16}{\alpha^4} \sum_k \sum_l [\frac{1}{\Delta x^2} \sin^2(\frac{k \Delta x}{2}) + \frac{1}{\Delta y^2} \sin^2(\frac{l \Delta y}{2})]^2 X_{k,l}^n e^{i(kx+ly)} \quad (2.1.4.12)$$

The response function is then

$$R_{k,l} = X_{k,l}^{n+1}/X_{k,l}^{n-1} = 1 - \frac{32\Delta t K}{a^4} \left[\frac{1}{\Delta x^2} \sin^2\left(\frac{k\Delta x}{2}\right) + \frac{1}{\Delta y^2} \sin^2\left(\frac{l\Delta y}{2}\right) \right]^2 \quad (2.1.4.13)$$

and for numerical stability $|R_{k,l}| \leq 1$ is required.

The e-folding time T_e is the time interval where the amplitude of a wave number k, l is damped to $1/e$. For practical purposes, let $\Delta x = \Delta y$ and $k = l$ in the following. In the leap-frog time stepping we require that

$$(R_{k,k})^{n/2} = 1/e \quad (2.1.4.14)$$

where $T_e = n\Delta t$. This leads to the following equations for T_e and K .

$$T_e = - \frac{2\Delta t}{\ln \left[1 - \frac{128\Delta t K}{a^4 \Delta x^4} \sin^4\left(\frac{k\Delta x}{2}\right) \right]} \quad (2.1.4.15)$$

$$K = \frac{a^4 \Delta x^4}{128\Delta t \sin^4\left(\frac{k\Delta x}{2}\right)} (1 - e^{-2\Delta t T_e}) \quad (2.1.4.16)$$

2.1.5 Boundary relaxation scheme

The interior variables X are adjusted towards prescribed boundary values X after the semi-implicit corrections are done.

We use the technique developed by Davies (1976) and Kallberg (1977):

$$X^{n+1} = (1 - \alpha_b)X_i^{n+1} + \alpha_b X_b^{n+1} \quad (2.1.5.1)$$

where X_b is linearly interpolated in time between boundary data sets in 12-hourly (or e.g. 6-hourly) intervals.

The default value of α_b is chosen as

$$\alpha_b = 1 - \tanh\left(\frac{2}{N-4}j\right) \quad (2.1.5.2)$$

where j is the number of gridpoints from the boundary point and N being the width of the boundary relaxation zone. Optionally, a cosine shape can be used, according to the following formula

$$\alpha_b = \frac{1}{2}\left[1 + \cos\left(\pi \frac{j}{N}\right)\right] \quad (2.1.5.3)$$

This option is used in the normal mode initialization and in the semi-Lagrangian time scheme.

Due to the staggering of the variables in the horizontal directions, separate relaxation functions are computed for the mass points and the velocity points, so that the weight is given according to the physical distance from the lateral boundaries. The physical boundary points with full weight given to the external fields is situated at the velocity points (u points at the western and eastern boundaries, v points at the southern and northern boundaries). Optionally, the same relaxation function can be applied to those staggered values of mass and velocity variables which are coded with the same horizontal index.

2.1.6 Statistics

The model can compute and print some diagnostic measures about mass, energy, surface pressure tendency and maximum wind speed every time step. The parameters about mass and energy are all computed as the area mean value of the vertically integrated quantity per unit horizontal area. E.g. the potential energy due to gravity of a column of air per unit horizontal area, P , is defined as

$$P = \int_0^{\infty} g z \varrho dz = \int_0^{p_s} \frac{\varphi}{g} dp = \frac{1}{g} (\varphi_s p_s + \int_0^{p_s} R_d T dp) \quad (2.1.6.1)$$

The corresponding internal energy, I , is written as

$$I = \int_0^{\infty} c_v T \varrho dz = \frac{1}{g} \int_0^{p_s} c_v T dp \quad (2.1.6.2)$$

The following energy measures are computed:

Total potential energy, $TPE = P + I$;

$$TPE = \frac{1}{g} (\varphi_s p_s + \int_0^{p_s} c_p T dp) \quad (2.1.6.3)$$

Kinetic energy, KE ;

$$KE = \int_0^{\infty} \frac{1}{2} v^2 \varrho dz = \frac{1}{g} \int_0^{p_s} \frac{1}{2} v^2 dp \quad (2.1.6.4)$$

Total energy, $TE = TPE + KE$

The following mass measures are computed:

Total mass, M ;

$$M = \int_0^{\infty} \varrho dz = \frac{1}{g} \int_0^{p_s} dp = \frac{p_s}{g} \quad (2.1.6.5)$$

Mass of water vapour, M_v

The mass of water vapour, m_v , is related to the total mass, m , by $m_v = qm$, where q is the specific humidity. Since $m = \varrho dx dy dz$, the integrated mass per unit horizontal area is

$$M_v = \int_0^{p_t} q \varrho dz = \frac{1}{g} \int_0^{p_t} q dp \quad (2.1.6.6)$$

Mass of liquid water, M_l

In a similar fashion as M_v , $m_l = q_l \varrho dx dy dz$, where q_l is the specific liquid water content. Then M_l is written as

$$M_l = \int_0^{p_t} q_l \varrho dz = \frac{1}{g} \int_0^{p_t} q_l dp \quad (2.1.6.7)$$

The corresponding area mean value \bar{F} of variable F is computed as

$$\bar{F} = \frac{1}{\iint h_x h_y dx dy} \iint F h_x h_y dx dy \quad (2.1.6.8)$$

and their respective units are

$$\begin{aligned} [TPE] &= [KE] = [TE] = Jm^{-2} \\ [M] &= [M_v] = [M_l] = kg m^{-2} \end{aligned}$$

An addition to mass and energy, the model computes the area mean value of the absolute surface pressure tendency

$$\left| \overline{\frac{\partial p_s}{\partial t}} \right| = \frac{1}{NLON \cdot NLAT} \sum_{j=1}^{NLAT} \sum_{i=1}^{NLON} \left| \frac{\partial p_s}{\partial t} \right|_{i,j} \quad (2.1.6.9)$$

in unit hPa/3hours and the magnitude and location of the maximum horizontal wind speed

$$|\vec{v}|_{max} = \max [(u_{i,j,k}^2 + v_{i,j,k}^2)^{1/2}] ; i=1,NLON, j=1,NLAT, k=1,NLEV$$

Code

The statistics is computed in routine STATIS and printed in routine PRSTAT. The output is written with the following names.

<u>quantity</u>	<u>name in program</u>
-----------------	------------------------

TPE	STPE
-------	------

KE	STKE
------	------

\overline{TE}	STTE
\overline{M}	STPS
$\overline{M_v}$	STQ
$\overline{M_l}$	STS

2.1.7 Extra scalars

In addition to the present prognostic variables (u , v , T , p_s , q and q_f) the forecast model has been prepared for any number of additional scalar variables, in the following called extra scalars. These scalars can be turbulent kinetic energy and dissipation, which are prognostic variables in a higher order vertical diffusion scheme, and air pollutants. The extra scalars are included in the interface to the physical parameterization package. In the post-processing they can be written to the output files, both by extraction of values at model levels and after vertical interpolation to pressure levels. In an assimilation experiment, the forecasted values of the extra scalars will be written to the model history file, which contains the prognostic variables, and read from the corresponding data file at the initial time of the next assimilation cycle.

In the eulerian time scheme, the prognostic equation for an extra scalar, S , is

$$\frac{\partial S}{\partial t} = - \frac{u}{h_x} \frac{\partial S}{\partial x} - \frac{v}{h_y} \frac{\partial S}{\partial y} - \dot{\eta} \frac{\partial S}{\partial \eta} + P_S + K_S \quad (2.1.7.1)$$

This equation is extrapolated in time by a second order three-time-level leap-frog scheme, in the same way as for the specific humidity, i.e.

$$\begin{aligned} \overline{\delta_t S}_k' &= -\frac{1}{\Delta p_k} \left\{ \frac{1}{h_x h_y} [\overline{U h_y \delta_x S^x} + \overline{V h_x \delta_y S^y}]_k + \overline{[(\dot{\eta} \frac{\partial p}{\partial \eta}) \Delta_\eta S]}_k \right\} \\ &\quad + P_S + K_S \end{aligned} \quad (2.1.7.2)$$

The advection terms are evaluated at time step n and the horizontal diffusion term is evaluated at time $n-1$ or $n+1$ depending on the choice of diffusion scheme (see section 2.1.4). At present there is no physical forcing included, i.e. $P_S = 0$. After time extrapolation, the values are relaxed towards boundary values of S as described in section 2.1.5. After boundary relaxation, the Asselin timefilter is applied to the values at time n , using the same coefficient as for the other prognostic variables.

In the two-time-level semi-Lagrangian time scheme the corresponding prognostic equation is

$$\frac{dS}{dt} = P_S + K_S \quad (2.1.7.3)$$

This equation is discretized in the same manner as for specific humidity, i.e. the physical forcing

and the implicit horizontal diffusion terms are evaluated at the arrival point after adiabatic time extrapolation. In case of explicit diffusion, evaluated at time n, this term is averaged between the departure and arrival points, i.e.

$$S_k^{n+1} = (S^n + \frac{\Delta t}{2} K_S^n)_{*3,k} + \frac{\Delta t}{2} (K_S^n)_k \quad (2.1.7.4)$$

After time extrapolation, boundary relaxation of S is performed as described in section 2.1.5.

2.1.8 The semi-Lagrangian time scheme

This scheme is documented in HIRLAM Technical Report No. 17: "The HIRLAM two time level, three dimensional semi-Lagrangian, semi-implicit, limited area, grid point model of the primitive equations." by Adian McDonald.

2.2 Physical parameterization

The purpose of the physical parameterization is to determine the terms P_x in the hydrodynamical equations (2.1.1.4 to 2.1.1.11) governing the state of the atmosphere. The physical processes contributing to P_x are basically of sub-grid scale. They are parameterized in terms of model variables describing the atmospheric state on resolvable scales.

In section 2.2.1-2.2.6 the various physical parameterizations (i.e. radiation, vertical diffusion, stratiform condensation, convection and surface processes) are described, partly from a theoretical point of view.

The computational strategy (subroutines and order of calculations) is described in section 2.3.2.

2.2.1 Radiation

The main purpose of a radiation scheme used in an atmospheric forecast model is to produce a net radiative flux profile as a result of infrared (longwave) radiation and solar (shortwave) radiation. The corresponding temperature tendency is proportional to the net flux divergence according to (1)

$$\frac{\partial T}{\partial t} = -\frac{g}{c_p} \times \frac{\partial F_{net}}{\partial p} \quad (1)$$

g is the acceleration of gravity and c_p the specific heat capacity at constant pressure. Both longwave and shortwave radiation are computed as a sum of a clear air contribution and a cloudy contribution.

a. Shortwave radiation

The temperature tendency $\frac{\partial T}{\partial t_s}$ as a result of shortwave radiation is formulated according to (2)

$$\frac{\partial T}{\partial t_s} = \frac{\partial T}{\partial t_{sa}} (1 - C_M) + \frac{\partial T}{\partial t_{sc}} C_M \quad (2)$$

The first term represents absorption in clear air, and the second term absorption in cloudy air. As an approximation we consider the cloudy fraction C_M to be equal to the maximum fractional cloud cover of all levels in a vertical column. The computations assume that the maximum cloud cover applies to all levels below the uppermost cloud layer. Hence, the cloud water content valid for clouds in a given model layer is scaled by a factor of C/C_M . A cloud water content $q'_c = q_c/C_M$ inside the cloud will then guarantee consistency between vertically integrated cloud water from the grid box model variable q_c and the vertical integral of cloud water used for the cloudy part of the shortwave computation. Above the uppermost cloud level the shortwave heating equals the clear air contribution.

In order to determine the effects of solar radiation at a given time it is necessary to specify the solar constant S and the solar zenith angle θ . These are computed according to (3) and (4) following Paltridge and Platt (1976).

$$S = S_0 [1 + b_1 \cos(2\pi r) + b_2 \sin(2\pi r) + b_3 \cos(4\pi r)] \quad (3)$$

$$r = j/365 \quad S_0 = 1365 \text{ J/(m}^2\text{s)}$$

A small term involving $\sin(4\pi r)$ has been neglected. - j is the running date from 1 January. The values of the constants b_1 - b_3 are given in 2.2.1c

$$\cos(\theta) = \cos(\phi) \cos(d(j)) \cos(t_0 + \lambda) + \sin(\phi) \sin(d(j)) \quad (4)$$

ϕ is the geographical latitude and $d(j)$ is the declination of the sun as a function of the current day. The geographical longitude is λ , and the local hour angle of the sun at Greenwich is

t_0 .

The heating rate of the clear air is computed according to (5), (6), (7) and (8) following Savijärvi (1990).

$$\frac{\partial T}{\partial t_{sa}} = S \times \frac{q}{c_p} \times \frac{p}{p_0} [Y(u_s) + b_4 \alpha \cos\theta - Y(u_*)] + b_5 \cos\theta^{0.3} \quad (5)$$

$$Y(u_s) = \begin{cases} b_6 \times u_s^{-0.81}, & u_s \geq 0.05 \text{ cm} \\ b_7 \times u_s^{-0.63}, & u_s < 0.05 \text{ cm} \end{cases} \quad (6)$$

α is the shortwave albedo of the ground. The constants b_4 - b_7 are defined in the 2.2.1c. - u_s is the linearly pressure scaled vertical water vapour (in cm) divided by $\cos\theta$.

$$u_s = \frac{1}{\cos\theta} u(0, p) \quad u(p_1, p_2) = \frac{1}{g} \times \int_{p_1}^{p_2} q \times \frac{p}{p_0} dp \quad (7)$$

$p_0 = 1013$ hPa is a reference pressure. Similarly, the path length for isotropically reflected beams is

$$u_* = \frac{1}{\cos\theta} u(0, p_s) + b_4 u(p, p_s) \quad (8)$$

The two terms in the bracket of (5) represent absorption by water vapour. The second of these describes the heating from reflected beams. Good agreement is obtained with the broadband scheme of Chou (1986) based on comparisons with line-by-line calculations. The last term in (5) is a parameterization of absorption by CO_2 , O_2 and O_3 using the standard absorption curves of Sasamori et al. (1972).

The cloud absorption is given by (9)

$$\frac{\partial T}{\partial t_{sc}} = \hat{T}(p_Z, p) \frac{\partial T}{\partial t_{sa}} + \frac{g}{c_p} F_{sZ} \frac{\partial}{\partial p} \hat{A}(p_Z, p) \quad (9)$$

F_{sZ} is the solar flux density at the top of the uppermost cloud layer. The absorption is determined as the clear air absorption modified by the transmittance factor \hat{T} for the cloudy atmosphere above (first term), plus a major contribution from cloud drop absorption and increased path lengths due to scattering (second term). F_{sZ} is parameterized according to (10) as given for clear sky conditions at the ground ($p=p_s$) by Savijärvi (1990).

$$F_{sZ} = S \times \cos\theta [1 - 0.024 \cos\theta^{-0.5} - b_8 0.11 u_s^{0.25} - b_9 \frac{p}{p_0} (0.28/(1+6.43 \cos\theta) - 0.07 \alpha_*)] \quad (10)$$

The first term in (10) depending on θ concerns the stratospheric absorption due to ozone. A major contribution to the extinction of solar radiation comes from tropospheric absorption due to water vapour, CO_2 and O_2 . This is parameterized according to the second term in (10) while the last term involving two contributions, describes the effect of scattering. The first

contribution arises from scattering of the incoming solar beam while the second one is a compensating effect due to reflected radiation (albedo α_*) , which is back-scattered from the atmosphere above. The coefficients b_8 and b_9 which are larger than 1 (see 2.2.1c) represent a crude inclusion of effects due to aerosol absorption and scattering , respectively. \hat{A} and \hat{T} are absorptance and transmittance respectively, from the top of the uppermost cloud layer to a level below. Simple formulas, (11) and (12), have been fitted to cloud absorption and transmission computations obtained with more detailed schemes (Slingo et al. 1982, Stephens 1978, Liou and Wittmann 1979) .

$$\hat{A} = b_{10} \times (b_{11} + \cos\theta) \times \ln(1 + b_{12} \times M_t) \quad (11)$$

$$\hat{T} = \hat{T}_1 / (\hat{T}_1 + M_t) \quad \hat{T}_1 = b_{13} \times (b_{14} + \cos\theta) \quad (12)$$

M_t is the vertically integrated cloud water from the top of the uppermost cloud layer to a level below. b_{10} - b_{14} are constants (see 2.2.1c)

Finally, the net solar flux density at the ground F_{s0} is computed according to (13) as a product of $(1 - \alpha)$ and a weighted sum of a clear air contribution F_{s0a} determined from a formula similar to (10) and a cloudy contribution F_{s0c} according to (14)

$$F_{s0} = (1 - \alpha) [F_{s0a} (1 - C_M) + F_{s0c} C_M] \quad (13)$$

$$F_{s0c} = F_{sZ} \times \frac{\hat{T}(p_Z, p_s)}{1 - \alpha \times (1 - \hat{T}(p_Z, p_s)) \times b_{15}} \quad (14)$$

The denominator in (14) takes into account multiple reflections between ground and cloud. The factor b_{15} accounts for absorption in reflected beams. This heating term, usually small, is included as a uniformly distributed heat source below the cloud top.

b. Longwave radiation

The longwave scheme is based on an emissivity function $E(p_{i-1/2}, p_{j+1/2})$ defined according to (15). The two pressures $p_{i-1/2}$ and $p_{j+1/2}$ apply to model layer intersections (half-levels) .

$$E(p_{i-1/2}, p_{j+1/2}) = a_1 + a_2 X - a_3 X^2 - a_4 X^3, \quad (15)$$

$$X = a_5 \ln \left(\sum_{k=i}^j q_k \times \frac{p_k}{p_{00}} \times \frac{\Delta p_k}{10g} \right)$$

The water vapour path has been linearly pressure scaled . The values of the constants a_1 to a_5 are given in 2.2.1c. The clear air cooling rate based on (15) takes into account cooling due to the water vapour line spectrum. It may be shown (Savijärvi 1990) that the cooling rate may be written as a sum of three terms. One term expressing local "cooling to space" is

usually significantly larger than the other terms in the free atmosphere. A second term expressing interaction between the surface and the atmosphere is of significance close to the ground. A third term which determines the exchange of radiation with all other layers is usually small and is neglected in the present version of the scheme. Also a complete interaction between levels is avoided for the cloudy part of the computations. However the significant effect of radiative interaction between clouds is retained as described below. These simplifications may be defended only with reference to computational efficiency which is considerably increased since the number of calculations become proportional to the number of model levels.

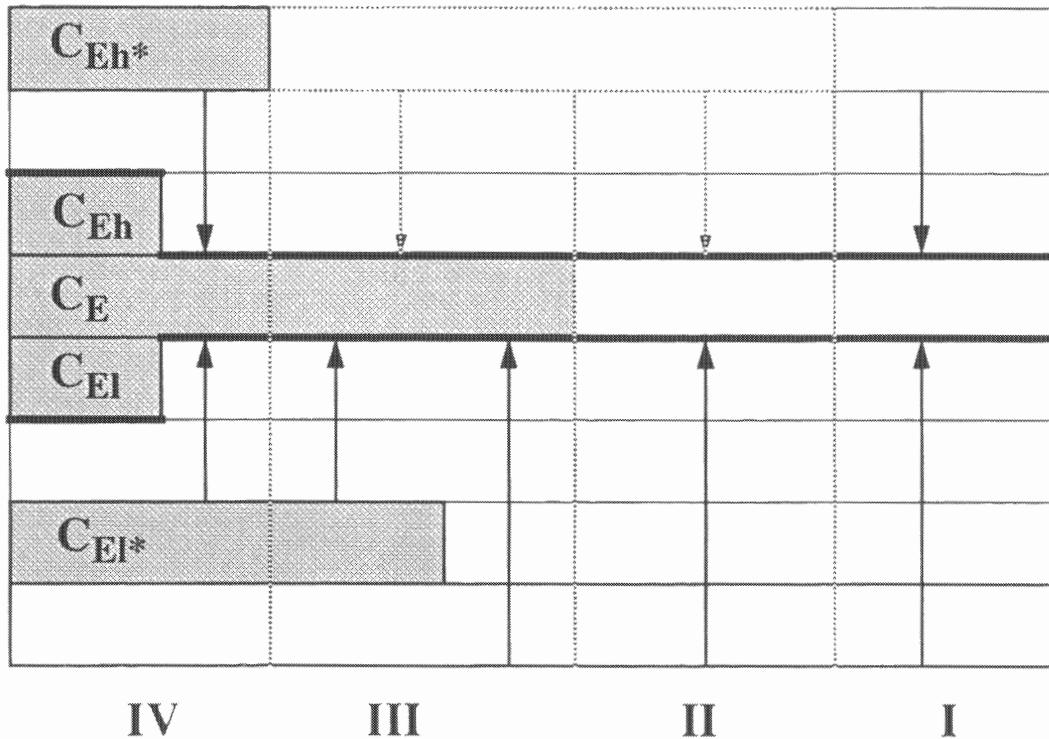


Fig.1 Computational scheme for infrared computations

Formally, the computation of the radiative temperature tendency due to thermal radiation is split up into four parts according to the scheme shown in Fig.1

$$\frac{\partial T}{\partial t_i} = \frac{\partial T}{\partial t_{i1a}} + \frac{\partial T}{\partial t_{i2a}} + \frac{\partial T}{\partial t_{i1c}} + \frac{\partial T}{\partial t_{i2c}} \quad (16)$$

Each of the terms in (16) is a weighted contribution to the total tendency. The scheme applies to a maximum cloud overlap assumption. The first term is the heating rate contribution (I) from the clear air part of the grid box with no clouds above while the second term (II) is the clear air contribution with clouds above. Similarly, the third term (III) is the cloudy contribution to the heating rate without clouds above and the fourth term (IV) a contribution with clouds above. The cloud covers (shaded) are effective cloud covers. The customary assumption is applied, namely that "grey" clouds may be introduced by reducing the normal grid box fractional cloud cover (C) with the cloud emissivity according to (17a)

$$C_E = C (1 - \exp(-f_1 M)) \quad (17a)$$

$$f_1 = \text{Max} \{ 0.05 + 0.20 (\eta - \eta_1) / (1 - \eta_1), 0.05 \}, \eta_1 = 0.25 \quad (17b)$$

As expressed in (17b), the parameter f_1 depends on cloud type. It varies linearly with the vertical η -coordinate becoming 0.05 (m^2/g) for high clouds and 0.25 for low clouds close to the ground. M is the vertically integrated cloud water (g/m^2) in the cloud. The radiative impact of cloud layers above are taken into account by computing radiation from a maximum effective cloud cover C_{Eh^*} . In order to treat high vertical resolutions in a realistic way this maximum effective cloud cover is determined in the following way: For any level above the layer in consideration an effective cloud cover is assigned by (17a) using local values of cloud cover and f_1 while M is determined as the vertical integral of cloud water up to the uppermost atmospheric model layer. The maximum of these effective cloud covers is C_{Eh^*} which is assigned to the level where the maximum occurs. A similar computation is carried out for the maximum effective cloud cover C_{El^*} for clouds below the layer considered. Also the effect of the cloud covers C_{Eh} and C_{El} of the neighbouring layers above and below, respectively, is taken into account by assuming zero net flux contribution at the intersection of overlapping clouds. The individual terms in (16) are given in (18)-(21)

$$\frac{\partial T}{\partial t_{i1a}} = W_1 (R_1 + R_2 + R_3) \quad (18)$$

$$\frac{\partial T}{\partial t_{i2a}} = W_2 f_2 (R_1 + R_3) + (W_2 - W_3) R_2 \quad (19)$$

$$\frac{\partial T}{\partial t_{i1c}} = W_4 R_4 + W_5 R_5 + W_6 R_6 \quad (20)$$

$$\frac{\partial T}{\partial t_{i2c}} = W_7 R_7 + W_8 R_5 + W_9 R_6 \quad (21)$$

W_1-W_9 are weight functions related to the cloud overlap assumption. (See 2.2.1 c) R_1-R_7 are given in finite difference form. (Eq. 22-30).

$$R_1 = B(T_k) [E(0, p_{k-1/2}) - E(0, p_{k+1/2})] g / (c_p \Delta p_k) \quad (22)$$

$$R_2 = [B(T_{E0}) - B(T_k)] [E(p_{k-1/2}, p_s) - E(p_{k+1/2}, p_s)] g / (c_p \Delta p_k) \quad (23)$$

$$R_3 = -a_6 q_k^3 - a_7 \quad (24)$$

$$R_4 = [-B(T_{k-1/2}) + B(T_{k-1}) E(0, p_{k-1/2}) + a_9 - f_3 + a_{10} q_{k-1}] g / (c_p \Delta p_k) \quad (25)$$

$$R_5 = [B(T_{l^*}) - B(T_{k+1/2})] [f_4 + G_{k+1} (1-f_4)] g / (c_p \Delta p_k) \quad (26)$$

$$R_6 = [B(T_{E0}) - B(T_{k+1/2})] [f_5 + G_{k+1} (1-f_5)] g / (c_p \Delta p_k) \quad (27)$$

$$R_7 = [B(T_{h^*}) - B(T_{k-1/2})] [f_6 + G_{k-1} (1-f_6)] g / (c_p \Delta p_k) \quad (28)$$

$$G_j = 1 / (1 + a_{12} (\Delta E) / (\Delta p)_j), \quad 1 \leq j \leq N \quad (29)$$

$$f_2 = \text{Min}\{ (p_{k-1/2} - p_{h^*}) / a_8, 1 \} \quad (30a)$$

$$f_3 = a_{11} (1 - \eta) / (1 - \eta_0), \quad \eta_0 = 0.05 \quad (30b)$$

$$f_4 = \text{Min}\{ a_{13} / (p_l^* - p_{k+1/2}), 1 \} \quad (30c)$$

$$f_5 = \text{Min}\{ a_{13} / (p_s - p_{k+1/2}), 1 \} \quad (30d)$$

$$f_6 = \text{Min}\{ a_{13} / (p_{k-1/2} - p_{h^*}), 1 \} \quad (30e)$$

$B(T_j)$ is the Planck black body radiation at temperature T_j . $-R_1$ is the "cooling to space" term and R_2 accounts for the radiative effect of ground radiating with an effective temperature T_{E0} . The first term in R_3 represents crudely the effect of cooling due to water vapour continuum which is of major importance only in a moist and warm atmosphere (Savijärvi 1990). The cooling effect of other gases like CO_2 and O_3 seems to be rather small and fairly constant in the troposphere (Liou 1980, Paltridge and Platt 1976). Hence a constant cooling term a_7 is introduced to represent this.

The factor f_2 in (19) represents reduced cooling below clouds. Paltridge and Platt (1976) note that the assumption of zero cooling below 'black' clouds might be compatible with a "cooling to space" approximation. However an assumption like that seems to be somewhat too drastic (Sass et al. 1994). In the present scheme the cooling rate is simply assumed to vary linearly with a pressure increment below cloud, from zero to the clear air value through a certain depth $a_8 = 4 \times 10^4 \text{ Pa}$. The last term in (19) proportional to $W_2 - W_3$ represents the effect of radiation from the ground attenuated by overlapping low level clouds.

In (20) the first term estimates the net radiation balance at cloud top. The downward radiation due to gases other than water vapour including aerosols is described by $a_9 - f_3$ taking into account a flux correction consistent with the integrated cooling in clear air due to these constituents. The term involving a_{10} parameterizes an additional flux due to water vapour continuum.

The additional terms in (20) and (21) involve a determination of effective radiation temperatures describing radiative interaction either between clouds or between ground and cloud. The first bracket in the formulations of $R_5 - R_7$ involving differences between related Planck-functions is multiplied by a factor between 0 and 1 according to the expression in the second bracket. An empirical function G_j expressing sensitivity to the emissivity slope in clear air next to the cloud has been introduced (29). From a qualitative point of view it is obvious that the level of an effective radiation temperature for upward radiation comes closer to the ground as the humidity below cloud base decreases. In the extreme (unrealistic) situation of no absorption in the clear atmosphere the formulation automatically guarantees transmission of radiation between ground and cloud without attenuation.

The downward surface flux from the clear atmosphere is obtained from (31). In addition to the summation of the contributions from all levels expressed from the Planck function and the emissivity function (first term) two additional terms are added corresponding to the contributions already mentioned in relation to the cloud top radiation budget.

$$F_{i0a} = \sum_{k=1}^N B(T_k) (E(p_{k-1/2}, p_s) - E(p_{k+1/2}, p_s)) + a_9 + a_{10} q_N \quad (31)$$

Index N refers to the lowest model layer. The total downward flux from the atmosphere is obtained by adding a cloudy contribution F_{i0c} determined as radiation transmitted to the ground from the effective cloud cover according to the empirical expression in (32). The term in the brackets is a fractional transmission estimated from the clear air contribution F_{i0a} and a low tropospheric effective temperature T_{Ef} .

$$\begin{aligned} F_{i0c} &= B(T_{h*}) C_{Eh*}(N) [1 - F_{i0a} / (\sigma T_{Ef}^4)] \\ T_{Ef} &= T_N - a_{14} \left(\frac{\partial T}{\partial p} \right)_N \end{aligned} \quad (32)$$

The net radiation at the ground due to longwave radiation follows from (32)

$$F_{i0} = \epsilon_0 (F_{i0a} + F_{i0c} - \sigma T_{E0}^4) \quad (33)$$

σ is Stefan Boltzmann's constant and ϵ_0 is the emissivity of the surface.

c. Weight factors and constants.

The weight factors (W_1-W_9) related to the computational scheme of Fig.1 are given below:

$$W_1 = 1 - \text{Max}\{C_E, C_{Eh*}\}$$

$$W_2 = \text{Max}\{C_{Eh*} - C_E, 0\}$$

$$W_3 = \text{Min}\{\text{Max}\{C_{EI*} - C_E, 0\}, W_2\}$$

$$W_4 = \text{Max}\{C_E - C_{Eh*}, 0\}$$

$$W_5 = \text{Max}\{\text{Min}\{C_{EI*}, C_E\} - C_{Eh*}, 0\} - \text{Max}\{\text{Min}\{C_{EI}, C_E\} - C_{Eh*}, 0\}$$

$$W_6 = W_4 - \text{Max}\{\text{Min}\{C_{EI*}, C_E\} - C_{Eh*}, 0\}$$

$$W_7 = \text{Max}\{\text{Min}\{C_{Eh*}, C_E\} - C_{Eh}, 0\}$$

$$W_8 = \text{Max}\{\text{Min}\{\text{Min}\{C_{EI*}, C_E\}, C_{Eh*}\} - C_{EI}, 0\}$$

$$W_9 = \text{Min}\{C_{Eh*}, C_E\} - \text{Min}\{\text{Min}\{C_{EI*}, C_E\}, C_{Eh*}\}$$

Values of constants applied in the H2 radiation scheme.

$a_1=0.60$	$a_2=0.17$	$a_3=0.0082$	$a_4=0.0045$
$a_5=0.4343$	$a_6=11.5 \text{ (K/s)}$	$a_7=2.3 \cdot 10^{-6} \text{ (K/s)}$	$a_8=4 \cdot 10^4 \text{ (Pa)}$
$a_9=35 \text{ J/(m}^2\text{s)}$	$a_{10}=3000 \text{ J/(m}^2\text{s)}$	$a_{11}=24 \text{ J/(m}^2\text{s)}$	$a_{12}=2 \cdot 10^4 \text{ (Pa)}$
$a_{13}=2.5 \cdot 10^3 \text{ (Pa)}$	$a_{14}=a_{13}$		
$b_1=0.03422$	$b_2=0.0013$	$b_3=0.000719$	$b_4=1.67$
$b_5=1.7 \cdot 10^{-6} \text{ (K/s)}$	$b_6=0.03$	$b_7=0.05$	$b_8=1.20$
$b_9=1.25$	$b_{10}=0.013$	$b_{11}=1.4$	$b_{12}=0.10 \text{ (m}^2\text{/g)}$
$b_{13}=40$	$b_{14}=0.5$	$b_{15}=0.80$	

Stefan-Boltzmann's constant $\sigma = 5.67 \cdot 10^{-8} \text{ J/(m}^2\text{sK}^4)$

REFERENCES

- Chou, M.-D., 1986: Atmospheric solar heating in the water vapour bands. *J.Appl. Meteor.*,25,1532-1542.
- Liou, K-N, 1980: An Introduction to Atmospheric Radiation. - Academic, 392 pp.
- Liou, K-N and Wittmann, G.D., 1979: Parameterization of the radiative processes of clouds. *J.Atmos. Sci.*, 36, 1261-1273
- Paltridge ,P.W. ,and C.M.R. Platt, 1976: Radiative processes in Meteorology and Climatology. Elsevier Scientific Publishing Company, 318 pp
- Sasamori, T., J. London and D. Hoyt, 1972: Radiative budget of the Southern Hemisphere. *Meteor. Monogr.*, 35, 9-23.
- Sass, B.H. , Rontu, L.R. and Räisänen, P. ,1994: HIRLAM2 Radiation Scheme: Documentation and tests. HIRLAM Technical report no.16 [Available from SMHI]
- Stephens, G.L., 1978, I and II: Radiative properties of extended water clouds Part I. *J. Atmos. Sci.*, 35, 2111-2132

2.2.2 Vertical diffusion

The vertical diffusion is a parameterization of the vertical fluxes of momentum, sensible heat and moisture, caused by turbulence (sub-grid scale motions)

2.2.2 A Theory and general procedure

The calculation of surface fluxes are based on a drag coefficient formulation, using the Monin-Obukhov similarity theory for the atmospheric surface layer. The calculation of fluxes at model levels in the atmosphere is based on an exchange coefficient formulation, using a consistent extension of the similarity theory to layers above the surface layer. The fluxes are zero at the top of the model atmosphere.

The drag coefficients are functions of the static stability and a roughness length, the exchange coefficients functions of static stability and a mixing length.

It is not possible to derive neither the drag coefficient nor the exchange coefficient formula from similarity theory. Empirically determined formulae for the drag coefficients depending on z/L (z : height above surface and L : the Monin-Obukhov stability length) and z/z_0 (z_0 : roughness length) are used in some numerical weather prediction models. However they have the disadvantage that the surface fluxes have to be calculated by a computational expensive iterative procedure.

Here analytic formulas depending on the Richardson number (which can be calculated directly from model variables) and the roughness length/mixing length proposed by Louis et al. 1981 are used. They have the advantage that surface fluxes can be calculated directly. However, in unstable cases over a rough surface they diverge from the empirically derived drag coefficients (N Woetmann Nielsen 1987)

Dry static energy $s = c_p T + \varphi$ is diffused vertically instead of potential temperature. The contribution to s from the vertical diffusion is then converted into a contribution to T

Equations

The vertical diffusion equations have the form

$$\frac{\partial \varphi}{\partial t}_{vdiff} = \frac{1}{\rho} \frac{\partial}{\partial z} (\rho K_\varphi \frac{\partial \varphi}{\partial z}) \quad \varphi = u, v, q, s \quad (2.2.2.1)$$

$$J_\psi = \rho K_\psi \frac{\partial \psi}{\partial z} \quad (2.2.2.2)$$

is the vertical turbulent flux of ψ . The boundary conditions for J are

$$\begin{aligned} J_{\psi} &= 0 \quad \text{for } p=0 \\ J_{\psi} &= \rho C_{\psi_s} |\vec{v}(z)| (\Psi(z) - \Psi_s) \quad \text{for } p=p_s \end{aligned} \tag{2.2.2.3}$$

Here C_{ψ_s} is a drag coefficient. Its functional form is given in (2.2.2.6). The relation between $(\partial s/\partial t)_{vdif}$ and $(\partial T/\partial t)_{vdif}$ is

$$(\frac{\partial s}{\partial t})_{vdif} = c_p (\frac{\partial T}{\partial t})_{vdif} + T (\frac{\partial c_p}{\partial t})_{vdif} + (\frac{\partial \Phi_v}{\partial t})_{vdif} \tag{2.2.2.4}$$

In (2.2.2.4), the last term involving Φ_v is currently neglected. The drag coefficients C_{ψ_s} can be written as

$$C_{\psi_s} = [\frac{k}{\ln(z/z_o)}]^2 f_{\psi_s}(Ri, z/z_o) \tag{2.2.2.6}$$

where the analytic expressions (2.2.2.7) for f_{ψ_s} proposed by Louis et al. 1981 are used.

$$\left. \begin{aligned} f_{\psi_s} &= 1 - \frac{a_{\psi} \cdot b \cdot Ri}{1 + 3 \cdot b \cdot c \cdot [k/\ln(z/z_0)]^2 \cdot \sqrt{(1+z/z_0) - Ri}} && \text{for } Ri \leq 0 \\ f_{ms} &= \frac{1}{1 + 2 \cdot b \cdot Ri / \sqrt{1 + d \cdot Ri}} \\ f_{ss} &= f_{qs} = \frac{1}{1 + 3 \cdot b \cdot Ri \cdot \sqrt{1 + d \cdot Ri}} \end{aligned} \right\} \quad \text{for } Ri > 0 \tag{2.2.2.7}$$

The constants used at present are:

$$a_m = 2, \quad a_s = a_q = 3$$

$$b = c = d = 5$$

The Richardson number Ri is defined by

$$Ri = \frac{g \cdot \partial S_v / \partial z}{c_p \cdot T_v \cdot |\partial \vec{v} / \partial z|^2} \approx \frac{\Delta \Phi_v \cdot (c_p \cdot \Delta T_v + \Delta \Phi_v)}{c_p \cdot T_v \cdot |\Delta \vec{v}|^2} \tag{2.2.2.8}$$

and S_v is the virtual static energy; $S_v = c_p T_v + \Phi_v$. Here T_v is the virtual temperature defined

by

$$T_v = T(1 + (1/\epsilon - 1)q) \quad (2.2.2.9)$$

and φ_v the geopotential obtained by vertical integration of $\Delta \varphi_v = g\Delta z = -R_d T_v \Delta \ln p$. The drag coefficients K_ψ in (2.2.2.2) are given by

$$K_\psi = (l_\psi)^2 \cdot \left| \frac{\partial \vec{v}}{\partial z} \right| \cdot f_\psi(Ri, \frac{\Delta z}{z}) \quad (2.2.2.10)$$

f_ψ is identical with f_{ψ_s} except that

$$[k/\ln(z/z_o)]^2 \cdot [(z/z_o)^{1/3} - 1]^{3/2} \approx [k/\ln(z/z_o)]^2 \cdot \sqrt{z/z_0 + 1}$$

in (2.2.2.7a) is replaced by its equivalent

$$(l_\psi/\Delta z)^2 \cdot \sqrt{\Delta z/z} \cdot \{[(z + \Delta z)/z]^{1/3} - 1\}^{3/2} \quad (2.2.2.11)$$

In (2.2.2.10) and (2.2.2.11) l_ψ is a mixing length. The formulation of Blackadar (1962) is modified according to (2.2.2.12a) and (2.2.2.12b) by defining a height dependency of the mixing length λ above a layer of scale height H .

$$l_\psi = \frac{kz}{1 + kz/\lambda(z)} \quad (2.2.2.12a)$$

$$\lambda(z) = \begin{cases} \lambda_B & \text{for } z < H \\ \lambda_o + (\lambda_B - \lambda_o) \exp(-\frac{(z-H)}{H}) & \text{for } z \geq H \end{cases} \quad (2.2.2.12b)$$

where λ_o is an asymptotic mixing length representative of conditions in the upper troposphere. Currently $\lambda_o = 30$ m, $H = 1000$ m. Values of λ_B between 100 m and 300 m have been applied successfully in operational context.

Shallow convection

A parameterization of shallow convection is introduced in the vertical diffusion scheme. It is introduced to compensate partly for the absence of liquid water in the model and partly to compensate for the local nature of the simulation of turbulence in the unstable PBL (the fluxes being proportional to the local gradients of model variables, which is known to be unrealistic in the unstable PBL) leading to an underestimation of the magnitude of the turbulent fluxes at the top of the PBL.

Consequently, without shallow convection parameterization the model tends to create a too

shallow and too moist PBL particularly over sea areas with large scale subsidence.

The shallow convection is parameterized by introducing a modified Richardson number Ri_* (Geleyn, 1986) defined by

$$Ri_* = Ri + \frac{g \cdot L \cdot \min[0, \frac{\partial}{\partial z}(q - q_{sat}^*)]}{c_p \cdot T_v \cdot |\partial \vec{v}/\partial z|^2} \quad (2.2.2.13)$$

whenever specific humidity q exceeds a threshold value q_c assumed to be

$$q_c = A' \cdot q_{sat}^*$$

where q_{sat}^* is the saturation specific humidity applied by the condensation scheme. Currently, $A' = 0.95$. The stability functions f_s and f_q in (2.2.2.7) are made dependent on both Ri and Ri_* by introducing (Woetmann Nielsen and Hansen Sass, 1987) the new stability functions \tilde{f}_ψ defined by

$$\begin{aligned} (\tilde{f}_m)^2 &= f_m(Ri) \\ \tilde{f}_q \cdot \tilde{f}_m &= f_q(Ri, Ri_*) \\ \tilde{f}_s \cdot \tilde{f}_m &= f_s(Ri, Ri_*) \end{aligned} \quad (2.2.2.14)$$

It is assumed that

$$\tilde{f}_s(Ri_*) = \tilde{f}_q(Ri_*) = \sqrt{f_s(Ri_*)} \quad (2.2.2.15)$$

By this type of shallow convection parameterization the momentum flux is not affected.

Time stepping

In the diffusion equation (2.2.2.1) an implicit time stepping is used.

$$\frac{\Psi^*(t+\Delta t) - \Psi(t-\Delta t)}{2\Delta t} = \frac{1}{\rho} \frac{\Delta}{\Delta z} \{ \rho \cdot K_\psi [\Psi(t-\Delta t)] \cdot \frac{\Delta}{\Delta z} [\Psi(t+\Delta t)] \} \quad (2.2.2.16)$$

Where $\Psi(t+\Delta t) = 1.5 \cdot \Psi^*(t+\Delta t) - 0.5 \cdot \Psi(t-\Delta t)$ and $\Psi^*(t+\Delta t)$ the updated value of $\Psi(t-\Delta t)$ due to vertical diffusion.

Solution of the diffusion equations

The diffusion equations (2.2.2.1) can be written in the form

$$\begin{aligned}\psi_k^*(\Delta t) - \psi_k(-\Delta t) &= 1.5 \cdot [A_k \cdot (\psi_{k+1}^*(\Delta t) - \psi_k^*(\Delta t)) + C_k \cdot (\psi_k^*(\Delta t) - \psi_{k-1}^*(\Delta t))] \\ &\quad - 0.5 \cdot [A_k \cdot (\psi_{k+1}(-\Delta t) - \psi_k(-\Delta t)) + C_k \cdot (\psi_k(-\Delta t) - \psi_{k-1}(-\Delta t))]\end{aligned}\quad (2.2.2.17)$$

where A_k, C_k are diffusion coefficients defined in section 2.2.2 B. For convenience the time t has been omitted in $\psi_k(\Delta t)$ and $-\Delta t$ should be understood as $t+\Delta t$ and $t-\Delta t$ respectively). By rearrangement in (17) we obtain the following set of linear equations:

$$\psi_k^*(\Delta t) = E_k(-\Delta t) \psi_{k+1}^*(\Delta t) + F_{\psi_k}(-\Delta t) \quad \text{for } k=1, NLEV \quad (2.2.2.18)$$

where

$$\begin{aligned}E_k(-\Delta t) &= \frac{A_k}{2/3 + A_k + C_k \cdot (1 - E_{k-1})} \\ F_{\psi_k}(-\Delta t) &= \frac{C_k \cdot F_{\psi_{k-1}} + 1/3 \cdot [\Psi_k \cdot (2 + A_k + C_k) - (A_k \cdot \Psi_{k+1} + C_k \cdot \Psi_{k-1})]}{2/3 + A_k + C_k \cdot (1 - E_{k-1})}\end{aligned}$$

The set of linear equations (2.2.2.18) is closed at the lowest model level by

$$\psi_n^*(t+\Delta t) = E_n(t-\Delta t) \cdot \psi_s(t-\Delta t) + F_{\psi_n}(t-\Delta t)$$

(See section 2.2.2 C, equations (2.2.2.22) - (2.2.2.25)).

It is noted, that ψ_s -values at this stage is for time level $t-\Delta t$. New surface values are later determined by the surface parameterization scheme. (See section 2.2.5).

For momentum, however, $u_s = v_s = 0$ for all time levels. The value of $q_s(t-\Delta t)$ is determined by the following formula:

$$q_s(\text{over land}) = \min[G \cdot q_{sat}(T_s) + (1-G) \cdot q_n, q_{sat}(T_s)]$$

$$q_s(\text{over sea}) = q_{sat}(T_{sat})$$

T_s is the surface temperature over land, T_{sea} is surface temperature over sea. (See section 2.2.2C). G is surface wetness, which is parameterized in the following way:

$$G = \begin{cases} F_{sn} + (1-F_{sn}) \cdot (a_1 + (1-a_1) \cdot (W_s/W_{smax})^{a_2}) & \text{over land} \\ 1 & \text{over sea} \end{cases}$$

where $a_1=0.05$ and $a_2=8$.

W_s is the soil water content in the uppermost soil layer. W_{smax} is the maximum soil water content in the uppermost soil layer. (See section 2.2.5).

F_{sn} is the fraction of snow cover, which is also calculated in the vertical diffusion scheme and parameterized as

$$F_{sn} = \min(H_{sn} / 0.015, 1.)$$

H_{sn} is the snow height in meters of equivalent amount of water.

2.2.2 B Diffusion coefficients

The diffusion coefficients $A_k(\psi), C_k(\psi)$ in (2.2.2.17) become

$$\begin{aligned} A_k(\psi) &= \frac{g^2 \cdot \eta_{k+1/2} \cdot p_{k+1/2} \cdot 2 \cdot \Delta t}{p_s \cdot \Delta \eta_k \cdot R \cdot T_{k+1/2} \cdot \Delta \eta_{k+1/2}} \cdot K_{\psi_{k+1/2}} \\ C_k(\psi) &= \frac{g^2 \cdot \eta_{k-1/2} \cdot p_{k-1/2} \cdot 2 \cdot \Delta t}{p_s \cdot \Delta \eta_k \cdot R \cdot T_{k-1/2} \cdot \Delta \eta_{k-1/2}} \cdot K_{\psi_{k-1/2}} \end{aligned} \quad (2.2.2.20)$$

with

$$K_{\psi_{k\pm1/2}} = (l_{\psi_{k\pm1/2}} / \Delta Z_{k\pm1/2})^2 \cdot \Delta Z_{k\pm1/2} \cdot |\Delta \vec{v}|_{k\pm1/2} \cdot f_{\psi_{k\pm1/2}}$$

From (2.2.2.19) follows

$$\begin{aligned} C_{k+1}(\psi) &= A_k(\psi) \cdot \Delta \eta_k / \Delta \eta_{k+1} \\ A_k(\psi) &= A_k(u) \cdot (K_\psi / K_m)_{k+1/2} \\ (K_\psi / K_m)_{k+1/2} &= (l_\psi / l_m)^2 \cdot f_\psi / f_m \end{aligned} \quad (2.2.2.20)$$

At k=NLEV, using the lower boundary condition (See section 2.2.2, equation (2.2.2.3))

$$\begin{aligned} A_k(\psi) &= A_k(u) \cdot C_h / C_m \\ A_k(u) &= 2 \cdot \Delta t \cdot g / p_s \cdot 1 / \Delta \eta_n \cdot p_n \cdot C_m \cdot |\vec{v}|_n \end{aligned} \quad (2.2.2.21)$$

2.2.2 C Additional computations at the ground

Information about the distribution of land and sea/ice at the lower boundary of the atmosphere is used in order to determine surface fluxes with sufficient accuracy.

In every surface grid square of the model integration area the fraction of land and ice is specified and the fraction of sea calculated. The fraction of land (F_{land}) is defined as the fraction of the grid square covered by land. The fraction of ice (F_{ice}) is defined as the fraction of sea covered by ice. F_{ice}

is constant during model integration. How these fields are defined can be seen in sections 4.6, 8.2 and 8.3. Ice is treated as land in the surface energy budget calculations (See section 2.2.5).

The fraction of sea (F_{sea}) is similarly defined as the fraction of the surface grid square covered by open water. (sea or lake).

From these definitions follows

$$F_{sea} = (1 - F_{land}) \cdot (1 - F_{ice}),$$

and the 'total fraction of land' (including lake- and sea ice) becomes

$$F_{littot} = 1 - F_{sea}$$

Surface fluxes of momentum and heat (sensible and latent) are calculated separately for 'fraction of sea' and 'total fraction of land'.

The diffusion equations (2.2.2 eq (2.2.2.18)) are linear and solved by backward substitution starting at the lowest model layer, where the diffusion equations takes the form

$$\Psi_n^*(t+\Delta t) = E_n(t-\Delta t) \cdot \Psi_s(t-\Delta t) + F_{\Psi_n}(t-\Delta t) \quad (2.2.2.22)$$

$\Psi_n^*(t+\Delta t)$ is a provisional prognostic variable at time level $t+\Delta t$. ($\Psi_n(t-\Delta t)$ modified by vertical diffusion).

$E_n(t-\Delta t)$ and $F_{\Psi_n}(t-\Delta t)$ are diffusion coefficients. (See 2.2.2 eq. (2.2.2.18) and (2.2.2.24)).

Eq.(2.2.2.22) is solved for both $\Psi_n^*(sea)$ and $\Psi_n^*(land)$.

$$\Psi_n^*(sea) = E_n(sea) \cdot \Psi_s(sea) + F_{\Psi_n}(sea) \quad (2.2.2.23)$$

$$\Psi_n^*(land) = E_n(land) \cdot \Psi_s(land) + F_{\Psi_n}(land)$$

In eq. (2.2.2.23)

$$E_n(sea) = \frac{A_n(sea)}{2/3 + A_n(sea) + C_n(sea) \cdot (1 - E_{n-1})} \quad (2.2.2.24)$$

$$F_{\Psi_n}(sea) = \frac{C_n(sea) \cdot F_{\Psi_{n-1}} + 1/3 \cdot [\Psi_n \cdot (2 + A_n(sea) + C_n(sea)) - (A_n(sea) \cdot \Psi_s(sea) + C_n(sea) \cdot \Psi_{n-1})]}{2/3 + A_n(sea) + C_n(sea) \cdot (1 - E_{n-1})} \quad \text{Similar}$$

expressions hold for $E_n(land)$ and $F_{\Psi_n}(land)$.

For simplicity the vertical profiles of the prognostic variables are only allowed to diverge (towards the surface), in the surface layer (here defined as the layer between the surface and the lowest

model level). Therefore E_{n-1} , $F_{\psi_{n-1}}$, ψ_n and ψ_{n-1} in (24) are the same for sea and land. The backward solution of the diffusion equations (18) is started at the lowest level with

$$\psi_n^*(t+\Delta t) = (1-F_{sea}) \cdot \psi_n^*(land) + F_{sea} \cdot \psi_n^*(sea) \quad (2.2.2.25)$$

The procedure shortly outlined above requires two surface temperatures and two roughness lengths, for 'total fraction of land' and 'fraction of sea', respectively

The doubling is only required in grid squares, where $0 < F_{sea} < 1$, but for convenience the doubling is done everywhere in the integration area.

The surface temperature field T_{sea} , which represents the temperature of the fraction of the grid area covered by sea, is defined in section 4.6

T_{sea} is kept constant during model integration. The surface temperature T_s over the fraction of the grid area covered by land is a prognostic variable, initially assigned the same value as the analyzed temperature at the lowest model layer. In points with $F_{sea} = 1$ the T_s -and T_{sea} -fields are equal.

The roughness length (z_{osca}) over 'fraction of sea' ($0 < F_{sea} \leq 1$) is diagnostically determined from Charnock's formula (Charnock 1955).

$$z_{osca} = \max [\beta \cdot u_*^2 / g, 1.5 \cdot 10^{-5}], \quad \beta = 0.018$$

A 'first forecast' having no prognostic first guess fields carries a constant initial field for z_{osca} ($= 1.5 \times 10^{-5}$ m). In the forecast model z_{osca} is treated like a one time level prognostic variable. Another roughness field defines roughness over land ($F_{sea} < 1$). This field varies with geographical location, but is constant in time.

2.2.2 D Transfer of information

From the vertical diffusion subroutine **vdiff** the derivative of total surface heat flux with respect to 'total fraction of land' surface temperature is transferred to subroutine **surf**, where it is used in the surface energy budget calculations. (See section 2.2.5 for details)

The diffusion coefficients at the lowest model level calculated for 'total fraction of land' and 'fraction of sea' are applied in subroutine **anemlv**. This routine performs diagnostic calculations of near surface wind, temperature and specific humidity. The sensible and latent heat fluxes at the surface are calculated from the following formulas:

$$\varphi_s(t-\Delta t) = \rho_n \cdot |\vec{v}_s|_s \cdot S_{ss} = \rho_n \cdot C_{s_s} \cdot |\vec{v}(N)| \cdot [S_n(t-\Delta t) - S_s(t-\Delta t)]$$

$$\varphi_q(t-\Delta t) = \rho_n \cdot |\vec{v}_s|_s \cdot q_{ss} = \rho_n \cdot C_{q_s} \cdot |\vec{v}(N)| \cdot [q_n(t-\Delta t) - q_s(t-\Delta t)] \cdot L$$

The derivatives with respect to 'total fraction of land' surface temperature T_s become

$$\frac{\partial \varphi_s}{\partial T_s} = -\rho_n \cdot C_{s_s} \cdot |\vec{v}(N)| \cdot [C_{pd} \cdot 1 + (\delta - 1) \cdot q_s] + T_s \cdot C_{pd} \cdot (\delta - 1) \cdot G \cdot \frac{\partial q_{sat}}{\partial T_s}$$

$$\frac{\partial \Psi_s}{\partial T_s} = - \rho_n \cdot C_{q_s} \cdot |\vec{v}(N)| \cdot L \cdot G \cdot \frac{\partial q_{sat}}{\partial T_s}$$

$C_{s_s} = C_{q_s}$ are the drag coefficients for dry static energy and moisture. G is soil wetness ($0 \leq G \leq 1$)

2.2.3 Condensation and precipitation

2.2.3 A General

- * Cloud water content is a prognostic variable.
- * Two main parts constitute the cloud parameterization scheme.
 1. One part is the parameterization of the release of latent heat (production of condensate) and connected subgrid-scale circulation (fractional cloud cover). Two different schemes including cloud cover are developed, one for the convective and one for the stratiform condensation.
 2. The other part is the parameterization of microphysical processes, i.e. the release of precipitation, the evaporation and the melting of the hydrometers. Here, the parameterization schemes are identical for the convective and the stratiform cases but with the different parameter values.
- * Either convective or stratiform condensation is allowed in a grid box. No (stratiform) condensation is allowed beneath the base of convective clouds.
- * Conditions for convective condensation are first tested. If such conditions are not fulfilled, then a test for possible stratiform condensation will follow.

2.2.3 B Convective release of latent heat and associated cloud cover

A modified Kuo parameterization is applied based on buoyant surface air. The base of cloud is taken to be the lifting condensation level (LCL) of surface air, and a moist adiabat is calculated starting from this level. The intersection between the moist adiabat and the actual stratification determines the top of the cloud.

If T_k is a grid point temperature at level k and T_c is the temperature on the moist adiabat through LCL at the same level, the conditions for convection are that $(T_c - T_k) > 0$ for at least two model levels above LCL and that there is a convergence of water vapor in the column above LCL.

The governing equations (Sundqvist et al., 1989) are

$$\begin{aligned}\frac{\partial T}{\partial t} &= A_T + \xi(T_c - T) - \frac{L}{C_p} \xi H m \\ \frac{\partial q}{\partial t} &= \xi H(q_c - q) + \xi H m \\ \frac{\partial m}{\partial t} &= A_m + \frac{C_p}{L} \xi(T_c - T) - \xi H m - G_p\end{aligned}\tag{2.2.3.1}$$

where $\xi = \xi_0 f(k)$ is the quantity that results from Kuo's assumption and closure relation. ξ_0 is proportional to the convergence of vapor in the column divided by the amount of water vapor necessary to produce the model cloud. The function $f(k)$ has been introduced to govern the

vertical distribution of convective heating according to the following expression:

$$f(k) = \begin{cases} \frac{T_c - T_k}{(T_c - T_k)_{\max}} & \text{for } k \leq k_{dTmax} \\ 1 & \text{for } k > k_{dTmax} \end{cases} \quad (2.2.3.2)$$

where k_{dTmax} is the level, where $(T_c - T_k)$ has its maximum.

H is a function which deals with the partitioning of available vapor between condensation and moistening and is given by

$$H = (1 + U)^{pmoist} \quad (2.2.3.3)$$

The parameter $pmoist$ has been given the value of 3. The A-terms denote tendencies from all processes other than condensation (advection, diffusion etc.). The evaporation of detrained cloud water is parameterized by $\xi H m$, where m is the cloud water.

In order to guarantee numerical stability during the integration of the thermodynamic equation as well as for the moisture equations the semi-implicit method is used. This leads to the following expression

$$\xi_{imp} = \xi \frac{1}{1 + \xi \Delta t} \quad (2.2.3.4)$$

The cloud cover, b , is calculated according to the relation

$$b = \xi_{imp} \tau \frac{1 + (p_B - p_T)/hcunrm}{1 + 2.5 \xi_{imp} \tau} (1 + U) \quad (2.2.3.5)$$

where τ is a characteristic time scale for convection (~ 3600 s) and B denotes cloud base and T cloud top of the cumulus clouds. $hcunrm$ is a reference value of convective clouds, at present 3×10^4 .

Anvil

If the top level of the convection has a temperature of -20°C or below, the condensation at this level is treated as stratiform to allow an anvil to develop.

2.2.3 C Stratiform condensation

The prognostic equations for temperature, specific humidity and cloud water mixing ratio are, respectively,

$$\begin{aligned}
 \frac{\partial T}{\partial t} &= A_T + \frac{L}{C_p} Q - \frac{L}{C_p} E_0 \\
 \frac{\partial q}{\partial t} &= A_q - Q + E_0 \\
 \frac{\partial m}{\partial t} &= A_m + Q - (G_p - E_r) - E_0
 \end{aligned} \tag{2.2.3.6}$$

The A-terms denote tendencies from all processes other than condensation (advection, diffusion etc.). We also define the total evaporation E_0

$$E_0 = E_r + E_c \tag{2.2.3.7}$$

where E_r is the evaporation of precipitating water and E_c is the evaporation of cloud water advected into a grid box where no condensation is taking place. Since we assume that the release of latent heat takes place at the location of the water vapor convergence, we may put the Q and E_0 terms together into a net heating term.

This is obtained as

$$Q - E_0 = \frac{M - q_s \frac{\partial U}{\partial t}}{1 + US_q} \tag{2.2.3.8}$$

Here q_s is the saturation specific humidity and M is the convergence of the available latent heat into the grid box, (Sundqvist et al. 1989), given as

$$M = A_q - US_q \frac{C_p}{L} A_T + \frac{U q_s}{p} \frac{\partial p}{\partial t} \tag{2.2.3.9}$$

where

$$S_q = \frac{e q_s L^2}{R C_p T^2} \tag{2.2.3.10}$$

The factor $(1 + S_q)^{-1}$ is a reflection of an assumed wet-bulb process in which the converging vapor is used in part for the latent heat release and in part for the maintenance of saturation with respect to the changing temperature.

To close the system, we need an expression for the U-tendency. To obtain this, the available latent heat $M + E_0$ is divided into one part, $b \cdot M$, which condenses in the already cloudy portion of a grid box, and another part $(1 - b)M + E_0$, which is used to increase the relative humidity of the cloud free portion and to increase the cloudiness in the grid box.

From the aforementioned it is derived that

$$\frac{\partial U}{\partial t} = \frac{2(1 - b)(U_s - U_{00})[(1 - b)M + E_0]}{2q_s(1 - b)(U_s - U_{00}) + \frac{m}{b}} \quad (2.2.3.11)$$

Here we have adopted a diagnostic relation for stratiform cloud cover:

$$b = 1 - \sqrt{\frac{U_s - U}{U_s - U_{00}}} \quad (2.2.3.12)$$

where U is the grid averaged relative humidity, U_s is unity and U_{00} is the threshold value for condensation to start in a cloud free grid box. In a layer nearest to the ground (with the thickness of $0.1 p_s$) the threshold value changes linearly from the maximum value $U_{00\max}$ at the surface to the prescribed value $U_{00\text{land}}$ or $U_{00\text{ocean}}$ at the top of this layer. The effect of the topography is roughly taken into account by letting $U_{00\text{land}}$ and $U_{00\text{ocean}}$ differ by the parameter duoirl . Further, we increase the threshold value with the increasing stability in the two lowest model layers. We also have a correction that is implemented in the upper model layers. To prevent cirrus from appearing unrealistically frequently, we let U_{00} increase asymptotically to the value $U_{00\max}$ for temperatures lower than 238K.

2.2.3.D Parameterization of microphysical processes

The rate of release of precipitation, G_p , is described by the expression

$$G_p = C_0 m \left[1 - \exp \left(- \left(\frac{m}{bm_r} \right)^2 \right) \right] \quad (2.2.3.13)$$

The rate of conversion of cloud droplets to precipitation drops is governed by C_0 , which can be regarded as a function of autoconversion, coalescence and the Bergeron-Findeisen effects.

$$C_0 = C_{00} F_{co} f_{co} \quad (2.2.3.14)$$

$$F_{co} = 1 + C_1 \left(\frac{P_{coal}}{b} \right)^{0.5} + C_2 f_{B-F} \quad (2.2.3.15)$$

The first term of F_{co} expresses the growth by diffusion(autoconversion), involving processes of coalescence of cloud particles forming precipitation size particles; the second term expresses the growth of the precipitation particles through collection of cloud particles i.e collision-coalescence; the third term expresses the B-F mechanism i.e the growth of the ice particles at the expense of the droplets in a cloud where ice crystals and water droplets coexist.

Since there is no contribution in the uppermost cloud layer from precipitation entering from above

the coalescence effect is not strong enough. Therefore a local coalescence effect is introduced by adding a half of the rate of precipitation forming in the layer in question, based on the water content from the previous time step. The earlier rate of precipitation, P , is thus modified to enhance the coalescence effect and becomes

$$P_{coal}^{\tau} = P^{\tau} + 0.5 C_{00} \frac{\Delta P}{g} m^{\tau-1} \left[1 - \exp \left(- \left(\frac{m^{\tau-1}}{bm_r} \right)^2 \right) \right] \quad (2.2.3.16)$$

The probability of ice crystal existence as a function of temperature is calculated with a reference distribution function obtained by fitting an empirical curve (Matveev 1984).

Thus

$$f_{ice} = 1 - A(1 - e^{-x^2}) \quad (2.2.3.17)$$

with

$$f_{ice} = \begin{cases} 0 & T \geq 273 \\ f_{ice} & 273 > T > 232 \\ 1 & T \leq 232 \end{cases} \quad (2.2.3.18)$$

$$A = \frac{1}{1 - \exp \left[- \left(\frac{T_1 - T_{ci}}{(T_2 - T_{ci})\sqrt{2}} \right)^2 \right]}$$

$$T_1 = 273K$$

$$T_2 = 299K$$

$$T_{ci} = 232K$$

Accounting for an ice probability implies that the latent heat, L , also varies:

$$L = L_{vw} + f_{ice} \cdot L_{wi} \quad (2.2.3.19)$$

where L_{vw} is the latent heat of condensation and L_{wi} is the latent heat of freezing.

However, the resulting fraction of ice does not depend on the local temperature alone but also on the amount of ice crystals brought into the layer in question by precipitation from above. This modification is given by

$$f_{ice}^{(mod)} = f_{ice} + (1 - f_{ice}) \frac{P_{ice}}{P_{tot}} \quad (2.2.3.20)$$

This assumption implies that, if precipitation starts at a level with full glaciation ($P_{ice}/P_{tot}=1$), the modified probability of ice is unity, even if the local f_{ice} is less than unity, and the whole cloud consists of ice precipitation particles.

The Bergeron-Findeisen process is parameterized with the following expression

$$f_{B-F} = f_{ice}^{(mod)} (1 - f_{ice}) \Delta E_{w-i}^{(n)} \quad (2.2.3.21)$$

where $\Delta E_{w-i}^{(n)}$ is the difference in saturation vapor pressure over plain water and ice surfaces, normalized by its own maximum value.

Looking back at the original equation for calculation of the precipitation it is found that m_r indicates the amount of cloud water, around which the conversion becomes effective. This threshold is obtained from

$$m_r = m_{ro} \frac{f_{mr}}{F_{co}} \quad (2.2.3.22)$$

where f_{mr} is a function of temperature introduced to obtain realistically small amounts of cloud water for low temperatures.

$$f_{mr} = (1 - f_{ice})^2 + f_{ice} \cdot hmrost \quad (2.2.3.23)$$

where

$$hmrost = \begin{cases} \frac{4}{3} \cdot \exp \left[- \left[(T - 273) \cdot \frac{2}{30} \right]^2 \right] & T \geq 250K \\ 0.075 \cdot \left[1.07 + \frac{y}{1+y} \right] & 232 \leq T \leq 250 \\ 0.075 \cdot \left[1.07 - \frac{y}{1+y} \right] & T < 232K \end{cases} \quad \begin{cases} y = x + x^2 + \frac{4}{3}x^3 \\ x = \frac{|T - 232|}{18} \end{cases} \quad (2.2.3.24)$$

For similar reasons as for introducing the function f_{mr} we also introduce a function F_{co} , which multiplies F_{co} :

$$f_{co} = \begin{cases} 1 & T > 238 \\ 1 + \frac{238 - T}{2} & 230 < T \leq 238 \\ 5 & T \leq 230 \end{cases} \quad (2.2.3.25)$$

Finally, the amount of ice precipitation leaving the actual layer is determined from

$$P_{ice}(out) = P_{ice}(in) + f_{ice}^{(mod)} G_p \rho \Delta z \quad (2.2.3.26)$$

Evaporation and melting

The local rate of evaporation of precipitating water is given by

$$E_r = k_E(U_s - U) \sqrt{\frac{P}{b_k V}} \quad (2.2.3.27)$$

Here, \wedge denotes the in-cloud value. The cloud cover b_k (bar) from which precipitation falls, is assumed to be randomly overlapping with cloud free areas ($1-b_k$) in which the evaporation takes place. The grid box average effect of evaporation, thus is

$$P_{out} = P_{in} - \left[P_{in} - \left[\sqrt{P_{in}} - \frac{k_E}{2}(U_s - U)\sqrt{b_k V} \min\left(\frac{\Delta p}{g\rho V}, 2\Delta t\right) \right]^2 \right] \cdot (1 - b_k)^2 \quad (2.2.3.28)$$

where k_E is the evaporation coefficient and V is the terminal velocity of the precipitation. The evaporative cooling in $\text{kg/kg} \cdot \text{s}^{-1}$ is thus

$$\frac{(P_{in} - P_{out})g}{\Delta p} \quad (2.2.3.29)$$

The average, or bird's eye view, cloud cover \bar{b}_k (bar) is obtained from

$$\bar{b}_k = 1 - \prod_{k=1}^K \frac{1 - \max(b_{k-1}, b_k)}{1 - b_{k-1}} \quad (2.2.3.30)$$

which is derived from an assumption of maximum/random overlapping.

The derivation for the rate of melting ($T > 273\text{K}$) is analogous to the evaporation, but instead of a subsaturation we now have a temperature excess ($T - 273\text{K}$) and no cloud cover overlapping is involved.

$$P_{iceout} = P_{icein} - \left[P_{icein} - \left[\sqrt{P_{icein}} - \frac{k_m}{2}(T - 273)\sqrt{b_k V} \min\left(\frac{\Delta p}{g\rho V}, 2\Delta t\right) \right]^2 \right] \quad (2.2.3.31)$$

where k_m is the melting coefficient.

2.2.3 E Numerical treatment of the governing equations

As already discussed, in the convective case the tendency equations for temperature T and specific humidity q are solved semi-implicitly in contrast to the stratiform case where the equivalent equations are solved explicitly.

The tendency equation for cloud water is also obtained by semi-implicit time integration. In the latter, the term governing the rate of generation of precipitation is evaluated implicitly, while the remaining terms (RHS) are solved explicitly. Thus, the new value of m at time step ($\tau+1$) is obtained from

$$m^{\tau+1} = m^{\tau-1} + 2\Delta t \cdot RHS - 2\Delta t \cdot C_0 \frac{m^{\tau+1} + m^{\tau-1}}{2} \left\{ 1 - \exp \left[\left(\frac{m^{\tau+1} + m^{\tau-1}}{2b m_r} \right)^2 \right] \right\} \quad (2.2.3.32)$$

Then after introducing

$$x = \frac{m^{\tau+1} + m^{\tau-1}}{2b m_r} \quad (2.2.3.33)$$

and rearranging terms, we obtain a nonlinear equation in x , which is solved with the Newton-Raphson method.

$$x[1 + 2\Delta t \cdot C_0(1 - \exp^{-x^2})] - \frac{2m^{\tau-1} + 2\Delta t \cdot RHS}{2b m_r} = 0 \quad (2.2.3.34)$$

To ensure continuity the rate of precipitation from a layer k is calculated from

$$P_{out} = P_{in} + \frac{\Delta p}{g} \left(RHS - \frac{m^{\tau+1} - m^{\tau-1}}{2\Delta t} \right) \quad (2.2.3.35)$$

which in the continuous case is the same as

$$P(z) = \rho \int_z^{z'} G_p dz \quad (2.2.3.36)$$

References

- Sundqvist, H., Berge, E., and Kristjansson, J. E. (1989): Condensation and cloud parameterization studies with a mesoscale numerical weather prediction model. *Mon. Wea. Rev.* **117**, 1641 - 1657.
- Sundqvist, H. (1993): Inclusion of ice phase of hydrometeors in cloud parameterization for mesoscale and large-scale models. *Beitr. Phys. Atmosph.* **66**, 137 - 147.

2.2.5 Soil Processes

The surface parameterization scheme describes the evolution of soil temperature, soil water content and snow depth over land/ice by means of simple prognostic equations.

The surface belonging to every surface gridpoint may be either land, sea or a mixture of land and sea/lake and ice. To define this partitioning between different types of surfaces two fields are used: one field F_{land} describes the fraction of the grid area covered with land, and another field F_{ice} describes the fraction of ice relative to the area covered with sea/lake. (See sections 2.2.2, 4.6, 8.2).

The soil is represented by a 3 layer model. The depth of the top layer and the second layer are D_1 and D_2 respectively. The distance between the center of second and third layer is D_3 .

The horizontal fields defined over the whole domain are required for both temperature and soil water.

The first temperature field T_s , which is updated by the surface scheme, contains land surface temperature (upper soil layer) if $F_{\text{land}} + F_{\text{ice}} > 0$, otherwise sea surface temperature, which is constant during the forecast.

Another temperature field T_d represents temperature in the intermediate soil layer.

A third temperature field T_{scl} contains the deep soil layer temperature if $0 < F_{\text{land}} + F_{\text{ice}} \leq 1$ (climate field). If $F_{\text{land}} + F_{\text{ice}} = 0$ the sea surface temperature is stored.

Similarly the soil water content w_s in the uppermost layer and the water content w_d of the intermediate layer are updated by the surface scheme (provided, that $F_{\text{land}} + F_{\text{ice}} > 0$) while the soil water content w_{scl} of the deepest layer is kept constant during the forecast. (climate field. see chapter 8).

The surface temperature field T_s is initially equal to the analyzed temperature field T_n for the lowest model layer. ($F_{\text{sea}} < 1$). The other fields (T_d , w_s , w_d) are initialized to the first guess values from the previous forecast. If no forecast is available the fields are determined from climate information. (See chapter 8).

The evolution of temperature and soil water is assumed to obey a simple diffusion equation. (ice is treated as land).

Soil temperature:

The equations for temperature are written as follows.

$$(1) \left\{ \begin{array}{l} \frac{\partial T_s}{\partial t} = \frac{1}{\rho_s \cdot c_s \cdot D_1} \cdot \sum_i \Phi_i + \frac{\kappa_0 \cdot (1 - k_{sn} \cdot F_{sn}) \cdot (T_d - T_s)}{0.5 \cdot D_1 \cdot (D_1 + D_2)} \\ \sum_i \Phi_i = \Phi_r + \Phi_s + \Phi_q . \quad F_{sn} = \min \{ H_{sn}/H_{snc}, 1 \} \end{array} \right.$$

T_s is the soil temperature in the upper soil layer. (K)

HDM/Orig(June 1990)

T_d	is the soil temperature in the second soil layer. (K)
H_{sn}	is snow height in meters of equivalent water. (m)
H_{snc}	is a scaling snow height. (m)
ρ_s	is the density of the soil. (kg/m^3).
c_s	is the specific heat capacity of the soil. ($\text{J}/\text{kg}/\text{K}$).
κ_0	is the heat diffusivity of soil without snow cover. (m^2/s)
k_{sn}	is a dimensionless constant used to reduce heat diffusivity if snow cover is positive.
F_{sn}	is fraction of snow cover.
Φ_r	is net absorbed flux density of radiation at the surface. ($\text{J}/\text{m}^2/\text{s}$). (transferred from subroutine 'RADIA').
Φ_s	is sensible heat flux density at the surface. ($\text{J}/\text{m}^2/\text{s}$) (transferred from subroutine 'VDIFF').
Φ_q	is latent heat flux density at the surface. ($\text{J}/\text{m}^2/\text{s}$) (transferred from subroutine 'VDIFF').

Values of constants:

$$\rho_s \cdot c_s = 2.7 \cdot 10^6 \text{ (J/m}^3/\text{K}), \quad \kappa_0 = 7.5 \cdot 10^{-7} \text{ (m}^2/\text{s})$$

$$k_{sn} = \frac{2}{3}, \quad H_{snc} = 0.015 \text{ (m)}, \quad D_1 = 7.2 \cdot 10^{-2} \text{ (m)},$$

$$D_2 = 6 \cdot D_1$$

$$(2) \quad \frac{\partial T_d}{\partial t} = - \frac{(T_d - T_s) \cdot \kappa_0}{0.5 \cdot D_2 \cdot (D_1 + D_2)} + \frac{(T_{s,1,i} - T_d) \cdot \kappa_0}{D_2 \cdot D_3}$$

$$D_3 = D_2 = 6 \cdot D_1 \text{ (m)}$$

Similarly the equations for soil water can be written as

$$(3) \quad \frac{\partial w_s}{\partial t} = (1 - F_{sn}) \cdot \Phi_q + P_{rn} + M_{sn} + \frac{(w_d - w_s) \cdot \lambda}{0.5 \cdot D_1 \cdot (D_1 + D_2)}$$

$$(4) \quad \frac{\partial w_d}{\partial t} = - \frac{(w_d - w_s) \cdot \lambda}{0.5 \cdot D_2 \cdot (D_1 + D_2)} + \frac{(w_{s,1,i} - w_d) \cdot \lambda}{D_2 \cdot D_3}$$

w_s is soil water in the top layer (meters of water).

- w_d is soil water in the second soil layer times (D_1/ D_2).
 (meters of water). – It is scaled to the depth of the top layer.
- w_{scli} is the soil water amount of the bottom layer scaled to the depth of the top layer. (meters of water).
- P_{rn} is the rain flux density reaching the ground. ($\text{kg/m}^2/\text{s}$)
 (transferred from subroutines 'KUO' and 'COND').
- M_{sn} is the change in snow height per unit time.
- λ is a diffusivity for conduction of soil water.
 $(\lambda = 1.10^{-7} \text{ m}^2/\text{s})$

Snow height H_{sn} is updated if $F_{land} + F_{ice} > 0$, and the equation is written as follows.

$$(5) \quad \frac{\partial H_{sn}}{\partial t} = F_{sn} \cdot \Phi_q / \rho_{h2o} + P_{sn} - M_{sn}$$

ρ_{h2o} is the density of water (kg/m^3).

P_{sn} is the change in snow height pr. unit time as a result of precipitation.
 (transferred from subroutine 'KUO'/'COND').

Corrections due to snow melting and water runoff.

A given soil layer can only contain a limited amount of soil water. w_s and w_d are thus constrained to be less than a specified maximum value:

$$w_s \leq w_{smax}, w_d \leq w_{dmax}, w_{smax} = 0.02 \text{ m}$$

Furthermore both w_s and w_d must be greater than zero.

$$(w_s > 10^{-10} \text{ m}, w_d > 10^{-10} \text{ m})$$

Snow melt affects snow height H_{sn} and soil water w_s . The model surface temperature is only affected by the melting process if ice is present in the grid area. The area mean surface temperature should be able to rise above the freezing point before all snow has melted, since the surface temperature in snow free areas may be considerably above 0°C. The temperature of sea ice is however not expected to increase above the freezing point of sea water taken to be -1.9°C . Thus the area covered by ice is forced not to exceed the freezing point of sea water.

If the fraction of the grid area covered by land is F_{land} then the fraction of land + ice covered by ice is given by

$$(6) \quad F_{ice}^* = F_{ice} \cdot (1 - F_{land}) / (F_{land} + F_{ice} \cdot (1 - F_{land}))$$

In (6) F_{ice} is the fraction of ice relative to the fraction of sea/lake.

HDM/Orig(June 1990)

The snow melt is determined after the provisional values \hat{T}_s and \hat{H}_{sn} have been determined by application of (1) and (5):

$$(7) \quad M_{sn} = \min\left\{\frac{\rho_s \cdot c_s \cdot D_1 \cdot (\hat{T}_s - T_{melt})}{2 \cdot \Delta t \cdot L_i \cdot \rho_{h_2o}}, H_{sn}\right\}$$

The associated updating of temperature, snow height and soil water is then given by

$$(8) \quad T_s = \hat{T}_s - F_{ice}^* \cdot \max\{(\hat{T}_s - T_{seaf}), 0\}$$

$$(9) \quad H_{sn} = \hat{H}_{sn} - M_{sn}$$

$$(10) \quad w_s = \tilde{w}_s + M_{sn}$$

T_{seaf} is the freezing point of sea water. (K)

Numerical aspects:

In order to prevent possible numerical instability, equation (1) is solved by the following implicit time stepping procedure: The equations are written in a general way as

$$(11) \quad \frac{\partial \psi}{\partial t} = f(\psi), \quad (\psi = T_s)$$

$f(\psi)$ can be linearized in the following way:

$$(12) \quad f^{t+1} = f^{t-1} + \frac{\partial f^{t-1}}{\partial \psi} \cdot (\psi^{t+1} - \psi^{t-1})$$

If (12) is substituted into (11) we get for a centered implicit time step:

$$(13) \quad \frac{\psi^{t+1} - \psi^{t-1}}{2 \cdot \Delta t} = \frac{1}{2} \cdot (f^{t-1} + f^{t+1})$$

From this we obtain

$$(14) \quad \psi^{t+1} = \psi^{t-1} + \frac{2 \cdot \Delta t \cdot f^{t-1}}{(1 - \Delta t \cdot \partial f^{t-1} / \partial \psi)}$$

This implies, that in order to update variable ψ it is necessary to determine the partial derivatives of the right hand side of equation (1) with respect to ψ . Considering the parameters transferred from other subroutines, Φ_r from 'RADIA', Φ_s and Φ_q from 'VDIFF', P_{rn} and P_{sn} from 'KUO' and 'COND' we determine $\partial \Phi_r / \partial T_s$:

$$\frac{\partial \Phi_r}{\partial T_s} = 4 \cdot \epsilon_s \cdot b_0 \cdot T_s^3$$

The remaining partial derivatives are determined in 'SURF' except the derivative of sensible heat and moisture flux. It is only necessary to determine the partial derivatives of the sum $\Phi_s + \Phi_q$ with respect to T_s . This is done in subroutine 'VDIFF'.

In addition to this implicit procedure a time filter is used to connect the odd time steps with the even ones. This filter uses only the forward part of the operator used for the atmospheric variables:

$$(15) \quad \psi_{\text{filt}}^t = \psi^t + \epsilon_f \cdot (\psi^{t+1} - \psi^t), \quad \epsilon_f = 0.05$$

The reason for using the forward operator only is to allow snow to melt completely and soil moisture to reach saturation.

2.2.6 Diagnostic calculation of near surface parameters

These calculations are performed in subroutine 'ANEMLV'. This routine needs information about $A_n(u)$ and C_h/C_m (See 2.2.2 B). This information is transferred (in 'WCA' and 'WCC' respectively) to subroutine 'ANEMLV' FROM 'VDIFF'.

The parameters calculated are T_2 (temperature 2 m. above the surface), \vec{v}_{10} (wind velocity 10 m above the surface) and q_2 (specific humidity 2 m. above the surface).

The stable surface layer

In case of a stably stratified surface layer the near surface parameters are obtained from (Woetmann Nielsen and Hansen Sass, 1987)

$$(1) \quad \begin{aligned} u(z) &= (u_*/k) \cdot \ln(z/z_0) + u_n \cdot (1 - \exp(-b_m \cdot k^{-1} \cdot u_*/u_n \cdot z/L)) \\ v(z) &= (v_*/k) \cdot \ln(z/z_0) + v_n \cdot (1 - \exp(-b_m \cdot k^{-1} \cdot v_*/v_n \cdot z/L)) \\ T(z) &= T_s + (\theta_*/k) \cdot \ln(z/z_0) + \Delta\theta_n \cdot (1 - \exp(-b_h \cdot k^{-1} \cdot \theta_*/\theta_n \cdot z/L)) \\ q(z) &= q_s + (q_*/k) \cdot \ln(z/z_0) + \Delta q_n \cdot (1 - \exp(-b_q \cdot k^{-1} \cdot q_*/q_n \cdot z/L)) \end{aligned}$$

where subscript n refers to the lowest model level and

$$\begin{aligned} \Delta\theta_n &= \theta_n - \theta_s \quad , \quad \Delta q_n = q_n - q_s \\ b_m &= 1/Ricr \approx 4. \end{aligned}$$

It is assumed that $b_h = b_q = b_m$.
The formulae (1) are a modified form of the log-linear, integrated Businger profiles.

The unstable surface layer

In case of an unstably stratified surface layer the near surface parameters are obtained from

$$(2) \quad \begin{aligned} u(z) &= u_*/k \cdot \left\{ \ln(z/z_0) - \left(\ln\left(\frac{1+x^2}{2}\right) + 2 \cdot \ln\left(\frac{1+x}{2}\right) - 2 \cdot \tan^{-1}x + \frac{\pi}{2} \right) \right\} \\ v(z) &= v_*/k \cdot \left\{ \ln(z/z_0) - \left(\ln\left(\frac{1+x^2}{2}\right) + 2 \cdot \ln\left(\frac{1+x}{2}\right) - 2 \cdot \tan^{-1}x + \frac{\pi}{2} \right) \right\} \\ T(z) &\approx T_s + c_\theta \cdot \theta_*/k \cdot \left\{ \ln(z/z_0) - 2 \cdot \ln\left(\frac{1+y}{2}\right) \right\} \\ q(z) &= q_s + c_q \cdot q_*/k \cdot \left\{ \ln(z/z_0) - 2 \cdot \ln\left(\frac{1+v}{2}\right) \right\} \end{aligned}$$

where

$$x = (1 - 15 \cdot z/L)^{\frac{1}{4}} , \quad y = (1 - 9 \cdot z/L)^{\frac{1}{2}}$$

$$c_\theta = c_q = 1$$

(2) is obtained by vertical integration of the Businger profiles (Paulson, 1970) and neglecting the dependence on z_0/L . Thus (2) is an approximation to

$$(3) \quad \begin{aligned} u(z) &= u_* / k \cdot \left\{ \ln(z/z_0) - \left(\ln\left(\frac{1+x^2}{1+x_0^2}\right) + 2 \cdot \ln\left(\frac{1+x}{1+x_0}\right) - 2 \cdot \tan^{-1}\left(\frac{x-x_0}{1+x \cdot x_0}\right) \right) \right\} \\ v(z) &= v_* / k \cdot \left\{ \ln(z/z_0) - \left(\ln\left(\frac{1+y^2}{1+y_0^2}\right) + 2 \cdot \ln\left(\frac{1+y}{1+y_0}\right) - 2 \cdot \tan^{-1}\left(\frac{y-y_0}{1+y \cdot y_0}\right) \right) \right\} \\ T(z) &= T_s + \theta_* / k \cdot \left\{ \ln(z/z_0) - \ln\left(\frac{1+v}{1+y_0}\right) \right\} \\ q(z) &= q_s + q_* / k \cdot \left\{ \ln(z/z_0) - \ln\left(\frac{1+v}{1+y_0}\right) \right\} \end{aligned}$$

It should be pointed out that the "Businger profiles" (whatever it is (1), (2) or (3)) are not fully consistent with the vertical diffusion formulae used in the model. However it is not believed to have serious consequences.

2.3.1 Dynamic part

General overview

Table 2.3.1 summarizes the algorithm for the adiabatic part of the forecast model. The normal mode initialization scheme, which is coded as a part of the forecast model, is described in Section 3.2, and the organization of the physics is described in section 2.3.2. Usually, the initialization and the forecast steps are performed in the same run. This is done by calling the main routine, `gemini`, twice (initialization and forecast) from the main program, `hlprog`, with different input steering parameters in the namelists. Routine `gemini` is organized as follows:

- (a) Input of namelist, `namdev`, `namrun`, `namtun`, `namos` and `namppp`. A complete description of their contents is given in Section 2.4.
 - `namdev` contains information about the input file units for the start data set and the boundary data sets and output file units for the model level history files.
 - `namrun` contains the steering parameters for the run; timestep, forecast length, switches between different formulations etc.
 - `namtun`, read in routine `contun`, defines tuning constants in the physics.
 - `namos`, read in routine `inipos`, defines the postprocessed files. Several sets of output files can be defined, so called postprocessing streams, where the content of each stream is given in following namelists `namppp`.
- (b) Calls to several routines which define time independent constants for boundary relaxation (`bdinit`), horizontal diffusion (`difini`) and implicit schemes (`impini`).
- (c) The input start data set and the first pair of boundary data in grib format are read in routine `getdat`. The information about the geographical location of the integration area and the definition of model levels are taken from the file descriptor in the start data file. In order to save some space only values in the boundary relaxation zone are stored during the forecast.
- (d) Routine `initsf` is called in order to initiate the output of timeseries data in geographical points, where the locations and output frequency is given in namelist `namtsf`.
- (e) Routine `iniphy` inserts default values for the surface fields for the physics if not available from the start data set.
- (f) The experiment name is read in order to overwrite the experiment name in the start data file.
- (g) For a normal mode initialization run, the routine `normod` is called for each iteration in the initialization procedure.
- (h) The time step loop. For normal mode initialization, the purpose of this loop is only to create output at the initial time. For the time integration, this loop organizes the computations during one time step.
 - Routine `euler` is called for the eulerian explicit or semi-implicit scheme.
 - Routine `sl2tim` is called for the Semi-Lagrangian, semi-implicit two-time level scheme.
 - Routines `statis` and `prstat` are called for computation and output of the area mean absolute surface pressure tendency and the area mean integrated values.

- The fields at time n are timefiltered and then the variables for the next time step are prepared.
- Routine getdat reads the next set of boundary data and extracts values in the boundary relaxation zone.
- Routine puttsf is called for output of time series data in geographical points.
- Routine putdat is called for output of model level data sets and for output of postprocessed data.

If the forecast is found to be unstable, recognized from the pressure tendency and the maximum velocity, a model level data set is written and the integration is stopped.

- (i) Routine fintst finalizes the output of time series data.

Eulerian scheme

Routine euler organizes the time extrapolation during one timestep, for the eulerian explicit or semi-implicit scheme:

- Routine dyn computes the explicit time tendency of surface pressure, horizontal velocity components, temperature and humiditys.
- Routine dyn is followed by an explicit leapfrog time extrapolation.
- Routine hdiff4 computes the changes due to horizontal diffusion of the horizontal velocity components, temperature and specific humidity, and adds these to the explicitly forecasted values.
- For the semi-implicit scheme, routine expadj creates the right hand side of the Helmholtz equation for the divergence variable, routine hhsolv computes the solution of the Helmholtz equations and routine impadj adds the correction terms to the tendencies from the solution for the divergence variable. The final output is the adjusted adiabatic values at the next step.
- Routine bndrel is performing the boundary relaxation of the fields at time step n+1.
- Routine phcall adds the tendency due to physical parameterization to the dynamic tendency. Note that these changes are added after completion of the semi-implicit adjustment. For this purpose, the tendency from the dynamics is computed before phcall, followed by time extrapolation after routine phcall. The changes from the physics are added with less weight in the boundary relaxation zone, in order to keep the correct boundary values.

Extra scalar prognostic variables

Extra scalar prognostic variables are included in the following way:

1. Compile the libraries (grdy, prpo, phys) with the variable msvar > 0 where msvar is the max. number of extra scalars.
2. Specify nsvar and nwmosv in the input namelist namrun in the forecast model (script Forecast) where nsvar is the actual number of extra scalars and nwmosv is a list of nsvar wmo code numbers. Some variables concerning diffusion may also have to be changed from the default values in namrun: ak4lev, ndifx, cdifx, nlldifx.
3. Output of extra scalars are set in the postprocessing streams in the script Forecast by adding

the corresponding wmo code numbers in the list of output code numbers at model levels and/or pressure levels.

According to the list of nsvar values of wmo code numbers in the array nwmosv in the input namelist namrun, the model will search for these data in the initial data file and the boundary data files. If they are not present in the file, they are set to zero (in subroutine getdat), in the same way as implemented for cloud liquid water. The boundary relaxation is treated in the same way as for cloud liquid water: If the eulerian time scheme is applied, the boundary relaxation is performed only if the boundary values are available. If the semi-Lagrangian scheme is applied, the extra scalars are relaxed towards zero if the data are not present in the boundary files.

Change of integration area

The following variables, given in parameter statements, are used for compilation of the model code:

mlon	max. number of grid points in x-direction.
mlat	max. number of grid points in y-direction.
mlev	max. number of model full levels.
mbdpts	max. number of grid points in the boundary relaxation zone (including the extra passive lines with full weight to the boundary field used in the semi-Lagrangian scheme).
mmlfld	max. number of multi level boundary field variables (u, v, T, q, s and in addition sdot in the semi-Lagrangian scheme).
mmodes	max. number of vertical modes in the normal mode initialization scheme.
mtask	max. number of tasks in the physics (for multi-tasking).
msvar	max. number of extra scalar prognostic variables (default=0)

In the code, there are similar variables defined from the input namelist namrun parameters:

nlon	number of points in x-direction.
nlat	number of points in y-direction.
nlev	number of model full levels.
nbdpts	number of points in the boundary relaxation zone.
npbpts	number of extra passive lines with full weight to the boundary fields (used in the semi-Lagrangian scheme in order to prevent the trajectories from going outside the integration area).
nmlfld	number of multilevel boundary field variables.
nmodes	number of vertical modes in the normal mode initialization scheme.
nhorph	number of gridpoints in the subarea for physics computations.
nsvar	number of extra scalar prognostic variables (default=0)

The following restrictions must be fulfilled:

$$\begin{aligned} \text{mlon} &\geq \text{nlon} \\ \text{mlat} &\geq \text{nlat} \\ \text{mlev} &\geq \text{nlev} \\ \text{mbdpts} &\geq \text{nbdpts} + \text{npbpts} \\ \text{mmlfld} &\geq \text{nmlfld} \\ \text{mmodes} &\geq \text{nmodes} \\ \text{mtask} &\geq (\text{nlon} * \text{nlat} + \text{nhorph} - 1) / \text{nhorph} \\ \text{msvar} &\geq \text{nsvar} \end{aligned}$$

In addition, $(nlon-2-2*npbpts)/2$ must be factors of 2, 3 and 5 for the Fourier transform routine used in the Helmholtz-solver.

Finally, nbdpts should be an even number ≥ 4 if the tanh-shape is chosen for the boundary relaxation function.

Table 2.3.1 Algorithm and calling tree for the dynamics

Main algorithm

```
hlprog - forecast model main program
second - accumulate cpu time for forecast
gemini - hirlam main routine
    <read input namelists namdev and namrun>
    second - accumulate cpu time for preparations
    contun - initiate physical tuning constants
        <read input namelist namtun>
    conphys - initiate physical constants
    inipos - initiate output of history files
        <read input namelists nampos and namppp>
    <check actual against compiled dimensions>
    bdinit - initiate boundary relaxation function
    getdat - read input start data set and initiate parameters
    initfs - initiate output of time series data
        <read input namelist namtsf>
    <extract experiment name from start data file or input>
    iniphy - initiate physics surface parameters
        minmax - compute min- and max-values of surface fields
        getdat - read the input boundary data set and initiate parameters
        <initiate variables for time step loop>
        difini - initiate the fourth order diffusion scheme
        impini - initialization of implicit schemes
            matin1 - matrix inversion routine
            geteig - main routine for eigenvalues and -vectors
                eisrg1 - routine for eigenvalues and -vectors
                balanc - routine for eigenvalues and -vectors
                elmhes - routine for eigenvalues and -vectors
                eltran - routine for eigenvalues and -vectors
                hqr2 - routine for eigenvalues and -vectors
                balbak - routine for eigenvalues and -vectors
            matin1 - matrix inversion routine
            fax - compute the factors in the Fourier transformation
            fftrig - compute the trigonomometric functions for fft
    second - accumulate cpu time for preparations
    <print cpu time for preparations>
    <implicit normal mode initialization:>
    <iteration loop:>
    do iter=1,nitnmi
        normod - normal mode initialization scheme
    enddo
    <time step loop:>
    do nstep=1,nstop
        dates - compute current date/time for physics
        euler - Eulerian time integration
        sl2tim - semi-Lagrangian semi-implicit 2-time-level scheme
        statis - statistics computations
```

```

prstat - print output from the statistics computations
    <time-filters and preparations for next time step>
getdat - read the next boundary data set and initiate parameters
puttsf - put time series data
putdat - postprocessing and output of the history files
enddo
fintsf - finish output of time series data
second - accumulate cpu time for forecast
    <print cpu time for forecast>

```

Auxilliary routines

ismin - index with minimum value of array elements
ismax - index with maximum value of array elements
abort - abort the execution

Input of data sets

```

getdat - read the input data set and initiate parameters
  getgrb - read the GRIB data set
  mapfac - compute the geographically dependent variables
    lonlat - compute latitude and longitude
  extrda - extract a column of data
  crihum - check critical values of humidity
  comped - compute vertical velocity sdot
    <compute statistics from data set:>
  statis - statistics computations
  prstat - print output from the statistics computations
    <extract data for boundary relaxation>

```

Output of data sets

```

putdat - postprocessing and output of the history files
  <output of pressure levels:>
  extend - extend data array in the boundary zone
  extzer - extend data array with zeroes in the boundary zone
  postp - postprocessing
    omcomp - computation of the vertical velocity omega
    postpp - postprocessing to pressure levels
  <output of model levels:>
  extrda - extract data
  extend - extend data array in the boundary zone
  extzer - extend data array with zeroes in the boundary zone
  putgrb - write GRIB data set

```

Eulerian semi-implicit scheme

euler - Eulerian time integration
dyn - explicit dynamical tendency
 <explicit Leapfrog time stepping>
extend - extend data array for computation of lnp
hdiff4 - fourth order horizontal diffusion
expadj - eulerian explicit adjustment
hhsolv - master routine for solution of Helmholtz-equations
divav - compute average divergence
impadj - implicit semi-implicit adjustment
bndrel - boundary relaxation
 <physical parameterization:>
 <compute dynamic tendency for phcall>
phcall - main routine for physical parameterization
 <explicit timestepping after phcall>

Solution of Helmholtz-equations

hhsolv - master routine for solution of Helmholtz-equations
matmul - matrix multiplication (vertical decoupling)
trimat - triangular matrix for Helmholtz solver
hhsolx - Helmholtz-solver
fft44 - fast Fourier sine-transformation
vpassm - fast Fourier sine-transformation
fft44a - fast Fourier sine-transformation
fft44b - fast Fourier sine-transformation
matmul - matrix multiplication (vertical coupling)

Semi-Lagrangian semi-implicit 2-time-level-scheme

sl2tim - semi-Lagrangian semi-implicit 2-time-level scheme
compx - compute fields at timestep n+1/2
sldyn - explicit dynamical tendency
 <explicit forward time stepping>
hdiff4 - fourth order horizontal diffusion
sldynam - semi-Lagrangian, semi-implicit integration
preint - preparation for interpolation routines
 <compute nonlinear terms at time n+1/2>
 <compute the displacements:>
bivint - vertical interpolation of vertical velocity sdot
destag - interpolation from velocitypoints to masspoints
mimagr - statistics computations from sdot
calpqr - calculate the three-dimensional displacement
bixint - main interpolation routine
 verint - three dimensional interpolation
 horint - horizontal interpolation
 <interpolation at the departure points:>
bixint - main interpolation routine
 verint - three dimensional interpolation
 horint - horizontal interpolation

intpqr - interpolation from masspoints to velocitypoints
slexpa - semi-Lagrangian explicit adjustment
hholv - master routine for solution of Helmholtz-equations
divav - compute average divergence
impadj - implicit semi-implicit adjustment
bndrel - boundary relaxation
<physical parameterization:>
<compute dynamic tendency for phcall>
phcall - main routine for physical parameterization
<explicit timestepping after phcall>

Table 2.3.2 Parameter statements and common blocks in the dynamics

Parameter statements for definition of the main dimensions

mlon	max. number of gridpoints in x-direction
mlat	max. number of gridpoints in y-direction
mlev	max. number of model full levels
mbdpts	max. number of gridpoints in boundary relaxation zone including passive lines (mbdpts >= npbpts+nbdpts)
mmlfld	max. number of multilevel boundary variables (u,v,T,q,s,sdot)
mmodes	max. number of vertical modes in initialization
mtask	max. number of tasks in physics (for multi-tasking) (mtask >= (nlon*nlat+nhorph-1)/nhorph)
msvar	max. number of extra scalar prognostic variables (default=0)

Common blocks

comhkp - house keeping parameters

nlon	number of gridpoints in the x-direction
nlat	number of gridpoints in the y-direction
nlev	number of full model levels
nstep	time step counter (nstep=0,1,...,nstop)
nstop	number of time steps
nyear	current year (at time step nstep)
nmonth	current month (at time step nstep)
nday	current day (at time step nstep)
nhour	current hour (at time step nstep)
nmin	current minute (at time step nstep)
nsec	current second (at time step nstep)
ncbase	time for start data set (yymmdd)
ntbase	hour for start data set
ntdata	second for start data set
lamcho	choice of lam formulation concerning discretization at the uppermost level =1 for Eulerian time scheme =2 for semi-Lagrangian time scheme
nexp	switch for experiment name =0 : experiment name taken from the start data set =1 : experiment name given as input (last input line)
nwppt	pointer to the elements of array nwtime
nwtime	list of time steps for output model level data sets
tdata	time of the day (in seconds)
twodt	double timestep (2*dt) (in seconds)
eps	coefficient for Asselin time filtering and time filtering for surface physics fields
nleul	.true. if Eulerian time scheme
nlsimp	.true. if semi-implicit scheme
nlphys	.true. if physical parameterization
nlstat	.true. if computation and print of statistics
nlhdif	.true. if horizontal diffusion scheme in dynamics
nlvdif	.true. if vertical diffusion scheme in physics
nrad	.true. if radiation scheme in physics
nlkuo	.true. if Kuo convection scheme in physics

nlcond	.true. if large scale condensation scheme in physics
nlsurf	.true. if surface parameterization scheme in physics
nlhumc	.true. if check of critical humidity values for input data
nltvir	.true. if virtual temperature in dynamics
nhorph	number of gridpoints in each sub-area for the physics (the loop length in each physics routine is nhorph and the physics routines are multitasked for all sub-areas)
dtphys	time step for physics (seconds)
dtdif	time step for vertical diffusion in physics (seconds)
timesu	spinup time in seconds (the physics is called every dynamic time step during spinup time)
nldynvd	.true. if dynamic tendency used in the vertical diffusion scheme
nlsund	.true. if Sundqvist condensation scheme in physics
nlstph	.true. if statistics computations in physics
nlpost	.true. if postprocessing at current time step nstep
nsvar	number of extra scalar prognostic variables (default=0)

commap - geographical dependent variables and level parameters

anorth	northern boundary (degrees)
west	western boundary (degrees)
south	southern boundary (degrees)
east	eastern boundary (degrees)
aplon	longitude position of rotated south pole (degrees)
aplat	latitude position of rotated south pole (degrees)
dlamda	grid distance in the x-direction (degrees)
dtheta	grid distance in the y-direction (degrees)
rdlam	inverse grid distance in the x-direction (radians)
rdth	inverse grid distance in the y-direction (radians)
ra	inverse radius of earth (meter)
ahyb	a-parameter for model half level definition ($p=a+b*ps$)
bhyb	b-parameter for model half level definition ($p=a+b*ps$)
rhxu	inverse map-factor in the x-direction (in u-points)
rhyv	inverse map-factor in the y-direction (in v-points)
hxv	map-factor in the x-direction (in v-points)
hyu	map-factor in the y-direction (in u-points)
phis	surface geopotential ($(m*m)/(s*s)$)
hybi	reference hybrid half level definition ($hybi=a/p0+b$)
hybk	reference hybrid full level definition ($hybk=a/p0+b$)

comimp - constants for implicit scheme etc.

sit0	reference temperature (Kelvin)
sip0	reference surface pressure (Pa)
sitau1	list of level dependent coefficients for the linear terms in the thermodynamic equation
sitau2	list of level dependent coefficients for the linear terms in the thermodynamic equation
sigam1	list of level dependent coefficients for the linear terms in the hydrostatic equation
sigam2	list of level dependent coefficients for the linear terms in the hydrostatic equation
sidpk0	list of level dependent coefficients for the linear terms in the continuity equation
eig	eigenvector matrix for vertical decoupling
eiginv	eigenvector matrix for vertical coupling
c2	eigenvalues for vertical normal modes
sk	sine coefficient for Helmholtz solver

cosu	cosine(latitude) in u-points for Gaussian elimination
cosv	cosine(latitude) in v-points for Gaussian elimination
nfax	list of factors for the Fourier transformation
trigs	list of trigonometric function for the Fourier transformation
epsg	coefficient for the gravity wave damper in the semi-Lagrangian scheme

combdy - boundary relaxation constants

lubfi1	unit for first boundary data set (=lusdfi)
lubfi2	unit for second boundary data set (following boundary data sets are assumed to be lubfi2+1, lubfi2+2,...)
nbdnum	pointer to current boundary data set
nbdpts	number of gridpoints in the boundary relaxation zone for the boundary relaxation function alfab. alfab=1.0 in gridpoint i=1+npbpts alfab=0.0 in gridpoint i=nbdpts+1+npbpts
nbdtim	time counter for interpolation between boundary data sets
nmlfld	number of multi level boundary variables (u,v,T,q,s,sdot)
nlnltb	number of gridpoints where the boundary relaxation function alfab!=0.0
nbdptr	pointer to boundary buffer in common comibv
timrat	time rate for interpolation between boundary data sets
bdfunc	list of values for the boundary relaxation function alfab if nltanh=.false. in the interval [1.0,...0.0]> bdfunc=1.0 in gridpoint i=1+npbpts bdfunc=0.0 in gridpoint i=nbdpts+1+npbpts
alfab	boundary relaxation function
nalfa	pointer to gridpoints where alfab!=0.0 (nalfa(i),i=1,nlnltb)
nltanh	.true. if tanh-shape of boundary relaxation function
nlpwei	.true. if print of boundary relaxation function alfab
nlbduv	.true. if the lateral boundaries are situated at the velocity points

comsta - computed statistics parameters

stps	area mean surface pressure
stq	area mean vertically integrated specific humidity
sts	area mean vertically integrated cloud water
stpe	area mean vertically integrated potential energy
stke	area mean vertically integrated kinetic energy
stte	area mean vertically integrated total energy
cmaxwi	maximum horizontal velocity (m/s)
nmaxwx	gridpoint in the x-direction with maximum horizontal velocity
nmaxwy	gridpoint in the y-direction with maximum horizontal velocity
nmaxwk	vertical level with maximum horizontal velocity
cabsdp	mean absolute pressure tendency (hPa/3hours)

comext - extraction of data to be printed

nextrd	number of gridpoints for print of model level data
nextrx	list of gridpoints in the x-direction, i=1,nextrd
nextry	list of gridpoints in the y-direction, i=1,nextrd

comdif - constants in horizontal diffusion schemes

rprers	reference surface pressure prs
rtemrs	reference temperature Trs
rtemrt	reference temperature Trt
ak4	diffusion coefficient for horizontal diffusion scheme
atcref	list of coefficients for temperature along pseudo pressure levels
ak4lev	list of weight factors for the level dependent horizontal diffusion coefficients for the fourth order scheme. Effective diffusion coefficient is ak4lev*ak4. The dimension is ak4lev(nlev,5) where the first index is the level k=1,nlev and the second index corresponds to t, u, v, q and s. The default value is ak4lev=1..
nlcrf	.true. if correction for horizontal diffusion of temperature and humidity along pseudo pressure levels

comnmi - constants used in implicit normal mode initialization

nitnmi	number of iterations for normal mode initialization
nmodes	number of vertical modes to be initialized
edepth	list of depths for vertical normal modes
gmat	vertical structure matrix G
gmati	inverse vertical structure matrix
pmat	vertical coupling matrix gamma for the auxilliary variable P
pmati	inverse vertical coupling matrix gamma
ffmean	mean value of the Coriolis parameter
rhxumm	inverse mean value of mapfactor in the x-direction
rhyvmm	inverse mean value of mapfactor in the y-direction
nlinmi	.true. if implicit nonlinear normal mode initialization

comsla - constants used in semi-Lagrangian scheme

npbpts	number of extra (passive) boundary lines where the boundary relaxation function alfab=1.0 (the total width of the relaxation zone is npbpts+nbdpts, but the list bdfunc(i) should only be given for i=1,nbdpts as if npbpts=0) nslext type of extrapolation to time n+1/2 in the semi-Lagrangian scheme =1 : f(n+1/2)=f(n) =2 : f(n+1/2)=f(n,n-1) =3 : f(n+1/2)=f(n,n-1,n-2)
nslpqi	number of iterations for calculation of displacement in the semi-Lagrangian scheme
nslinc	type of interpolation at the midpoint in the semi-Lagrangian scheme (not used)
nslind	type of interpolation at the departure point in the semi-Lagrangian scheme =1 : linear =2 : quadratic =3 : cubic =4 : mixed linear/cubic
nslint	list of interpolation types for each iteration, i=1,nslpqi =1 : linear =2 : quadratic =3 : cubic =4 : mixed linear/cubic
nslan	.true. if semi-lagrangian advection
nsl3d	.true. if 3-dimensional semi-Lagrangian advection

combig - dynamics variables

apsx	surface pressure (Pa)
alnpsx	log(surface pressure)
tx	temperature (K)
ux	velocity component in the x-direction (m/s)
vx	velocity component in the y-direction (m/s)
qx	specific humidity (kg/kg)
sx	cloud water content (kg/kg)
edotx	vertical velocity sdot (1/s)
svvarx	extra scalars where x for apsx, alnpsx, ux, vx, qx, sx, edotx and svvarx refers to x=m : previous time step (n-1) x=z : current time step (n) x=p : following time step (n+1) in the eulerian time scheme and in addition x=x : saved values from time step n-2 and extrapolated values to time step n+1/2 in the semi-Lagrangian time scheme

comdtp - physical tendencies saved during longer physical timesteps

dtdtph	tendency of temperature
dqdtph	tendency of specific humidity
dsdtph	tendency of cloud water content
dtsdt	tendency of surface temperature
dswdt	tendency of soil water content
dsndt	tendency of snow depth
dtddt	tendency of deep soil temperature
dsddt	tendency of deep soil water content

comppf - fields in physical parameterization schemes

tsx	surface temperature
tdx	deep surface temperature
swx	soil water content
sdx	deep soil water content
snx	snow depth where x for tsx, tdx, swx, sdx and snx refers to x=m : previous time step (n-1) x=z : current time step (n) x=p : following time step (n+1)
tsc	climatological deep surface temperature
tss	sea surface temperature
swc	climatological deep soil water content
snc	climatological snow depth
rou	roughness length
roc	climatological roughness length
alb	albedo
frl	fraction of land
fri	fraction of ice
aln	longitude
slt	sine(latitude)
clt	cosine(latitude)
prl	accumulated large scale precipitation

prc	accumulated convective precipitation
u10	10-meter velocity in x-direction
v10	10-meter velocity in y-direction
t2m	2-meter temperature
q2m	2-meter specific humidity
totcov	total cloud cover
cucov	cumulus cloud cover
cov2d	2-dimensional cloud cover
cwpath	vertically integrated cloud water content
senf	sensible heat flux
latf	latent heat flux
momf	momentum flux
acsnoc	accumulated convective snow fall
acsnol	accumulated large scale snow fall
asenf	accumulated surface sensible heat flux
alatf	accumulated surface latent heat flux
amomf	accumulated surface momentum flux
aslwr	accumulated surface longwave radiation flux
asswr	accumulated surface shortwave radiation flux
apreta	accumulated total precipitation at model levels

comphc - time independent and statistics variables in physics

phcoef	time independent coefficient in physics
phstat	statistics variables in physics

comdtx - tendency for initialization and 3-dimensional displacements

dudt	tendency of velocity component in x-direction
dvdt	tendency of velocity component in y-direction
dtdt	tendency of temperature
dpdt	tendency of log(surface pressure)
alfa	displacement to departure point in x-direction
beta	displacement to departure point in y-direction
gama	departure point in vertical direction

equivalence (dudt,alfa),(dvdt,beta),(dtdt,gama)

comwrk - work space in dynamics

w	work space array used for several purposes, e.g. in the helmholz solver
wb	workspace for boundary relaxation of physical tendencies

comibv - incore boundary values

parameter statements:

mbdpts	max. total number of points in the boundary relaxation zone inclusive passive boundary lines
bndval	boundary values of ps,u,v,T,q,s and sdot in the boundary relaxation zone for the current boundary data sets

bndvsv	boundary values of extra scalar prognostic variables in the boundary relaxation zone
lcw	indicator of cloud water in the boundary data
lsv	indicator of extra scalars in the boundary data

comput - postprocessing (pp) variables and output history data sets

parameter statements:

jpstr	max. number pp-streams
jpppt	max. number of output times per pp-stream
jplev	max. number of output levels per pp-stream
jpmlf	max. number of output multi level fields per pp-stream
jpslf	max. number of output single level fields per pp-stream
jprefix	max. length of prefix character strings for file names
jsuffix	max. length of suffix character strings for file names
jppstr	list of pointers to current time
nppstr	number of pp-streams
lunlis	unit for input namelist namppp
lundip	unit for direct access file for output
lprint	.true. if print of variables in gridpoint (i,j)=(iprint,jprint)
iprint	gridpoint in x-direction to be printed
jprint	gridpoint in y-direction to be printed
lposton	.true. if post-processing only
lphys	.true. if physics fields available
lomega	.true. if computation of vertical velocity omega
ntimep	list of number of output times
nlevml	list of number of levels for multi level fields
nwmoml	list of multi level field wmo-parameters
nsl	list of number of single level fields
npplon	list of number of output gridpoints in x-direction
npplat	list of number of output gridpoints in y-direction
iminpp	list of first output gridpoint in x-direction
jminpp	list of first output gridpoint in y-direction
lunarc	list of units for archive file (not used)
lunppf	list of units for postprocessed file
ltypml	list of multi level field type (model or pressure levels)
timepp	list of postprocessing times (seconds)
alevml	list of levels for multi level fields
alevsl	list of levels for single level fields
iwmoml	list of multi level field wmo-parameters
ltypsl	list of multi level field types
iwmosl	list of single level field wmo-parameters
arakawa	definition of model grid in Arakawa convention
filnam	list of file names of output file (not used)
lintyp	list of vertical interpolation type in postprocessing
prefix	list of prefixes for postprocessed file
sufix	list of suffixes for postprocessed file
hisnam	prefix for model level history file
hissub	suffix for model level history file

2.3.2 Physical Parameterization

Some preparations of variables and fields are done before entering the physics package. The main routine **gemini** calls **contun**, **conphys**, **tabdef** and **iniphy** before the first call of **phcall**. The routine **contun** is only giving values of tuning parameters in the common block **ctun**, and **tabdef** is computing tables for saturation water vapour pressure and its derivative with respect to temperature, **estab** and **destab**. In the reference version the resolution of these tables are 100 values per degree, this can be changed in the comdeck **escom**. Those tables are used in the physics routines by the inline functions **esat(T)** and **desat(T)**. Note that nearest neighbour interpolation is performed. The routine **conphys** defines values of the physical constants stored in the common block **confys**.

In the forecast model the tendencies due to physical processes are calculated after the explicit tendencies due to advective processes and dynamical adjustment calculated by **dyn**. Then the tendencies due to physics are added to the dynamical ones followed by the semi-implicit adjustment.

The physics calculations are controlled by the subroutine **phcall**, which organizes the input and output fields of the physical package for each timestep. This routine is calling **iniphys** the first timestep, where all local constants and vertical arrays, used by the different physics routines, are calculated. To parallelize the physics calculations the integration area is divided into subareas, each of them containing **norph** gridpoints. The complete calculation of physics for each of these subareas constitutes a 'task' and the administration of this is handled by **phcall**. This means that the full fields are "cut" down to subareas (for full fields new subarea arrays are used, and for one-dimensional fields, only manipulation of start and stop indices is done) and the subroutine **phtask** is called by **phcall** for each task. Routine **phcall** also takes care of the destaggering and staggering of the wind tendencies. The destaggering of the winds is done within **phtask**. After all tasks have been calculated (in parallel) the full field tendencies are recreated within **phcall**.

For each task **phtask** is calling **phys**, which is the actual routine calling the different physics routines. All these routines are called by "cover" routines named a+routine name (i.e. **aradia** etc.) which is allocating work space. This is done with "automatic" arrays for machines with that possibility, and with "parameter" statements for other machines.

The first subroutine called by **phys**, is subroutine **hybrid**, which determines some quantities involving pressure and geopotential for later use in other subroutines.

The pressure between the model layers ('half levels') is determined by means of the two vertical arrays $A_{k+1/2}, B_{k+1/2}$ of the hybrid coordinate system.

$$p_{k+1/2} = A_{k+1/2} + B_{k+1/2} p_s \quad k=0, \dots, NLEV \quad (2.3.2.1)$$

The pressure of the 'full' model layers is computed as the mean value of the nearest 'half levels':

$$p_k = \frac{1}{2}(p_{k+1/2} + p_{k-1/2}) \quad k=1, \dots, NLEV \quad (2.3.2.2)$$

Also the corresponding pressure differences are computed in **hybrid**:

$$\begin{aligned}\Delta p_{k1/2} &= p_{k1} - p_k \quad k=1,\dots,NLEV \\ \Delta p_k &= p_{k1/2} - p_{k-1/2}\end{aligned}\tag{2.3.2.3}$$

Finally the geopotential Φ_k , ($k=1,\dots,nlev$) for 'full levels' is determined according to:

$$\begin{aligned}\Phi_{nlev} &= - R_d T_{vnlev} \ln(\eta_{nlev}) \\ \Phi_k &= \Phi_{k+1} + 0.5 R_d (T_{vk+1} + T_{vk}) \ln(p_{k+1}/p_k)\end{aligned}\tag{2.3.2.4}$$

The next routine to be called from **phys** is the vertical diffusion **vdiff**. Note that it is possible to use "substepping" of the vertical diffusion by setting **ndtvdif** to a smaller value than **ndtphys**, and it is also possible to add the dynamical tendencies to the modelstate ($\tau=1$ values) by setting **nldynvd** to .true. When substepping is used, the tendency of surface temperature is extrapolated to the actual time, while the soil moisture is kept constant during the different calls of **vdiff**.

It is also possible to use longer timesteps in the physics than in the dynamics, except for the call of **vdiff**, which is called every dynamical timestep, in this case. To assure non-negative humidities the subroutine **qnegat** is called after **vdiff** if the rest of the physics is called at greater intervals. The administration of this procedure is handled by **phtask** and **phys**.

The results of **vdiff** are tendencies of temperature, humidity, and wind. Also (for the points where fraction of sea is not unity), the sensible - and latent heat fluxes, together with the derivative of total heat flux with respect to temperature, are computed for transfer to **surf**. The exchange coefficients are also transferred to the argument list of **phys**, to make postprocessing possible (i.e. the near surface variables like 10m wind and 2m temperature and humidity, must first be computed in **anemlv**, which is called by **phtask** via a routine called **srffld**). The virtual temperature, used by **kuo** is also computed.

The next routine called by **phys** is the radiation scheme **radia**, which now doesn't contain computation of cloudcover. This means that, if we don't use a scheme where cloudwater is present, (i.e. if **nlsund** is .false.) some diagnostic cloud cover, based on the humidity, must be computed. In this case we call **cloud** just before the radiation scheme. The output of **radia** is the tendency of temperature, due to both short- and long wave processes, as well as the net radiation at the surface which is transferred to **surf**, and to the argument list of **phys** (for post processing).

Note that the total tendencies are updated after each physical process, which is important when we enter into the condensation processes of the model, which use preliminary values of temperature and humidity.

For the condensation processes we have two possibilities, either the convective and stratiform condensation routine **condens**, developed by H.Sundqvist, or the older convective scheme **kuo** followed by the stratiform scheme **cond**. In the first case we get, in excess of the tendencies of temperature and humidity, also the tendency of cloud water mixing ratio, and diagnostic fields of cloudcover i.e. total cloudcover (totcov), convective cloudcover (cucov) and a two-dimensional interpretation field (cov2d). The accumulated stratiform precipitation **accpr1** and convective precipitation **accprc** are also computed by the new as well as the old condensation routines. Also the precipitation intensities **draindt** and **dsnowdt** are computed for transfer to **surf**.

After the condensation routines we must be sure that the humidity is still positive, and therefore we call **qnegat**, which is redistributing the humidity in the vertical, in case of negative values:

$$(5) \quad q_{adj} = q_k + (q^*_{k-1} - q_{min}) \Delta p_{k-1}/\Delta p_k, \quad q_{min} = 10^{-6}$$

where k is going from 1 to $nlev$. Here q^* means the "unadjusted" value.

The final routine called by **phys** is the surface routine **surf**. It is updating the ground variables (i.e. delivering tendencies) surface temperature, deep surface temperature, surface moisture, deep surface moisture and snow cover. It utilizes some transferred information from the other physical schemes (see above).

The different physical processes also produce values used in the statistics, i.e. the area mean average of the tendencies and also the standard deviations. These arrays (1-D vertical arrays) are handled by **phcall**, since the different tasks can be computed in parallel.

In order to evaluate the performance of the physical parameterization, some statistical calculations are done if the variable **nlstph** is set to true. Area averages of tendencies due to the different physical processes are computed for each task and summed up in **phcall**. Also the square of these tendencies are computed in the same way. The results are printed by the routine **stphys** called by **phcall**.

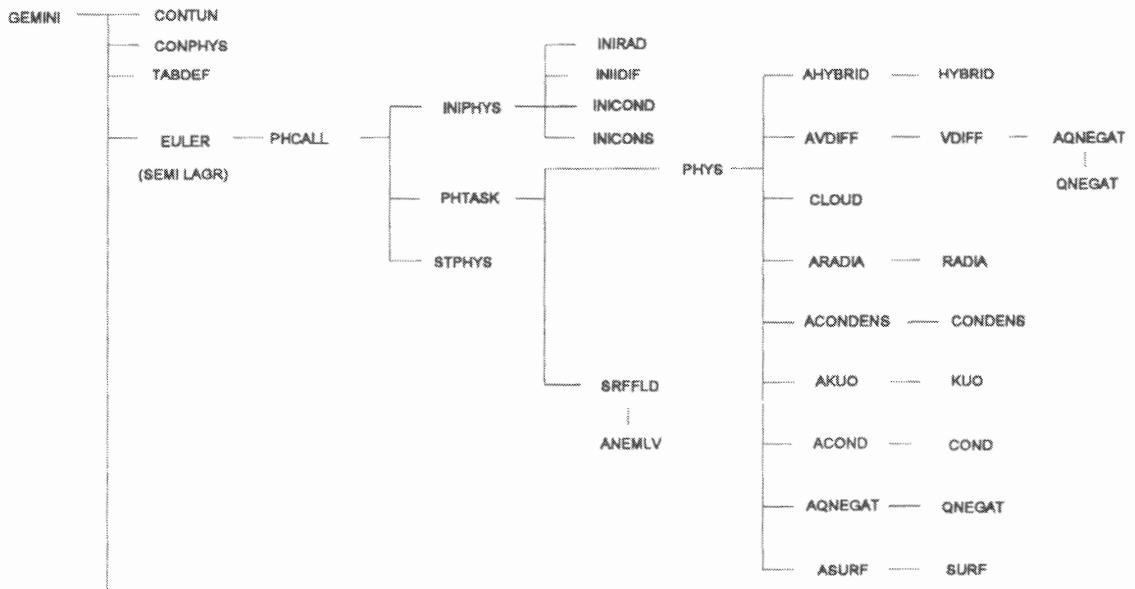


Diagram showing the subroutines involved in the physical parameterization.

Common blocks

confys - physical constants

The commonblock **confys** is containing the necessary physical constants. It is defined in the subroutine **conphys**.

pi	3.1415926...
latvap	latent heat for evaporation of water
rair	specific gas konstant for dry air
cpair	specific heat for dry air at constant pressure
ccpq	factor to multiply the specific humidity, when computing current specific heat at constant pressure ($c_p = c_{pd} * (1 + ccpq * q)$)
epsilo	0.622
gravit	aceleration of gravity
tmelt	melting temperature for ice
lattice	latent heat for melting of ice
rhos	density of the soil
rhoh2o	density of water
solar	solar constant
stebol	stefan bolzmanns constant
carman	von karman constant
rearth	radius of earth
omega	angular velocity of earth

comsrf - surface variables

The common block **comsrf** contains surface variables used in the physics package. This common block is used by the dynamics only (variables are passed to the physics via argument list).

tsm,tsz,tsp	surface temperatures at timestep $\tau-1$, τ and $\tau+1$ respectively
tdm,tdz,tdp	intermediate layer soil temperatures at timestep $\tau-1$, τ and $\tau+1$ respectively
tsc	climatological deep soil layer temperture
tss	climatological sea surface temperature
swm,swz,swp	surface soil water content at timestep $\tau-1$, τ and $\tau+1$ respectively
sdm,sdz,sdp	intermediate layer soil water content at timestep $\tau-1$, τ and $\tau+1$ respectively
swc	climatological deep soil water content
snm,snz,snp	snow depth at timestep $\tau-1$, τ and $\tau+1$ respectively
snc	climatological snowdepth
rou	surface roughness
roc	climatological roughness
alb	albedo
frl	fraction of land
fri	fraction of sea that is iced covered by ice
aln	true longitude
slt	sine(true latitude)
ctl	cosine(true latitude)
prl, prc	accumulated stratiform and convective precipitation respectively
u10, v10	10 meter (interpreted) wind components
t2m	2 meter (interpreted) temperature
q2m	2 meter (interpreted) specific humidity
senf	sensible heat flux at the ground

latf	latent heat flux at the ground
momf	momentum flux at the ground
radf	radiation flux at the ground
draindt	rain intensity at the ground
dsnowdt	snow intensity at the ground
runoff	runoff

comrd - constants for the radiation scheme

The common block **comrd** contains the constants that are used by the radiation scheme. It is defined in the routine **inirad** called by **iniphys**.

pzero	reference pressure
ctuns	tuning constant for clear air cooling, unstable surface layer
cerg	constant used in connection with absorber mass
cdumin	scaled absorber mass minimum
cle1,cle2,cle3	coefficients to calculate longwave clear air cooling rate
ctsta	tuning constant for clear air cooling, stable surface layer
cuk1	coefficient used to define effective absorber mass, shortwave radiation
cdpcl	coefficient used in calcualtion of clear air cooling below clouds
ab0	absorptance constant
re0	reflectance constant
cuk2	coefficient used to define effective absorber mass, shortwave radiation
cbedo, cbeco	coefficients used when computing ground albedo
alsnow	snow albedo constant
csndep	critical snow depth
cn1, cn2	tuning constants used to determine an effective temperature at the ground, longwave radiation
ctcu	coefficient determinening the temperature curvature cooling rate
ctdtrad	cooling rate constant for clear air

comdif - constants for the vertical diffusion scheme

The common block **comdif** contains the constants that are used by the vertical diffusion scheme. It is defined in the routine **inidif** called by **iniphys**.

cueps	security constant
cams3, cms1,cchrn	constants used for flux calculations at the surface
qc, qb	constants used in exchange coefficients
sncv	inverse of critical snow depth
qsgn	constant used for flux calculations at the surface

econs - constants for the sundqvist scheme

The common block **econs** contains the constants that are used by the sundqvist condensation scheme. It is defined in the routine **inicons** called by **iniphys**.

hdl	latent heat of melting
ht273	Temperature of 273 K
he273	saturation vapour pressure at 273 K
tvirtc	coefficient used when computing virtual temperature
stpevp	coefficient for evaporation from stratiform precipitation

u00max	maximum allowable value of modified hu00
hu00	threshold relative humidity for stratiform condensation
aecon, conae	constant used when computing equivalent potential temperature
cfreez	constant used to increase conversion rate for temp lower than 273 K
coales	constant used to increase conversion rate due to precipitation coming in from above
hccu	conversion rate of cloud droplets to precipitation in a convective cloud
hcunrm	constant used in determining the cloud cover
hmrcu, hmrst	cloud water mixing ratio at which conversion becomes efficient in convective and stratiform cloud respectively
htaucu	characteristic time used in convective cloud cover scheme
hp0	reference pressure (=100 kpa)
cbfeff	coefficient used in the parameterization of the bergeron-findeisen effect
hvterm, hvsnow	terminal fall velocities for rain and snow/ice respectively
hk melt	coefficient for melting of ice in precipitation
tanvil	critical temperature for formation of stratiform anvil cloud
hcst	conversion rate of cloud droplets to precipitation in stratiform cloud
pmoist	coefficient used in the partitioning between heating and moistening in the convective parameterization
prbice(1750)	table containig the probability for existance of ice crystals
pfeff(1750)	table used in the paramerization of the bergeron-findeisen effect
hdewi(1750)	table containing difference in saturation vapour pressure over water and ice
hmroft(1750)	table containing function used to multiply hmrcu and hmrst for temperatures lower than 273 K
dttabl	coefficient used in the connection with the tables above

2.4 Data organization

The model reads an input start data set (in ASIMOF/grib format) which contains the dynamic prognostic variables; surface pressure, temperature, specific humidity and horizontal velocity components. The cloud liquid water content is either read from the start data set or initialized to zero if not present in the input file. The start data set also contains surface geopotential, sea surface temperature, snow depth, ice cover and all the climatological fields which are needed in the physics. The boundary data sets are read at the time when they are needed. The first type of output files are history files at model levels (in grib format) containing the same fields (all model variables) as the start data set (the six hour forecast is used as guess field for the analysis) and in addition some prognostic/diagnostic fields from the physics: 2 meter temperature and specific humidity, 10 meter velocity, accumulated large scale and convective precipitations and cloud cover. The second type of output files, the postprocessed files, contains either fields interpolated to pressure levels or fields extracted from model levels and the surface fields. The output frequency of the model level fields is specified in namelist namrun, while the content and output frequency of postprocessed files are specified in namelists namps and namppp. The names of the input and output files are built internally using the standard hirlam filenaming conventions, which consists of a 2 character prefix, the date/time group and a 2 character suffix. The filenames are linked to the their appropriate fortran file units specified in namelists namdev and namppp. The unit numbers for the input data sets and the output model level data sets in namrun namdev are

lusdfi	unit for start data set
lubfi1	unit for first boundary data set (= lusdfi)
lubfi2	unit for second boundary data set. The following boundary dat sets are assumed to be lubfi2+1, lubfi2+2, ...
luhfi1	unit for first output model level data set. The following output data sets are assumed to be luhfi1+1, luhfi1+2, ...

A third type of output data are the timeseries data in geographical points. The geographical points and output frequency is given in input namelist namtsf.

Table 2.4.1. The input namelists to the forecast model

namdev - i/o-units and file names

lunaml	unit for namelist namrun
lusdfi	unit for start data set
lubfi1	unit for first boundary data set (=lusdfi)
lubfi2	unit for second boundary data set (following boundary data sets are assumed to be lubfi2+1, lubfi2+2,...)
luhfi1	unit for first output model level data set (following output data sets are assumed to be luhfi1+1, luhfi1+2,...)
lundir	unit for direct access grib file (work file)
luntun	unit for namelist namtun
fgtype	prefix for start data set (lusdfi)
fgsub	sufix for start data set (lusdfi)
bdtype	prefix for boundary data sets (lubfi2, lubfi2+1,...)
bdsb	sufix for boundary data sets (lubfi2, lubfi2+1,...)

namrun - input steering parameters

nlon	number of gridpoints in the x-direction
nlat	number of gridpoints in the y-direction
nlev	number of full model levels
nstop	number of time steps
ndtime	time step in seconds
ndtphys	time step for physics
ndtvdif	time step for vertical diffusion in physics
ntimesu	spinup time in seconds (the physics is called every dynamic time step during spinup time)
nexp	switch for experiment name =0 : experiment name taken from start data set =1 : experiment name given as input (last input line)
nlgrb	.true. if grib format for input/output data sets
nwtme	list of time steps for output model level data sets
nextrd	number of gridpoints for print of model level data
nextrx	list of gridpoints in the x-direction, i=1,nextrd
nextry	list of gridpoints in the y-direction, i=1,nextrd
nbdpts	number of gridpoints in the boundary relaxation zone for the boundary relaxation function alfab. alfab=1.0 in gridpoint i=1+npbpts alfab=0.0 in gridpoint i=nbdpts+1+npbpts
nltnh	.true. if tanh-shape of boundary relaxation function
bdfunc	list of values for the boundary relaxation function alfab if nltnh=.false. in the interval [1.0,...,0.0]> bdfunc=1.0 in gridpoint i=1+npbpts bdfunc=0.0 in gridpoint i=nbdpts+1+npbpts
npbpts	number of extra (passive) boundary lines where the boundary relaxation function alfab=1.0 (The total width of the relaxation zone is npbpts+nbdpts, but the list bdfunc(i) should only be given for i=1,nbdpts as if npbpts=0)
nlpwei	.true. if print of boundary relaxation function alfab
nlinmi	.true. if implicit nonlinear normal mode initialization
nmodes	number of vertical modes to be initialized

nitnmi	number of iterations for normal mode initialization
neul	.true. if Eulerian time scheme
nlsimp	.true. if semi-implicit scheme
nslan	.true. if semi-Lagrangian advection
nlstat	.true. if computation and print of statistics
nlstph	.true. if statistics computations in physics
nltvir	.true. if virtual temperature in dynamics
nlhumc	.true. if check of critical humidity values for input data
nlphys	.true. if physical parameterization
nlvdif	.true. if vertical diffusion scheme in physics
nrrad	.true. if radiation scheme in physics
nlkuo	.true. if Kuo convection scheme in physics
nlcond	.true. if large scale condensation scheme in physics
nlsund	.true. if Sundqvist condensation scheme in physics
nlsurf	.true. if surface parameterization scheme in physics
nhorph	number of gridpoints in each sub-area for the physics (the loop length in each physics routine is nhorph and the physics routines are multitasked for all sub-areas)
nldynvd	.true. if dynamic tendency used in the vertical diffusion scheme
eps	coefficient for the Asselin time-filter and time-filter for physics surface fields
epsg	coefficient for the gravity wave damper in the semi-Lagrangian scheme
epsn	coefficient for time-filter in the semi-Lagrangian 3-time-level extrapolation scheme (nslext=3)
nlhdif	.true. if horizontal diffusion scheme in dynamics
ak4	diffusion coefficient for horizontal diffusion scheme
nlterf	.true. if correction for horizontal diffusion of temperature and humidity along pseudo pressure levels
ak4lev	list of weight factors for level dependent horizontal diffusion coefficients for the fourth order scheme. Effective diffusion coefficient is ak4lev*ak4. The dimension is ak4lev(nlev,5) where the first index is the level k=1,nlev and the second index corresponds to t, u, v, q and s. The default value is ak4lev=1..
nsl13d	.true. if 3-dimensional semi-Lagrangian advection
nslpqi	number of iterations for calculation of displacement in the semi-Lagrangian scheme
nslint	list of interpolation types for each iteration, i=1,nslpqi =1 : linear =2 : quadratic =3 : cubic =4 : mixed linear/cubic
nslinc	type of interpolation at the midpoint in the semi-Lagrangian scheme (not used)
nslind	type of interpolation at the departure point in the semi-Lagrangian scheme =1 : linear =2 : quadratic =3 : cubic =4 : mixed linear/cubic
nslext	type of extrapolation to time $n+1/2$ in the semi-Lagrangian scheme =1 : $f(n+1/2)=f(n)$ =2 : $f(n+1/2)=f(n,n-1)$ =3 : $f(n+1/2)=f(n,n-1,n-2)$
nlbduv	.true. if laterat the boundaries are situated at the velocity points
ndify	choise of implicit horizontal diffusion scheme of variables y where ndify = 2 : second order scheme ndify = 4 : fourth order scheme ndify = 6 : sixth order scheme and

$y = u$: horizontal velocity component u
$y = v$: horizontal velocity component v
$y = t$: temperature
$y = q$: specific humidity
$y = s$: specific cloud water
$y = x$: extra scalar prognostic variables
cdify	list of levels dependent diffusion coefficient for the implicit scheme ofr variable y
nldify	.true. if implicit horizontal diffusion of variables y
nsvar	number of extra scalar prognostic variables (default=0)
nwmovs	list of wmo code numbers for extra scalars (the data input routines will search for these wmo codes in the input data sets)
sit0	reference temperature in the semi-implicit and normal mode initialization schemes (default = 300K)
sip0	reference pressure in the semi-implicit and normal model initialization schemes (default = 80000 Pa)

nampos - main post processing parameters

nppstr	number of pp-streams
lunlis	unit for input namelist namppp
lundip	unit for direct access file for output
lposton	.true. if post-processing only
lphys	.true. if physics variables are available
lomega	.true. if computation of vertical velocity omega
lprint	.true. if print of variables in gridpoint (i,j)=(iprint,jprint)
iprint	gridpoint in the x-direction to be printed
jprint	gridpoint in the y-direction to be printed

namppp - post processing stream parameters

ntimepp	number of output times
nlevmlp	number of levels for multi level fields
nwmomlp	list of multi level field wmo-parameters
nslp	number of single level fields
npplonp	number of output gridpoints in the x-direction
npplatp	number of output gridpoints in the y-direction
iminppp	first output gridpoint in the x-direction
jminppp	first output gridpoint in the y-direction
lunarcp	unit for archive file (not used)
lunppfp	unit for postprocessed file
lintypp	vertical interpolation type in postprocessing
timeppp	list of postprocessing times (seconds)
ltypmlp	multi level field type (model or pressure levels)
alevmlp	list of levels for multi level fields
iwmomlp	list of multi level field wmo-parameters
ltypslp	list of single level field types
iwmoslp	list of single level field wmo-parameters
alevslp	list of levels for single level fields
prefixp	prefix for postprocessed file
sufixp	suffix for postprocessed file
hisnam	prefix for model level history file
hissub	suffix for model level history file

namtsf - output timeseries files in gridpoints

ifreqa	sample frequency of sl fields
zlon _a	list of output ml longitude positions (decimal degrees with pole at 90 degrees)
zlat _a	list of output ml latitude positions (decimal degrees with pole at 90 degrees)
ifreqb	sample frequency of sl fields
zlon _b	list of output sl longitude positions (decimal degrees with pole at 90 degrees)
zlat _b	list of output sl latitude positions (decimal degrees with pole at 90 degrees)

3. INITIALIZATION

3.1 The implicit normal mode initialization scheme

3.1.1 Introduction

For the purpose of initialization of weather prediction models, many scheme will meet the main requirement of such schemes. That is; the forecast should already from the start of the integration be free from the high frequency oscillations with unrealistically high amplitudes. The nonlinear normal mode initialization (NMI) scheme of Machenhauer (1977) gives the best possibility to separate the fast (gravity) modes from the slow (Rossby) modes, and it controls the unwanted gravity part by setting its initial tendency to zero. When the NMI scheme is formulated in terms of spherical coordinates for a global or hemispheric spectral model the normal modes are easily derived from a linear system of equations without further approximations. In addition they have the same spatial structure as the basis functions which are used in the spectral representation. On the other hand, when the NMI scheme in the explicit form is developed in a regional model some simplifications must be introduced in the separation of the equations due to the variation of the Coriolis term and the map factors. In addition a suitable spatial representation of the normal modes is needed within the limited area geometry. It has been demonstrated by Temperton (1988) that NMI can be expressed without explicit knowledge of the models normal mode solutions. The implicit form of the NMI, which is algebraically equivalent with the conventional NMI scheme is particularly useful for limited area models where the horizontal normal modes are not readily found.

The implicit NMI scheme is in the following derived for the HIRLAM model. The method follows closely the ideas described by Temperton with some simplification and changes in order to facilitate the implementation. In Tempertons scheme the equation are expressed in terms of vorticity and divergence. Then the initialization increments for the u and v horizontal velocity components can be derived from stream function and velocity potential by use of the Helmholtz decomposition theorem. In the present scheme the resulting equations for the initialization increments are expressed directly in terms of u and v, which are prognostic variables in the model. In order to achieve this both the metric coefficients in the derivatives and the Coriolis factor must be divided into a mean value and a deviation. The area mean value is used in the linear terms and will implicitly determine the structure of the model normal mode solutions. The deviation is absorbed into the remaining nonlinear terms. These simplifications are only reasonable when the scheme is applied in a limited area model which is situated well outside the tropical region. The use of constant Coriolis parameter and metric coefficients have in addition the comfortable consequence that the resulting Helmholtz equations allow utilization of subroutines which already have been implemented for use in the semi-implicit time scheme of the model. Futhermore there is no need for storage of the horizontal normal modes and the scheme is computationally efficient. The scheme has been implemented as a part of the forecast model and expressed in terms of general orthogonal coordinates as the model equations themselves.

3.1.2 Linear equation system

The dynamic prognostic variables in the model are the horizontal velocity components u and v, temperature T and log(surface pressure) lnp. From u and v the expressions for the divergence d and the relative vorticity ζ are

$$d = \frac{1}{h_x h_y} [\frac{\partial}{\partial x}(h_y u) + \frac{\partial}{\partial y}(h_x v)] \quad (3.1.2.1)$$

$$\zeta = \frac{1}{h_x h_y} [\frac{\partial}{\partial x}(h_y v) - \frac{\partial}{\partial y}(h_x u)]$$

where h_x and h_y are the metric coefficients in the x- and y-direction, respectively and a is the radius of the earth. These expressions are valid for general orthogonal coordinates. However, in order to derive a separable system of equations for the model formulation, h_x and h_y are in the following replaced with their respective area mean values, H_x and H_y . In addition the Coriolis factor f will be replaced with its area mean value F . With these assumptions the set of linearized equations becomes

$$\begin{aligned} \frac{\partial u}{\partial t} &= Fv - \frac{1}{H_x} \frac{\partial P}{\partial x} + N_u \\ \frac{\partial v}{\partial t} &= -Fu - \frac{1}{H_y} \frac{\partial P}{\partial y} + N_v \\ \frac{\partial T}{\partial t} &= -\tau d + N_T \\ \frac{\partial \ln p_s}{\partial t} &= -v^T d + N_{bps} \end{aligned} \quad (3.1.2.2)$$

where τ and v depend on the vertical discretization and N_u , N_v , N_T and N_{bps} are the nonlinear terms. The deviations from the linearized state which is defined by the reference pressure $p_r=800\text{hPa}$, the reference temperature $T_r=300\text{K}$, H_x , H_y and F are absorbed into the nonlinear terms. However, there is no need to compute the deviation explicitly, as shown in the following.

The thermodynamic equation and the continuity equation are combined into one prognostic equation for the auxiliary variable P , where the linearized part of the pressure gradient term in the momentum equation is the gradient of P . It is defined as

$$P = \gamma T + R_d T_r \ln p_s \quad (3.1.2.3)$$

where γ depends on the vertical discretization. The prognostic equation for P is then

$$\frac{\partial P}{\partial t} = -Gd + N_P \quad (3.1.2.4)$$

where $G=\gamma\tau+R_d T_r v^T$ is the vertical structure matrix (the definition of γ , τ and v is given in the documentation of the semi-implicit time scheme) and N_P is the nonlinear terms.

The vertical normal modes of the model are used for transformation of the three dimensional primitive equations into nlev shallow water equations, where nlev is the number of model levels carrying the prognostic variables u , v and T . They are derived in the same way as in the semi-implicit time scheme of the model. The linear terms in this system can be vertically decoupled if they are pre-multiplied with the matrix E^{-1} , where E consists of the eigen vectors of the G matrix.

Then $E^{-1}GE$ is a diagonal matrix with the corresponding eigenvalues $c^2=gH$ as elements, where g is the acceleration due to gravity and H is the equivalent depth. In terms of vertical normal modes the system of equation becomes

$$\begin{aligned}\frac{\partial \tilde{u}}{\partial t} &= F\tilde{v} - \frac{1}{H_x} \frac{\partial \tilde{P}}{\partial x} + \tilde{N}_u \\ \frac{\partial \tilde{v}}{\partial t} &= -F\tilde{u} - \frac{1}{H_y} \frac{\partial \tilde{P}}{\partial y} + \tilde{N}_v \\ \frac{\partial \tilde{P}}{\partial t} &= -c^2 \tilde{d} + \tilde{N}_p\end{aligned}\quad (3.1.2.5)$$

where the tilde denotes the vertically transformed field.

The horizontal normal modes in these shallow water equations could be found by setting the nonlinear terms to zero. This is equivalent to linearizing the equations about a basic state of rest with constant height H and constant Coriolis parameter F . The use of constant f is not a serious simplification for application of the NMI procedure in the this model, since it is a regional model and operates well outside the tropical region.

3.1.3 Implicit NMI scheme

The horizontal normal modes of the model are classified as slow (Rossby) modes and fast (gravity) modes on the basis of their frequencies. In the conventional NMI a procedure is needed to separate the variable X_o into its two components:

$$X_o = X_R + X_G \quad (3.1.3.1)$$

were X_R and X_G correspond to the slow and fast modes respectively. The balanced state in Machenhauer's first order scheme (where the nonlinear terms are assumed independent of time) is found by setting the time tendencies of the fast gravity modes in Eq. (3.1.2.5) to zero. Since the nonlinear terms are functions of the fast components themselves, the system must be solved iteratively. Each iteration of the conventional NMI scheme can in the equivalent implicit form be written as

$$\begin{aligned}-F\Delta \tilde{v} + \frac{1}{H_x} \frac{\partial \Delta \tilde{P}}{\partial x} + (\delta \tilde{v})_G \\ F\Delta \tilde{u} + \frac{1}{H_y} \frac{\partial \Delta \tilde{P}}{\partial y} + (\delta \tilde{u})_G \\ c^2 \Delta \tilde{d} + (\delta \tilde{d})_G\end{aligned}\quad (3.1.3.2)$$

Here in symbolic form, ΔX is the initialization increments to the variable X and $(\delta, X)_o$ is the time tendency obtained by running the model one forward time step. In order to express the equations in terms of the total time tendency, $(\delta, X)_o$, including fast and slow modes, the following useful properties of the normal modes are utilized:

Property 1: The slow (Rossby) modes are geostrophic and non-divergent:

$$F\tilde{u}_R = -\frac{1}{H_y} \frac{\partial \tilde{P}_R}{\partial y}, \quad F\tilde{v}_R = \frac{1}{H_x} \frac{\partial \tilde{P}_R}{\partial x} \quad (3.1.3.3)$$

Property 2: The fast (gravity) modes have zero linearized potential vorticity:

$$c^2 \tilde{\zeta}_G = F \tilde{P}_G \quad (3.1.3.4)$$

Consider first the momentum equations in (3.1.3.2). By use the decomposition (3.1.3.1), property 1 and the last equation in (3.1.3.2) the right hand sides can be written as

$$\begin{aligned} (\delta \tilde{\mu})_G &= (\delta \tilde{\mu})_o + \frac{1}{FH_y} \frac{\partial}{\partial y} (\delta \tilde{P})_o - \frac{c^2}{FH_y} \frac{\partial \Delta \tilde{d}}{\partial y} \\ (\delta \tilde{v})_G &= (\delta \tilde{v})_o - \frac{1}{FH_x} \frac{\partial}{\partial x} (\delta \tilde{P})_o + \frac{c^2}{FH_x} \frac{\partial \Delta \tilde{d}}{\partial x} \end{aligned} \quad (3.1.3.5)$$

Since the change during the initialization should project entirely on the fast modes, property 2 implies that

$$\Delta \tilde{P} = \frac{c^2}{F} \Delta \tilde{\zeta} \quad (3.1.3.6)$$

The momentum equations in (3.1.3.2) can then be written using (3.1.3.5) in the equivalent form

$$\begin{aligned} \Delta \tilde{u} &+ \frac{c^2}{F^2} \left[\frac{1}{H_y} \frac{\partial \Delta \tilde{\zeta}}{\partial y} - \frac{1}{H_x} \frac{\partial \Delta \tilde{d}}{\partial x} \right] = \frac{1}{F} (\delta \tilde{v})_o - \frac{1}{F^2 H_x} \frac{\partial}{\partial x} (\delta \tilde{P})_o \\ \Delta \tilde{v} &- \frac{c^2}{F^2} \left[\frac{1}{H_x} \frac{\partial \Delta \tilde{\zeta}}{\partial x} + \frac{1}{H_y} \frac{\partial \Delta \tilde{d}}{\partial y} \right] = -\frac{1}{F} (\delta \tilde{\mu})_o - \frac{1}{F^2 H_y} \frac{\partial}{\partial y} (\delta \tilde{P})_o \end{aligned} \quad (3.1.3.7)$$

This leads to the following Helmholtz equations for the increments of u and v expressed in terms of the computed tendencies:

$$\begin{aligned} (1 - \frac{c^2}{F^2} \nabla^2) \Delta \tilde{u} &= \frac{1}{F} (\delta \tilde{v})_o - \frac{1}{F^2 H_x} \frac{\partial}{\partial x} (\delta \tilde{P})_o \\ (1 - \frac{c^2}{F^2} \nabla^2) \Delta \tilde{v} &= -\frac{1}{F} (\delta \tilde{\mu})_o - \frac{1}{F^2 H_y} \frac{\partial}{\partial y} (\delta \tilde{P})_o \end{aligned} \quad (3.1.3.8)$$

(The last step going from Eq. (3.1.3.7) to Eq. (3.1.3.8) is the only place in the derivations which calls for the use of constant metric coefficients.)

Consider then the last equation in (3.1.3.2) for the increments of P . Eq. (3.1.3.6) could be used directly to compute the changes to P from the computed increments of u and v . However, the

variables are carried in the Arakawa C-grid. Consequently the velocity components must be averaged in the computation of the vorticity in the P-points. Therefore an alternative equation is used. From (3.1.3.2) the corresponding equation involving the increments of the relative vorticity is derived:

$$F \Delta \tilde{\zeta} - \nabla^2 \Delta \tilde{P} = -(\delta_t \tilde{\rho})_G \quad (3.1.3.9)$$

The tendency of the divergence projected on the slow modes is zero and after substitution of (3.1.3.6) the resulting equation is

$$(1 - \frac{c^2}{F^2} \nabla^2) \Delta \tilde{P} = -\frac{c^2}{F^2} (\delta_t \rho)_o \quad (3.1.3.10)$$

The equations (3.1.3.8) and (3.1.3.10) form the three Helmholtz equations which have to be solved for each equivalent depth H. The changes to u, v and P can then be found by vertical transformation of the increments (pre-multiplication of the E-matrix). The total algorithm implies that the changes to u, v and P are derived directly from the total tendencies $(\delta_t u)_o$, $(\delta_t v)_o$ and $(\delta_t P)_o$. The increments to P must be divided into increments to T and $\ln p_s$. This is done in the following way:

$$\begin{aligned} \Delta \ln p_s &= \text{yepsilon}^T G^{-1} \Delta P \\ \Delta T &= \gamma^{-1} (\Delta P - R_d T_r \Delta \ln p_s) \end{aligned} \quad (3.1.3.11)$$

3.1.4 Boundary values

The Helmholtz equations are solved in the points carrying their respective variables, i.e. the equation for u is solved in u-points. They are solved by a direct solver which assumes zero boundary values. Looking at the right hand sides of Eq. (3.1.3.8) and (3.1.3.10) this corresponds to geostrophically balanced boundary values. At the first point inside the boundary some values must be specified for the velocity tendencies in (3.1.3.8) due to the computations in the limited area grid. These are given from the external boundary fields X_b as

$$(\delta_t X)_o = \frac{1}{\Delta t} [X_b(t + \Delta t) - X_b(t)] \quad (3.1.4.1)$$

The initialization increments after each iteration m for $X=(u,v,T,\ln p_s)$ are added to the initial fields with a reduced weight along the boundaries, according to the equation

$$X^{(m+1)} = X^{(m)} + (1 - \alpha_b) \Delta X^{(m)} \quad (3.1.4.2)$$

where α_b is the boundary relaxation function. The functional form of α_b during the initialization step is a cosine shape. The width of the relaxation zone is smaller (4 grid distances) than usually specified for the subsequent forecast in order to initialize the fields also close the lateral boundaries.

References

- Machenhauer, B., 1977: On the dynamics of gravity oscillations in a shallow water model with applications to normal mode initialization. *Beitr. Atmos. Phys.*, **50**, 253-271.
- Temperton, C., 1988: Implicit normal mode initialization. *Mon. Wea. Rev.*, **116**, 1013-1031.
- Temperton, C., 1989: Implicit normal mode initialization for spectral models. *Mon. Wea. Rev.*, **117**, 436-451.
- Temperton, C., and D. L. Williamson, 1981: Normal mode initialization for a multilevel grid point model. Part I. Linear aspects. *Mon. Wea. Rev.*, **109**, 729-743.
- Temperton, C., and M. Roch, 1991: Implicit normal mode initialization for an operational regional model. *Mon. Wea. Rev.*, **119**, 667-677.
- Williamson, D. L., and C. Temperton, 1981: Normal mode initialization for a multilevel grid point model. Part II: Nonlinear aspects. *Mon. Wea. Rev.*, **109**, 744-757.

3.2 Computer code organization

The normal mode initialization scheme is coded as a part of the forecast model and the initialization is performed in the same step as the forecast. This is done by calling the main routine **gemini** twice (initialization and forecast) from the main program **hlprog** with different input steering parameters. The initialized data are written to the model level history file, which is read in as the initial condition for the forecast in the second call to **gemini**. The main initialization routine, **normod**, is called in **gemini** after the setup for the run and just before the timestep loop. **Normod** is called in a loop for each iteration in the initialization procedure. After initialization, timestep 0 is performed in order to create the data for the output model level or postprocessed files.

A calling tree for the dynamic part of the forecast model is given in Section 2.3 (Table 2.3.1) and the main routines for initialization is displayed in table 3.2.1. The procedure is organized as follows:

- (a) Routine **dyn** computes the explicit time-tendency of the dynamic prognostic variables; surface pressure, temperature and horizontal velocity components.
- (b) The tendencies are stored for use in the right hand sides of the Helmholtz equation.
- (c) An explicit forward timestepping is done followed by boundary relaxation in routine **bndrel**. The boundary values are used in routine **delnmi** in order to compute tendencies at the boundary points not given from routine **dyn**.
- (d) Routine **impnms** controls computation and addition of initialization increments for surface pressure, temperature and horizontal velocity components. Routines **statis** and **prstat** are called in order to compute and print statistics before and after initialization.
- Routine **delnmi**, called from routine **impnms**, computes the right hand side of the three Helmholtz equations for the horizontal velocity components and the auxiliary pressure variable, P. Routine **hhsolx** is called for the solution of these equations. Routine **matmul** performs the vertical decoupling and coupling of fields between physical space and vertical normal modes. The increments of P is distributed into increments of surface pressure and temperature.
- The initialization increments are then added to the uninitialized fields and the fields are prepared for the next iteration.

Table 3.2.1. Algorithm and calling tree for the initialization

```

<iteration loop:>
do iter=1,nitnmi
    normod - normal mode initialization scheme
        dyn - explicit dynamical tendency
        <save dynamic time tendency>
        <explicit forward time stepping>
        extend - extend data array for computations of lnp
        bndrel - boundary relaxation
        impnms - implicit normal mode schemes
        <compute statistics before initialization:>
            statis - statistics computations
            prstat - print output from statistics computations
        delnmi - routine for implicit nonlinear normal mode schemes
            matmul - matrix multiplication (vertical decoupling)
            trimat - triangular matrix for helmholtz solver
            hhsolx - helmholtz-solver
                fft44 - fast fourier sine-transformation
                vpassm - fast fourier sine-transformation
                fft44a - fast fourier sine-transformation
                fft44b - fast fourier sine-transformation
            matmul - matrix multiplication (vertical coupling)
        <compute statistics after initialization:>
            statis - statistics computations
            prstat - print output from statistics computations
            statis - computation average tendency of surface pressure
            <print average tendency of surface pressure>
            <preparations for next iteration>
enddo

```

3.3 Data organization

The input and output datasets used during the initialization are defined exactly as the files for the forecast. The scheme needs a start dataset, a boundary dataset for computation of tendencies close to the boundaries. The main output data set is the model level history file at the initial time, which is used as the initial condition for the subsequent forecast. In addition it is possible to define postprocessed output files.

The input namelists are the same as for the forecast. See section 2.4 for a similar more detailed description for the forecast model Table 3.3.1 displays the main variables in the input namelist namrun used for the initialization scheme.

Table 3.3.1. The input namelists namrun for initialization

namrun - input steering parameters

nlon	number of gridpoints in x-direction
nlat	number of gridpoints in y-direction
nlev	number of model full levels
nstop	number of time steps
ndtime	time step in seconds
nexp	switch for experiment name =0 : experiment name taken from start data set =1 : experiment name given as input (last input line)
nwtime	list of time steps for output model level data sets
nexrd	number of gridpoints for print of model level data
nexrx	list of gridpoints in x-direction, i=1,nexrd
nexry	list of gridpoints in y-direction, i=1,nexrd
nbdpts	number of gridpoints in the boundary relaxation zone for the boundary relaxation function alfab. alfab=1.0 in gridpoint i=1 alfab=0.0 in gridpoint i=nbdpts+1
nltnanh	.true. if tanh-shape of boundary relaxation function
nlpwei	.true. if print of boundary relaxation function alfab
nlinmi	.true. if implicit nonlinear normal mode initialization
nmodes	number of vertical modes to be initialized
nitnmi	number of iterations for normal mode initialization
nleul	.true. if eulerian time scheme
nlsimp	.true. if semi-implicit scheme
nlstat	.true. if computation and print of statistics
nltvir	.true. if virtual temperature in dynamics
nlhumc	.true. if check of critical humidity values for input data
nlphys	.true. if physical parameterization
eps	coefficient for asselin time-filter and time-filter for physics surface fields
nlhdif	.true. if horizontal diffusion scheme in dynamics

4. THE ANALYSIS SYSTEM

The HIRLAM Level 1 analysis system consists of the scheme for analysis of wind, mass and humidity fields, based on the ECMWF analysis scheme, and the scheme for analysis of sea surface temperature and ice coverage, based on a successive correction method. These two analysis schemes are described below in sections 4.1 and 4.2 respectively.

4.1 Analysis of wind, mass and humidity fields

The HIRLAM Level 1 analysis scheme for wind, mass and humidity fields is a limited area version of the ECMWF analysis scheme.

The following ECMWF reports are relevant for the HIRLAM analysis scheme:

- 1 Meteorological Bulletin 1.5/1
Research Manual 1.
ECMWF Data Assimilation Scientific Documentation
Edited by P.Lönnberg and D.Shaw
2. Meteorological Bulletin 4.0/1
Research Manual 5:
ECMWF Data Assimilation Program Documentation
Edited by J.Haseler
3. Tech. Note 23.3
Format of the analysis observation file
J.Martellet (1978)

A general introduction to the ECMWF analysis scheme and to HIRLAM adaptations for higher resolutions is given in section 4.1.1. More detailed descriptions of HIRLAM adaptations and auxiliary software are given as follows:

- (section 4.1.2) Adaptations from the global to a local area version, on a possibly non-standard projection.
- (section 4.1.3) Adaptations to the HIRLAM file structure
- (section 4.1.4) Adaptations to the setting of file unit numbers.
- (section 4.1.5) Introduction of look-up tables for horizontal correlation functions.
- (section 4.1.6) Adaptations to saturation water vapour pressure calculations.
- (section 4.1.7) Modifications to the auxiliary job to sort the observations.
- (section 4.1.8) Auxiliary software to de-stagger wind component first guess fields and to stagger wind component analysis increment fields.

(section 4.1.9)

Auxiliary software to create analysis coefficient files to be used by the HIRLAM analysis scheme

For descriptions of analysis algorithms and program documentation including details of e.g. file formats is referred to the ECMWF publications mentioned above.

4.1.1 The ECMWF analysis scheme and HIRLAM modifications - an introduction

The basic analysis technique of the ECMWF scheme is optimum interpolation (Eliassen (1954), Gandin (1963)) extended to multivariate three-dimensional interpolation of observed deviations from forecast first guess fields (Lorenc (1981)). Analyzed parameters are geopotential, wind components and relative humidity. Analysis increments are evaluated directly on the forecast model levels.

Geopotential and wind components are analyzed simultaneously with an assumption of near linear geostrophic balance between the rotational part of the wind analysis increment field and the geopotential analysis increment. The incremental streamfunctions and geopotentials are assumed to be correlated with a factor 0.95° north of 30 N, 0.0 at the equator and with a gradual transition in the zone between. Relative humidity is analyzed univariately. Three-dimensional correlations of true minus forecast first guess values are modeled by assuming them to be products of horizontal and vertical correlations. In order to simplify the use of non-standard level data and the evaluation of analysis increments at the forecast model levels, the vertical correlations are modeled by fitting analytic functions to empirically determined vertical correlations. Pressure is used as vertical coordinate for the analysis. Isotropic correlation models for geopotential, stream function, velocity potential and relative humidity are used in the horizontal. Series of zero order Bessel functions, fitted to empirical correlations from time series of historical ECMWF forecast errors (Hollingsworth and Lennberg (1986)), are used to represent these isotropic horizontal correlations. The series of zero order Bessel functions include 9 terms in order to represent also smaller scale features of interest for high resolution analysis. As regards the variance of the wind field first guess errors, 90% of the variance is assumed to be non-divergent while 10% is assumed to be divergent. Generally, the variance of forecast first guess errors is obtained from the estimated variance of the analysis errors of the previous analysis cycle by assuming a linear growth in time towards the climate error variance over a specified time period. This time period has been set to 36 hours in HIRLAM experiments and tests so far, while ECMWF uses a corresponding time period of 144 hours. This increase of assumed error growth rate will result in increased assumed variances of first guess errors and consequently in a closer fit of the analysis fields to the observations for the HIRLAM analysis scheme compared to the operational ECMWF analysis scheme. Intuitively, this is what is required for higher resolution analysis, but a more rigorous motivation based on statistics of forecast errors certainly is needed.

A unique feature of the ECMWF analysis scheme is that all grid points within three-dimensional volumes (analysis boxes) are analyzed together in the sense that the same observed values are utilized to influence all grid point values within these sub-volumes. This "box" data selection scheme makes it possible to use much more observed data values to influence each grid point value than in analysis schemes based on more localized data selection. As an example, no more than 12 observed values are allowed to influence each grid point value in the SMHI (HIRLAM Baseline) analysis scheme, while the same parameter for the ECMWF scheme is set to 451. As a consequence, multivariate relationships are more efficiently utilized and the scheme is well suited for vector processing computers. To be able to analyze horizontal scales larger than the horizontal

box size and in order to have a smoother transition in the analysis fields from box to box, observations from outside the analysis box are also selected to influence the analysis of grid point values within the box. In addition, a further averaging of analysis increment values within the box overlap zones, between neighboring boxes, is also carried out in order to smooth out possible discrepancies between analysis values near the box boundaries. There is also an option to use vertical boxes for different data selection in 2-3 vertical layers. The largest horizontal box size for most HIRLAM experiments has been set to 330 km with a limit distance of influencing observations of 930 km away from the center of the analysis box. The corresponding values used operationally at ECMWF are 660 km and 1500 km respectively. These reductions in box size and data selection radius were motivated by the relatively denser network of data within the HIRLAM area compared to the global ECMWF area. In case the number of selected data values becomes larger than the maximum number (451), the "basic" box is automatically divided into four sub-boxes. If needed, a further division of these sub-boxes is also carried out.

Other important features of the ECMWF analysis scheme are the very flexible data structure that allows inclusion and use of observed data from very many different sources and the comprehensive quality control of all observed data to be used. For radiosonde geopotential and wind, observed data from levels being the closest ones to 15 predetermined pressure levels are utilized while for radiosonde humidity, 5 levels of data in the troposphere only are utilized. PILOT wind observations are treated in the same way as radiosonde wind observations. Observed sea-level pressure observations or, if available, station level pressure observations from SYNOP, SHIP and DRIBU reports as well as single-level wind reports from SHIP, AIREP, and SATOB reports are utilized at the reporting levels. Because of the low vertical resolution of satellite sounding observations, thicknesses from 7 pre-determined layers only are extracted from the SATEM reports. SATEM data below 100 hPa are not used for land areas except for opolar areas (north of 70°) where SATEM data down to 400 hPa are used over land. Reported precipitable water observations for three tropospheric layers in the SATEM reports are utilized and converted to relative humidities of the same levels as humidities are extracted from radiosonde reports. Additional "indirect" humidity observations are derived from SYNOP and SHIP cloudiness observations. Each observed value to be used by the analysis scheme is associated with an estimated standard deviation of the observational error. For satellite sounding data, different standard deviations are used depending on the degree of cloud contamination (clear, partly cloudy or micro-wave). Radiosonde and satellite sounding data are assumed to have vertically correlated observational errors and a model for this correlation is used in the analysis scheme. Satellite sounding data are assumed to have horizontally correlated observational errors, whereas all other observational errors are assumed to be horizontally independent. The horizontal correlation of satellite sounding observational errors has been modeled by an isotropic Gaussian correlation function. All observational data are assumed to be uncorrelated with the first guess fields.

The comprehensive quality control of all observed data includes check against forecast first guess values, multi-level vertical consistency check and a powerful check of the observed value against analysis values derived without the influence of the observed values to be checked ("analysis check"). The quality control system also includes a possibility to "black-list" observations from stations or platforms which, in the data monitoring, have been found to be systematically in error or contaminated by frequent large random errors. To be able to carry out an efficient monitoring of the performance of the analysis scheme, quality control information in the form of quality flags and quality control event flags in addition to observed minus first guess, observed minus analysis and observed minus initialized analysis values are archived for all observed values used or checked by the analysis and stored in comprehensive "statistics" files. These statistics files have turned out

to be very useful for "black-listing", i.e. finding observation sites and observation platforms with large bias errors or with high frequency of data rejections due to large random observational errors. Another important application of the statistics files is for the development of spatial covariance models for forecast first guess and observational errors.

The analysis computations are carried out in the following sequence

- (1) Extraction of observational data.
- (2) Interpolation of first guess fields and first guess error standard deviation fields to the observational positions. Bilinear interpolation is used in the horizontal and tension spline interpolation is used in the vertical.
- (3) Formation of superobservations by averaging observed values of the same data types over areas corresponding to the analysis grid resolution. Superobservation formation is preceded by a quality control to avoid superobservation values to be influenced by erroneous data.
- (4) Analysis check and final rejection/acceptation of observed data for the grid point analysis. This is done by inversion of the large covariance matrices for each analysis box, reduction of the inverted matrices to suppress influence of the values to be checked and successive rejection of those observed values which have the largest deviations from the analyzed values.
- (5) Computation of an analysis "vector" for each analysis box by solving a linear system of equations, the left hand side matrix being the covariance matrix corresponding to all observed values accepted by the final quality control and the right hand side being the vector of observational increments.
- (6) Evaluation of analysis increments for all forecast model gridpoints by scalar multiplication of the proper analysis vector with a vector containing covariances between the observed values and the grid point quantity to be analyzed

The analysis code is highly vectorized and also efficiently multi-tasked, the latter being possible due to the linear nature of the analysis computations

4.1.2 Adaptations to the limited area version

The following changes were required to run the ECMWF global scheme on a limited area

1. The east and west boundaries of the area no longer coincide
2. The north and south boundaries (the poles in the global area) no longer collapse into single points.
3. The gridpoint numbering now runs from (2,2) (NW corner), through (NLON,2) (NE), (2,NLAT) (SW) to (NLON,NLAT) (SE). This deviates from the ECMWF scheme, where the numbering starts at the north pole and at the Greenwich meridian.

The limited area is defined on a projection which may deviate from the geographical lat/lon system. Therefore a coordinate transformation is to be performed. Coordinates enter the scheme in the following places.

- 1 Coordinates of the observation positions, wind direction and ship courses. These transformations are described in section 4.1.7
- 2 The Coriolis parameter, the geostrophy factor (see section 4.1.1) and other parameters describing the analysis structure functions. All these parameters are supplied to the analysis scheme as tables in coefficient files. Due to the coordinate transformation, these coefficient tables were changed from one-dimensional tables in the original code into two-dimensional tables in the HIRLAM code. Programs to construct HIRLAM analysis coefficient files from the corresponding ECMWF coefficient files are described in section 4.1.9

The following two variables were added to the COMMON-area COMMDL in order to describe the shifted pole of the HIRLAM transformed grid

<u>Variable</u>	<u>Type</u>	<u>Meaning</u>
SINPOL	Real	Sine of latitude of shifted pole
COSPOL	Real	Cosine of latitude of shifted pole

The following two variables were added to the COMMON-area EXTPAR in order to have access to the original lat.long. coordinates of observations

<u>Variable</u>	<u>Type</u>	<u>Meaning</u>
RLAT90	Real	Obs. latitude in old coordinate system (pole at 90° N).
RLON90	Real	Obs. longitude in old coordinate system (pole at 90° N)

Code changes to run the ECMWF analysis scheme on limited areas in the transformed grid were introduced into the following subroutines BASTRE, CNTLAN, DATAUCH, DATNUM, EXPCLR, FGERR, FNDBOX, GETERR, HEDAOF, INIERR, INIFG, INTCOE, MEMMAN, MODDES, MODSUR, MULTIL, NEARPT, PREAMB, SETCOM, SINGLL, SPRDER, TANAEV, TDATCH, TGRPEV, THGRPEV, THKFGE, THSUPER, THUMCH, THUMEV, TSUPER and WMOMESS

The Cray "update" modifications required to change the ECMWF scheme according to above are available as a file ANAMODS in the fileset HIRLAMSET

4.1.3 Adaptations to the HIRLAM file structure

The input first guess and output analysis fields in the original ECMWF code are given as global spectral coefficient files. These input and output parts of the ECMWF analysis scheme have been changed to fit the HIRLAM unpacked gridpoint model slab format as described in section 5.1. Due to the general structure of this format, this adaptation involved rather extensive code changes in adjusting the internal pointer system of the analysis code to the pointer system of the HIRLAM file format. Significant changes were introduced to the following subroutines:

MDLDDR Extraction of horizontal geometry and vertical coordinate information from the data

description records of the input HIRLAM first guess file and preparation of the corresponding COMMON areas in the analysis code. The COMMON area DDRHL , see section 5.1 was added to the analysis code

BUFMDR: Allocation of space for forecast first guess fields and construction of tables of pointers to various parameters and levels of these forecast first guess field.

GETROW: Input of forecast first guess field values. A separate subroutine READSD was introduced for reading of the HIRLAM slab format files.

GPCALC: Preparation of first guess field values including calculation of values not included in the input first guess file

ENDFG: Finishing the first guess field processing.

A subroutine WSPECO was introduced for the output of the gridpoint analysis fields to a file in the HIRLAM slab format.

In order to facilitate the addressing of the various fields in the HIRLAM first guess fields file records, the following common area was introduced

COMMON/FGPNT/ MPTFIS, MPTTS, MPTTN, MPTUN, MPTVN, MPTQN,
MPTTS, MPTZ0, MPTWS, MPTSN, MPTT2, MPTQ2, MPTU10, MPTV10,
MPTLSM, LT2, LQ2, LU10, LV10, LLSM, NAMFGN, NAMFGS

The variables in COMMON/FGPNT/ have the following meaning

<u>Variable</u>	<u>Type</u>	<u>Meaning</u>
MPTFIS	I	Pointer to s, surface geopotential
MPTPS	I	Pointer to Ps, surface pressure
MPTTN	I	Pointer to T, model level temperatures
MPTUN	I	Pointer to U, model level u-components of wind
MPTVN	I	Pointer to V, model level v-components of wind
MPTQN	I	Pointer to Q, model level specific humidities
MPTTS	I	Pointer to Ts, surface temperature
MPTZ0	I	Pointer to Z0, surface roughness
MPTWS	I	Pointer to Ws, surface wetness
MPTSN	I	Pointer to Sn, snow depth
MPTT2	I	Pointer to T2, 2 meter temperature
MPTQ2	I	Pointer to Q2, 2 meter specific humidity
MPTU10	I	Pointer to U10, 10 meter u-component of wind
MPTV10	I	Pointer to V10, 10 meter v-component of wind
MPTLSM	I	Pointer to LSM, the land-sea mask
LT2	L	Flag, T if T2 is available in the f.g. file, else F
LQ2	L	Flag, T if Q2 is available in the f.g. file, else F
LU10	L	Flag, T if U10 is available in the f.g. file, else F
LV10	L	Flag, T if V10 is available in the f.g. file, else F
LLSM	L	Flag, T if LSM is available in the f.g. file, else F

NAMFGN	I	String containing the name of the first variable of the first guess file for the northern row of the two rows in core
NAMFGS	I	String containing the name of the first variable of the first guess file for the southern row of the two rows in core

The subroutine BUFDMDR sets proper values to the variables of COMMON/FGPNT/ based on the content of the Data Description Record of the first guess field file

The vertical levels of the HIRLAM model are given as coefficients for model "full" levels in the Data Description Record of the HIRLAM files. To calculate the model "half" levels properly, code modifications were introduced into the following subroutines: GPCALC, INIANA and INIVER.

4.1.4 Setting of file unit numbers

In order to have some flexibility in the setting of unit numbers to the input and output files, an additional NAMELIST has been introduced into the analysis code.

NAMLIST/NAUNIT/ NUNIN, NAOF, NSPECI, NAERR, NCLERR,
NFCOUN, NCFIO, NSPECO, NEVAER

The correspondence between these variables, containing file unit numbers, and the input-output files is as follows

<u>Variable</u>	<u>File</u>
NUNIN	Main namelist NAMANA containing analysis parameters
NAOF	Analysis observation file (AOF)
NSPECI	Input first guess fields
NAERR	Analysis errors of the previous analysis cycle
NCLERR	Climatological first guess error coefficients
NFCOUN	Forecast first guess error coefficients
NCFIO	Comprehensive observation file (COF)
NSPECO	Output analysis fields
NEVAER	Output analysis errors

File unit number 5 is used for input of NAMLIST/NAUNIT/

4.1.5 Look-up tables for horizontal correlation functions

To be able to represent small scale horizontal correlation functions also for large distances, look-up tables of these correlation functions have been introduced. The look-up tables are constructed by the aid of subroutines for Bessel functions in the NAG library (S17AEF and S17AFF). A subroutine IBESTB has been introduced for the preparation of the correlation tables. The distance increment of the look-up tables is hard-coded into the variable BESRES of the subroutine INIBES. (At present this distance increment has been set to 500 meters.) Original ECMWF calculation of Bessel functions by polynomial approximations in the statement function BESSFUN has been replaced by looking up in the tables. Distances are simply truncated to the

closest distances represented in the look-up tables.

(Note: After the introduction of look-up tables in the HIRLAM version of the analysis code, ECMWF has introduced a more accurate approximation of Bessel-functions by the aid of rational functions. It is probably no longer useful to have look-up tables for the reason given above. The computing time efficiency of the different ways of getting the correlation model values has not been evaluated thoroughly.)

4.1.6 Adaptations to water vapour saturation pressure calculations

In order to be consistent with the forecast model formulations, a revised calculation of water vapour saturation pressure has been introduced. For temperatures below -15°C a formulation for saturation pressure valid above an ice surface is used, for temperatures above 0°C a formulation valid over a water surface is used and in the intermediate interval coefficients for the calculation of water vapour saturation pressure are obtained by linear interpolation in temperature between the coefficients of the two formulations.

Water vapour saturation pressure is obtained by the following formula:

$$E_s = C_1 EXR \frac{C_3 (T - T_{melt})}{T - C_4}$$

where

$$C_1 = 610.78$$

$$C_3 = \begin{cases} 17.269 & \text{if } T_{melt} \\ 21.875 & \text{if } T \leq T_{melt} - 15.0 \end{cases}$$

T = Temperature

T_{melt} = Melting point temperature = 273.16

$$C_4 = \begin{cases} 35.86 & \text{if } T \geq T_{melt} \\ 7.66 & \text{if } T \leq T_{melt} - 15.0 \end{cases}$$

For temperatures in the interval $(T_{melt} - 15.0) \leq T \leq T_{melt}$ the coefficients C_3 and C_4 are obtained by linear interpolation in temperature.

The water vapour saturation pressures are stored in a look-up table as functions of temperature with an temperature increment of 0.1 K. A subroutine ESTAB is used to prepare the look-up table which is stored in the COMMON-area ESCOM.

PARAMETER (NESTAB=2521)
COMMON/ESCOM/TFIRST,ES(NESTAB)

where

NESTAB = Length of the look-up table

TFIRST = the temperature of the first value in the look-up table (=123.16 K)

ES = the table of water vapour saturation pressures

The following ECMWF analysis subroutines were modified to make them consistent with the HIRLAM water vapour saturation pressure calculations: SETCOM, SPCHMD, THGRPEV, SOUNDG, MULTIL and GPCALC.

4.1.7 Preparation of the observation file from the ECMWF AOF

The observation file is obtained by sorting the observations in the AOF (Analysis Observation File) according to the gridpoint number. An auxiliary job in the ECMWF system ("SORTAOF") precedes the analysis to accomplish this

This auxiliary job was adjusted for HIRLAM experiments on CRAY

- a) The lat/lon coordinates of the observations are transformed to the HIRLAM grid (subroutine TRNSCO). The transformed coordinates override the original ones
(Note : The direction of vector elements (e.g. wind, ship course) is adjusted to the new coordinate axes orientation later during analysis computations by the subroutine UVGRID)
- b) The gridpoint number of the observations is calculated from the transformed coordinates, according to the gridpoint numbering described in section 4.1.2
- c) Observations outside the limited area are rejected

The program REDAOF rejects observations outside the limited area and the program SORTAOF "quicksorts" the observations by means of the Cray library routine ORDERS.

4.1.7 The Script Preop

The script Preop is responsible for all processing of observations from the input BUFR file to the point where the Analysis Observation File (AOF) is ready for the analysis program.

It starts with determining the analysis time and the time interval for which observations should be retrieved (based on the DTG for the present run, and FCINT - the forecast interval - for this experiment)

At ECMWF, the observation file is retrieved from the MARS archive; this implies that at

ECMWF, no experiments can be run that are more recent than 2 days ago, because the observation data is not stored in MARS before 2 days since observation time have elapsed.

If running at ECMWF, or if the observation file was sent from ECMWF (and not generated locally), a reduction of the file is attempted based on observation time and position relative to the area of forecast. This is accomplished by running the MAOF (Make Analysis Observation File) program with MODE=3.

Subsequently, the script MakeBUFRtabs is called, to create the BUFR tables - or simply copy them to the directory pointed to by the environment variable HL_DATA (see also 5.2.2.)

Finally, MAOF is called to produce the AOF from the BUFR input.

4.1.8 De-staggering of wind components of the first guess and staggering of wind components of analysis increments

Wind components in the HIRLAM model slab files are given in the staggered Arakawa C grid. The ECMWF analysis program assumes that wind components are given in un-staggered grids. A revision of the analysis code to take the staggered geometry in account would have resulted in significantly increased computing time (new distances and correlations valid for the staggered u and v points). Therefore, the wind components of the analysis first guess are de-staggered before the analysis computations and the wind component analysis increments are staggered after the analysis computations have been carried out. These de-staggering and staggering calculations are done in separate program modules DESTAG and UVSTAG respectively

The de-staggering as well as the staggering are done by a simple averaging:

$$\begin{aligned} U_{i,j}^u &= 0.5(U_{i-1,j}^s + U_{i,j}^s) \quad \text{for } 2 \leq i \leq NLON \wedge 1 \leq j \leq NLAT \\ U_{1,j}^u &= U_{1,j}^s \quad \text{for } 1 \leq j \leq NLAT \\ V_{i,j}^u &= 0.5(V_{i-1,j}^s + V_{i,j}^s) \quad \text{for } 1 \leq i \leq NLON \wedge 2 \leq j \leq NLAT \\ V_{i,1}^u &= V_{i,1}^s \quad \text{for } 1 \leq i \leq NLON \end{aligned}$$

and

$$\begin{aligned} \delta U_{i,j}^s &= 0.5(\delta U_{i+1,j}^u + \delta U_{i,j}^u) \quad \text{for } 1 \leq i \leq NLON \wedge 1 \leq j \leq NLAT \\ \delta U_{nlon,j}^s &= \delta U_{nlon,j}^u \quad \text{for } 1 \leq j \leq NLAT \\ \delta V_{i,j}^s &= 0.5(\delta V_{i,1+j}^u + \delta V_{i,j}^u) \quad \text{for } 1 \leq i \leq NLON \wedge 1 \leq j \leq NLAT-1 \\ \delta V_{i,nlat}^s &= \delta V_{i,nlat}^u \quad \text{for } 1 \leq i \leq NLON \end{aligned}$$

Here, U^s, V^s denotes staggered wind components, U^u, V^u destaggered wind components, $\delta U^u, \delta V^u$ un-staggered wind component analysis increments and $\delta U^s, \delta V^s$ staggered wind component analysis increments. NLON and NLAT denotes the number of latitude

gridpoints and longitude gridpoints respectively

The program DESTAG for destaggering of first guess wind components has the following input parameters on unit 5

(1) NAME (CHARACTER*3) containing the experiment name of the output slab file. If NAME='EXP', the experiment name in the output slab file is taken as the same as given in the input slab file.

(2) Namelist NAMSTA with the following two variables ILUIN Integer Unit number of the input slab file ILUOUT Integer Unit number of the output slab file.

The program UVSTAG for staggering of wind component analysis increments has the following input parameters on unit 5

(1) NAME (CHARACTER*3) containing the experiment name of the output analysis slab file. If NAME='EXP', the experiment name of the output analysis slab file is taken as the same as on the input analysis slab file.

(2) Namelist NAMUVS with the following four variables: ILUFGU Integer Unit number of the input de-staggered first guess file ILUFGS Integer Unit number of the input staggered first guess file ILUNAU Integer Unit number of the input un-staggered analysis file ILUNAS Integer Unit number of the output staggered analysis file

Note that the programs DESTAG and UVSTAG contain PARAMETER-statements for the following variables describing maximum dimensions.

JDN	!	Number of gridpoints in the longitudinal direction
JPI	!	Number of gridpoints in the latitudinal direction
JPELEV	Integer	Maximum number of model levels
JPMLF	Integer	Maximum number of multi-level fields in the slab files
JPSLF	Integer	Maximum number of single-level fields in the slab files

4.1.9 Creation of HIRLAM analysis coefficient files

A program ANAFIL for creation of HIRLAM analysis coefficient files as well as "cold start" analysis error (standard-deviation) files from the corresponding ECMWF files has been established. The analysis coefficients of the ECMWF scheme are only functions of latitude. Since some of the coefficients, e.g. the Coriolis parameter, also have a longitudinal dependence in transformed HIRLAM grids, the analysis coefficients of the HIRLAM scheme have to be functions of latitude and longitude. The ECMWF formats for the analysis coefficient files allows for such two-dimensional tables, although the analysis code had to be modified for the use of these tables.

Linear interpolation in latitude is used to derive the HIRLAM analysis coefficients from the corresponding ECMWF coefficients. Bilinear interpolation in latitude and longitude is used to create the HIRLAM "cold start" analysis errors from the corresponding ECMWF errors.

The program ANAFIL has the following two input namelist statements, the namelist LAMP is read from file unit 5.

NAMELIST/LAMP/ LFGIN,LFGOUT,LCLIN,LCLOUT,LAEIN,LAEOUT,RESERR,LNM

NAMELIST/NAMGEO/FLONW,FLONE,FLATS,FLATN,DFLON,DFLAT,POLE,NFLON,NFLAT

The variables in the namelist statements have the following meaning.

LFGIN	Integer	Unit number of the input (ECMWF) forecast first guess analysis coefficients file
LFGOUT	Integer	Unit number if the output (HIRLAM) forecast first guess analysis coefficient file
LCLIN	Integer	Unit number of the input (ECMWF) climatological first guess analysis coefficient file
LCLOUT	Integer	Unit number of the output (HIRLAM) climatological first guess analysis coefficient file
LAEIN	Integer	Unit number of the input (ECMWF) analysis error file
LAEOUT	Integer	Unit number of the output(HIRLAM) analysis error file
RESERR	Real	Grid resolution of the HIRLAM grid of analysis coefficients and analysis errors. This grid resolution should roughly correspond to the resolution of the analysis boxes.
LNM	Integer	File unit number of NAMELIST/NAMGEO/.
FLONW	Real	Western boundary, o E, of the HIRLAM transformed area
FLONE	Real	Eastern boundary, oE, of the HIRLAM transformed area
FLATS	Real	Southern boundary, o N, of the HIRLAM transformed area
FLATN	Real	Northern boundary, o N, of the HIRLAM transformed area
DFLON	Real	Longitudinal grid resolution, long., of the HIRLAM grid
DFLAT	Real	Latitudinal grid resolution, lat., of the HIRLAM grid
POLE	Real	Latitude of the North Pole, N, of the transformed HIRLAM grid
NFLON	Integer	Number of longitudes of the transformed HIRLAM grid
NFLAT	Integer	Number of latitudes of the transformed HIRLAM grid

The subroutine CONAER, called from ANAFIL to generate the HIRLAM analysis error file, includes a PARAMETER statement for the following parameter

JPMLON	Integer	Maximum number of longitudes of the HIRLAM analysis error and analysis coefficient grid (=50 at present)
--------	---------	--

The subroutine COCOEF, called from ANAFIL to generate the HIRLAM analysis coefficient files, includes a parameter statement for the following parameters

JPMLON	Integer	See above!
JPMLAT	Integer	Maximum number of latitudes of the HIRLAM analysis error and analysis coefficient grid (=50 at present)

4.2 Analysis of sea surface temperature and ice coverage

4.2.1 General features

A system to analyze the sea surface temperature (SST) and ice coverage in a grid square has been developed in the HIRLAM project. The basic idea of the system is that many kind of observations can be used - real observations, manually picked up "observations" from manually analyzed SST or ice coverage maps or "observations", which are grid point values from another objective analysis. The first guess for SST analysis is either climate or another analysis. The first guess for the ice analysis is created from the analyzed sea surface temperature.

Another essential feature is that the analyzed parameter always is the sea surface temperature. For instance, if half of a grid square is covered by land and half by water, the analysis describes the temperature of the water part, not a mean value of land and water temperatures in the grid square. This means that in the model there must be two surface temperatures, one for land and one for water. On the other hand this makes it possible to handle water and land surfaces in one grid square separately, for instance, when calculating the surface fluxes. In grid squares, which are totally land covered, the analysis of sea surface temperature or ice coverage is meaningless.

The analyzed parameter in ice analysis is the percentage of ice cover in a grid square. This again makes it possible to treat open and ice covered parts separately in the forecast model.

Because sea surface temperature is changing rather slowly, it is not necessary to create SST analysis in every data assimilation cycle. For this reason the SST and ice coverage analysis is run separately from the data assimilation cycle and the same analysis can be used several times. For the same reason observations from a time period of 4 days from actual date are accepted as valid observations.

4.2.2 Input of sea-surface temperature and ice coverage data from the Swedish Meteorological and Hydrological Institute

The Marine Forecasting Office of the SMHI carries out a very careful manual analysis of sea-surface temperature (SST) and ice coverage in the Baltic Sea, in Skagerack, in Kattegatt and in the large lakes of Sweden. This analysis is done three times a week. The manually analyzed charts are "digitized" in the following way:

- (1) SST values are inserted for a number (30) of pre-selected positions. In addition, values in any number of "significant" points, e.g. values in areas with steep gradients, can also be inserted.
- (2) The ice-charts are digitized to a horizontal grid with a resolution of 0.2×0.2 with each gridpoint value showing only ice/ no ice.

The digitized information on SST and ice coverage is transmitted to the ECMWF computer system in the following way and in the data format described below.

- (1) All SST values inserted by the Marine Forecasting office are transmitted.
- (2) The ice coverage grid data (ice / no ice) is averaged in the horizontal to construct grid pointvalues of "percentage ice cover" with a horizontal resolution of 0.5×0.5 .

4.2.3 Method of computations

The first guess for the analysis of sea surface temperature is either climatological SST of the month or previous analyses or some other analysis of SST. Because the sea surface temperature is changing rather slowly it is not necessary to create a new analysis every day and the analysis can be kept constant during the forecast. It is important that also the first guess describes the analysis of water surfaces to get a proper analysis in grid squares which are only partly sea covered.

The analysis method is a successive correction method, where the first guess value is iteratively corrected with the observed deviation from the first guess. The analysis value is given as

$$T_{an} = T_{fg} + \sum_i w(i) (T_{ob}(i) - T_{fg}(i))$$

where the weighting value $w(i)$ is calculated as

$$w(i) = \frac{(R(i) - Rmax)^2}{q(R(k) - Rmax)^2}$$

$R(i)$ is the distance between i 'th observation and the grid point and $Rmax$ is the maximum influencing radius. All the observations within the radius $Rmax$ from the grid point are used.

For SST analysis five iterations are made with decreasing maximum influencing radius. The values used for $Rmax$ are 1000, 500, 250, 80 and 50 km in the different iterations.

The first guess for the ice coverage is calculated from the analyzed SST in the following way:

1. if $T(i) > Tmax$ then 0 % of ice
2. if $T(i) < Tmin$ then 100 % of ice
3. between $Tmin$ and $Tmax$ interpolate the ice coverage linearly so that $Tmin$ gives 100 % and $Tmax$ 0 % of ice.

The proper values for $Tmin$ and $Tmax$ in the North Atlantic are 271.3 K and 272.16 K, respectively. However, these values give too little of ice in the areas, where the salinity is small, for example in the Baltic Sea.

The marginal ice zone, where the sea is partly ice covered, is usually rather narrow. This means that the maximum influencing radius $Rmax$ must be much smaller for ice analysis than for SST analysis. In the present version we use four iterations for ice analysis with influencing radius of 250, 125, 75 and 20 km, respectively. To get a good analysis on the area of small salinity we need a good coverage of observations.

4.2.4 Computer code organization

The calculations are made in a modified latitude/longitude grid in a limited area.

Subroutines

The main subroutine for SST and ice coverage analysis is "SST", which calls for the following subroutines:

GOPEN, GREAD: opening and reading the files of meteorological fields

INITPA: printing headers etc.

~~ROM~~ reading, sorting and selecting, concerning date and area, of SST and ice coverage observations.

SSTPRE: control the analysis of SST

SSTANA: called by SSTPRE, makes the analysis of SST.

ICEANA making the analysis of ice coverage

STOSST create a file containing the analysis of SST and ice coverage

PBLINA, GRDINT: used for interpolation

Common blocks

~~SDM~~ Gives, via parameter statement, the maximum size of fields and maximum number of observations

GRID: Description of geographical area

DATE: Date of the analysis

WORKSP: Working arrays

SSTFIL: Logical numbers of files

SSTOBS Arrays for observations

DDRHL: Ddr of files in HIRLAM format

4.2.5 Data organization

Observation file:

The observation file consists of coded data records, which are read in using free field format. The structure of a record is following:

xx. yy. zz.

A record is interpreted in the following way:

1. normally:
 - xx = observed SST value in Celsius
 - yy = latitude in degrees
 - zz = longitude in degrees
2. if $80 < xx < 150$:
 - xx = year
 - yy = month
 - zz = day
3. if $200 < xx < 300$:
 - xx - 200 = ice observation, percentage of ice
 - yy = latitude in degrees
 - zz = longitude in degrees

The observation record refers always to the previous date record. This means that the first record in a file should always be a date record. The observation file can contain records for many dates.

Namelists.

Namelist LULIST gives the logical units for the files used in the run:

nlm	= parameter file (default 5)
nlp	= printer file (default 6)
nlnd	= output analysis file, random (default 99)
nlnd	= output analysis file, sequential (default 92)
nlfgg	= first guess file, sequentia (default 95)
nlfg	= first guess file, random (default 91)
nlob1	= observation file (default 93)

Namelist DALIST gives the date of the run

idate	= date in the form 'yymmddhh', where
yy	= year
mm	= month
dd	= day
hh	= hour

5. Data structures

5.1 Fields

5.1.1 General

All HIRLAM field data are kept and archived in packed form. The packing is done according to the WMO standard, 'FM 92-VIII GRIB' gridded binary (GRIB) format. Details on the GRIB format may easiest be found in an ECMWF Publication, (Binary Data Representation. ECMWF May 1988). This format provides a high degree of efficiency in the packing and allows easy transfer of fields between computers with different wordlengths.

A field file consists of a data description record (DDR) followed by a sequence of grib records, one for each field. The DDR contains all information needed for complete identification of the file and its contents. The DDR is described in sect. 5.1.2. Normally a field file contains data from one 'atmospheric state', i.e. only one analysis or one forecast time.

Each individual field is identified by three parameters, the field type, the parameter and the level. In addition common information such as origin, date/time, geometry etc. for the particular state is included in each grib record of the file. Each GRIB record thus contains all information essential for its identification and use. The small amount of duplication of information caused by this is considered worthwhile, since it makes each single field record fully selfcontained. It may be read, decoded and understood without any external information.

5.1.2 Definition of mandatory descriptor record. (DDR)

The DDR contains the following consecutive sections. Each section is composed of either integers or real numbers, but never mixed.

Horizontal grid definition section (integer part)

word name	type/dim	meaning
1 NCODHL	I	code definition number=1000
2 NLDRHL	I	record length=1000
3 NLNXHL	I	length of next descriptor record (0=none)
4 MRCLHL	I	maximum record size
5 MDRLHL	I	maximum data record size
6 NTYPHL	I	file organization type 0 slab file from north to south, in words 10 gribcode file with hirlam conventions, eg. ordered w to e, then s to n, all units are SI, etc 90 general gribcode file 100 field file (w to e, then s to n), in words
7 NEXPHL	I	experiment/model identification use CCITT-ITA no.5 (=ASCII), 4 bytes - 0=48, 9=57, A=65,

.., Z=90, eg. 'A01' = hex(00,41,30,31) - if the first byte of these four is not 0, it identifies the operational model version.

8 NMDIHL	I	missing data indicator (equivalenced to AMDIHL)
9 NIDFHL	I(12)	0: NMDIHL is not supplied spare, to define file origin

Time level definition section (integer part)

21 NDTVHL	I	verification date/time (YYMMDDHH,clim=00MM0000)
22 NSCVHL	I	verification time, seconds after NDTVHL
23 NDTBHL	I	data base date/time (YYMMDDHH, climate=00MM0000)
24 NSCBHL	I	data base time, seconds after NDTBHL
25 NFLHHL	I	forecast length, hours
26 NFLSHL	I	forecast length, seconds after NFLHHL
27 NDORHL	I	data origin 0 analysis 1 initialized analysis or forecast spare, to define date and times
28 NIDTHL	I(13)	

Horizontal grid definition section (integer part)

41 NPRJHL	I	projection type 0 regular lat/lon 1: Mercator 2: stereographic 3 Lambert conformal 4: Gaussian lat/lon 50 spherical harmonics 60 regular lat/lon with shifted pole
42 NPRCHL	I	indicator of presence of grid descriptor records 0: no grid descriptor records present
43 NLONHL	I	number of gridpoints along x (lon) axis
44 NLATHL	I	number of gridpoints along y (lat) axis

Vertical grid definition section (integer part)

45 NLTPHL	I	vertical level type 0:as used in HIRLAM Level 1 100: pressure 103 z above msl 105 z above ground 107 109 hybrid, not 356 log(pressure) add 1000 for polynomial expansion coefficients
46 NLEVHL	I	number of levels
47 NPPLHL	I	number of parameters per level (eg. 2 for eta)
48 NRFLHL	I	number of reference levels (eg., 0 for eta-coords)
49 NLPTHL	I(40)	pointers to levels for multi-level fields

Multi-level fields definition section (integer part)

89 NMLFHL I	number of multi-level fields
90 NWMMHL I(40)	WMO-codes of multi-level fields. See 5 1 5
130 NMPTHL I(40)	pointers to first level of multi-level fields

Single level fields definition section (integer part)

170 NSLFHL I	number of single level fields
171 NWMSHL I(40)	WMO-codes of single level fields See 5 1 5
211 NSPTHL I(40)	pointers to single level fields
251 NSLTHL I(40)	types of levels of single level fields

Horizontal grid definition section (real part)

291 APLOHL R	longitude of (shifted) pole (degrees)
292 APLAHL R	latitude of (shifted) pole (degrees)
293 AWESHL R	western boundary of area (degrees)
294 AEASHL R	eastern boundary of area (degrees)
295 ALALHL R	last lat of area in file (degrees)
296 ALAFHL R	first lat of area in file (degrees)
297 DLONHL R	longitudinal gridspacing (degrees)
298 DLATHL R	latitudinal gridspacing (degrees) negative for files north to south
299 GRIDHL R(28)	spare, to define other projection types

Vertical grid definition section (real part)

327 ALEVHL R(40,4)	parameters to define levels
487 RLEVHL R(4)	reference levels

Multi-level fields definition section (real part)

491 STMXHL R(40)	x-stagger of multi-level fields (griddistances)
531 STMYHL R(40)	y-stagger of multi-level fields (griddistances)
571 STMZHL R(40)	z-stagger of multi-level fields

Single level fields definition section (real part)

611 STSXHL R(40)	x-stagger of single level fields (griddistances)
651 STSYHL R(40)	y-stagger of single level fields (griddistances)
691 SLEVHL R(40,4)	level parameters of single level fields

Record lengths of the GRIB records.

851 LRECHL I(150)	Record lengths of GRIB records, in the order the records are written.
-------------------	---

5.1.3 GRIB format

The GRIB format is designed to allow efficient packing of meteorological field data on computers of different wordlengths and other characteristics. The field values (gridpointvalues or spectral coefficients) are packed in an arbitrary, preselected accuracy. One field is packed into one GRIB record, which is completely self-contained, i.e. it contains all information needed to unpack, identify and use the field.

Each GRIB record consists of 5 (or optionally 6) blocks and is organized in basic units of 8 bits, called octets. All numbers are integers, suitably scaled. The contents of the blocks is outlined below.

Some minor additions have been added to cater for special HIRLAM features, such as rotated lat-long coordinates. A star (*) below indicates these modifications.

block 0 indicator block

the word 'GRIB' in ASCII code	4 octets
-------------------------------	----------

block 1 product definition block

length of block (=24)	3 octets
experiment name in ASCII characters	3 octets *
grid definition (not defined. =0)	1 octet
indicator. Presence of blocks 2 and 3	1 octet
parameter (WMO code, see 5.1.5)	1 octet
level type (see 5.1.2)	1 octet
height/pressure/number of level	2 octets
year/month/day/hour/minute	5 octets
indicators for time range etc.	4 octets
not set	2 octets

block 2 grid description block

length of block (36 or 40 (eta))	3 octets
no. of unused bits at end of block 2	1 octet
not set.	1 octet
data representation type (=60)	1 octet *
no. gridpoints along latitude	2 octets
no. gridpoints along meridian	2 octets
latitude of origin.	3 octets
longitude of origin.	3 octets
resolution flag. (= 1)	1 octet
latitude of extreme point.	3 octets
longitude of extreme point.	3 octets
direction increment along latitude	2 octets
direction increment along longitude	2 octets
scanning mode, (= 2, i.e. w-e,s-n)	1 octet
longitude of geometric pole.	2 octets *

	latitude of geometric pole	2 octets *
	exponent of 1st vertical coordinate	1 octet
	mantis of 1st vertical coordinate	3 octets
	exponent of 2nd vertical coordinate	1 octet
	mantis of 2nd vertical coordinate	3 octets
block 3	bit map block	
	not used block is absent indicated by flag in block 1	
block 4	binary data block	
	length of block (=11+N)	3 octets
	flag (see GRIB manual)	1 octet
	scale factor	2 octets
	reference value	3 octets
	no. of bits for each packed value	1 octet
	flag (= 0)	1 octet
	packed data	N octets
block 5	end bloc	
	'7777' in ASCII characters	4 octets

5.1.4 Programs for packing and unpacking

A special library containing packing and unpacking routines has been prepared. In the following a short account is given of those routines the user may need to understand

Storage

Packing of a field into the GRIB format is done by a subroutine called GWRITE. GWRITE packs one field, making use of information in the DDR, which has to be set in COMDDR according to 5.1.2. After packing, the field is written to a temporary direct access file (D.A.-file).

When all fields have been packed, they are written sequentially to the permanent GRIB-file, in the order defined by the pointers in the DDR. This order is not necessarily the order in which the actual packing happened. Furthermore, in the sequential file the fields are written with the varying record lengths that may result from the packing operation. These lengths are recorded in the DDR (LRECHL).

Before starting to write a GRIB-file, the temporary D.A.-file must be opened. This is done by

SUBROUTINE GWOPEN(LUDIR,IBUF,LENREC)
where:

LUDIR	unit number
IBUF	the workspace to receive the GRIB records.
LENREC	length of workspace.

After opening LUDIR, each file is packed and written to the D.A. file by

SUBROUTINE GWRITE(LUDIR,KTYP,KWMO,ALEV,F,MAX,MY,IBUF,LENREC)
where:

KTYP	field type (see 5.1.2)
KWMO	field parameter (see 5.1.2 and 5.1.5)
ALEV	level (in hPa for pressure levels, in meter for z-levels and as FLOAT (model level) for model levels)
F(MX,MY)	the field to be packed.
MX	max. dimension in x-direction
MY	max. dimension in y-direction Actual dimensions are taken from the DDR (5.1.2)

When all fields have been packed and written to LUDIR, they are copied sequentially to another file, LUSEQ. This is done in GWCLOS

SUBROUTINE GWCLOS(LUDIR,LUSEQ,IBUF,LENREC)

where:

LUSEQ unit number of the sequential file.

Retrieval.

A similar procedure is used for the retrieval. First the sequential GRIB-file is copied to a direct access file (D.A.-file). From this file any requested field is fetched and unpacked. Up to a maximum of four GRIB-files may be used simultaneously, by successively opening the files by GROPEN. This maximum is due to the dimensioning of some arrays, and may easily be changed by recompilation. The DDR of the GRIB-file is copied to COMDDR in GROPEN. Hence DDR will always contain the data description record of the most recently opened GRIB-file. In order to retain the information from the previous DDR's, a subset of their information is kept in another COMMON block, COMDDA. The parameters in COMDDA have the same names as those in COMDDR (see 5.1.2), except that the last two characters have been changed from 'HL' to 'DA', e.g. in the DDR the number of levels is found in NLEVHL (see 5.1.2), while in COMDDA it is found in NLEVDA(4), where the index indicates the GRIB-file order number

SUBROUTINE GOPEN(LUSEQ,LUDIR,IBUF,LENSEQ)

where.

LUSEQ	unit number of the original sequential GRIB-file.
LUDIR	D.A. workfile. Obviously different numbers must be used for different files
IBUF	work area
LENSEQ	is set, by GOPEN, to the max. length of the GRIB records.

After opening any required field may be read and unpacked by

SUBROUTINE GREAD(LUDIR,KTYP,KWMO,ALEV,F,MAX,MY,IBUF,LENREC)
where:

LUDIR	unit number of the D.A. file
KTYP	field type
KWMO	field parameter

ALEV	field level, see GWRITE
F	receiving field
MX	max x-dimension
MY	max y-dimension
IBUF	workspace
LENREC	dimension of IBUF

After use, the D.A. file may be returned

SUBROUTINE GRCLOS(LUDIR)

LUDIR may now be used again in another GOPEN

5.1.5 Indicator of Parameter

The code figures used to indicate the parameter is based on the standard WMO code figures as used in e.g. the GRIB code. However, for HIRLAM it has been decided to use SI-units strictly, while the WMO-tables frequently use so-called "meteorological" units such as geopotential meters, Celsius etc.

Code figures from 101 to 199 are used, they are normally obtained by adding 100 to the basic WMO-codes. There are however several parameters not defined in the WMO table, for such parameters the HIRLAM codes have been invented, using available numbers. Furthermore parameters have to be used in conjunction with "level type" and "level value", as defined in the DDR, in order to fully identify a parameter. Thus, for instance, the so called 2-meter temperature (T2m) is identified by level type = 103, level = 2 and 2m parameter = 104.

Fieldcode	Parameter	Unit
101	Pressure	Pa
102	Geopotential	m ² /s ²
104	Temperature	Kelvin
110	Dew-point temperature	Kelvin
112	Specific humidity	kg/kg
113	Relative humidity	0.0 to 1.0
123	U-component of wind	m/s
124	V-component of wind	m/s
130	Relative vorticity	s ⁻¹
134	Horizontal divergence	s ⁻¹
140	Vertical velocity (om)	Pa/s
147	Soil wetness	kg/m ²
151	Snow depth	kg/m ²
179	Cloud cover	0.0 to 1.0
180	Surface roughness	m
181	Fraction of land	0.0 to 1.0
182	Urbanization	0.0 to 1.0
183	Fraction of ice	0.0 to 1.0
184	Albedo	0.0 to 1.0
185	Fraction of vegetation	0.0 to 1.0
187	X-component of stress	N/m ²

5.2 Data Structures - Observations (BUFR)

5.2.1 Introduction

BUFR is a WMO standard for encoding variable length (non-gridded) data. It is especially suited for the encoding of observation data. A BUFR 'message' consists of the following 6 parts:

Section 0. Indicator section.

Octet No.	Contents
1 - 4	'B', 'U', 'F', 'R' in CCITT IA5 encoding
5 - 7	Length of the entire BUFR message (in octets)
8	BUFR edition number

Section 1 Identification section.

Octet No	Contents
1 - 3	Length of section (in octets)
4	BUFR master table (zero if std WMO FM 94-IX Ext. BUFR tables)
5 - 6	Originating centre
7	Update sequence number (zero for original messages)
8	Bit 1 = 0 No optional section Bit 1 = 1 Optional section (section 2) follows Bits 2 - 8 Set to zero (reserved)
9	BUFR message type
10	BUFR message subtype (defined locally)
11	Version number of master table used (currently 2 for std WMO FM 94-IX Ext. BUFR tables)
12	Version number of local tables
13	Year most typical for the BUFR message contents
14	Month
15	Day
16	Hour
17	Minute
18 -	Reserved for local use

Section 2. Optional section.

Octet No.	Contents
1 - 3	Length of section (in octets)
4	Set to zero (reserved)
5 -	Reserved for local use

Section 3. Data description section.

Octet No.	Contents
1 - 3	Length of section (in octets)
4	Set to zero (reserved)
5 - 6	Number of data subsets
7	Bit 1 = 1 Observed data Bit 1 = 0 Other data Bit 2 = 1 Compressed data Bit 2 = 0 Non-compressed data Bits 3 - 8 Set to zero (reserved)
8 -	A collection of 16 bit element descriptors, which define the form and content of individual data elements in each data subset in the Data section.

Section 4. Data section.

Octet No.	Contents
1 - 3	Length of section (in octets)
4	Set to zero (reserved)
5 -	Binary data as defined by sequence descriptors in section 3.

Section 5. End section.

Octet No.	Contents
1 - 4	'7', '7', '7', '7' in CCITT IA5 encoding

5.2.2 BUFR decoding; Program Library BUFR

At present, only BUFR decoding software is available, so it is not possible (except by creating the bitstream 'by hand') to encode observation data in BUFR.

The routine central to the decoding of BUFR messages is:

Subroutine BUFREX (KBUFL, KBUFF, KSUP, KSEC0, KSEC1, KSEC2, KSEC3, KSEC4,
KELEM, CNAMES,CUNITS,KVALS, VALUES,CVALS, KERR)

KBUFL	Integer	I	Length of BUFR message (in words)
KBUFF	Integer	I	Array containing BUFR message (length KBUFL)
KSUP	Integer	O	Array to contain supplementary information KSUP(1) - IDIM1, dimension of KSEC1 KSUP(2) - IDIM2, dimension of KSEC2 KSUP(3) - IDIM3, dimension of KSEC3 KSUP(4) - IDIM4, dimension of KSEC4 KSUP(5) - M (number of elements in VALUES array, first index) KSUP(6)-(number of subsets, second index in VALUES array) KSUP(7)-JVC (number of elements in CVALS array)

			KSUP(8)-Total BUFR message length in octets KSUP(9)-IDIM0, dimension of KSEC0
KSEC0	Integer	O	Array to contain BUFR section 0 information KSEC0(1)-Length of section 0 (in octets) KSEC0(2)-Total length of BUFR message (octets) KSEC0(3)-BUFR edition number
KSEC1	Integer	O	Array to contain BUFR section 1 information KSEC1(1)-Length of section 1 (in octets) KSEC1(2)-BUFR edition number KSEC1(3)-Originating centre KSEC1(4)-Update sequence number KSEC1(5)-Flag (presence of section 2) KSEC1(6)-BUFR message type KSEC1(7)-BUFR message subtype KSEC1(8)-Version number of local table KSEC1(9)-Year most typical for contents KSEC1(10)-Month KSEC1(11) - Day KSEC1(12) - Hour KSEC1(13) - Minute KSEC1(14) - BUFR master table KSEC1(15) - Version number of master table KSEC1(16) - KSEC1(40) - information defined locally
KSEC2	Integer	O	Array to contain BUFR section 2 information KSEC2(1) - Length of section 2 (in octets) KSEC2(2) - KSEC2(47) Database key (at ECMWF)
KSEC3	Integer	O	Array to contain BUFR section 3 information KSEC3(1) - Length of section 3 (in octets) KSEC3(2) - Reserved KSEC3(3) - Number of subsets KSEC3(4) - Flag (data type, compression)
KSEC4	Integer	O	Array to contain BUFR section 4 information KSEC4(1) - Length of section 4 (in octets) KSEC4(2) - Reserved
KELEM	Integer	I/O	Number of elements contained in this message
CNAMES	C*64	O	Array to contain KELEM BUFR table B element names
CUNITS	C*24	O	Array to contain KELEM BUFR table B element units
KVALS	Integer	I/O	Number of data values contained in this message Note that KVALS must be large enough for all N * M values (subsets * values per subset)
VALUES	Real	O	Array to contain KVALS expanded data values
CVALS	C*80	O	Array to contain KVALS BUFR code table or CCITT IA5 element entries
KERR	Integer	O	Return error code

A routine to read variable length FORTRAN unformatted records and retrieve its length is available in the library 'port'

Subroutine READ (KUNIT, KWORD, KOUNT, KSTAT, KUBC)

KUNIT	Integer	I	Unit number of file to read
KWORD	Integer	O	Array to contain the record read (sequentially)
KOUNT	Integer	I/O	On input Length of KWORD On output. Number of words returned in KWORD
KSTAT	Integer	O	Error return code -1 Data read, KWORD to small 0 Data read successfully 1 Empty record read 2 Reserved 3 End of file, or empty file 4 I/O error 5 Error on record structure
KUBC	Integer	O	Unused bit count

5.2.3 BUFR tables

BUFR encoding is based on the use of three types of tables. BUFR tables, code tables and flag tables. Tables containing information used to describe, classify and define the contents of a BUFR message are called BUFR tables. The following four BUFR tables are defined

TABLE A.

Defines the BUFR message types (octet 9 of section 1)

TABLES B, C and D

Define the element descriptors. An element descriptor is a 16 bit item, divided as follows

F (2 bits) X (6 bits) Y (8 bits)

F	Specifies the function of this descriptor	
= 0	BUFR table B	Descriptor classifies an element
= 2	BUFR table C.	Descriptor specifies an operation on other descriptors
= 3	BUFR table D	Descriptor defines a sequence of TABLE B and/or TABLE D descriptors
X	Specifies the group the descriptor belongs to, or the subfunction in case of TABLE C descriptors	
Y	Specifies the descriptor within the group, or the variable information of the operation in case of TABLE C descriptors.	

The BUFR tables must be present for the decoding software to work. Their names are BUFR{A|B|C|D}<edition><version>{32|64}, where the fifth character determines the specific table, the <edition> and <version> are 3 digit numbers and the trailing 32 or 64 discern between 32- and 64-bit formats. The ASCII (readable) versions of the tables have (in stead of 32 or 64 suffix) the suffix AS. The directory where the tables are to be found is set in the environment variable BUFRTAB_DIR.

5.2.4 Script Make BUFRtabs

The script MakeBUFRtabs is responsible for the creation of the (binary version) of the BUFR

tables. These might exist already, in which case they are simply copied to the directory indicated by the environment variable PARKDIR. In case they do not already exist, they are made by executing the program BUTA

5.2.5 Program BUTA

The program BUTA is responsible for the generation of the binary versions of the BUFR tables from the ASCII variants. It calls four subroutines (BTAV2, BTBV2, BTDV2 and BCTV2) to construct the tables A, B, D and C respectively. A certain amount of cross-consistency checks are made:

- 1 The code tables mentioned in table C must be associated with a table B descriptor
2. The table B and D descriptors mentioned in table D must be defined

5.3 Statistics files

5.3.1 General features of the statistics files

In a data assimilation system it is important to get feedback about the overall usage of observations. For this purpose so-called statistics files are generated as a part of the data assimilation cycle. These files contain observed data and quality control flags conveyed to each datum by the analysis system. Furthermore, they contain the deviations of the observed value from the first guess, analyzed and initialized value in the observation point.

The statistics files are used for several purposes. The observation minus analysis statistics is used to monitor the overall performance of the analysis system. This statistics can be calculated separately for each observation system. Such calculations can reveal minor systematic errors in the observations not easily detected in any other way. They can also reveal weak points in the assimilation system itself to handle certain types of observations. This statistics can also be calculated separately for every observation station. Stations producing systematically bad observations can be revealed in this way. When doing these calculations it is important that the statistics files contain all the quality control flags given at various stages of the analysis. So statistical evaluation of observation minus analysis fit give valuable information on the performance of the assimilation system as well as on the quality of the original observations.

Statistics files can also be used to developing new, flow-dependent structure functions. A general weakness in the structure functions used in the present analysis systems is that they are not geographically dependent or flow-dependent. So they are not suitable for analyzing, for instance, baroclinic structures in the atmosphere. Statistics files can provide data for empirical development of more realistic structure functions.

5.3.2 Structure and contents

Before creating the statistics files it is necessary to run the analysis in verification mode. In this run the deviations from observation are added to the COF (Comprehensive Observation File). The deviation from observed value is calculated from the field which is given as a first guess field for

the analysis program. Normally the deviations are calculated from first guess, from analyzed and from initialized analysis fields. Two statistics files are created in every analysis run, one for mass/wind data and another for humidity data. The files are created from a COF file in a separate job step. The structure of the statistics files is described in detail in the ECMWF data assimilation program documentation by Haseler (1988)

5.3.3 Transmission of statistics files

A special software has been developed for extraction and transmission of subsets of data from the statistics files to member states. This was necessary, because it seemed impossible to transmit data in binary from ECMWF Cyber to member states (except for Finland which uses Cyber). The software extracts subsets or total of the statistics file and writes every record in a formatted way. The file structure is basically the same as in the original statistics files. First the file contains one FDR (File Description Record) and one DDR (Data Description Record). The contents of these records is the same as in the original files. After that comes the data records, one for every observation, containing observation header, which is the same for all observation types, followed by observation body unique to every observation type. The format depends on the record type. This file can be sent to member states as normal output file. To make it easy to find automatically the begin and end of the file the first line of the file contains only the word "ECTODMI" and the last line only the word "END". The content of each kind of record is described in the following. The formats of records are given at the end of this chapter. A subroutine to read statistics files in formatted form can be found in the program library with the name "SISAB".

5.3.4 Contents of the statistics files

Below are listed the contents of the different types of records in the statistics files. Their format is described in section 5.3.5, which follows.

Statistics file. Observation header

Word	Type	Meaning
1	I	record length
2	I	old style analysis box number
3	I	record number
4	I	undefined
5	I	old style analysis box number
6	I	observation type
7	I	code type
8	I	latitude
9	I	longitude
10	I	observation date
11	I	synoptic time of obs
12	I	actual time of obs.
13		
14	I	db-key 1
15	I	db-key 2
16	I	station altitude

17	I	station characteristics
18	I	instrument type (0-23)
19.	I	instrument type (24-49)
20.	I	RDB-flags

Mass and wind statistics file body for SYNOP

Word	Type	Meaning
1	I	analysis pressure level
2	I	observed pressure
3	I	wind direction
4	I	wind speed
5	I	flags 1 (0-21)
6	I	flags 2 (36-53)
7	I	u departure from first guess
8	I	u departure from analysis
9	I	u departure from initialization
10	I	v
11	I	v
12	I	v
13	I	z
14	I	z
15	I	z
16	I	dq-information (0-26)
17	I	dq-information (27-44)
18	I	dq-information (45-59)

Mass and wind statistics file body for AIREP

Word	Type	Meaning
1	I	analysis pressure
2	I	observed pressure
3	I	wind direction
4	I	wind speed
5	I	temperature
6	I	height
7	I	flags 1 (0-29)
8	I	flags 2 (36-53)
9	I	departure from first guess
10	I	u departure from analysis
11	I	departure from initialization
12	I	
13	I	v
14	I	
15	I	
16	I	z
17	I	
18	I	dq-information (0-26)

19	I	dq-information (27-44)
20	I	dq-information (45-59)

Mass and wind statistics file body for SATOB

Word	Type	Meaning
1	I	analysis pressure level
2	I	observed pressure
3	I	wind direction
4	I	wind speed
5	I	cloud top temperature
6	I	flags 1 (0-17)
7	I	flags 2 (36-47)
8	I	departure from first guess
9	I	u departure from analysis
10	I	departure from initialization
11	I	
12	I	v
13	I	
14	I	dq-information (0-26)
15	I	dq-information (27-44)
16	I	dq-information (45-59)

Mass and wind statistics file body for DRIBU

Word	Type	Meaning
1	I	analysis pressure level
2	I	observed pressure
3	I	wind direction
4	I	wind speed
5	I	temperature
6	I	flags (0-23)
7	I	flags (36-53)
8	I	departure from first guess
9	I	u departure from analysis
10	I	departure from initialization
11	I	
12	I	v
13	I	
14	I	
15	I	z
16	I	
17	I	dq-information (0-26)
18	I	dq-information (27-44)
19	I	dq-information (45-59)

Mass and wind statistics file body for TEMP

Word	Type	Meaning
1	I	observed pressure
2	I	wind direction
3	I	wind speed
4	I	temperature
5	I	dew point
6	I	height (+1000)
7	I	flags (0-29)
8	I	flags (30-53)
9	I	u departure from first guess
10	I	u departure from analysis
11	I	departure from initialization
12	I	
13	I	v
14	I	
15	I	
16	I	z
17	I	
18	I	dq-information (0-26)
19	I	dq-information (27-44)
20	I	dq-information (45-59)

Mass and wind statistics file body for PILOT

Word	Type	Meaning
1	I	observed pressure
2	I	wind direction
3	I	wind speed
4	I	height (+1000)
5	I	flags (0-23)
6	I	flags (36-53)
7	I	departure from first guess
8	I	u departure from analysis
9	I	departure from initialization
10	I	
11	I	v
12	I	
13	I	
14	I	z
15	I	
16	I	dq-information (0-26)
17	I	dq-information (27-44)
18	I	dq-information (45-59)

Mass and wind statistics file body for SATEM or TOVS

Word	Type	Meaning
------	------	---------

1	I	analysis pressure level
2	I	reference level pressure
3	I	observed layer top pressure
4	I	mean temperature
5	I	precipitable water content
6	I	thickness or height
7	I	flags (0-29)
8	I	flags (36-41)
9	I	thickness
10	I	or
11	I	height departure
12	I	dq-information (0-26)
13	I	dq-information (27-44)
14	I	dq-information (45-59)

Humidity statistics file body for SYNOP

Word	Type	Meaning
1	I	analysis pressure level
2	I	observed pressure/ geopot
3	I	temperature
4	I	dew point temperature
5	I	ww
6	I	Cloud (0-22) as in AOF
7	I	Cloud (23-34) as in AOF
8	I	Empty
9	I	Empty
10	I	RDB Flags (0-21)
11	I	RDB Flags (36-41)
12	I	depart. from first guess
13	I	pwc depart from analysis
14	I	depart from init. anal
15	I	dq-information (0-26)
16	I	dq-information (27-44)
17	I	dq-information (45-59)

Humidity statistics file body for TEMP

Word	Type	Meaning
1	I	observed pressure
2	I	wind direction
3	I	wind speed
4	I	temperature
5	I	dew point
6	I	height (+1000)
7	I	empty
8	I	empty
9	I	RDB-flags (0-29)

10	I	RDB-flags (30-54)
11	I	departure from first guess
12	I	pwc departure from analysis
13	I	departure from initialization
14	I	dq-information (0-26)
15	I	dq-information (27-44)
16	I	dq-information (45-59)

Humidity statistics file body for SATEM or TOVS

Word	Type	Meaning
1	I	analysis pressure level
2	I	reference level pressure
3	I	observed layer top pressure
4	I	mean temperature
5	I	precipitable water content
6	I	thickness or height
7	I	empty
8	I	empty
9	I	flags (0-29)
10	I	flags (36-41)
11	I	departure from first guess
12	I	pwc departure from analysis
13	I	departure from initialization
14	I	dq-information (0-26)
15	I	dq-information (27-44)
16	I	dq-information (45-59)

5.3.5 FORTRAN formats for records in the statistics files

Below are listed the FORTRAN FORMAT statements used to write the statistics file information.

FDR

FORMAT(I3,I3,I2,I4,I2,I7,I5,I2,I2,I3,I3,I2,I2)

DDR

FORMAT(I3,I3,I2,I4,I2,I7,I5,I2,I3,I3,I7,I3,I4,I3,I3,I2,I5)

OBSERVATION HEADER

FORMAT(I4,I2,I5,I2,I2,I1,I3,I5,I5,I6,I4,I4,A5,I2,I2,I9,I2,I8,I8,I9)

MASS/WIND RECORDS

SYNOP

FORMAT(I6,I5,I3,I3,I7,I6, 3I7, 3I7, 3I7, I8, I6, I6)

AIREP

FORMAT(I6,I5,I3,I3,I4,I5, I9,I6, 3I7, 3I7, 3I7, I8, I6, I6)

SATOB
FORMAT(I6,I5,I3,I3,I4,I6,I4, 3I7, 3I7, I8, I6, I6)

DRIBU
FORMAT(I6,I5,I3,I3,I4,I8,I6, 3I7, 3I7, 3I7, I8, I6, I6)

TEMP
FORMAT(I6,I3,I3,I4,I4,I5, I9,I8, 3I7, 3I7, 3I7, I8, I6, I6)

PILOT
FORMAT(I6,I3,I3,I4,I8,I6, 3I7, 3I7, 3I7, I8, I6, I6)

SATEM
FORMAT(I6,I5,I5,I4,I3,I5, I9, I3, 3I7, I8, I6, I6)

PAOB
FORMAT(I6,I5,I5,I4,I3,I5, I9, I3, 3I7, I8, I6, I6)

HUMIDITY RECORDS

SYNOP
FORMAT(I6,I5,I4,I4,I2,I7,I4,I2,I2, I7, I3, 3I7, I8, I6, I6)

TEMP
FORMAT(I6,I3,I3,I4,I5,I4,I2,I2, I9, I3 ,3I7, I8, I6, I6)

SATEM
FORMAT(I6,I5,I5,I4,I9,I4,I2,I2, I9, I4, 3I7, I8, I6, I6)

6. Pre-processing

6.1 Horizontal interpolation

A set of procedures exist to convert the fields generated by a large-area model to boundary conditions for the HIRLAM forecast model, or to a set of data usable as initial conditions (to the forecast model if the analysis is missing; to the analysis scheme, as first-guess, if the appropriate forecast from an earlier analysis is missing). In practice, these procedures are applied to ECMWF products. They organise the selection of the right boundary file, the conversion of file formats and the horizontal interpolation to the HIRLAM grid. These transformations of a rather computational nature are described in this Section. The vertical interpolation, on the other hand, contains several decisions based on physical considerations. Therefore it is described separately (Section 6.2).

6.1.1 Selection of boundary files

The HIRLAM system requires the presence of boundary data that are valid at or after the validity time of the forecast. In an operational environment, this is achieved by requesting a sufficient number of forecast timesteps from operational dissemination by ECMWF. For experiments, usually carried out in 'hindcast' mode, the user may choose to use either ECMWF analyses or forecasts; the latter may be of varying length (e.g. +24 hours forecasts from the previous day versus +48 hours forecasts from the day before). The user specifies to the HIRLAM reference system what boundary files he wants to use by means of a so-called 'boundary strategy file', which gives, in a specified format, the type (analysis or forecast) and forecast lengths of the files to be used. Within the HIRLAM reference system, there is a script, `MakeStrategy`, to create a boundary strategy file for some commonly used 'strategies'. The user passes his preferred strategy to the system by means of the environment variable 'STRATEGY' (See also Section 10).

The first strategy, 'available', selects the available ECMWF products of the shortest forecast length, that are valid at or after the times required by the forecast model. It contains several fallback procedures to ensure the undisrupted continuation in case ECMWF products were not received in time. This strategy thus has been designed to be used for operational use of the HIRLAM system. In such circumstances, the ECMWF products should have been converted and interpolated to the HIRLAM file structure and grid by a process, that runs detached from the normal HIRLAM suite. Therefore, the strategy 'available' looks for ECMWF products in already converted form. Because this strategy does not make sense for experiments at ECMWF, it is not available for HIRLAM runs at the ECMWF computer system. The recovery algorithm is described in the inline documentation of the program '`mbdstr.x`', a copy of which is included in the Appendix.

The other strategies have been designed to be used in 'hindcast' situations. The HIRLAM reference system will look for ECMWF products in converted and interpolated form. If these are not available, the system will try to generate them from the ECMWF archives (`mars`). This generation can only succeed for runs at the ECMWF computer system. If the HIRLAM system is run on a local computer, the required ECMWF products must have been generated before at ECMWF by a 'mirror run', and transmitted to the local computer. The following strategies are available:

- analysis_only	Use ECMWF analyses throughout.
- operational	Mimic an operational situation, i.e. select ECMWF products that probably would have been available if the run had been done in an operational context. Note that in reality, due to e.g. transmission problems, the files may have been unavailable at that moment. It is repeated here that this strategy is not the one to be used for operational production. It is assumed that only 12 or 00 GMT runs are available.
- most_recent.	As 'operational', but allow the use of products from intermediate ECMWF runs (e.g. the 18 GMT analyses).
- parallel	Use the same boundaries as a different experiment, so as to run a parallel experiment. With this option, the boundary strategy file is simply copied from that other experiment (that could have been the operational HIRLAM run).

6.1.2 File format transformation

The program 'ec2asi.x' converts a file, containing one GRIB message per record, to the ASIMOF format, as used by the HIRLAM system (Section 5.1). It creates a list of eta-levels, in terms of A and B, for the full eta-levels, and adds that information to the ASIMOF file. Levels are recognised only if the information on that level is complete, i.e., if the input file contains fields of the two horizontal wind components, temperature and humidity, for that level. A manpage describes the program. It is available from the HIRLAM system through the documentation extraction facilities (Section 10.4.8). A copy of the manpage is added in the Appendix of this Section.

6.1.3 Horizontal interpolation

The horizontal interpolation of the boundary fields is performed by the program 'intp.x'. A manpage to this program is available through the HIRLAM reference system (see Section 10.3), a copy is included in the Appendix to this Section. The action of the program intp.x is controlled by namelists. Within the HIRLAM reference system its application to the boundary fields is through the script Bdries (Section 10). The default mode of the operation is to apply bi-linear interpolation to the dynamic variables (temperature, wind components and humidity at all levels, the logarithm of surface pressure and orography). The program 'rotates' the coordinate axes, such that the wind components describe the wind vector relative to the coordinate axes of the grid that the interpolation is to

Warning:

Because intp.x is applied frequently, it has been optimised for HIRLAM use. As a consequence, in the default mode of operation, it is avoided to check if the HIRLAM area is fully contained within the ECMWF area. If this condition is violated, the program will happily start to extrapolate the ECMWF fields to obtain values in all gridpoints, even those outside the ECMWF grid. This extrapolation will cause insurmountable problems (like noise) in the forecast model. So the user would do wiser to choose the ECMWF area large enough!

The treatment of surface pressure and orography requires further explanation. In the pressure gradient term in the dynamical equations there is an almost exact cancellation between the logarithm of surface pressure (lnps) and orography (phis). It is essential that this cancellation is

preserved during the interpolation process.

On one hand, this means that the gridpoint values of these two fields in the boundary model (ECMWF) be obtained from the model variables in exactly the same way. On the other hand, these two fields must be interpolated horizontally in exactly the same way. This latter requirement explains why `intp.x` is applied to the logarithm of surface pressure and not to surface pressure itself.

The consistency requirement leads to a complicated procedure to obtain the ECMWF orography. The model at ECMWF is spectral, i.e. the model variables are spectral components. The dissemination product for `lnps`, however, is gridded to a regular latitude / longitude grid of user specified resolution. Therefore, the orography that is to be interpolated to the HIRLAM grid should be a gridpoint representation of the spectral orography, obtained in exactly the same way as `lnps` from its spectral coefficients. Furthermore, the gridpoints at which the orography is given, should coincide with those of `lnps`. Within the reference system this is achieved by the script `Extr_oro_gp`. This script consists of two parts, the first applies the ECMWF spectral transformation routine (`SPTOGP`), which ECMWF uses to transform `lnps` from spectral to gridpoint space in their dissemination and archiving procedures, to the spectral representation of the ECMWF orography. This spectral representation has been made available by ECMWF to HIRLAM users for two triangular truncations, T213 and T106. The user of the HIRLAM system should ensure that he specifies the proper value of the truncation that is valid for his ECMWF products (through the environment variable `TRUNCN`; default is 213). The transformation is to a grid with gridpoints that coincide with those to be used to retrieve `lnps` from ECMWF. The user must communicate that grid to `Extr_oro_gp` as well (through the environment variables `DXEC` etc). The normal action of the first part of `Extr_oro_gp` is to produce a file in standard format (ASCII), with values of the orography on a global grid with specified resolution. (Warning: If this resolution is very high, you may find that array sizes in `Extr_oro_gp` are not sufficient for global coverage. Then please think of some solution).

The ASCII file is used in the second part of `Extr_oro_gp` to construct a file in ASIMOF format, which can be input to `intp.x`.

If the reference system is run at ECMWF, `Extr_oro_gp` will be invoked automatically. If you want to run the HIRLAM system on a local computer, you should run the first part of `Extr_oro_gp` at ECMWF (to ensure the exact correspondence of the transformation of orography to that of `lnps`), then transmit the resulting ASCII file to your local system, and there run the second part of `Extr_oro_gp`. This procedure has to be repeated every time you want to change the spectral resolution of the ECMWF model used for boundary generation, or the gridpoint resolution at which you request the boundary data from ECMWF. If you applied `Extr_oro_gp` to generate a global grid (as per default) you do not have to go through this procedure when you change the size or location of the area you request from ECMWF, provided you stick to the same grid distances. Also, you can use the same gridpoint values of orography when you change the HIRLAM resolution. The program `intp.x` ensures the consistency of the interpolation of `lnps` to that of `phis`.

The HIRLAM reference system requires the presence of surface pressure, rather than its logarithm, in the history files (see Section 6.1). Therefore, the horizontal interpolation program `intp.x` is followed by the program '`explnp.x`', to add surface pressure to the files, calculated from the logarithm as already available within those files.

6.1.4 Bitmaps

If the data from the larger area are only used to generate boundaries, and not as a first-guess or initial conditions to the forecast (see also the introduction to this Section), then there is no need to import complete fields. In these circumstances, transmission costs from ECMWF can be reduced by only transmitting the ECMWF data points that are in or near the HIRLAM area boundaries. ECMWF support this form of transmission: Member states may request data according to a so-called 'bitmap'. The format in which such a bitmap is to be specified may be rather tedious to generate. To facilitate this, the script 'Bitmap' was developed. The output generated by 'Whatis Bitmap' is included in the Appendix (see also Section 10.4.3). The algorithm to select the points in or near the boundary zone is closely related to the algorithm used in the horizontal interpolation scheme. All points from the horizontal interpolation is performed must at least be included. Also, some extra points are needed due to the de-staggering algorithms used in the vertical interpolation scheme. For ease of implementation, it was decided that 'Bitmap' should generate one bitmap, valid for un- and staggered variables alike. Therefore, the number of points to be selected is slightly higher than what would have been obtained with three separate bitmaps for temperature and the two wind components.

The script Bitmap will be run very occasionally only. It was possible to include a check on the validity of the ECMWF area as compared to that of HIRLAM. If the ECMWF area does not fully cover the HIRLAM boundary zone (plus neighbouring points), the script will fail gracefully. (Note that one may conceive of a combination of a HIRLAM area and an ECMWF area, such that the HIRLAM boundary zone is, but the full HIRLAM area is not, fully contained within the ECMWF area.)

Appendix: Manpages and inline documentation

```
!!execute InlineDoc mbdstr
!!execute Man ec2asi
!!execute Man intp
!!execute Whatis Bitmap
```

6.2 Vertical interpolation from pressure levels to HIRLAM hybrid levels

Lateral boundary fields are normally interpolated from ECMWF hybrid levels directly to HIRLAM hybrid levels, following the method described in section 6.3 below. For some purposes however, there may be a need for an interpolation from pressure levels to the HIRLAM hybrid levels. The algorithms for this vertical interpolation are essentially identical to those used operationally by the SMHI and the DMI. The vertical interpolation is carried out through the following steps:

- (1) Calculation of surface pressure in the mass field grid-points from geopotentials of three subsequent pressure levels.
- (2) Horizontal interpolation of surface pressure from the mass-field grid-points to the staggered wind-field grid-points.
- (3) Vertical interpolation of geopotential from the pressure levels to the model hybrid "half" levels by cubic splines in $\ln(\text{pressure})$

- (4) Calculation of virtual temperatures of the model hybrid "full" levels from the geopotentials by the aid of the hydrostatic equation.
- (5) Vertical interpolation of wind components and relative humidity from the pressure levels to the model hybrid "full" levels by linear interpolation in ln (pressure).
- (6) Computation of temperatures and conversion of the relative humidities to specific humidities of the model hybrid "full" levels.

More details of the algorithms are described in sections 6.2.1-6.2.3 below.

6.2.1 Notations

NPRE	Number of pressure levels
ϕ_i^{pre} , i=1,NPRE	Geopotential of the pressure levels
R_i^{pre} , i=1,NPRE	Relative humidities of the pressure levels
T_{vi}^{pre} , i=1,NPRE	Virtual temperatures of the pressure levels
U_i^{pre} , i=1,NPRE	U-components of the pressure levels
V_i^{pre} , i=1,NPRE	V-components of the pressure levels
P_i^{pre} , i=1,NPRE	Pressures of the pressure levels
ϕ_s^{hir}	HIRLAM surface geopotential
P_s^{hir}	HIRLAM surface pressure
NHIR	Number of HIRLAM model levels
T_i^{hir} , i=1,NHIR	HIRLAM temperature profile
Q_i^{hir} , i=1,NHIR	HIRLAM specific humidity profile

R_i^{hir} ,i=1,NHIR	HIRLAM relative humidity profile
T_{vi}^{hir} ,i=1,NHIR	HIRLAM virtual temperature profile
U_i^{hir} ,i=1,NHIR	HIRLAM U-component profile
V_i^{hir} ,i=1,NHIR	HIRLAM V-component profile
$P_i^{hir}=A_i^{hir}+B_i^{hir}P_s^{hir}$,i=1,NHIR	Pressures of the HIRLAM model levels
A_i^{hir} , B_i^{hir} ,i=1,NHIR	Coefficients defining the HIRLAM forecast model hybrid levels
$\phi_{i+1/2}^{hi}$,i=0,NHIR	Geopotentials of the HIRLAM i+1/2 half levels
R_d	Gas constant for dry air
R_v	Gas constant for water vapour
G	Acceleration of gravity

6.2.2 Calculation of surface pressure

For calculation of pressure P_s^{hir} at the HIRLAM surface level with surface geopotential ϕ_s^{hir} we will utilize the geopotentials ϕ_{i-1}^{pre} , ϕ_i^{pre} and ϕ_{i+1}^{pre} of three subsequent pressure levels $P_{i-1}^{pre} < P_i^{pre} < P_{i+1}^{pre}$. The calculation of surface pressure is carried out differently depending on the position of the surface level in relation to the lowest pressure level.

$$(I) : \underline{\phi_i^{pre} > \phi_s^{hir} \geq \phi_{i+1}^{pre}} :$$

In case the HIRLAM surface level is above the lowest pressure level, the surface pressure is calculated by assuming a virtual temperature lapse rate of -0.0065 K/m below the surface and

another lapse rate DT above the surface within the two layers between the three pressure levels. The assumption of a linear virtual temperature variation in height and the hydrostatic equation will give:

$$\begin{aligned}
 T_{vi-1}^{pre} &= T_{vs}^{hir} + DT (\phi_{i-1}^{pre} - \phi_s^{hir}) / G \\
 T_{vi}^{pre} &= T_{vs}^{hir} + DT (\phi_i^{pre} - \phi_s^{hir}) / G \\
 T_{vi+1}^{pre} &= T_{vs}^{hir} - 0.0065 (\phi_{i+1}^{pre} - \phi_s^{hir}) / G \\
 \phi_{i-1}^{pre} - \phi_i^{pre} &= 0.5R_d (T_{vi}^{pre} + T_{vi-1}^{pre}) \ln (P_i^{pre} / P_{i-1}^{pre}) \\
 \phi_i^{pre} - \phi_s^{hir} &= 0.5R_d (T_{vi}^{pre} + T_{vs}^{pre}) \ln (P_s^{hir} / P_i^{pre}) \\
 \phi_s^{hir} - \phi_{i+1}^{pre} &= 0.5R_d (T_{vs}^{hir} + T_{vi+1}^{pre}) \ln (P_{i+1}^{pre} / P_s^{hir})
 \end{aligned}$$

The solution of these equations for P_s^{hir} is given by

$$P_s^{hir} = EXR \frac{-D_1 + \sqrt{(D_1)^2 - 4D_2 D_0}}{2D_2}$$

where

$$\begin{aligned}
 D_0 &= -C_6 \ln (P_i^{pre}) - C_8 \ln (P_{i+1}^{pre}) + C_9 \ln (P_i^{pre}) \ln (P_{i+1}^{pre}) \\
 D_1 &= C_6 + C_8 - C_9 \ln (P_{i+1}^{pre}) - C_9 \ln (P_i^{pre}) \\
 D_2 &= C_9
 \end{aligned}$$

and

$$C_1 = \frac{G(\phi_{i-1}^{pre} - \phi_i^{pre})}{R_d (\phi_s^{hir} - \phi_{i-1}^{pre}/2 - \phi_i^{pre}/2) \ln(P_i^{pre}/P_{i-1}^{pre})}$$

$$C_2 = -\frac{G}{\phi_s^{hir} - \phi_i^{pre}/2 - \phi_{i-1}^{pre}/2}$$

$$C_3 = (\phi_i^{pre} - \phi_s^{hir})$$

$$C_4 = R_d (1 + C_2 (\phi_s^{hir} - \phi_i^{pre}) / 2G)$$

$$C_5 = R_d C_1 (\phi_s^{hir} - \phi_i^{pre}) / 2G$$

$$C_6 = \phi_s^{hir} - \phi_{i+1}^{pre}$$

$$C_7 = 0.0065 R_d (\phi_s^{hir} - \phi_{i+1}^{pre}) / 2G$$

$$C_8 = \frac{R_d C_3}{C_4}$$

$$C_9 = C_7 - \frac{R_d C_5}{C_4}$$

(I) : $\phi_s^{hir} < \phi_{i+1}^{pre}$:

In case the HIRLAM surface level is below the lowest pressure level, the surface pressure is calculated by assuming a virtual temperature lapse rate DT within the two layers between the three pressure levels. The assumption of a linear virtual temperature variation in height and the hydrostatic equation will give:

$$T_{vi-1}^{pre} = T_{vi}^{pre} + DT (\phi_{i-1}^{pre} - \phi_s^{pre}) / G$$

$$T_{vi+1}^{pre} = T_{vi}^{pre} + DT (\phi_{i+1}^{pre} - \phi_s^{pre}) / G$$

$$T_{vs}^{hir} = T_{vi}^{pre} + DT (\phi_s^{hir} - \phi_i^{pre}) / G$$

$$\phi_{i-1}^{pre} - \phi_i^{pre} = 0.5 R_d (T_{vi}^{pre} + T_{vi-1}^{pre}) \ln (P_i^{pre}/P_{i-1}^{pre})$$

$$\phi_i^{pre} - \phi_{i+1}^{pre} = 0.5 R_d (T_{vi}^{pre} + T_{vi+1}^{pre}) \ln (P_{i+1}^{pre}/P_i^{pre})$$

$$\phi_{i+1}^{pre} - \phi_s^{hir} = 0.5 R_d (T_{vs}^{hir} + T_{vi+1}^{pre}) \ln (P_s^{hir}/P_{i+1}^{pre})$$

The solution of these equations for P_s^{hir} is given by:

$$P_s^{hir} = P_{i+1}^{pre} \text{ EXR} \frac{2(\phi_{i+1}^{pre} - \phi_s^{hir})}{R_d(T_{vi+1}^{pre} + T_{vs}^{hir})}$$

where T_{vi+1}^{pre} and T_{vs}^{hir} are calculated from the formulas given above and

$$DT = \frac{C_3 - C_1}{C_4 - C_2}$$

$$T_{vi}^{pre} = \frac{C_1 C_4 - C_2 C_3}{R_d(C_4 - C_2)}$$

and

$$C_1 = \frac{\phi_{i-1}^{pre} - \phi_i^{pre}}{\ln(P_i^{pre}/P_{i-1}^{pre})}$$

$$C_2 = R_d(\phi_{i-1}^{pre} - \phi_i^{pre})/2G$$

$$C_3 = \frac{\phi_i^{pre} - \phi_{i+1}^{pre}}{\ln(P_{i+1}^{pre}/P_i^{pre})}$$

$$C_4 = R_d(\phi_{i+1}^{pre} - \phi_i^{pre})/2G$$

6.2.3 Vertical interpolations

Once the surface pressures P_s^{hir} of the HIRLAM mass-field gridpoints are available, surface pressures of the staggered wind-field points are obtained by bi-linear interpolation. Then the pressures of model hybrid "full" as well as "half" levels are computed, see notations above.

The geopotentials of the HIRLAM model "half" levels are interpolated from pressure levels by means of cubic splines. General subroutines from the NAG-library, E02BAF and E02BBF, are utilized for the cubic spline interpolation with $\ln(\text{pressure})$ used as the vertical coordinate. In case of extrapolation below the lowest pressure level or above the highest pressure level, linear extrapolation is used.

Virtual temperatures of the HIRLAM model "full" levels are obtained from the "half" level

geopotentials by the hydrostatic equation:

$$T_{vi}^{hir} = -\frac{(\phi_{i+1/2}^{hir} - \phi_{i-1/2})}{R_d \ln(P_{i+1/2}^{hir}/P_{i-1/2}^{hir})} \text{ for } i=1, NHIR$$

The relative humidities and the wind components of the HIRLAM "full" levels are interpolated from the pressure levels by means of linear interpolation in $\ln(\text{pressure})$. Temperatures of the HIRLAM full levels are calculated from the virtual temperatures and the relative humidities by means of an iterative scheme:

$$(T_i^{hir})^{\epsilon=0} = T_{vi}^{hir}$$

$$(T_i^{hir})^\epsilon = T_{vi}^{hir} - \frac{(R_v/R_d - 1) R_i^{hir} (T_i^{hir})^{\epsilon-1}}{R_v/R_d P_i^{hir}/E_s ((T_i^{hir})^{\epsilon-1}) - (R_v/R_d - 1)}$$

and

$$T_i^{hir} = (T_i^{hir})^{\epsilon=5}$$

The water vapour saturation pressure $E_s(T)$ is calculated in the same way as in the forecast model and in the analysis (see section 4.1.6).

The relative humidities of the HIRLAM model "full" levels, finally, are converted to specific humidities:

$$Q_i^{hir} = \frac{R_i^{hir}}{R_v/R_d P_i^{hir}/E_s (T_i^{hir}) - (R_v/R_d - 1)}$$

6.2.4 Program organization and data organization

The program system for vertical interpolation from pressure levels to HIRLAM model hybrid levels consists of the following subroutines:

VINT: Main program which handles all input/output and organizes the calculations.

PTOETA: Main subroutine for the vertical interpolation.

PSTAR	Surface pressure calculations.
SPLINT.	Cubic spline interpolations.
LININT:	Linear interpolation
TVTOT	Temperature is computed from virtual temperature, relative humidity and pressure
RQ:	Specific humidity is computed from temperature, relative humidity and pressure

There are two namelist-statements, read by VINT, that determines the flow of input/output and computations:

NAMELIST/NAMV2E/ ILUIN1,ILUIN2,ILUUT,ILUDIR,NLURPL,ILURPL,
 LSPLIN,LPRINT,IPRINT,JPRINT,ILUNL,
 NCLIM,NWMOCL,NLTPCL,SLEVCL

NAMELIST/NAMETA/ NELEV,AHALF,BHALF,REFPRE

The variables in these namelists have the following meaning

Variable(Type)	Meaning
ILUIN1 (I)	File unit number of the input pressure level fields
ILUIN2 (I)	File unit number of climatological fields including the surface geopotential field.
ILUUT (I)	File unit number of the output model level data
ILUDIR (I)	File unit number of the direct access file to be used by the reading/ungribbing routine GREAD, see chapter 5
NLURPL (I)	Number of extra GRIB files to be searched for fields to replace climatological field read from file unit ILUIN2
ILURPL(NLURPL) (I)	File unit numbers of the extra GRIB files
LSPLIN (L)	Switch for spline interpolation(T.) or linear interpolation(.F)
LPRINT (L)	Switch for obtaining extra printing during execution of VINT (T for extra printing)
IPRINT (I)	X-coordinate of the gridpoint for printing
JPRINT (I)	Y-coordinate of the gridpoint for printing
ILUNL (I)	File unit number of namelist NAMETA
NCLIM (I)	Number of climatological fields to be read from file unit ILUIN2
NWMOCL(NCLIM) (I)	Variable indicators of the climatological fields
NLTPCL(NCLIM) (I)	Type of level of the climatological fields
SLEVCL(NCLIM) (R)	Vertical coordinate values of the climatological fields
NELEV (I)	Number of HIRLAM model levels
AHALF(NELEV+1) (R)	A-coefficients defining the HIRLAM model "half" levels
BHALF(NELEV+1) (R)	B-coefficients defining the HIRLAM model "half" levels
REFPRE (R)	Reference pressure of the HIRLAM vertical eta-coordinate system

The namelist NAMV2E is read from file unit 5. Variables describing the climatological fields to be utilized are in agreement with those specified for input/output and archiving of fields (chapter 5). Note that the surface geopotential is read by default from file unit ILUIN2 and should not be specified in the list of climatological fields.

6.3 Vertical interpolation from one hybrid level system to another hybrid level system

A procedure for deriving HIRLAM boundary condition fields from ECMWF forecast model data is described below. Vertical interpolation is carried out directly from the ECMWF model level data to the HIRLAM model levels and the interpolation methods are based on ideas of Majewski(1985). The planetary boundary layer is treated separately during the vertical interpolation with the emphasis to preserve stability properties of the temperature profile close to the earth's surface also when there are large differences between the HIRLAM and the ECMWF orographies. Tension splines are utilized for the vertical interpolation, while bi-linear interpolation is used in the horizontal.

The interpolation is carried out in the following steps

- Step I: The ECMWF temperature, humidity and wind component fields of the hybrid model levels together with the surface pressure and the surface geopotential fields are evaluated from their spectral representations to a regular lat.-long. grid
- Step II: The ECMWF fields are interpolated bi-linearly in the horizontal from the regular lat.-long. grid to the transformed HIRLAM lat.-long. grid. See section 6.1 for details.
- Step III: A preliminary HIRLAM surface pressure field is estimated by the aid of the HIRLAM surface geopotential field, the ECMWF surface pressure, surface geopotential, temperature and humidity fields. From this preliminary surface pressure field, preliminary pressures of the HIRLAM model levels are obtained
- Step IV: Vertical interpolation is carried out separately in the planetary boundary layer and above the planetary boundary layer. Above the boundary layer, vertical interpolation is carried out for temperature, relative humidity and wind components with logarithm of pressure as the vertical coordinate. In the boundary layer, potential temperature, relative humidity and wind components are interpolated vertically in a ground-following vertical coordinate system. Potential temperatures in the boundary layer, finally, are adjusted by a constant value to coincide with potential temperatures obtained from the free atmosphere vertical interpolation at the top of the boundary layer. In this way, stability in the planetary boundary layer is preserved
- Step V: A final HIRLAM surface pressure field is computed by correcting its preliminary value in such a way that the geopotentials of pressure surfaces well above the highest mountains are equal to the same geopotentials obtained from the original ECMWF fields. The aim of this correction is to preserve the mass-wind balances in the free atmosphere.

Steps (III) - (V) of the scheme are described in more detail below

6.3.1 Notations

φ_s^{ec}	ECMWF surface geopotential
P_s^{ec}	ECMWF surface pressure
NEC	Number of ECMWF model levels
T_i^{ec} , i=1,NEC	ECMWF temperature profile
θ_i^{ec} , i=1,NEC	ECMWF potential temperature profile
Q_i^{ec} , i=1,NEC	ECMWF specific humidity profile
R_i^{ec} , i=1,NEC	ECMWF relative humidity profile
T_{vi}^{ec} , i=1,NEC	ECMWF virtual temperature profile
U_i^{ec} , i=1,NEC	ECMWF U-component profile
V_i^{ec} , i=1,NEC	ECMWF V-component profile
$P_i^{ec} = A_i^{ec} + B_i^{ec} P_s^{ec}$, i=1,NEC	Pressures of the ECMWF model levels
$\eta_i^{ec} = A_i^{ec} / P_{ref}^{ec} + B_i^{ec}$, i=1,NEC	-coordinate values of the ECMWF levels
$P_{ref}^{ec}, A_i^{ec}, B_i^{ec}$, i=1,NEC	Constant reference pressure and coefficients defining the ECMWF model levels
$\varphi_{i+1/2}^{ec}$, i=0,NEC	Geopotentials of the ECMWF +1/2 half levels

Φ_s^{hir}	HIRLAM surface geopotential
\hat{P}_s^{hir}	HIRLAM preliminary surface pressure
P_s^{hir}	HIRLAM surface pressure
NHIR	Number of HIRLAM model levels
T_i^{hir} , i=1,NHIR	HIRLAM temperature profile
Θ_i^{hir} , i=1,NHIR	HIRLAM potential temperature profile
Q_i^{hir} , i=1,NHIR	HIRLAM specific humidity profile
R_i^{hir} , i=1,NHIR	HIRLAM relative humidity profile
T_{vi}^{hir} , i=1,NHIR	HIRLAM virtual temperature profile
U_i^{hir} , i=1,NHIR	HIRLAM U-component profile
V_i^{hir} , i=1,NHIR	HIRLAM V-component profile
$\hat{P}_i^{hir} = A_i^{hir} + B_i^{hir} \hat{P}_s^{hir}$, i=1,NHIR	Preliminary pressures of the HIRLAM model levels
$P_i^{hir} = A_i^{hir} + B_i^{hir} P_s^{hir}$	Pressures of the HIRLAM model levels
$\eta_i^{hir} = A_i^{hir} / P_{ref}^{hir} + B_i^{hir}$, i=1,NHIR	η -coordinate values of the HIRLAM model levels
$P_{ref}^{hir}, A_i^{hir}, B_i^{hir}$, i=1,NHIR	Constant reference pressure and coefficients defining the HIRLAM forecast model levels

$\phi_{i+1/2}^{hir}$, i=0,NHIR

Geopotentials of the HIRLAM +1/2 half levels

η_{pblt}

η -value defining the top of the planetary boundary layer
(=0.8)

P_{ref}

Reference pressure used in calculation of potential temperature (=1000hPa)

P_1

Reference pressure level used for fixing HIRLAM geopotentials to ECMWF geopotentials (=500hPa)

6.3.2 Estimation of a preliminary HIRLAM surface pressure

A preliminary HIRLAM surface pressure value is estimated by an integration of the hydrostatic equation from the ECMWF surface level to the HIRLAM surface level using a virtual temperature profile expressed as a linear function of $\ln(\text{pressure})$ in the vicinity of the HIRLAM surface level. In case the HIRLAM surface level is below the ECMWF surface level, the virtual temperature profile is obtained by extrapolation from the 3 lowest ECMWF levels (Figure 6.3.2.1(A)), else the temperature profile is obtained by interpolation from the three closest ECMWF levels to the HIRLAM surface level (Figure 6.3.2.1(B)). The linear virtual temperature profile is obtained by regression from the three selected ECMWF temperatures.

Figure 6.3.2.1: (A) Extrapolation and (B) Interpolation

We introduce

$$T_v(P) = B \ln(P) + C$$

where the coefficients B and C are to be determined by minimization of the mean square error E

$$E = 1/3 \sum_{i=k-1}^{k+1} (T_{vi}^{ec} - B \ln(P_i^{ec}) - C)^2$$

Here, in the case of extrapolation below the ECMWF surface level the three temperature levels are defined by k=NEC-1 while in the case of interpolation, k is selected such that

$$\Phi_{k+1/2}^{ec} \leq \Phi_s^{hir} \leq \Phi_{k-1/2}^{ec}$$

The condition for minimum of E is $\partial E / \partial B = \partial E / \partial C = 0$ and we will have the following values of the coefficients B and C:

$$B = \frac{\sum_{i=k-1}^{k+1} T_{vi}^{ec} \ln P_i^{ec} - 1/3 (\sum_{i=k-1}^{k+1} T_{vi}^{ec}) (\sum_{i=k-1}^{k+1} \ln P_i^{ec})}{\sum_{i=k-1}^{k+1} (\ln P_i^{ec})^2 - 1/3 (\sum_{i=k-1}^{k+1} \ln P_i^{ec})^2}$$

$$C = 1/3 \sum_{i=k-1}^{k+1} T_{vi}^{ec} - B/3 \sum_{i=k-1}^{k+1} \ln P_i^{ec}$$

In case of extrapolation below the ECMWF surface level, integration of the hydrostatic equation between ECMWF and HIRLAM surface levels will give

$$\phi_s^{ec} - \phi_s^{hir} = - \int_{\hat{P}_s^{hir}}^{P_s^{ec}} R_d T_v(P) d\ln P$$

and after some manipulation we will have when $B \neq 0$

$$\ln \hat{P}_s^{hir} = \frac{-C' + \sqrt{C^2 + B(B(\ln P_s^{ec})^2 + 2Cl \ln P_s^{ec} + 2(\phi_s^{ec} - \phi_s^{hir})/R)}}{B}$$

and when $B=0$

$$\ln \hat{P}_s^{hir} = \ln P_s^{ec} + \frac{\phi_s^{ec} - \phi_s^{hir}}{RC}$$

In case the HIRLAM surface level is above the ECMWF surface level, the hydrostatic equation is integrated from the ECMWF $k+1/2$ level to obtain the preliminary HIRLAM surface pressure and we will obtain correspondingly

$$\ln \hat{P}_s^{hir} = \frac{-C + \sqrt{C^2 + B(B(\ln P_{k+1/2}^{ec})^2 + 2Cl \ln P_{k+1/2}^{ec} + 2(\phi_{k+1/2}^{ec} - \phi_s^{hir})/R)}}{B}$$

when $B \neq 0$ and

$$\ln \hat{P}_s^{hir} = \ln P_{k+1/2}^{ec} + \frac{\phi_{k+1/2}^{ec} - \phi_s^{hir}}{RC}$$

when $B=0$

6.3.3 Vertical interpolation of temperature, humidity and wind components.

In order to preserve the stability structure in the planetary boundary layer, vertical interpolation will be carried out with the terrain-following vertical coordinate $\ln(\)$ within this layer. Above the boundary layer, $\ln(\text{pressure})$ will be used as vertical coordinate for the interpolation. The preliminary HIRLAM surface pressure \hat{P}_s^{hir} obtained as described above, is thereby used to calculate \hat{P}_s^{hir} , $i=1, \text{NHIR}$ of the HIRLAM levels. To obtain the HIRLAM values on a level i by vertical interpolation with $\ln(\text{pressure})$ as the vertical coordinate, the following two conditions should be fulfilled:

$$\eta_i^{hir} < \eta_i^{pblt}$$

and

$$\hat{P}_i^{hir} < \eta_i^{pblt} P_s^{ec}$$

The second condition is added in order to avoid that two boundary layer structures are inserted into the HIRLAM profiles in case of deeper valleys in the HIRLAM orography compared to the ECMWF orography (Figure 6.3.2.1 (A)). Assume that two conditions above are fulfilled for $i=1, \text{imax}$. Then temperatures T_i^{hir} , relative humidities R_i^{hir} and wind components U_i^{hir}, V_i^{hir} for $i=1, \text{imax}$ are obtained by tension spline interpolation from the corresponding ECMWF values with $\ln(\text{pressure})$ as the vertical coordinate. For levels $i=\text{imax}, \text{NHIR}$, which are considered to be the HIRLAM levels within and at the top of the boundary layer, preliminary potential temperatures Θ_i^{hir} , relative humidities R_i^{hir} and wind components U_i^{hir}, V_i^{hir} are obtained by tension spline interpolation from the ECMWF values with \ln as the vertical coordinate. In order to obtain a continuous temperature profile at the top of the boundary layer, a constant correction is applied to the preliminary potential temperatures within the HIRLAM boundary layer, for $i=\text{imax}, \text{NHIR}$:

$$\Theta_i^{hir} = \Theta_i^{hir} + (T_{\text{imax}}^{hir} \left(\frac{P_{\text{ref}}}{\hat{P}_{\text{imax}}^{hir}} \right) R/C_p - \Theta_{\text{imax}}^{hir})$$

Finally, the potential temperatures of the HIRLAM boundary layer are converted to temperatures and the results of the two parts of the vertical interpolation are merged by taking values from the pressure-based interpolation for $i=1, \text{imax}$ and values from the η -based vertical interpolation

for i=imax+1,NHIR

6.3.4 Computation of the final HIRLAM surface pressure.

The adjustment of the preliminary HIRLAM surface pressure \hat{P}_s^{hir} is done in such a way that the resulting geopotential $\phi^{hir}(P_1)$ of a reference pressure level P_1 is as close as possible to the corresponding ECMWF geopotential $\phi^{ec}(P_1)$. (For the present HIRLAM areas P= 500 hPa has been used). This surface pressure is obtained through the following computational steps

- 1 Obtain geopotential $\phi_{i+1/2}^{ec}$ of the ECMWF model half levels by integration of the hydrostatic equation
- 2 Obtain $\phi^{ec}(P_1)$ by linear interpolation in ln(pressure)
- 3 Obtain preliminary geopotential $\phi_{i+1/2}^{hir}$ of the HIRLAM model half levels by integration of the hydrostatic equation starting from the preliminary HIRLAM surface pressure P at the surface
- 4 Obtain $\phi^{hir}(P_1)$ by linear interpolation in ln(pressure)
- 5 Correct the preliminary HIRLAM surface pressure by adding or subtracting mass at the bottom of the HIRLAM vertical column in such a way that the thickness of this added(subtracted) mass is equal to the difference between the two geopotential values $\phi^{ec}(P_1)$ and $\phi^{hir}(P_1)$. The virtual temperature of this added mass is assumed to be the same as that of the lowest HIRLAM model level T_{vhir}^{hir} . This will give:

$$P_s^{hir} = \hat{P}_s^{hir} \exp\left(\frac{\phi^{ec}(P_1) - \phi^{hir}(P_1)}{R T_{vhir}^{hir}}\right)$$

-

The corrections applied to \hat{P}_s^{hir} obtain P_s^{hir} are generally quite small, of the order of 0.1 hPa, while larger values of the order 1-2 hPa may occur in areas of large orography gradients (e.g. Greenland). Because of this, no attempt is done to re-interpolate the HIRLAM temperature,

humidity and wind component profiles to account for the re-definition of the pressures of the HIRLAM model levels

6.3.6 Program organization and data organization.

The program system for vertical interpolation from one hybrid level system to another hybrid level system consists of the following subroutines

VINETA	Main program which handles all input/output organizes the calculations
ETAETA.	Main subroutine for surface pressure computation and vertical interpolation
EVALTS, FITTS, INIVER	Tension spline subroutines, the same as used in the ECMWF analysis scheme
RHUM	Relative humidity is computed from specific humidity, temperature and pressure
RHTOQ.	Specific humidity is computed from temperature, relative humidity and pressure

Water vapour saturation pressure formulas as well as the iterative algorithm to determine temperature from virtual temperature are the same as described in section 6.2.

There are two namelist-statements, read by VINETA, that determines the flow of input/output and computation:

NAMELIST/NAMV2E/ ILUIN1,ILUIN2,ILUUT,ILUDIR,NLURPL,ILURPL,
LPRINT,IPRINT,JPRINT,ILUNL,NCLIM,NWMOCL,NLTPCL,SLEVCL

NAMELIST/NAMETA/ NELEV,AHALF,BHALF,REFPRE

The variables in these namelists have the following meaning

Variable(Type)	Meaning
ILUIN1 (I)	File unit number of the ECMWF hybrid level fields
ILUIN2 (I)	File unit number of climatological fields including the surface geopotential field
ILUUT (I)	File unit number of the output model level data
ILUDIR (I)	File unit number of the direct access file to be used by the reading/ungribbing routine GREAD, see chapter 5
NLURPL (I)	Number of extra GRIB files to be searched for fields to replace climatological field read from file unit ILUIN2
ILURPL(NLURPL) (I)	File unit numbers of the extra GRIB files
LPRINT (L)	Switch for obtaining extra printing during execution of VINETA (T for extra printing)
IPRINT (I)	X-coordinate of the gridpoint for printing
JPRINT (I)	Y-coordinate of the gridpoint for printing

ILUNL	File unit number of namelist NAMETA
NCLIM (I)	Number of climatological fields to be read from file unit ILUIN2
NWMOCL(NCLIM) (I)	Variable indicators of the climatological fields
NLTPCL(NCLIM) (I)	Type of level of the climatological fields
SLEVCL(NCLIM) (R)	Vertical coordinate value of the climatological fields
NELEV (I)	Number of HIRLAM model levels
AHALF(NELEV+1) (R)	A-coefficients defining the HIRLAM model "half" levels
BHALF(NELEV+1) (R)	B-coefficients defining the HIRLAM model "half" levels
REFPRE (R)	Reference pressure of the HIRLAM vertical -coordinate system

The namelist NAMV2E is read from file unit 5. Variables describing the climatological fields to be utilized are in agreement with those specified for input/output and archiving of fields (chapter 5). Note that the surface geopotential is read by default from file unit ILUIN2 and should not be specified in the list of climatological fields.

7. Postprocessing

The postprocessing package has two main purposes. First it provides an interface between the model slab file and the model (archived) field file, and secondly it postprocesses the slab file into a pressure level field file. The former only changes the data organization from the slab format (one latitude line of data, including all variables, at all model levels in a record) into the field format (one horizontal field of one variable in a record), and packs the data with the space-saving GRIB format. The latter computes the variables which are not model variables, interpolates or extrapolates the data from model levels to pressure levels, and packs the data with the GRIB format. In the following, the postprocessing is described in further details.

7.1 Vertical interpolation and extrapolation

In the postprocessing, the model level data can be interpolated or extrapolated onto the constant pressure levels or left as such, i.e. as model level fields (in model coordinates). By default, cubic splines are used for the interpolation in the vertical from model coordinates to pressure levels for geopotential and temperature, and linear interpolation for wind components, humidity and vertical velocity. Cubic spline fitting and interpolation is carried out by using NAG library routines. $\ln(\eta)$ is the vertical coordinate system used in both cubic spline and linear interpolation.

Before the data can be interpolated with respect to $\ln(\eta)$ to pressure p , the value of η corresponding to p must be computed for each grid point. Value of η is calculated by linear interpolation as follows:

First find k such that $p_{k-1/2} < p \leq p_{k+1/2}$, for $1 \leq k \leq \text{NLEV}$, then,

$$\eta = \frac{(p - p_{k-1/2}) (\eta_{k+1/2} - \eta_{k-1/2})}{p_{k+1/2} - p_{k-1/2}}$$

Geopotential and one component of vertical velocity are input to the vertical interpolation from model half levels and all other variables are given on model full levels.

Extrapolation is used if data has to be estimated above the highest model level or below the lowest model level. The general rule for extrapolation is that above the highest model level the estimated value is the same as on the highest model level (constant profile is kept above highest level). Similarly, below the lowest model level the values of the lowest level are retained. Exceptions for this rule are given in the following:

- (1) Wind components u and v are extrapolated linearly (in $\ln(\eta)$) above the highest model level
- (2) The full level part of the vertical velocity is set to 0 below the earth's surface

- (3) Geopotential is extrapolated linearly in $\ln(\eta)$ above the highest model level
- (4) For extrapolation of geopotential below the earth's surface, see section 7.3
- (5) Between the lowest model level and the surface, the temperature is interpolated linearly (in p) by

$$T = \frac{(p_s - p) T_{nlev} + (p - p_{nlev}) T_s}{p_s - p_{nlev}}$$

where T_s is

the surface temperature as defined in section * 7.2

- (6) The extrapolated temperature below the earth's surface is approximated by

$$T = T_s \left[1 + \alpha \ln \frac{P}{p_s} + \frac{1}{2} (\alpha \ln \frac{P}{p_s})^2 + \frac{1}{6} (\alpha \ln \frac{P}{p_s})^3 \right]$$

with T_s as defined in section 7.2, and the reduction factor

$$\alpha = 0.0065 R_d/g \text{ for } \phi_s/g < 2000 \text{ m},$$

but is modified for high orography to limit the extrapolated mean sea level temperature. The formulae used are

$$\alpha = R_d (T'_0 - T_s) / \phi_s$$

where defining

$$T_{plat} = \min(T_0, 298.)$$

T' is given by

$$T'_0 = T_{plat} \text{ for } \phi_s/g > 2500 \text{ m}$$

and

$$T'_0 = 0.002 [(2500 - \phi_s/g) T + (\phi_s/g - 2000) T_{plat}]$$

for $2000 \leq \phi_s/g < 2500$

T is given in section 7.2. If $T'_0 < T_*$, α is set to zero

7.2 Mean sea level pressure

The calculation of the mean sea level pressure (p_{msl}) is based on assumption of a dry, hydrostatic subterranean atmosphere and a uniform lapse rate of 6.5 K/km (or 0.0065 K/m). This lapse rate is modified for warm (or cold) surface temperatures in order to prevent too low (or high) mean sea level pressures. If surface geopotential equals zero, mean sea level pressure equals surface pressure p .

The computations proceed as follows. First the surface temperature T and the temperature reduction factor α are defined by

$$T_* = T_{nlev} + 0.0065 \frac{R_d}{g} T_{nlev} \left(\frac{P_s}{p_{nlev}} - 1 \right)$$

$$\alpha = 0.0065 R_d / g$$

Now knowing also the surface geopotential ϕ_s, p_{msl} can be calculated by

$$p_{msl} = P_s \exp \left(- \frac{\phi_s}{R_d T_*} \left[1 - \frac{1}{2} \left(\frac{\alpha \phi_s}{R_d T_*} \right) + \frac{1}{3} \left(\frac{\alpha \phi_s}{R_d T_*} \right)^2 \right] \right),$$

However, for cold and hot surface temperatures the values of T_* are changed as follows.

For cold surface temperatures ($T_* < 255 K$) T_* is replaced by

$$T_s = 0.5 (T_s + 255.)$$

For surface temperatures above 255 K, T_s is defined as

$$T_0 = T_s + 0.0065\varphi_s/g$$

Now, if $T_s \leq 290.5$ and $T_0 > 290.5$ the reduction factor α is changed to

$$\alpha = R_d (290.5 - T_s) / \varphi_s$$

But, if $T_s > 290.5$ and $T_0 > 290.5$, α and T_s are modified to

$$\alpha = 0$$

and

$$T_s = 0.5 (T_s + 290.5)$$

7.3. Time Series Files

Time Series Files are a way to store model state and derived parameters for a selected number of latitude/longitude points with a high time resolution (down to once every dynamical time step)

The Time Series contained in the file are identified by their (BUFR compatible) descriptor, the latitude and longitude of the data point and whether it contains single level (mostly surface) data or multi level data. Multi level data has to be accompanied by a vertical coordinate (e.g. pressure) corresponding to the levels stored.

Time Series Files are a good way to supply data for the display of meteograms

7.3.1. Generating Time Series Files

7.3.1.1 Initialising - INITRF

This routine initialises the data structures for the definition of this Time Series File

Subroutine INITRF(KUNIT, KTYP, KMDI, KDRT, KPDI, KPDIL,
 KIYY, KIMM, KIDD, KIHH, KIMI, KISS, KTIN, KTUN,
 KRTL, KRDP, KRML, KACC, KIAF, KWAF, KRSE, KSLE,
 KRME, KMLE, KTIMSF, KLATTS, KLONTS, KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KTYP	Integer	I	File organisation type 1 = Normal TSF 2 = Trajectory TSF
KMDI	Integer	I	Missing data indicator (equivalenced to real)
KDRT	Integer	I	Data representation type 0 = words 1 = bits (implementation deferred) 2 = hex chars (implementation deferred) 3 = dec chars (implementation deferred)
KPDI	Integer	I	BUFR identification section 1 (length KPDIL) 1 Length of section (octets) Actual value ignored; assumes 26 2 BUFR edition number 3 Originating centre Actual value ignored, assumes 0 (KNMI) 4 Update sequence number (0 = original) 5 Integer value of flag bits Actual value ignored; assumes 0 6 BUFR message type Actual value ignored, assumes 255 7 BUFR message subtype (locally defined) 1 = Normal TSF 2 = Trajectory TSF 8 Local table version number 0 = Standard BUFR tables 1 = KNMI BUFR tables 9 Year most typical for message contents 10 Month 11 Day 12 Hour 13 Minute 14 Model identification (local) 15 Product definition number (local)
KPDIL	Integer	I	Length of KPDI (15)
KIYY	Integer	I	Year; Initial State, Analysis
KIMM	Integer	I	Month
KIDD	Integer	I	Day
KIHH	Integer	I	Hour
KIMI	Integer	I	Minute
KISS	Integer	I	Second
KTIN	Integer	I	Time increment (in units KTUN)
KTUN	Integer	I	Units of time increment 11 = Years 12 = Months

				13 = Days 14 = Hours 15 = Minutes 16 = Seconds
KRTL	Integer	I		Number of time points (maximum 512)
KRDP	Integer	I		Number of data points (maximum 32)
KRML	Integer	I		Number of multiple levels
KACC	Integer	I		Accuracy of horizontal coordinates 1 = high 0.00001 degree 2 = coarse 0.01 degree
KIAF	Integer	I		Indicator additional fields Actual value ignored, assumes 0
KWAF	Integer	I		Data width additional fields Actual value ignored, assumes 0
KRSE	Integer	I		Number of single level descriptors (max 31)
KSLE	Integer	I		Array of single level descriptors (len KRSE)
KRME	Integer	I		Number of multi level descriptors (max 31)
KMLE	Integer	I		Array of multi level descriptors (len KRME)
KTIMSF	Integer	I		Array of time points (length KRTL)
KLATTS	Integer	I		Array of data point latitudes (length KRDP) Range: -9000000 / +9000000 for KACC .eq. 1 -9000 / +9000 for KACC .eq. 2
KLONTS	Integer	I		Array of data point longitudes (length KRDP) Range: -18000000 / +18000000 for KACC .eq. 1 -18000 / +18000 for KACC .eq. 2
KERR	Integer	O		Error return code 0 = No errors < 0 = Fatal error > 0 = Warning
YAERR	C*60	O		Error return message

7.3.1.2 Copy data to Time Series buffer - CPDTRF

This routine copies data from a user buffer to the correct position in the Time Series buffer
 Note that the buffer to gather the Time Series in (PBUF) must be provided by the caller of this routine (and of WRITRF)

Subroutine CPDTRF (KUNIT, KTIMC, KLAT, KLON, PDATA, PADDAT, PBUF,
 KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KTIMC	Integer	I	Time point sequence number (Data time = initial time + KTIMC * KTIN)
KLAT	Integer	I	Latitude of data point, Range: -9000000 / +9000000 for KACC .eq. 1 -9000 / +9000 for KACC .eq. 2
KLON	Integer	I	Longitude of data point, Range.

			-18000000 / +18000000 for KACC eq 1 -18000 / +18000 for KACC eq 2
PDATA	Real	I	Array with data for this data point and this time point
PADDAT	Real	I	Additional fields associated with PDATA (implementation deferred, value ignored)
PBUF	Real	I/O	Buffer into which the data are to be copied Minimum length. $KRDP * (KRSE + KRML * KRME) * KRTL$
KERR	Integer	O	Error return code 0 = No errors < 0 = Fatal error > 0 = Warning
YAERR	C*60	O	Error return message

7.3.1.3 Writing the Time Series File - WRITRF

This routine writes the buffer filled by CPDTRF to the indicated Fortran unit.

Subroutine WRITRF (KUNIT, KMODE, PBUF, KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KMODE	Integer	I	File mode: 0 = TSF format 1 = BUFR format (implementation deferred)
PBUF	Real	I	The buffer filled by CPDTRF
KERR	Integer	O	Error return code 0 = No errors < 0 = Fatal error > 0 = Warning
YAERR	C*60	O	Error return message

7.3.2 Reading Time Series Files

7.3.2.1 Get information on Time Series File - GINFSF

This routine retrieves the data structures defining this Time Series File.

Subroutine GINFSF (KUNIT, KTYP, KMDI, KDRT, KPDI, KPDIL, KIYY, KIMM, KIDD, KIHH, KIMI, KISS, KTIN, KTUN, KRTL, KRDP, KRML, KACC, KIAF, KWAF, KRSE, KSLE, KRME, KMEL, KLATTS, KLONTS, KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KTYP	Integer	O	File organisation type 1 = Normal TSF 2 = Trajectory TSF

KMDI	Integer	O	Missing data indicator (equivalenced to real)
KDRT	Integer	O	Data representation type 0 = words 1 = bits (implementation deferred) 2 = hex chars (implementation deferred) 3 = dec chars (implementation deferred)
KPDI	Integer	O	BUFR identification section 1 (length KPDIL) 1 Length of section (octets) Actual value ignored; assumes 26
		1	BUFR edition number
		2	Originating centre Actual value ignored; assumes 0 (KNMI)
		3	Update sequence number (0 = original)
		4	Integer value of flag bits Actual value ignored, assumes 0
		5	BUFR message type Actual value ignored, assumes 255
		6	BUFR message subtype (locally defined) 1 = Normal TSF 2 = Trajectory TSF
		7	Local table version number 0 = Standard BUFR tables 1 = KNMI BUFR tables
		8	Year most typical for message contents
		9	Month
		10	Day
		11	Hour
		12	Minute
		13	Model identification (local)
		14	Product definition number (local)
KPDIL	Integer	O	Length of KPDI (15)
KIYY	Integer	O	Year; Initial State, Analysis
KIMM	Integer	O	Month
KIDD	Integer	O	Day
KIHH	Integer	O	Hour
KIMI	Integer	O	Minute
KISS	Integer	O	Second
KTIN	Integer	O	Time increment (in units KTUN)
KTUN	Integer	O	Units of time increment 11 = Years 12 = Months 13 = Days 14 = Hours 15 = Minutes 16 = Seconds
KRTL	Integer	O	Number of time points (maximum 512)
KRDP	Integer	O	Number of data points (maximum 32)
KRML	Integer	O	Number of multiple levels
KACC	Integer	O	Accuracy of horizontal coordinates 1 = high 0.00001 degree

			2 = coarse 0.01 degree
KIAF	Integer	O	Indicator additional fields
			Actual value ignored, assumes 0
KWAF	Integer	O	Data width additional fields
			Actual value ignored, assumes 0
KRSE	Integer	O	Number of single level descriptors (max 31)
KSLE	Integer	O	Array of single level descriptors (len KRSE)
KRME	Integer	O	Number of multi level descriptors (max 31)
KMLE	Integer	O	Array of multi level descriptors (len KRME)
KLATTS	Integer	O	Array of data point latitudes (length KRD _P)
			Range -9000000 / +9000000 for KACC eq 1 -9000 / +9000 for KACC eq 2
KLONTS	Integer	O	Array of data point longitudes (length KRD _P)
			Range -18000000 / +18000000 for KACC eq 1 -18000 / +18000 for KACC eq 2
KERR	Integer	O	Error return code 0 = No errors < 0 = Fatal error > 0 = Warning
YAERR	C*60	O	Error return message

7.3.2.2 Reading Single Level Time Series - RDSESF

This subroutine reads a single level element time series from a TSF. It can be called without calling GINFSF first

Subroutine RDSESF (KUNIT, KLATR, KLONR, KACCR, KSLER, KTLVS, KRTL_V, PDATA, PADDAT, KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KLATR	Integer	I	Latitude of requested data point (see KACCR)
KLONR	Integer	I	Longitude of requested data point (see KACCR)
KACCR	Integer	I	Accuracy of position of data point 1 = high 0.00001 degree 2 = coarse 0.01 degree
KSLER	Integer	I	Descriptor of requested element
KTLVS	Integer	O	Array with time points (length KRTL _V)
KRTL _V	Integer	I/O	Number of time points
PDATA	Real	O	Array to contain the Time Series data (length KRTL _V)
PADDAT	Real	O	Array to contain additional data (length KRTL _V) (implementation deferred)
KERR	Integer	O	Error return code 0 = No errors < 0 = Fatal error
YAERR	C*60	O	Error return message

7.3.2.3 Reading Multi Level Time Series - RDMETF / RAMETF

These routines read a multiple level element for a given data point and time point (RDMETF) or all multiple level elements for a given data point and time point (RAMETF).

They can be called without calling GINFSF first

Subroutine RDMETF (KUNIT, KLATR, KLONR, KACCR, KMLER, KTUNR,
KTIMR, KRMLR, PTDATA,PADDAT, KERR, YAERR)

KUNIT	Integer	I	TSF unit number
KLATR	Integer	I	Latitude of requested data point (see KACCR)
KLONR	Integer	I	Longitude of requested data point (see KACCR)
KACCR	Integer	I	Accuracy of position of data point 1 = high: 0.00001 degree 2 = coarse: 0.01 degree
KMLER	Integer	I	Descriptor of requested element
KTUNR	Integer	I	Units of requested time distance 13 = Days 14 = Hours 15 = Minutes 16 = Seconds
KTIMR	Integer	I	Requested time distance, in units KTUNR
KRMLR	Integer	I/O	Number of multiple levels
PTDATA	Real	O	Array to contain the Time Series data (length KRMLR)
PADDAT	Real	O	Array to contain additional data (length KRMLR) (implementation deferred)
KERR	Integer	O	Error return code 0 = No errors < 0 = Fatal error
YAERR	C*60	O	Error return message

Subroutine RAMETF (KUNIT, KLATR, KLONR, KACCR, KTUNR, KTIMR,
KDIM1, KRMER, KRMRL, KMLER, PTDATA,PADDAT, KERR,
YAERR)

KUNIT	Integer	I	TSF unit number
KLATR	Integer	I	Latitude of requested data point (see KACCR)
KLONR	Integer	I	Longitude of requested data point (see KACCR)
KACCR	Integer	I	Accuracy of position of data point 1 = high: 0.00001 degree 2 = coarse: 0.01 degree
KTUNR	Integer	I	Units of requested time distance 13 = Days 14 = Hours 15 = Minutes 16 = Seconds

KTMR	Integer	I	Requested time distance, in units KTUNR
KDIM1	Integer	I	First dimension of actual parameter PTDATA
KRMER	Integer	I/O	Number of multiple level element descriptors
KRMLR	Integer	I/O	Number of multiple levels
KMLER	Integer	O	Array to contain the multiple level descriptors (length KRMER)
PTDATA	Real	O	Array to contain the Time Series data (dimensions KDIM1, KRMER)
PADDAT	Real	O	Array to contain additional data (dimensions KDIM1, KRMER) (implementation deferred)
KERR	Integer	O	Error return code 0 = No error < 0 = Fatal error
YAERR	C*60	O	Error return message

7.4 Vertical velocity

Computation of the vertical velocity (ω) takes place in two parts, one on the model half levels and the other on full levels:

$$\omega = \omega_h + \omega_f$$

where

$$\omega_h = - \int_0^n \nabla \cdot (V \frac{\partial p}{\partial n}) d\eta$$

$$\omega_f = V_{hor} \nabla p$$

The ω_h is calculated at the model half levels. Using the model's finite differences scheme, the half level part becomes:

$$\omega_{1/2} = 0$$

$$\omega_{k+1/2} = \omega_{k-1/2} - D_k \Delta p_k \quad \text{for pure pressure layers and}$$

$$\omega_{k+1/2} = \omega_{k-1/2} - D_k \Delta p_k - (V_{hor} \nabla p_s) \Delta B_k \quad \text{for hybrid layers,}$$

where V_{hor} and D_k are the horizontal wind and divergence at level k and

$$\Delta p_k = p_{k+1/2} - p_{k-1/2}$$

$$\Delta B_k = B_{k+1/2} - B_{k-1/2}$$

$$p_{k+1/2} = A_{k+1/2} + B_{k+1/2} p_s$$

The ω_f is calculated on the model full levels and the finite difference representation is given by

$$\omega_k = 0$$

for pure pressure levels, and

$$\omega_k = p_k (V_{hor} \nabla p_s) \frac{1}{\Delta p_k} (\Delta B_k + C_k \frac{1}{\Delta p_k} \ln(\frac{p_{k+1/2}}{p_{k-1/2}}))$$

for hybrid levels, and

$$\omega_k = p_k (V_{hor} \nabla p_s) \frac{1}{\Delta p_k} \Delta B_k$$

for pure sigma levels, where

$$C_k = A_{k+1/2} B_{k-1/2} - A_{k-1/2} B_{k+1/2}$$

7.5 Relative humidity

The relative humidity is calculated on the model full levels using

$$RH_k = \frac{q_k}{q_{sk}}$$

where q_s is the saturation specific humidity and is defined as

$$q_s = \frac{\frac{R_d}{R_v} \frac{e_s(T_k)}{P_k}}{1 - (\frac{R_v}{R_d} - 1) \frac{R_d}{R_v} \frac{e_s(T_k)}{P_k}}$$

and $e_s(T_k)$ is the saturation vapour pressure at temperature T_k . e_s is computed in the same way as in the analysis and in the forecast model, see section 4.1.6

7.6 Surface fluxes

The postprocessing package provides an opportunity to compute the surface fluxes of momentum, sensible and latent heat. The surface flux of ψ is determined by using the drag coefficient expression

$$F_\psi = \rho C_{\psi_s} |V(z)| (\psi(z) - \psi_s)$$

and by using model variables

$$F_\psi = CONS \cdot WCC(n) \cdot (\psi_n - \psi_s)$$

where

$$CONS = \rho_n k^2 WCA(n) T_n / CAMS 1,$$

k is the von Karman constant and indexes n and s refer to NLEV (the lowest level) and surface Coefficients WCA and WCC are defined in section 2.2.2 in connection with vertical diffusion. Further,

$$CAMS1 = \left[2\Delta t g / (\Delta n_n R) \right] n_n k^2$$

$$\rho_n = p_n / (R T_n)$$

For momentum flux ($\psi = u, v$) WCC(n) = 1. Term CONS can further be given by

$$\begin{aligned} CONS &= p_n WCA(n) \left[\Delta n_n / (2\Delta t g n_n) \right] \\ &= p_n WCA(n) ZFLUX \end{aligned}$$

Finally, the total flux is obtained as a linear combination of fluxes over fraction of land and fraction of sea:

$$\begin{aligned} F_\psi &= p_n ZFLUX + (1 - FRSEA) WCA(n)_1 WCA(n)_1 (\psi_n - \psi_s)_1 \\ &\quad + FRSEA WCA(n)_s (\psi_n - \psi_s)_s \end{aligned}$$

Here an approximation $(\psi_n)_s = (\psi_n)_1$ is made. Note that the time step Δt is included in the formula for the surface flux. At the moment Δt is hard-coded and is set to 240 s (4 min)

7.7 Computer code organization and namelist parameters

The postprocessing package contains two master subroutines POSTPM and POSTPP. The former just changes the data organization from the model slab file format into the packed GRIB field file format leaving the data still in the model coordinates. The latter performs the postprocessing of data onto constant pressure levels in the way described above.

POSTPM:

POSTPM reads the slab file data description record (DDR) and then all the (latitude) data records. The slab file DDR is modified to correspond to the horizontal field file. Then the data are organized into horizontal fields, and all multi-level and single-level fields are written in the GRIB format (one horizontal field in a record). POSTPM further provides a possibility to write out (archive) only a sub-area from the whole forecast area. This is controlled by the logical switch LSUB (default false; set true for sub-area postprocessing) and the grid point numbers of the new corners (IWE & IES between 1...NLON and JSO & JNO between 1...NLAT). POSTPM reads namelist NAMPOS from unit 5. An example procedure PPM performing postprocessing in model coordinates can be found in section 9.5. in job JCLEXP.

NAMPOS includes the following parameters

parameter	meaning	default
ILUIN	unit number of the slab file	11
ILUOUT	unit number of the output GRIB file	12
ILUDIR	unit number of the direct access file	71
LPRINT	logical switch for test print	TRUE
IPRINT	number of the grid point for test print in longitudinal (x) direction	10
JPRINT	number of the grid point for test print in latitudinal (y) direction	10
LSUB	logical switch for sub-area	FALSE
IWE	number of the western grid point and	0
IES	number of the eastern grid point for the sub-area (1..NLON)	0
JSO	number of the southern grid point and	0
JNO	number of the northern grid point for the sub-area (1..NLAT)	0

POSTPP.

Master routine POSTPP postprocesses the unpacked slab file into the GRIB field file with selected fields (given in the namelist). Multi-level fields can be either extracted as model level fields or interpolated onto constant pressure levels.

POSTPP reads the slab file DDR, and all slab data records into a 3-dimensional array. The DDR is then modified to correspond to the field file, which will contain only the selected fields given in the input namelist. The data are re-organized from one big array into arrays containing only one variable. Then the selected multi-level fields are extracted (as model level data) or interpolated onto the pressure levels. Single-level fields (near-surface-fields) are copied as such. Variables, which are not model variables are calculated. All selected fields are written in the GRIB format into the output file. POSTPP provides also an opportunity to calculate the surface flux fields, i.e., the momentum, sensible and latent heat fluxes (with parameter code numbers 175, 176 and 177, respectively). Specifying these code numbers (TWMOSL) and the level parameter (ALEVSL = 0.) in the input namelist activates the flux computations. POSTPP reads two namelists, NAMPOS and NAMPPP. NAMPOS is read from unit 5 and NAMPPP from unit ILUNL given in NAMPOS. An example of the postprocessing job (DOPP) can be found in section 9.5.

NAMPOS contains the following parameters.

parameter	meaning	default
ILUIN	unit number of the slab file	11
ILUOUT	unit number of the output GRIB file	12
ILUDIR	unit number of the direct access file	71
LPRINT	logical switch for test print	FALSE
IPRINT	number of the grid point for test print in longitudinal (x) direction	10
JPRINT	number of the grid point for test print	

	in latitudinal (y) direction	10
NAMPPP	contains the following parameters (no defaults)	
NLEVHL	number of vertical levels for multi-level variables	
LTPHHL	type of vertical coordinate system	
ILEVHL	list of selected eta (given in k) levels, if in model coordinates	
ALEVHL	list of selected pressure levels (in Pa), if in pressure coordinates	
NWMOHL	number of multi-level variables	
IWMOHL	list of multi-level variables	
NSL	number of single-level variables selected	
LTPSL	list of level types (for each s-l variable)	
ILEVSL	list of ordinal numbers for single-level fields	
(not used)		
ALEVSL	list of levels	
IWOSL	list of parameter codes	

8. The surface orography and climate system

The HIRLAM Level 1 surface orography and climate system is essentially a limited area version of the corresponding ECMWF system. The ECMWF system was modified to take the limited area and the transformed latitude/longitude geometry into account. In addition, the HIRLAM forecast model surface parameterization schemes use fractional land coverage and fractional ice coverage within each grid square, which had to be taken into account when modifying the ECMWF surface orography and climate system.

Documentation of the ECMWF surface orography and climate system is included in the following ECMWF reports

- 1 Meteorological Bulletin 1 6/2
Research Manual 3
ECMWF Forecast Model Physical Parameterization
Edited by J-F.Louis
- 2 ECMWF Research Department
Tech. Memorandum No 109
The ECMWF Climate System
Cedo Brankovic and Jan Van Maanen (October 1985)

A general description of the procedures for deriving the HIRLAM climate data fields is given in section 8.1. Then the software for deriving the surface orography, the surface roughness due to orography, the fraction of land and the fraction of urban area are described in section 8.2, followed by a description of the software for deriving the surface climate fields in section 8.3. In section 8.4, finally, the format and content of the HIRLAM climate data files are described.

8.1 Introduction

The input data for the ECMWF surface orography and surface climate systems have been collected from many sources. In summary, these sources are

- (1) The US Navy terrain data containing a range of terrain parameters, such as mean terrain heights, maximum heights, minimum heights, land/sea fractions, fraction of urban area etc. on a regular latitude/longitude grid with a resolution of $1/6^\circ \times 1/6^\circ$
- (2) Surface soil moisture fields defined on a global $4^\circ \times 5^\circ$ latitude/longitude grid for the 1st and 16th of each month.
- (3) Monthly sea-surface temperature fields and information on the extent of sea ice on a regular global latitude/longitude grid with a resolution of $1^\circ \times 1^\circ$.
- (4) Global precipitation fields representing monthly mean values in $5^\circ \times 5^\circ$ latitude / longitude boxes
- (5) Monthly mean air surface temperature fields on a regular $5^\circ \times 5^\circ$ latitude / longitude grid.

- (6) A yearly averaged albedo field given on a global regular latitude/longitude grid with a resolution of $1.875^\circ \times 1.875^\circ$
- (7) Roughness length due to vegetation given on a global regular latitude / longitude grid with a resolution of $5^\circ \times 5^\circ$

More detailed descriptions of the input data together with relevant references to the sources are included in the ECMWF publications referred to above

The generation of the HIRLAM climate data files from these input climate data files is carried out through the following steps

- (1) Generation of the HIRLAM mean orography, roughness due to orography, fraction of land and fraction of urban area from the US Navy terrain data set by averaging in HIRLAM grid boxes and Gaussian smoothing, see section 8.2 for details.
- (2) Horizontal interpolation of monthly mean air surface temperatures and the corresponding large-scale mean orography heights (resolution $5^\circ \times 5^\circ$) to the HIRLAM grid-points
- (3) Reduction of the monthly mean air surface temperatures from the large-scale mean orography heights to the HIRLAM orography heights assuming a vertical temperature lapse rate of 0.0065 K/m . The reduced air surface temperature is taken as the monthly mean surface soil temperature over land and ice covered areas.
- (4) Derivation of consistent global monthly mean ice coverage fields from the global monthly mean sea surface temperature fields and the global information on the extent of sea ice (resolution $1^\circ \times 1^\circ$)
- (5) Horizontal interpolation of the monthly sea surface temperature fields and the monthly mean ice coverage fields to the HIRLAM grid
- (6) Horizontal interpolation of the monthly mean global precipitation fields to the HIRLAM grid
- (7) Derivation of monthly mean snow depth fields from the monthly mean precipitation fields and the monthly mean surface temperature fields. The monthly surface temperatures are used to model two snow processes, the fraction of precipitation that is falling as snow and a snow melt function. The monthly mean snow depth fields are computed iteratively over a number of years until these depths have reached steady state values. The monthly mean surface temperature fields are modified in accordance with the derived snow depths.
- (8) Derivation of the monthly mean mid- and deep-layer soil temperature fields from the monthly mean surface soil temperature fields by introduction of seasonal phase lags and amplitude damping of deviations from annual means
- (9) Horizontal interpolation of the surface soil moisture fields from the global grid to the HIRLAM grid

- (10) Derivation of monthly mean mid- and deep-layer soil moisture fields from the surface soil moisture fields by proper scaling in accordance with the defined depths of these layers.
- (11) Interpolation of the global albedo field to the HIRLAM grid
- (12) Gauss-filtering of the interpolated albedo field.
- (13) Correction of the albedo field for the occurrence of sea ice.
- (14) Horizontal interpolation of the roughness field due to vegetation from the global grid to the HIRLAM grid.
- (15) Blending of the roughness field due to vegetation with the roughness field due to orography
- (16) Gauss filtering of the blended roughness field and resetting of the roughness field over pure sea points to a constant value.
- (17) Creation of the HIRLAM climate files in GRIB format, one file for each month.

Detailed descriptions of the algorithms used to create the HIRLAM surface climate fields are given in the ECMWF publications mentioned above.

8.2 Orography, surface roughness and fraction of land

In order to take the complications introduced by the transformed HIRLAM grid into account, the software for deriving the HIRLAM orography, roughness due to orography, fraction of land and fraction of urban area was re-written for the HIRLAM purposes. The HIRLAM values of these variables are derived by a simple averaging over all the US Navy terrain gridpoints falling within the HIRLAM grid-boxes:

$$\Phi_s^{hir}(x^{hir}, y^{hir}) = \frac{1}{N_k} \sum_{k=1}^{N_k} \Phi_s(x_k^{navy}, Y_k^{navy})$$

$$P_{land}^{hir}(x^{hir}, y^{hir}) = \frac{1}{N_k} \sum_{k=1}^{N_k} P_{land}(x_k^{navy}, Y_k^{navy})$$

$$P_{urban}^{hir}(x^{hir}, y^{hir}) = \frac{1}{N_k} \sum_{k=1}^{N_k} P_{urban}(x_k^{navy}, Y_k^{navy})$$

$$z_0^{hir}(x^{hir}, Y^{hir}) = \sqrt{\frac{N}{E} \left(\sum_{k=1}^{N_k} A_k (\phi_s(x_k^{navy}, Y_k^{navy}))^2 - \left(\sum_{k=1}^{N_k} A_k \phi_s(x_k^{navy}, Y_k^{navy}) \right)^2 \right) + \sum_{k=1}^{N_k} \sqrt{n_k/f_k} A_k \frac{(\phi_{sk} - \phi_{sk}^{\min}) (\phi_{sk}^{\max} - \phi_{sk})}{4}}$$

where summation is taken over all US Navy terrain data set grid-boxes falling within the HIRLAM grid boxes

$$x^{hir}-0.5 \leq x_k^{navy} < x^{hir}+0.5$$

and

$$Y^{hir}-0.5 \leq Y_k^{navy} < Y^{hir}+0.5$$

and where the following notations have been used

$$x = \frac{LONFLONW}{DFLON} + 1.0$$

x-coordinate in the HIRLAM grid geometry

$$y = \frac{LATFLATES}{DFLAT} + 1.0$$

y-coordinate in the HIRLAM grid geometry

LON Longitude in the transformed geometry

LAT Latitude in the transformed geometry

FLONW Western boundary of the HIRLAM area in the transformed geometry

FLATS Southern boundary of the HIRLAM area in the transformed geometry

DFLON Longitudinal grid distance

DFLAT Latitudinal grid distance

x^{hir} x-coordinates of HIRLAM gridpoints =1,2,...,NFLON

y^{hir}	y-coordinates of HIRLAM gridpoints =1,2,...,NFLAT
x^{navy}, y^{navy}	Coordinates of the mid-points of the US Navy terrain data boxes, expressed in HIRLAM units
ψ_s^{hir}	Unsmoothed HIRLAM surface orography height
P_{land}^{hir}	Fractional land cover of the HIRLAM grid boxes
P_{urban}^{hir}	Fractional urban area of the HIRLAM grid boxes
Z_0^{hir}	Roughness due to orography of the HIRLAM grid boxes
$\bar{\phi}_s$	Mean orography height of the US Navy terrain data boxes
ϕ_s^{\max}	Minimum orography height of the US Navy terrain data boxes
ϕ_s^{\min}	Maximum orography height of the US Navy terrain data boxes
P_{land}	Fraction of land of the US Navy terrain data boxes
P_{urban}	Fraction of urban area of the US Navy terrain data boxes
N_k	Number of US Navy grid boxes falling within the HIRLAM grid boxes
N	Number of relative $\bar{\phi}_s$ maxima in the HIRLAM grid boxes
F	Surface area of the HIRLAM grid box
A_k	Proportion of the HIRLAM grid box covered by the US Navy grid box
n_k	Number of significant ridges in the US Navy terrain data grid box

f_k

Surface area of the US Navy Terrain data grid box

$$\bar{\Phi}_{sk} = \bar{\Phi}_s(x_k^{navy}, Y_k^{navy})$$

$$\Phi_{sk}^{\min} = \Phi_s^{\min}(x_k^{navy}, Y_k^{navy})$$

$$\Phi_{sk}^{\max} = \Phi_s^{\max}(x_k^{navy}, Y_k^{navy})$$

A Gaussian smoothing operator is applied to the ϕ_s^{hir} -field to obtain the final HIRLAM orography field ϕ_s^{hir}

$$\phi_s^{hir}(i, j) = \sum_{k=1-n}^{i+n} \sum_{l=j-n}^{j+n} EXP\left(-\frac{(k-i)^2 + (l-j)^2}{2 DDD^2}\right) \phi_s^{hir}(k, l) / WSUM(i, j)$$

where

$$WSUM(i, j) = \sum_{k=1-n}^{i+n} \sum_{l=j-n}^{j+n} EXP\left(-\frac{(k-1)^2 + (l-j)^2}{2 DDD^2}\right)$$

and with DDD=0.5 and n=5 used in most HIRLAM experiments carried out so far.

The program (CLIFIL) for derivation of the physiographical parameters described above has one parameter statement defining the maximum dimensions JPLON and JPLAT of the HIRLAM grid and the following two input namelist statements:

```
NAMELIST/NAMCLI/ LUNORO,LUNNL,LUNFIS,LUNFIM,LUNROU,  
LUNLSM,LUNURB
```

```
NAMELIST/NAMGEO/ FLONW,FLONE,FLATS,FLATN,DFLON,DFLAT,POLE,  
NFLON,NFLAT
```

The variables in these namelist statements have the following meaning

LUNORO = File unit number of the US Navy terrain data set

LUNNL	=	File unit number of NAMELIST/NAMGEO/
LUNFIS	=	File unit number of e (output)
LUNFIM	=	File unit number of mean orography = e (output)
LUNROU	=	File unit number of Z (output)
LUNLSM	=	File unit number of P (output)
LUNURB	=	File unit number of P (output)
FLONW	=	Western boundary of the transformed HIRLAM grid
FLONE	=	Eastern boundary of the transformed HIRLAM grid
FLATS	=	Southern boundary of the transformed HIRLAM grid
FLATN	=	Northern boundary of the transformed HIRLAM grid
DFLON	=	Longitudinal grid distance
DFLAT	=	Latitudinal grid distance
POLE	=	Latitude of the north pole of the transformed HIRLAM grid
NFLON	=	Number of gridpoints in the longitudinal direction
NFLAT	=	Number of gridpoints in the latitudinal direction

The output data files are stored according to the ECMWF climate data formats in order to facilitate the further processing of the data.

NAMELIST/NAMCLI/ is read from unit 5

The following subroutines are utilized in the program package to derive the HIRLAM orography, roughness due to orography, fraction of land and fraction of urban area:

CLIFIL	Main subroutine, input/output etc
TOPPO	Averaging over the US Navy terrain data grid boxes
USNUNP:	Unpacking of US Navy terrain data
MAPPA	Test printing
SGAUSS	Gaussian smoothing
PTHDR,PTVAR,PTRES,PTLSM and	
PTFLD	Output of fields in the ECMWF climate data format

8.3 Surface climate fields

Most of the subroutines of the ECMWF climate data system were possible to use without any significant modifications. For these subroutines and for the formats of the various input and intermediate climate data files, reference is made to the ECMWF publications mentioned above. Significant changes were introduced into the following subroutines of the ECMWF climate data system:

PCGAUSF.

Administration of the Gaussian filtering. The same type of filtering as described in section 8.2 above is used for various climate fields. A subroutine SGAUSS is doing the actual smoothing. At

present, the same smoothing parameters as used for the smoothing of orography are applied (DDD=0.5 and n=5, see above)

PCSNOW

The HIRLAM transformed geometry had to be taken into account when looking for the areas permanently covered by snow, e.g. Greenland.

SETIND,INTERPU,INMANIA:

The horizontal interpolation subroutines were modified to fit the transformed HIRLAM latitude / longitude geometry

In addition, no blending of the sea surface temperature and the surface soil temperature fields is carried out for the HIRLAM purposes, since the HIRLAM surface parameterization schemes operate with fractional land cover and consequently also with two surface temperature fields

8.4 HIRLAM climate data files

The HIRLAM climate data fields are stored in GRIBbed monthly files containing the following variables:

WMO variable index	Level value	Variable
102	0	e surface orography height
104	0.	T surface soil temperature
104	999.	T mid-layer soil temperature
104	998	T deep-layer soil temperature
180	0.	Z surface roughness
147	0	W surface soil wetness
147	999.	W mid-layer soil wetness
147	998	W deep-layer soil wetness
151	0.	SN snow depth
184	0	Albedo
181	0.	Fraction of land ("Land/Sea mask")
182	0.	Fraction of urban area
161	0.	Sea-surface temperature
183	0.	Fraction of ice-covered area

All the climate fields are stored with level type = 105 corresponding to a ground-following vertical coordinate system (see chapter 5)

The gribbing and storing of the climate files are done by the program CLGRIB. This program has

one parameter-statement defining the maximum dimensions of the HIRLAM grid-point-fields (KPLON and KPLAT) and the following two namelist-statements

```
NAMELIST/NAMPAR/ LUNGEO,LUNOUT,MONTH,NFLD,
LUNIN,NMON,JWMO,LEVTyp,VERT,
IVAREC,LUNDIR
```

```
NAMELIST/NAMGEO/ FLONW,FLONE,FLATS,FLATN,
DFLON,DFLAT,POLE,NFLON,NFLAT
```

The variables in these two namelist statements have the following meaning

Variable	Type	Meaning
LUNGEO	Integer	File unit number of NAMLIST/NAMGEO/
LUNOUT	Integer	File unit number of the output GRIB file
MONTH	Integer	Month number (1-12) of the file to be gribbed
NFLD	Integer	Number of fields to be gribbed
LUNIN(NFLD)	Integer	Input file unit numbers of the fields to be gribbed
NMON(NFLD)	Integer	Number of months of the input fields (e.g. 1 for orography and 12 for surface soil temperature)
JWMO(NFLD)	Integer	WMO-indicators corresponding to the input fields
LEVTyp(NFLD)	Integer	Level types of the input fields (=105 so far)
VERT(NFLD)	Real	Level value of the input fields (see above)
IVAREC(NFLD)	Integer	ECMWF variable indicators of the input fields, see the ECMWF documentation
LUNDIR	Integer	File unit number of the direct access file used by GWRITE for intermediate storing (see chapter 5)
FLONW	Real	Western boundary of the HIRLAM grid
FLONE	Real	Eastern boundary of the HIRLAM grid
FLATS	Real	Southern boundary of the HIRLAM grid
FLATN	Real	Northern boundary of the HIRLAM grid
DFLON	Real	Grid resolution in the longitudinal direction
DFLAT	Real	Grid resolution in the latitudinal direction
POLE	Real	Latitude of the north pole of the transformed HIRLAM grid
NFLON	Integer	Number of gridpoints in the longitudinal direction
NFLAT	Integer	Number of gridpoints in the latitudinal direction

Namelist NAMPAR is read from file unit 5.

9.3 General verification against observations

The EWGLAM verification package is sometimes unsatisfactory, in particular in cases where verification over areas outside the European continent is sought. Also a limitation of the package is that it requires the observations to be available in a format which is internal to the HIRLAM analysis scheme (the so-called Analysis Observation File, see Section 4.1.7). This links the EWGLAM package tightly to the HIRLAM system, it will be difficult to apply it to other models or to partial HIRLAM implementations, where the HIRLAM analysis scheme is not used.

A package, supplementary to the EWGLAM package, is under development. In its current state it is available, within the HIRLAM reference system, for verification of selected near-surface fields against the corresponding observations, over a selection of areas. The selection of areas is rather easy to adapt, changing the selection of fields requires more work. In this Section, the package will be described, an indication will be given how the choices of fields and/or areas can be adapted.

The package consists of three steps. The correct order of execution of these steps is organised by the script `Compare`. The documentation to this script, which is online available by executing '`Whatis Compare`' (see Section 10.4.3), is added as an Appendix.

In the first step, the observations, encoded in BUFR (e.g. available at ECMWF through the archiving system `mars`), are decoded and the data to be used for verification are stored in an ASIMOF file (see Section 5.1.2). All observations of a certain quantity (for example, all surface observations of 10 m wind) are collected into one 'pseudo-GRIB' message (Section 5.1.3). Supplementary data are also written to the ASIMOF file: Pseudo-GRIB records of latitudes and longitudes (as required by this pseudo-GRIB format), and of station type etc, furthermore, one record is added with the station identifiers. This last record is not pseudo-GRIB, but it contains ASCII data, so here is an example of an ASIMOF file that is not interpretable by pure-grib programs (see also Section 5.1.1). This first step is performed by the program '`bfrx.x`', which is a modified form of the program '`maof.x`' (Section 4.1.7).

The second and third step are executed for each set of fields to be verified against the data. In this way, it is possible to verify forecasts of a range of forecast lengths and of a range of models, all verifying at the same time, namely the time at which the observations as used in the first step are valid.

In the second step, the fields to be verified are interpolated horizontally to the observation positions. Note that this implies that fields can only be verified if they do not need vertical interpolation to match them to observed quantities. Mean sea level pressure can be verified, but an eta level temperature cannot. The horizontal interpolation is done by the program '`intp.x`' (Section 6.1).

In the third step, the interpolated fields and the observations are compared. Some statistics (e.g. bias, rms, maximum deviation) are calculated for a range of areas, and printed to a standard format ASCII file. The format of this file defines a number of 'fields', separated by the character '='. Each such field refers to a parameter, e.g. 10 m wind. The fields are partitioned into subfields, separated by '|'. Each such subfield gives a statistical quantity (for 10 m wind, these may be 'number of data', 'maximum wind speed error', or 'rms of wind vector difference'). There is a special field with model and area identification. The file has been designed to be easily

interpretable with standard Unix utilities like 'grep' and 'awk'

The first time the third step is executed, the difference between the field and the observation is used to obtain a quality estimate of the observation. If it deviates too much from the field, it will not be used for verification. Grossly erroneous observations are thus avoided to blur the statistics. This quality control information is added, again in pseudo-GRIB format, to the file produced by the second step. Later executions of the third step will recognise the presence of this quality control information and use it to select the data to be used for verification. Therefore, all models and all forecast lengths that are verified within the same instance of Compare, use the same data for verification. The danger of this method lies in the required high quality of the field that is to be verified first, because that field is used for an absolute quality control check. In practice, it is advised to use the HIRLAM analysis as the first field, its quality is almost always far more than sufficient for gross error detection.

The selection of areas, or of statistical quantities to be calculated (e.g. mean absolute error versus rms), can simply be changed by modifying the (standard Fortran) program that executes the third step (vrfy.x). If other parameters than those currently selected (namely pmsl, 10 m wind, 2 m temperature, total cloud cover and precipitation, all from surface stations like SYNOP and SHIP) are to be verified, the program vrfy.x needs changing. But then also the program in the first step (bfrx.x) has to be modified. If the new parameters are to be verified against SHIP or SYNOP reports, the required modifications are straightforward. But if the new parameter requires the decoding of other observation types (for example, 200 hPa wind), then the changes become more involved. In that case, it is suggested to apply the script Compare separately for each new observation type.

Appendix

```
!!execute Whatis Compare
```

The HIRLAM system under Unix
A unified script system

G.J. Cats
T. Moene
B.G.J. Wicher Schreur

KNMI, De Bilt
May 1994

Summary

This note describes the HIRLAM level-2 system implementation under Unix, especially under Unicos on the ECMWF computer system. The reference HIRLAM system consists of (mainly) Fortran-77 programs, scripts to compile, load and execute those programs, and utilities to assist the porting of the system from ECMWF to a local computer and to generate documentation of the system.

Environment variables are used to define the default HIRLAM system and to define the user's particular deviations from the default. With proper settings of those variables, the local system manager can use the script system to maintain the local default version; the script system can be used to control the operational HIRLAM runs; and experiments with the HIRLAM system can be carried out on the basis of that system. The deviations of the operational and experimental systems from the local default system, and of the local default system from the reference default system are described in a limited number of small files. The script system automatically incorporates those deviations when it is invoked. Thus, it guarantees the consistency of the default system, and of the derived systems.

The script system provides a control system not only for the HIRLAM system, but also for the input files used for a particular run. Thus it completely defines the runs.

A simple-to-use news facility at ECMWF allows users of the system to remain informed of all recent modifications.

-

1. _I_n_t_r_o_d_u_c_t_i_o_n

The weather forecasting system produced by the HIRLAM project is maintained in its basic version on the ECMWF computer configuration. That system is referred to as the reference HIRLAM system. The system consists of Fortran programs, input files, documentation and jobs to run the system. This note describes the jobs. The reference HIRLAM system jobs are kept in Unix scripts. These scripts run on the ECMWF computer system, notably a Cray-C90 under the Unicos operating system.

The reference system is intended to be copied by the countries participating in the HIRLAM project to their local computers for use in their national weather forecasting procedures. The porting of the system usually consists of conversion of the Fortran programs and of conversion of the jobs to the local operating system. The porting of the programs is a relatively small task, as the programs are almost fully complying with the Fortran-77 standard; exceptions are mainly found in the use of Cray or ECMWF library functions in the analysis code.

To facilitate the porting of the Unix scripts, also in the scripts Cray or ECMWF specific features were avoided as much as possible, and, if they were required, identified and isolated with proper surrounding 'if'-clauses. An example is the ECMWF database utility 'ecfile'; in the Unicos scripts ecfile is invoked only if a test on the computer centre being ECMWF is satisfied.

The user will have to change but a few lines to organise the construction of a series of runs with the system, either the reference system itself, or a modified version for experimentation.

This note describes the general design of the reference system and what should be done to run the reference system or a modification. The actual form in which the reference system is run is prescribed by environment variables. They are listed in the first Appendix. An important design feature of the reference system is a convention for the file titles used internally by the HIRLAM system. The second Appendix describes this convention. The third describes the location of the files and the fourth lists the scripts in the reference system.

Two Appendices have been added to facilitate running the system and the transfer to a member state's local computer system.

2. System overview

The HIRLAM reference system is maintained on the ECMWF computer facility. The Fortran programs are kept as nupdate program libraries (nupdate is a Cray source code management system) ; the files are stored with ecfile into the 'Common File System' (CFS). The scripts of the current version are kept on-line on the Cray, and those will be used for HIRLAM experiments unless the user takes explicit actions to use a different version. To run the system at ECMWF, the user must create a working directory, and copy a start-up script from the reference system into this directory. This copy must then be edited to specify the environment in which the run is to be done.

To run the system at a national computer (which is assumed to run Unix), the user must copy all programs and scripts to a basic directory on that computer. A utility is provided to copy the reference HIRLAM programs, maintained with Cray's nupdate, to the local computer, in such a form that they can be input to a simulation of nupdate, written in C. By editing the start-up script the user identifies the local computer system. It is assumed that this conversion is done once and for all, until a new version of the reference system is to be implemented on the local computer; the conversion is done by the local HIRLAM system manager and from then on, the system is available to all users. Each user will have to create a working directory for his experiment, copy the start-up script into it and edit it as required.

Within the system, a distinction is made between the host, which is the computer running the programs, and the database, where large amounts of data can be stored (semi-)permanently. The reference system makes this distinction rigorously for runs at ECMWF. In principle, it is conceivable that also on a local host such a distinction is to be made. In that case, the reference system should be adaptable without much effort.

Every experiment with the HIRLAM system thus starts from a working directory, in which at least the start-up script is found. If this is the only file in it, the experiment will automatically run all reference default scripts and programs. If the user wants to run modified versions of some scripts, he will have to copy them from the reference system directory and edit them. The run will then automatically use the modified scripts.

Also, facilities exist to run modified Fortran

programs. To accomplish this, the user must write a nupdate correction set and store it in his working directory. The system will use the correction set to create, compile and load a modified program. The system will only recognise the correction set as such if it is on a file with a prescribed title.

When setting up an experiment, the user identifies through the start-up file whether he wants to use existing input data, or create them.

An experiment normally consists of a number of cycles, each consisting of analysis and forecast. It is simple to leave one of these components out, or start the cycle with a forecast. The design of the system offers the user in principle the choice between running one job per cycle or one job for the whole experiment. In practice, the system is almost exclusively run in batch mode, with one job per cycle. Output from batch jobs at ECMWF is routed back to the member state with the ECMWF utility 'sendtm', or, if available, through the Unix utility 'ftp'.

3. System implementation

In the start-up script the user specifies a number of environment variables to identify himself, to define the computer configuration, to define the default reference system and to set the experiment dependent parameters. The list of environment variables is given in Appendix A.

A few examples of these environment variables are:

COMPENTRE: The computer centre hosting the experiment.
HL_LEV: The CFS root directory of the reference system.
HL_SCR: The directory containing the reference scripts.
HL_LIB: The directory containing the reference nupdate libraries, archives and executables.
HL_WD: The working directory for the scripts and nupdate correction sets specific to the experiment.
HL_DATA: The directory to store the data specific to the experiment.
HL_EXP: The directory in the archival system to store the data from HL_DATA

Because the reference system is maintained in CFS at ECMWF, the variable HL_LEV will only be referenced if COMPENTRE is ECMWF, or by jobs to copy (part of) the reference system to a local host (these jobs are in the set of utilities to port the system).

At ECMWF, the reference scripts are kept in a permanent Unicos directory, the name of which is in HL_SCR. Programs and executables are too big to be stored in the permanent file space, so they are kept in a volatile directory, HL_LIB. When running an experiment, the user may find that these files do not exist, and therefore the files in HL_LIB are reconstructed from HL_LEV in the CFS database, whenever they are needed and not present already. This maintenance of the files, further complicated by properties of the current ecfile implementation, is organised by a utility called Boot. (NB Due to security problems at ECMWF, the directory HL_LIB is not used at present; instead, the system creates a directory owned by the user.)

The user would normally distribute his experiment specific files over the two filesystems, the permanent one (directory HL_WD) for the (usually short) files to define his experiments and the volatile one (directory HL_DATA) for his (usually large) data files. Construction of the files in HL_WD may require a large human effort (e.g. to write nupdate correction sets), and it is thus fitting that the files be stored permanently, with appropriate backup facilities; the data files, on the other hand, are usually created automatically by

the experiment suite of programs, and thus it is acceptable that those files be less permanent. Some files in this directory (HL_DATA) may be required as input to later programs, e.g. to postprocess model output; this is organised by storing those files in the archive, in directory HL_EXP. This archive directory is thus the place where the results of the run are stored permanently. At ECMWF, this is a CFS directory.

On a local computer, the distinction between permanent and volatile files may be less strict than at ECMWF. In such case, the directories HL_SCR and HL_LIB need not be different; and also HL_WD and HL_DATA may be the same. Even, if there is no archiving system, the directory HL_EXP could be the same as HL_DATA.

Nupdate program libraries are in fact directories; with the current version of ecfile, it is not well possible to store directories in the CFS database. Therefore it was decided to convert the nupdate directory to a single file using the Unix archiving utility 'ar'. In CFS, the nupdate program libraries are thus maintained as single files. They must be converted back with 'ar' before they can be read by nupdate; this conversion is done automatically by the utility Boot.

The utility Boot also 'nupdate's, compiles, loads and executes the HIRLAM programs. It checks if a possibly existing executable is out of date. If not, it will not be recreated, thus saving the compilation and loading overhead. But if one of the object libraries to be linked is older than the relevant nupdate correction set, the library will be recreated by a nupdate-compile sequence. And if an object library is more recent than the executable, the executable will be brought upto date by invoking the loader.

4. System analysis library

4.1. The reference system

All HIRLAM programs and utilities are available to HIRLAM group members in nupdate program library form. Of some often used utilities also the object libraries are available. Because the analysis code takes very long to compile and load, and because the default analysis executable is applicable to a very wide range of experiments, that default executable is also maintained in the reference HIRLAM system.

At ECMWF, the script to start a HIRLAM run is found in the file \$HL_SCR/Start, where HL_SCR is /ec/nkg/HL2_SCR. The default scripts become available to the user by including \$HL_SCR in his path. If the user submits the Start script then the path will be set correctly automatically (unless of course the user overrode the path explicitly).

The default scripts organise the access to the default HIRLAM system. At ECMWF, this is accomplished by the purpose-built utility ecfile.x; at other computer centres, the environment variable \$HL_SYS defines the default system.

4.2. Installation of a local system

4.2.1. Installation

A number of utilities have been developed to facilitate the porting of the HIRLAM system to a local computer. These utilities are available to all HIRLAM staff members. They have been collected in the CFS directory

/nkg/hirlam2/member/port_local

This directory contains subdirectories for

1. joicr: Undo the effect of sendtm to split long Cray records to match VAX record lengths requirements
2. nupdate: A KNMI written restricted simulation of Cray's nupdate
3. scripts: Several scripts to facilitate life locally
4. unicos_simula: Several scripts to simulate Unicos utilities, not available on Convex (and perhaps others...)
5. lsab: Simulations of Cray library routines used by the analysis
6. man/cat: Manpages for several utilities, available in Unix manpage form (directories man1, man3) and in processed form (directories cat1, cat3). The general documentation script Doc in directory \$HL_SCR gives

details on how to access the manpages.
These subdirectories will be described in separate sections.

Many of the utilities will be of use if the memberstate's computer runs under a Unix dialect, some are however more general. Where it is known, system dependent features are indicated.

All file titles in CFS (ecfile) are in upper case characters. In general, some will have to be converted (perhaps partially) to lower case.

Besides the general utilities there are some particularities of the Cray version of the analysis code. They mainly are Cray library references. They have been collected into two libraries,

```
/nkg/hirlam2/member/port_local/lsab/xtn.apl  
/nkg/hirlam2/convex/cvx.apl
```

The first is (we think) not computer specific, and can be accessed by all HIRLAM members. (Currently it contains a Convex specific version of a Fortran function to get the record length of the next record, LENNRC; we intend to move it to the library port, or to remove it altogether). The second contains code specific for Convex, developed by the Convex Corporation, and therefore accessible only after permission from Convex. Parts of the Convex code are in assembler. A makefile to compile the libraries xtn.apl and cvx.apl is available in

```
/nkg/hirlam2/member/port_local/lsab/makefile
```

Parts of this makefile are of necessity rather Convex specific.

4.2.2. j_o_i_c_r

In this directory file format conversion programs have been collected. Included are programs to convert files produced on a Cray or on a VAX/VMS to native blocking format. Furthermore, this directory contains the files to run the utility joicr, which is a utility to join the pieces produced by the CHOPPER utility in ECMWF's sendtm (See ECMWF's manpages for sendtm, ref. Norbert Kreitz). The utilities in this directory are probably useful to all users of the computer system, not just the HIRLAM members. It is therefore suggested the system manager make it available in the directory /usr/local/bin.

The files in this directory are C or Fortran-77 sources. (Note: Unix file titles in lower case). The

C programs unravel the input file, the Fortran subroutine fortranwrite.f ensures that the produced file has the native Fortran record structure. There is a makefile to facilitate the installation of the utilities. it currently contains Convex specific statements (notably the Fortran compiler is called fc). Also the C programs contain a possibly Convex-specific method to call Fortran modules from a C program. The man directory contains the manpages for the utilities; it contains some references to Convex and KNMI but they seem to be used in the sense of 'non-Cray' and 'non-ECMWF' respectively.

Ref: Toon Moene, KNMI.

4.2.3. nupdate

The directory nupdate contains the C source code for the Unicos nupdate simulation. The source contains instructions how to compile it. Use lower case file titles only. It is useful to create the executable nupdate in a directory in your path, and set the HIRLAM environment variable HL_UP to that directory name, with a / at the end.

Not all features of nupdate have been simulated. For example, the simulation does not keep an archive of deleted cards. It creates line numbers for inserted cards, but there is not an absolute guarantee that it uses the same algorithm to do so as Cray. Also note the following restrictions:

The current version supports only the directives BEFORE, CALL, COMDECK, DECK, DELETE, ELSE, ENDIF, IF DEF, IF -DEF, INSERT and their abbreviations. The DEF directives may test on the definition of a string only. Source lines lengths are by default limited to 72 characters, the option -w on the command line may override this.

The man directory contains a version of CRI's nupdate manpages, modified to fit KNMI's version of nupdate.

Also the Fortran program nsplitle.f is in this directory. It is used to split a Cray-nupdate source file into a program library with one deck per file, which is the format expected by the nupdate simulation. Compile it with the standard Fortran compiler, e.g. on Convex:

```
fc -o ${HL_UP}nsplitle nsplitle.f
```

Note lower case characters in nsplitle.f and also note

my suggestion to store the executable also in the nupdate directory HL_UP.

Ref: nupdate: Toon Moene, KNMI
nsplitpl: Ben Wijchers Schreur, KNMI.

4.2.4. Scripts

In this directory you find scripts that we suggest your local HIRLAM system manager put on your local machine in the directory with all HIRLAM-system scripts. HIRLAM users should put this directory in their path. Because HIRLAM scripts have their names start with a capital, followed by lower case characters, these scripts should have their names correspondingly.

The script Parec is run to process output sent by the HIRLAM script Sendtm. (Ref: Ben Wijchers Schreur, KNMI). Command line arguments to Parec come in pairs of '-option value'. The following options exist:

```
option value
-f Sendtm file(s) to be done (relative or full path)
-d directory where Sendtm files are to be sought
-s string to separate sections in sendtm file
-v switch for verbose mode
-l switch to avoid execution of joicr
```

If an argument is not a member of a pair of one of these options and a value, it is ignored. The -f flag may be repeated or contain a multitude of names (space separated). Defaults are:

```
option default
-f files with names *binary_?? or *ecmwfoutput_??
(where * and ?? are the shell metacharacters) in
the directory specified by the option -d
-d environment variable ECMWFFILES if set, else
$HOME/ecmwffiles where $HOME is the user's home
directory
-s the string EOS-par
-v no verbose mode
-l execute joicr
```

In normal execution mode, the option -l will not be set. So joicr will be executed. For ascii files, this will lead to an error condition, after which Parec will try to analyse the file with some Unix utilities. Both for ascii and binary files, Parec will check if the file matches a prescribed format, with a division into sections, separated by the string as given by the -s option. If the file fails this check, its title will

be extended with the character '#', so as to avoid reprocessing by Parec. But if the file passes the check, the file will be split into sections. The first section is skipped, the second is used to derive a new file title, the third is assumed to be a script, the fourth section is assumed to contain the actual data; it will be copied to a file of which the title was given in the second section. Then the script as present in the third section will be executed, with command line arguments as given in the second section. So the first command line argument is always the data file title.

The script Nmodex creates a nupdate correction set from a file passed through your preferred editor. (Ref: Gerard Cats, KNMI).

The script Get is useful to send (parts of) the HIRLAM system to your home computer. By specifying the environment variable APL you select to extract the HIRLAM archived nupdate program libraries. They are sent in a compacted form, and contain instructions how to decompact them. With Parec (see above) you execute those instructions. For this, you will need nsplipl (see above) and the script Decompress in your path. The script Decompress is also available in the CFS sub-directory port_local/scripts. Some of the HIRLAM nupdate program libraries will be turned into archives of compiled objects. The instructions to create the objects are in the script Boot, which should be in your path. The mother copy of Boot is available in the HIRLAM system directory at ECMWF. The target directory in which Parec will put the files gotten with Get is specified in subdirectories of the directory LOCAL_SYSTEM. The source codes (files apl) will be stored in \$LOCAL_SYSTEM/lib. Your local HIRLAM system manager could e.g. install the HIRLAM system libraries port and util by using

```
Get APL=port,util LOCAL_SYSTEM=<your value of $HL_SYS>
```

In this case, the following files will be created:

```
$HL_SYS/lib/port.apl      (ar of program library)
$HL_SYS/lib/util.apl      (ar of program library)
$HL_SYS/lib/port.a        (ar of objects)
$HL_SYS/lib/util.a        (ar of objects)
```

(In here, ar is the Unix archiver ar).

4.2.5. _U_n_i_c_o_s__s_i_m_u_l_a

This directory contains the Unicos utilities used by the HIRLAM system that were not available on the

Convex. These are: assign, cut, ecfile and ja. The last two, ecfile and ja, are in fact not simulations, but mere dummies. Note that they are recognised as simulations only if their names are in lower case. (Ref: Gerard Cats, KNMI).

4.2.6. L_S_A_B

4.2.6.1. P_r_e_p_c_v_x_m_e_m_m_a_n

The script Prepcvxmemman replaces the character strings by which the memory manager routines identify chunks of memory by integers. This can be used to design memory manager routines in a much more efficient way: The search for matching strings is replaced by a direct memory reference. This may save 5% of the analysis code execution time. The script creates and runs a perl script, so we hope your system supports perl. Usage:

```
Prepcvxmemman file.in file.out
```

(Ref: Gerard Cats, KNMI).

4.2.6.2. T_h_e _C_o_n_v_e_x _l_i_b_r_a_r_y

Currently the following persons are allowed access to the library cvx.apl:

Gerard Cats	KNMI
Nils Gustafsson	SMHI

(Ref: Ben Wijchers Schreur, KNMI).

4.2.6.3. H_o_w _t_o _c_r_e_a_t_e _a _H_I_R_L_A_M _a_n_a_l_y_s_i_s _e_x_e_c_u_t_a_b_l_e.

This subsection gives some short instructions, step by step (with an indication of Convex specific features):

- 1) Install the HIRLAM scripts in the directory \$HL_SCR.
- 2) Install the HIRLAM libraries in the directory \$HL_LIB. Do not forget to run Prepcvxmemman (Convex!).
- 3) Copy the files:

```
lsab.makefile (source: port_local/lsab/makefile)
lsab.cvx.apl (source: convex/cvx.apl) (Convex!)
lsab.xtn.apl (source: port_local/lsab/xtn.apl)
```

to the directory \$HL_LIB. The utility Get can be used for this:

```
Get APL=cvx,xtn LOCAL_SYSTEM=<your value of $HL_SYS>
```

The file will be created in \$HL_SYS/lib, which you should ensure to be the same as \$HL_LIB. To Get cvx, you need to have access to it.

4) Change directory to \$HL_LIB and run:

```
make -f lsab.makefile # (Convex!)
```

5) Make a working directory \$HL_WD and copy the files:

```
Start  
lsab.updin
```

6) Edit Start to make it reflect your requirements. In particular specify the archive directories that contain the boundary files, AOF-file, climatology files, error coefficients and such. Specify the cycle parameters.

7) Start.

Steps 1) and 2) are part of the general HIRLAM system installation. Steps 3) and 4) are specific for installation of the analysis code. Steps 5), 6) and 7) are usual steps to run parts of the HIRLAM system. (Ref: Ben Wijchers Schreur, KNMI).

4.2.7. Tree structure

The tree structure of the files described here is as below. The subdirectory structure of /nkg/hirlam2/member/port_local was created by executing the documentation utility Scan_ecfile, as follows:

```
Scan_ecfile -r.. /nkg/hirlam2/member/port_local
```

Tree structure:

```

convex | cvx.apl

/nkg/hirlam2 | member | port_local | (followed by:)
               processed manpages of executables
cat1/          diasim.1      diagnose the contents of an ASIMOF file
               ec2asi.1      create an ASIMOF file from GRIB data
               fromcosblocked.1 convert from Cray file format
               fromvaxblocked.1 convert from VAX file format
               intp.1        horizontal interpolation
               joicr.1       process files received through sendtm
               nmodex.1      create a nupdate correction set
               nupdate.1     KNMI nupdate simulation
cat3/          asimhc.3     processed manpages of subroutines
               asimhm.3     close ASIMOF file
               asimhr.3     get machine constants
               asimhw.3     read a record from an ASIMOF file
               bufrex.3     write a record to an ASIMOF file
               degrib.3      decode BUFR message
               engrib.3      decode GRIB message
               getfd.3       encode GRIB message
               grbmod.3      read and decode a GRIB message
               loadfd.3     set GRIB (de)coding mode
               putfd.3      open ASIMOF file, construct inventory
               read.3        code and write a GRIB message
               read.3        read a record of unknown length
joicr/         fortranwrite.f file format conversion utilities
               fromcosblocked.c write a record, Fortran style
               fromvaxblocked.c see cat1 above
               joicr.c       see cat1 above
               makefile      see cat1 above
lsab/          makefile      to create executables
               prepcvxmemman analysis code porting utilities
               xtn.apl      to create executables
               optimise memory manager
               simulations of Cray routines
man1/          diasim.1     raw manpages, see further cat1 above
               ec2asi.1
               fromcosblocked.1
               fromvaxblocked.1
               intp.1
               joicr.1
               nmodex.1
               nupdate.1
man3/          asimhc.3     raw manpages, see further cat3 above
               asimhm.3
               asimhr.3
               asimhw.3
               bufrex.3
               degrib.3

```

```

.. engrib.3
.. getfd.3
.. grbmod.3
.. loadfd.3
.. putfd.3
.. read.3
nupdate/
.. nsplipl.f      convert a Cray nupdate source file
.. nupdate.c      see cat1 above
scripts/
.. Decompress    expand a compressed Cray nupdate source
.. Get           send a compressed Cray nupdate source
.. Nmodex        see cat1 above
.. Parec          process a file sent from ECMWF
unicos_simula/   Unicos utilities, simulated
.. assign         symbolic link
.. cut            extract columns from a file
.. ecfiler       archiving utility
.. ja             job accounting

```

4.3. Hardware dependent features

The reference system runs on a Cray-C90, and therefore all Cray-specific features are part of the default system. The HIRLAM system has also been ported to Convex machines. Some Convex-specific code has been included in the default HIRLAM system, it is selectable by a 'nupdate define option'.

Both companies (Cray and Convex) have developed machine dependent optimisations. These are considered proprietary software, and not available in the default system.

Parts of the HIRLAM code have been ported also to other hardware platforms, but we are not aware of any operational experience with the full HIRLAM system outside Cray and Convex.

5. Appendices

5.1. Appendix: Environment variables

To extract the most upto date list of environment variables, use the utility Doc. From the C shell, do

```
setenv HL_PW your_hirlam_password  
/ec/nkg/HL2_SCR/Doc environment
```

and from the Bourne shell

```
HL_PW=your_hirlam_password  
export HL_PW  
/ec/nkg/HL2_SCR/Doc environment
```

Essentially, this will produce the Tables as shown below, but the output of Doc may need editing e.g. to reduce the length of some long lines, and to remove some garbage generated by Doc.

Environment variables cross-reference
=====

Derived on Wed May 4 12:16:09 GMT 1994, from scripts in
/ec/nkg/HL2_SCR

Variables referenced by each script:

:
Actions :BDDIR CLDIR COLLECT COMPCENTRE DTG DTGEND EXP FCINT
GVDB HL_DATA HL_WD HOME HOST LATEST LL MAXBACK OBDIR
OPTION PATH PROGRESS QUEUE SETENV Start TRDB USER
Add_sstice_obs:COMPCENTRE HL_PW
Analyse :AFDIR CLDIR COF COMPCENTRE CYCLE DTG FCINT HL_DATA
HL_PW HL_WD HOST IN INLINE NAGLIB NBROWS NLAT OUT
PP SCILIB SETENV VERIFY WD
AnalyseInput :BLDIR CLDIR DTG FAE FBL
Anaveri :COF COMPCENTRE HL_LIB IN OUT SHFLAGS WD
Archive :COMPCENTRE DIRS FILE HL_COMDAT HL_DATA HL_EXP HOME
PARKDIR PARKLIST RELYMD SAVEDIR SETENV
As2puregb :HOST IN OUT PROTOCOL SEND_FORMAT SETENV
Bdries :DATA_DIS EAST IN NLAT NLON NORTH ORO OUT POLAT POLON
SETENV SOUTH T WEST
Bitmap :DXEC DYEC EAST EASTEC NBNDRY NLAT NLON NORTH
NORTHEC
POLAT POLON SETENV SHFLAGS SOUTH SOUTHEC WEST WESTEC
Boot :ALT APL COMPCENTRE COST DEBUG DEFAULT FLAGS HL_DATA
HL_LEV1 HL_LIB HL_PW HL_SCR HL_UP HL_WD HOST
INCLUDE LDLIB LIB MAIN NUFLAGS PATH PGM PMD WD
Boundaries :BDDIR BDEXP BDLL COMPCENTRE CYCLE DATE DTG DXEC
DYEC
EASTEC ECFOPT FFILE HH HL_COMDAT HL_DATA HL_PW HL_SCR
NORTHEC PATH PATH PROTOCOL RELYMD REMOTE SEND SETENV
SOUTHEC TMPDIR TRUNCN TYPE USER WD WESTEC
Cleanup :EXP HL_COMDAT HL_DATA HL_EXP TEMP USER
Compare :BUFRTAB_DIR CONTROL_DTG FIELDS GRIB_VN OBS
Destag :ECLIB EXP IN OUT SETENV WD
Diagnose :ASIMOFFILE ASIM_OPTIONS HL_WD SETENV
Doc :HL_PW HL_SCR TMPDIR
Environment :COMPCENTRE HL_PW HL_SCR HL_SYS
Extr_oro_gp :DXEC DYEC ECLIB EMOSLIB HL_PW OUT SETENV TMPDIR
TRUNCN
ExtractArea :IN LOWLEFTLAT LOWLEFTLON NLAT NLON NLON*\$NLAT OUT
SETENV
Fg :BDEXP BDLL CYCLE DTGBEG FCINT HL_DATA IN OUT SETENV
TYPE WD
Forecast :DTG EXP JMINPP LL NDTIME NDTPHYS NLEV NLON NLONPP
SETENV TIMESU TYPE WD WRITUPTIMES
Ftput :HOME SETENV
Hirl2asi :HL_WD IN OUT SETENV
Hirlam :DATE] \$HLUTC EXP HLUTC HOME
Horint :IN OUT
Lock :FILE HOME LOCK LOCKLIST
Lp :BATCHOUT COMPCENTRE EXP FFILE MEMBERSTATE PATH
PROTOCOL QSUB_PATH REMOTE SETENV
SETENV TMPDIR WD
Main :BATCHOUT DTG HL_WD OUTDIR PATH QSUB_PATH
QSUB_WORKDIR
MakeBUFRtabs :BUFRTAB_DIR BUFR_ED BUFR_VN COMPCENTRE HL_EXP
HL_PW

```

SETENV
MakeStrategy :ATUTC BDDIR BDINT BDLENGTH_MIN BDLL BD_MODE DTG
ECDTG

ECUTC HH HLUTC LL NOW PARDIR SETENV STRATEGY TYPE WD
Man :HL_PW TMPDIR
Mkdir :HOST
News :COMPCENTRE SHFLAGS USER
Nextbd :BDEXP BDLL HH TYPE WD
Postpp :DTG EXP HL_WD IMINPP IN JMINPP NAGLIB NLAT NLATPP NLEV
NLON NLONPP SCILIB SETENV WD
Postproc :COMPCENTRE EXP FFILE PROTOCOL REMOTE SETENV
PrepareLibs :COMPCENTRE EXP HL_WD NLEV NLEVEC NLON SETENV WD
Prepob :COMPCENTRE CYCLE DTG ECLIB FCINT HL_COMDAT IN OBDIR
OBEXP OUT PROTOCOL REMOTE SEND SETENV
Preps :CLDIR CLDIR CLEXP COMPCENTRE EAST EMOSLIB EXP
HL_COMDAT

HL_PW HL_WD MEANOR MISSING NAGLIB NLAT NLON NORTH
POLAT POLON SETENV SOUTH WD
Prog :DTG IN NAM NUMB NUMH SCILIB SETENV WD
Run :CWD CYCLE DTG ENVIRONMENT EXP FCINT FFG HL_DATA

HOME LATEST MORE_CYCLES OPTION OUTDIR PROGRESS
QSUB_WORKDIR QUEUE SETENV STRATEGY WD
Scan_ecfile :HL_PW
Sendtm :COMPCENTRE EXT FFILE FMT FTP_ADDRESS HEAD HL_WD HOME
LFILE LOG MAXLEN MEMBERSTATE PROTOCOL REMOTE SETENV
STMOPTION TMPDIR USER
Sort :BDLL DATE DIRS DTG HL_WD SETENV TMPDIR TYPE WD
Span :CL FG OUT SETENV
Start :AEDIR AEEXP AFDIR ANACVX ANAXTN BATCHOUT BDDIR BDEXP
BDINT BDLL BLDIR BLEXP BUFRTAB_DIR CLDIR CLEAN CLEXP
COMPCENTRE CVX CWD CYCLE DTG DTGBEG DTGEND EAST ECLIB
EXP FCINT FFEXP FFG FFILE FTP_ADDRESS HL_ARC HL_COMDAT
HL_DATA HL_EXP HL_HS HL_lev HL_LIB HL_Pw HL_SCR HL_UP
HL_WD HL_WD HL_WD HOME HOST IMINPP INLINE JMINPP LL
MEMBERSTATE MORE_CYCLES NAGLIB NBNDRY NDTIME NDTPHYS
NDTVDIF NLAT NLATPP NLEV NLEVEC NLON NLONPP NORTH
OBDIR OBEXP OPTION OUTDIR PARDIR PAREXP PATH PMOUT
POLAT POLON PP PROTOCOL QSUB_WORKDIR QUEUE RELYMD
REMOTE SCILIB SEND SETENV SHFLAGS SLABOUT SOUTH
STRATEGY TEMP TIMESU TRUNCN USER VERBOSE VERIFY
VINOUT WEST WRITUPTIMES
Statis :COF COMPCENTRE ECLIB HL_EXP HUM MAS SHFLAGS SPECLIB WD
Trace :PARENT VERBOSE
Traps :HOST SETENV SIGQUIT SIGTERM SIGXCPU
Uvstag :AND ANS ECLIB EXP FGD FGS SETENV WD
Vineta :CLDIR DTG EXP IN MM MODE OUT SCILIB SETENV VINOUT
Wrapup :HL_WD HOME QSUB_REQNAME QUEUE SETENV VINOUT WD

```

Meaning of each variable and scripts referencing it:
 (referencing scripts are on the line starting with '<tab><tab>-',
 a variable is exported by scripts marked 'x>')

```

AEDIR      =Directory-name of first analysis error.
            - Start
AEEXP       =Experiment-name of first analysis error.
            - Start
AFDIR       =Directory-name of observation and fg errors statistics
            files.
            - Analyse Start
ALT         =Directory with alternative fortran sources and objects.
            - Boot
ANACVX     =Library with Convex extensions to the analysis code.
            - Start
ANAXTN     =Library with extensions to the analysis code.
            - Start
AND         =Destaggered file.
            - Uvstag
ANS         =Staggered file.
            - Uvstag
APL         =Directory of the source files in apl format.
            - Boot
ASIMOFFILE =File title(s) of ASIMOF files.
            - Diagnose
ASIM_OPTIONS=Options to diagnose ASIMOF files.
            - Diagnose
ATUTC       =Availability time (format
            - MakeStrategy
BATCHOUT    =Name of local file to be 'sendtm'-ed to member state.
            - Lp Main Start x>Actions x>Actions x>Hirlam x>Main
            x>Run x>Start
BDDIR       =Directory-name of boundary files.
            - Actions Boundaries MakeStrategy Start
BDEXP       =Experiment-name of boundary files.
            - Boundaries Fg Nextbd Start
BDINT       =Intervals at which boundaries are provided (hours,
            2 digits).
            - MakeStrategy Start
BDLENGTH_MIN=Minimum length of a boundary file to be recognised
            as such.
            - MakeStrategy
BDLL        =Forecast length of boundaries (hours, 2 digits).
            - Boundaries Fg MakeStrategy Nextbd Sort Start
BD_MODE     =Select recipe to create list of boundary files.
            - MakeStrategy
BL          =Name of black list history file.
BLDIR .     =Directory-name of first black list.
            - AnalyseInput Start
BLEXP       =Experiment-name of first black list.
            - Start

```

```

BUFRTAB_DEV =Fortran device number to process BUFR tables.
              - x>Prepob
BUFRTAB_DIR =Directory of BUFR tables.
              - Compare MakeBUFRtabs Start x>Compare x>MakeBUFRtabs
              - x>Prepob
BUFR_ED     =Edition number of BUFR tables.
              - MakeBUFRtabs
BUFR_VN     =Version number of BUFR tables.
              - MakeBUFRtabs
CL          =Climatology file title (obsolete!).
              - Span
CLDIR       =Directory-name of climate files.
              - Actions Analyse AnalyseInput Preps Preps Start
              - Vineta
CLEAN       =Redo or continue the experiment?
              - Start
CLEXP       =Experiment-name of climate files.
              - Preps Start
COF         =Title of comprehensive observation file.
              - Analyse Anaveri Statis
COLLECT     =Cut-off for observations, format hh
              - Actions
COMPCENTRE  =Name of computer centre where the job is to run.
              - Actions Add_sstice_obs Analyse Anaveri Archive Boot
              - Boundaries Environment Lp MakeBUFRtabs News
              - Postproc PrepareLibs Prepob Preps Sendtm Start
              - Statis x>Actions x>Hirlam x>Start
CONTROL_DTG =File to control date/time selection.
              - Compare
COST        =Function to invoke HIRLAM executables.
              - Boot
CRUTC       =Defines cron execution for a local version of
              - MakeStrategy.
CVX         =Library of Convex simulations of GATHR, GBYTE, etc.
              - Start
CWD         =Directory from which the experiment was started.
              - Run Start x>Main
CYCLE       =Definition of the constituents of a cycle of the
              - experiment.
              - Analyse Boundaries Fg Prepob Run Start
DATA_DIS    =Dissemination data?
              - Bdries
DATE        =Date of boundary file.
              - Boundaries
DEBUG       =Switch for compile and load with debug options, set
              - DEBUG=on.
              - Boot
DEFAULT     =Directory where the archives (.a files) are to be found.
              - Boot
DIR         =Directory of boundary file (local implementation).
DIRS        =Directives to process generated boundary files.
              - Archive Sort
DTG         =Current cycle date/time.
              - Actions Analyse AnalyseInput Boundaries Forecast

```

```

Main MakeStrategy Postpp Prepbob Prog Run Sort
Start Vineta x>Actions x>Hirlam x>Run x>Start
DTGBEG =Begin date of the experiment in yyymmddhh.
          - Fg Start x>Actions x>Hirlam x>Start
DTGEND =End date of the experiment in yyymmddhh.
          - Actions Start x>Actions x>Hirlam x>Start
DXEC =x-Resolution of boundary files.
          - Bitmap Boundaries Extr_oro_gp
DYEC =y-Resolution of boundary files.
          - Bitmap Boundaries Extr_oro_gp
EAST =East border of the area (in degrees).
          - Bdries Bitmap Preps Start
EASTEC =East border of the boundary file area (in degrees).
          - Bitmap Boundaries
ECDTG =yyymmddhh of ECMWF files, used by a local version of
          - MakeStrategy
ECFOPT =ecfile options for storage of boundary files.
          - Boundaries
ECLIB =ECMWF software library file title.
          - Destag Extr_oro_gp Prepbob Start Statis Uvstag
ECMWFFILES =Local system directory of files 'sendtm'-ed from ECMWF.
ECUTC =Time of ECMWF forecast (used by a local version of
          - MakeStrategy).
          - MakeStrategy x>Actions x>Actions x>Hirlam
EMOSLIB =File title of emoslib at ECMWF.
          - Extr_oro_gp Preps
ENVIRONMENT =Run environment (BATCH, INTERACTIVE).
          - Run
EXP =Experiment code.
          - Actions Cleanup Destag Forecast Hirlam Lp Postpp
              Postproc PrepareLibs Preps Run Start Uvstag Vineta
              x>Actions x>Hirlam
EXT =Extension of the file title 'sendtm'-ed at ECMWF.
          - Sendtm
FAE =First analysis error file title (used for first cycle
          only).
          - AnalyseInput
FBL =First black list history file title (used for first cycle
          only).
          - AnalyseInput
FCINT =Cycle length (hours), i.e. interval between two forecasts.
          - Actions Analyse Fg Prepbob Run Start x>Actions
FFEXP =Experiment-name of first first guess.
          - Start
FFG =First first-guess file title (used for first cycle only).
          - Run Start
FFILE =First 7 characters of title of file received from sendtm.
          - Boundaries Lp Postproc Sendtm Start
FG =first guess data file.
          - Span
FGD =Destaggered first guess file title.
          - Uvstag
FGS =Staggered first guess file title.

```

```

FIELDS      - Uvstag
            =List of files to be verified against observations.
            - Compare
FILE        =File title, esp. title of Main script in the experiment.
            - Archive Lock
FILEENV     =Title of file-environment (assign-statements) file on
            Cray.
FIRST       =Indicator of first job in sequence.
FLAGS       =Compiler and loader options set in Boot
            - Boot
FMT         =Format (us for binary, c8 for ascii) of file to be
            'sendtm'-ed.
            - Sendtm
FTP_ADDRESS Sendtm Start
            - To ftp files to (combined with ~/.netrc).
GRIB_VN     =GRIB version number of fields to be verified.
            - Compare
GVDB        =Directory with GRIB/ASIMOF files.
            - Actions
HEAD        =Title of file containing header for tile to be
            'sendtm'-ed.
            - Sendtm
HH          =Time (hours, 2 digits) of boundary file.
            - Boundaries MakeStrategy Nextbd
HLUTC       =Time of HIRLAM forecast (used by local version of
            MakeStrategy).
            - Hirlam MakeStrategy x>Actions x>Hirlam
HL_ARC      =Common root directory for input files.
            - Start x>Main
HL_COMDAT   =Directory for data files common to several experiments.
            - Archive Boundaries Cleanup Prepob Preps Start
HL_DATA     =Directory of output data, (at ECMWF, short-lived archive).
            - Actions Analyse Archive Boot Boundaries Cleanup Fg
            Run Start x>Actions x>Hirlam x>Main x>Start
HL_EXP      =At ECMWF, CFS directory of output data (long-lived).
            - Archive Cleanup MakeBUFRtabs Start Statis x>Hirlam
            x>Main
HL_HS       =Directory for user-specific scripts, part of the search
            path.
            - Start
HL_LEV1     Boot
HLLEV       =HIRLAM-level-2-main-directory, at ECMWF in CFS.
            - Start x>Main
HL_LIB      =Directory of libraries.
            - Anaveri Boot Start x>Actions x>Hirlam x>Start
HL_PW       =User's HIRLAM-password (only on ECMWF).
            - Add_sstice_obs Analyse Boot Boundaries Doc
            Environment Extr_oro_gp MakeBUFRtabs Man Preps
            Scan_ecfile Start x>Actions x>Hirlam x>Main
HL_SCR      =Directory with UNIX-scripts, in path after HL_HS.
            - Boot Boot Boundaries Doc Environment Start x>Actions
            x>Hirlam x>Main x>Start
HL_SYS      =Common root directory for the HIRLAM system.
            - Environment

```

```

HL_UP      =Directory of the nupdate utility, including closing /.
            - Boot Start x>Actions x>Hirlam x>Start
HL_WD      =Working directory for the experiment (e.g. scripts,
            corrections).
            - Actions Analyse Boot Diagnose Hirl2asi Main Postpp
              PrepareLibs Preps Run Sendtm Sort Start Wrapup
              x>Actions x>Diagnose x>Hirlam x>Main x>Postpp
              x>Start
HOME       =User's home directory.
            - Actions Archive Ftpput Hirlam Lock Run Sendtm Start
              Wrapup
HOST       =Type of machine (CRAY, CONVEX).
            - Actions Analyse As2puregb Boot Mkdir Start Traps
              x>Actions x>Boundaries x>Hirlam x>Start
HUM        =Humidity statistics output file title.
            - Statis
IMINPP     =First x-point for sub-area postprocessing.
            - Postpp Start
IN         =Input file title.
            - Analyse Anaveri As2puregb Bdries Destag ExtractArea
              Fg Hirl2asi Horint Postpp Prepob Prog Vineta
INCLUDE    =
            - Boot
INLINE     =Directory of routines to be inlined for Convex analysis
            code.
            - Analyse Start
JMINPP     =First y-point for sub-area postprocessing.
            - Forecast Postpp Start
LATEST     =Title of the last file in the cycle (analysis, forecast,
            etc.).
            - Actions Run x>Actions x>Run x>Start
LDLIB      =Non-HIRLAM libraries to be loaded.
            - Boot
LFILE      =File to be 'sendtm'ed.
            - Sendtm
LIB        =HIRLAM libraries to be loaded.
            - Boot
LIST       =List of variables to be read from standard input.
LL         =Forecast length (hours, 2digits).
            - Actions Forecast MakeStrategy Startx>Actions
LOCK       =File to be locked.
            - Lock
LOCKLIST   =List of locked files.
            - Lock x>Actions x>Actions x>Start
LOG        =Title of log file.
            - Sendtm
LOWLEFTLAT =Latitude of Southern most row of subarea to be extracted.
            - ExtractArea
LOWLEFTLON =Longitude of Western most row of subarea to be extracted.
            - ExtractArea
MAIN       =Main program (Boot).
            - Boot
MAS        =Mass/wind statistics output file title.
            - Statis

```

MAXBACK =Maximum recover period before coldstart is requested.
 - Actions
 MAXLEN =Maximum length (bytes) of files to be 'sendtm'-ed.
 - Sendtm
 MAXTRY =Maximum number of trials to recupe a failed 'sendtm' file.
 MEANOR =Indicator for mean orography.
 - Preps x>Preps
 MEMBERSTATE =Memberstate id (for sendtm at ECMWF etc.).
 - Lp Sendtm Start
 MISSING =File(s) to be created from the climate system.
 - Preps
 MM =Month (computed from DTG).
 - Vineta
 MODE =Mode indicator.
 - Vineta
 MORE_CYCLES =More cycles in this batch job?
 - Run Start
 NAGLIB =NAG - library.
 - Analyse Postpp Preps Start x>Postpp
 NAM =Title of file with forecast (and, obsolete, postpp)
 namelist.
 - Prog
 NBNDRY =With of boundary zone.
 - Bitmap Start
 NBROWS =Number of gridpoint rows within one analysis base box.
 - Analyse
 NDTIME =Length of time step in Forecast in seconds (default=300).
 - Forecast Start
 NDTPHYS =Timestep for physics.
 - Forecast Start
 NDTVdif =Timestep for vertical diffusion.
 - Start
 NLAT =Number of grid-points in latitudinal direction.
 - Analyse Bdries Bitmap ExtractArea Postpp Preps Start
 NLATPP =Number of grid-points in lat direction for sub-arera
 postprocessing.
 - Postpp Start
 NLEV =Number of levels.
 - Forecast Postpp PrepareLibs Start
 NLEVEC =Number of levels in boundary files.
 - PrepareLibs Start
 NLON =Number of grid-points in longitudinal direction.
 - Bdries Bitmap ExtractArea Forecast Postpp
 PrepareLibs Preps Start
 NLONPP =Number of grid-points in lon direction for sub-arera
 postprocessing.
 - Forecast Postpp Start
 NORTH =North border of area.
 - Bdries Bitmap Preps Start
 NORTHEC =North border of the boundary file area (in degrees).
 - Bitmap Boundaries
 NOW =Current date/time.
 - MakeStrategy x>Actions x>Hirlam x>Start
 NUFLAGS =Nupdate flags.

```

        - Boot
NUMB      =Number of boundary files.
        - Prog
NUMH      =Number of history files.
        - Prog
OBDIR     =Directory-name of observation files.
        - Actions Preprob Start
OBEXP     =Experiment-name of observations.
        - Preprob Start
OBS       =File with observations (BUFR) used for verification.
        - Compare
OPTION    =Options for qsub (nqs).
        - Actions Run Start
ORO       =Orography file title
        - Bdries
OUT       =Output file title.
        - Analyse Anaveri As2puregb Bdries Destag Extr_oro_gp
          ExtractArea Fg Hir12asi Horint Preprob Span Vineta
OUTDIR    =Directory for printer output files.
        - Main Run Start
PARDIR    =Directory of experiment to be parallelled.
        - MakeStrategy Start
PARENT    =Calling script.
        - Trace x>Trace
PAREXP    =Name of experiment to be parallelled.
        - Start
PARKDIR   =Directory for volatile storage.
        - Archive
PARKLIST  =Files archived in volatile storage.
        - Archive x>Start x>Start.931223
PATH      =Search path (including HL_WD, HL_HS, HL_SCR in this order).
        - Actions Boot Boundaries Lp Main Start
          x>Actions x>Boot x>Boundaries x>Hirlam x>Main
          x>Start
PGM       =Program name of executable (.x file) to be created and
          executed.
        - Boot
PMD       =Switch for post-mortem-dump, set PMD=on.
        - Boot
PMOUT     =Title of Postpm printer output file.
        - Start
POLAT     =Latitude of the South pole.
        - Bdries Bitmap Preps Start
POLON     =Longitude of the South pole.
        - Bdries Bitmap Preps Start
PP        =List of actions to be postprocessed (analysis,
          forecast,...).
        - Analyse Start
PROGRESS  =File containing experiment progress information.
        - Actions Run x>Actions x>Hirlam
PROTOCOL  =File transfer protocol.
        - As2puregb Boundaries Lp Postproc Preprob Sendtm Start
QSUB_PATH =Search path passed from RQS.
        - Lp Main

```

```

QSUB_REQID =Request-id from RQS.
QSUB_REQNAME=Request name from RQS.
    - Wrapup
QSUB_WORKDIR=Working directory from RQS.
    - Main Run Start
QUEUE      =Queue specification for qsub (nqs).
    - Actions Run Start Wrapup x>Actions x>Actions
RECFILE     =File to be recupered (Local implementation).
RELYMD      =Release date of files stored in ECFILE (default= 1 year).
    - Archive Boundaries Start
REMOTE      =Title of 'sendtm'-ed file at member state host.
    - Boundaries Lp Postproc Prepob Sendtm Start
SAVEDIR     =Permanent storage directory.
    - Archive
SCILIB       =Scientific library.
    - Analyse Postpp Prog Start Vineta x>Postpp
SEND        =List of actions from which output file will be
    'sendtm'-ed.
    - Boundaries Prepob Start
SEND_FORMAT =Format of file to be sent.
    - As2puregb
SETENV      =File containing experiment 'environment; default
    - Actions Analyse Archive As2puregb Bdries Bitmap
      Boundaries Destag Diagnose Extr_oro_gp ExtractArea
      Fg Forecast Ftpput Hirl2asi Lp Main MakeBUFRtabs
      MakeStrategy Postpp Postproc PrepareLibs Prepob
      Preps Prog Run Sendtm Sort Span Start Traps
      Uvstag Vineta Wrapup x>Actions x>Hirlam x>Start
SHFLAGS     =Flags for the Bourne shell; include at least -k.
    - Anaveri Bitmap News Start Statis
SIGQUIT     =Quit signal.
    - Traps
SIGTERM     =Terminate signal.
    - Traps
SIGXCPU     =CPU time limit excess signal.
    - Traps
SLABOUT     =Title of Slabba printer output file.
    - Start
SOUTH       =South border of area.
    - Bdries Bitmap Preps Start
SOUTHEC    =South border of the boundary file area (in degrees).
    - Bitmap Boundaries
SPECLIB     =Title of speclib.a (ECMWF only).
    - Statis
STMOPTION   =Additional options for sendtm.
    - Sendtm
STRATEGY    =Strategy to use boundary files (e.g. ECMWF analyses).
    - MakeStrategy Run Start
SUB         =Subtype of file to be processed.
TEMP        =Semi-permanent directory.
    - Cleanup Start
TIMESU      =Start-up time
    - Forecast Start
TMPDIR      =Short lived directory.

```

TRDB - Boundaries Doc Extr_oro_gp Main Man Sendtm Sort
=Directory with time series output files.
 - Actions

TRUNCN =ECMWF spectral truncation.
 - Boundaries Extr_oro_gp Start

TYPE =Boundary file type (analysis
 - Boundaries Fg Forecast MakeStrategy Nextbda Sort

UNIT =Fortran unit number of file to be processed.

USER =You (or me).
 - Actions Boundaries Cleanup News Sendtm Start
 x>Actions x>Start

VECLIB =Convex vector library.

VERBOSE =List of scripts to produce extensive (diagnostic) output.
 - Start Trace x>Trace

VERIFY =List of actions from which the output is to be verified.
 - Analyse Start

VINOUT =Title of the Vineta printer output file.
 - Start Vineta Wrapup

WAITFILE =File to be waited for (local implementation).

WAITLIMIT =Maximum time (seconds) to wait for the file WAITFILE.

WAITSLEEP =Time (seconds) between two tests of WAITFILE arrival.

WD =Very temporary working directory, only for one job.
 - Analyse Anaveri Boot Boundaries Destag Fg Forecast
 Main MakeStrategy Nextbda Postpp PrepareLibs Preps
 Prog Run Sort Statis Uvstag Wrapup x>Actions
 x>Hirlam x>Main x>Postpp

WEST =West border of area.
 - Bdries Bitmap Start

WESTEC =West border of the boundary file area (in degrees).
 - Bitmap Boundaries

WRITUPTIMES =List of (space-separated) write-up times (hours, 2 digits).
 - Forecast Start

Explicitly exported variables:

```

Actions      :BATCHOUT COMPCENTRE DTG DTGBEG DTGEND ECUTC EXP
              FCINT HLUTC HL_DATA HL_LIB HL_PW HL_SCR HL_UP HL_WD HOST
              LATEST LL LOCKLIST NOW PATH PROGRESS QUEUE SETENV USER

WD           Boot:          PATH
              Compare:       BUFRTAB_DIR
              Diagnose:     HL_WD
              Doc:          HL_PW
              Hirlam:       BATCHOUT COMPCENTRE DTG DTGBEG DTGEND ECUTC EXP
                            HL_DATA HL_EXP HL_LIB HL_PW HL_SCR HL_UP HL_WD HOST NOW
                            PATH PROGRESS SETENV WD
              Main:         BATCHOUT HL_ARC HL_DATA HL_EXP HL_LEV HL_PW HL_SCR

HLUTC        WD
              MakeBUFRtabs: BUFRTAB_DIR
              Postpp:       HL_WD NAGLIB SCILIB WD
              Prepob:      BUFRTAB_DEV BUFRTAB_DIR
              Preps:        MEANOR
              Run:          BATCHOUT DTG LATEST
              Scan_ecfile: HL_PW
              Start:        BATCHOUT COMPCENTRE DTG DTGBEG DTGEND HL_DATA HL_LIB
                            HL_SCR HL_UP HL_WD HOST LATEST LOCKLIST NOW PARKLIST
                            PATH SETENV USER
              Trace:        PARENT VERBOSE

```

Variables exported through command line keywords:

```

Analyse     :COF IN OUT
Anaveri     :IN
Archive     :PARKDIR SAVEDIR
As2puregb: IN OUT
Bdries      :IN ORO OUT
Boot:       FLAGS LDLIB LIB PGM
Completed:  MODEL SETENV
Destag:     IN OUT
Environment: HL_SCR
Extr_oro_gp: DXEC DYEC OUT TRUNCN
ExtractArea: IN OUT
Fg:          IN OUT
Horint:    IN OUT
Postpp:    IN
Prepob:    IN OUT
Prog:      IN NAM NUMB NUMH
Run:       PROGRESS
Slabba:    IN OUT
Span:      CL FG OUT
Uvstag:   AND ANS FGD FGS
Vineta:   IN MODE OUT

```

_5._2. _A_p_p_e_n_d_i_x: _F_i_l_e _t_i_t_l_e _c_o_n_v_e_n_t_i_o_n_s

As a general principle, file titles in the permanent data base (CFS at ECMWF), should be the same as on the host computer. Also the directory structure should be similar. The reference system does not comply fully with this for several reasons. E.g., file titles under Unix allow extensions (in e.g. file.ext ext is the extension), whereas perhaps permanent data base titles may consist of alphanumerics only. For this reason, distinctions by extensions under Unix are replaced by distinctions by subdirectories in the permanent data base.

_5._2._1. _T_h_e _r_e_f_e_r_e_n_c_e _s_y_s_t_e_m _f_i_l_e_s

The Fortran codes have been collected in libraries, identified by four-character names, e.g. 'lsab' for the analysis code, 'util' for utilities, etc. In the following two tables, 'file' stands for such a name. The following are available for file: (The list of files was created by executing

```
Scan_ecfile /nkg/hirlam2/member/apl  
)  
  
bfrx: Extract data to be verified against from BUFR files  
bufr: Decode BUFR  
clim: Construct climatological files  
gcod: (De)code GRIB  
grdy: Gridpoint dynamics of forecast model  
grwl: Interface to GRIB/ASIMOF  
intp: Horizontal interpolation  
lsab: Analysis  
maof: Create Analysis Observation File from BUFR file  
phys: Physics routines of forecast model  
port: Non-standard routines ('unportable')  
prpo: Pre/postprocessing of fields  
span: Surface parameter pseudo-analysis  
stat: Statistics of fits to observations  
tsfs: Create Time Series files for meteograms  
util: Unix callable utilities, e.g. date/time manipulation  
vari: Coordinate transformations  
vrfy: Compare analysis or forecast to observations
```

The following titles and directories are used:

within database	on host	description
\$HL_LEV/apl/file	(see note 1)	ar-archive of nupdate program library
\$HL_LEV/arc/file	(see note 1)	ar-archive of object modules
\$HL_LEV/exe/file (not available)	\$HL_LIB/file.npl	executable
(not available)	\$HL_LIB/file.src	new nupdate program library (used by the system manager to create a new program version)
(not available)	\$HL_LIB/file.l	nupdate source text diagnostics produced by the compiler
\$HL_BCKP/apl/file (not available)	(not available)	backup of \$HL_LEV1/apl/file
\$HL_BCKP/ucs/file	\$HL_LIB/file.updin	latest reference system update
	(not available)	directory of system updates

Note 1: The utility 'ecfile.x' provides the access method to the database for files in the reference system. Files, accessed by this utility, are available on the host by the name that is echoed by ecfile.x in its last line. The proper way of accessing files of the reference system is therefore:

```
file='ecfile.x $HL_PW kind database_name | tail -1'
```

where kind can be apl, arc, exe or doc, and database_name is bfrx, bufr, etc.

At ECMWF, not all of the above files will always be available. Plain files in HL_LIB may disappear because HL_LIB is on a volatile file system. Of most programs, executables or object modules will not be archived in the database. The only 'complete' system is \$HL_LEV/apl, which constitutes the reference system. This system is backed-up in HL_BCKP (accessible by the system manager only). The nupdate correction sets used to update the system since the conversion from COS to Unicos are stored in the CFS directory \$HL_BCKP/ucs/file; the latest of these sets may be available in \$HL_LIB/file.updin.

The names of the reference scripts are chosen to be as informative as possible, with names up to 14 characters. To distinguish them as scripts, their names start with an upper case character (however, simulations of Unix commands not available under all Unix implementations must start with lower case characters). The scripts are in HL_SCR (not available in CFS).

_5._2._2. _F_i_l_e_s _t_o _d_e_f_i_n_e _a_n _e_x_p_e_r_i_m_e_n_t

The files in HL_WD define an experiment, in terms

of scripts that differ from the reference versions, and of nupdate correction sets to modify reference programs. The scripts must have the same name as the reference scripts they replace (else they will not be recognised as substitutes). The nupdate correction sets must be named file.updin where 'file' is the four-character name of the program (see previous subsection). New scripts and/or programs can be organised and named by the user the way he prefers, but implementations of new developments into the reference system will be facilitated if the user follows the conventions outlined above.

5.2.3. Date-time files produced by an experiment

Files produced by a run are in HL_DATA (at ECMWF backed-up in the CFS directory HL_EXP). The titles of the field files are ftymmmddhh[ll]. In here is

```
ft      a two-character file type
ymmmddhh the date/time of the initial data (usually
           the analysis time)
[ll]    the forecast length (two digits, hours),
       not present for analysis/initialisation products
```

The following file types ft have been defined:

```
ae  analysis errors
an  uninitialised analysis
bd  information on boundary files used by the experiment
bl  (ships) black-list history
fc  forecast
in  initialised analysis
ma  ECMWF analysis (or any other model producing boundaries)
mf  ECMWF forecast (or any other model producing boundaries)
mi  ECMWF initialised analysis (or any other model ...)
ob  observations
```

Some files require a subtype specification, e.g. to distinguish unstaggered files from the (default) staggered, or slab files from the (default) grib files. Files with a non-null subtype specification are almost always considered to be non-permanent; they are not backed-up, and they reside in the directory WD, which is removed after the completion of the cycle (unless WD and HL_DATA coincide). Subtypes defined so far are:

```
cf  (for observation files:) COF file
d  destaggered slab-file
hl (for observation files:) limited to HIRLAM area and time
s  slab-file
sy (for bd files:) strategy file
v  vertically interpolated to HIRLAM model levels
x  extracted area
```

Combinations may occur, e.g. sv is a staggered slab-file on HIRLAM model levels.

We have made the following exceptions to this rule: Boundary files and history files, used by initialisation and forecast, are called bdry<n>x and hfil<n>x resp., with the above extensions. In here, <n> is a positive integer, counting over the files. We have chosen this exception for efficiency: In this way, date/time groups need not be calculated. The disadvantage is that the file titles are not unique, different cycles use the same titles.

Beside the field files, there are some more files in HL_DATA or WD. These are files containing namelists and standard input to programs (usually named with extension .stdin), aliases like fort.1, (not on YMP), executables specific to the experiment (e.g. the forecast model, compiled for the experiment specific number of gridpoints), etc.

5.2.4. Common data files

A number of data files are common to several experiments. In this class are observations, climatological files, the first first-guess, etc. In principle, they should be stored in directories not specific to the experiment. Currently, ecfile security problems prevent that at ECMWF, so (for the time being?) they are stored in directories owned by the user; the directories are subdirectories of a common directory, HL_COMDAT, with the subdirectory name derived from the experiment to create the files (e.g. boundary data are stored in \$HL_COMDAT/\$BDDIR). The file names follow the same conventions as given in the previous subsection. Climatological files have as date/time 00mm0000 where mm is the month of validity, or 00000000 if they refer to a year's climate. The following file types are used:

```
cl  surface climatology
io  monthly ice observations
so  monthly sst observations
```

The cl files depend on the area, the observation files

are experiment independent (unless, of course, the experiment is on different observation configurations etc.). The io and so files carry yyymm0000 as date/time group and they have the extension

F Finnish data
N Norwegian data
S Swedish data

The following files are input to an analysis run:

\$AFDIR/btfcst	barotropic forecast error coefficients
\$AFDIR/rszerr	radiosonde height errors
\$AFDIR/rswerr	radiosonde wind errors
\$AFDIR/saterr	SATEM/TOVS thickness errors
\$CLDIR/fc	forecast error coefficients
\$CLDIR/cc	climate-forecast error coefficients
\$AEDIR/ae(dtg)	analysis errors for the first cycle

The files in \$CLDIR and \$AEDIR depend on the area, those in \$AFDIR are experiment independent (except, of course, if the contents of those files are the subject of the experimentation).

_5._3. _A_p_p_e_n_d_i_x: _W_h_e_r_e _t_o _f_i_n_d _t_h_e _f_i_l_e_s

5.4. Appendix: The reference scripts In this Appendix
it is described how to get to the online
documentation, and some information is given on the main
scripts.

You can extract the most recent documentation on script *script_name* by use of the utility Doc. From the C shell, do

```
setenv HL_PW your_hirlam_password  
/ec/nkg/HL2_SCR/Doc script_name
```

From the Bourne shell, you can do

```
set -k  
/ec/nkg/HL2_SCR/Doc HL_PW=your_hirlam_password script_name
```

In these, *your_hirlam_password* is your HIRLAM password, and *script_name* is Main, Run, etc.

Also, you can extract documentation with the script Whatis. E.g.,

```
Whatis Whatis
```

will give a short description of the script 'Whatis'. The advantage over the first method is that it will work for almost every script; the disadvantage, on the other hand, is that the description is rather concise.

If your experiment involves a version of a script that deviates from the default HIRLAM version, then you must copy it to a experiment-specific directory, the name of which is in the environment variable HL_WD. Then edit it, and ensure HL_WD is in your PATH before the default directory HL_SCR. You must always at least do this for the start-up script, Start.

5.4.1. Calling tree

The calling tree of the scripts is:

```

Hirlam . . . . . Enter reference system
  Actions . . . . . Take action
    Start . . . . . Prepare & start an experiment
    Main . . . . . Continue an experiment
      PrepareLibs . . . Prepare experiment libraries
      Run . . . . . Run the experiment
        MakeStrategy . . Determine which boundaries to use
        Boundaries . . Create boundaries
          Extr_oro_gp . Obtain orography of boundary model
          Bdries . . Interpolate to HIRLAM grid
            Horint . . Horizontal interpolation
            Vineta . . Vertical interpolation
              Preps . . Create climatological input
            Fg . . . . Find first-guess file to use
              ExtractArea . Extract subarea for this experiment
            Analyse . . Perform analysis
              Preps . . Create climatological input
              Span . . Surface parameter update
              AnalyseInput . Find input files for analysis
                Preps . . Create climatological input
              Prepob . . Prepare analysis observation file
                MakeBUFRtabs . Make BUFR tables
              Destag . . Destagger guess field
              Make_lsab.x . Create analysis executable
              Uvstag . . Stagger analysis
              Postpp . . Stand-alone postprocessing
                Prog . . Run forecast model
                  Postproc . Inmodel postprocessing
                Anaveri . . Verify analysis
                Statis . . Quality control monitor
              Forecast . . Perform forecast
                Nextbd . . Find next boundary file to be used
                  ExtractArea . Extract subarea for this experiment
                Prog . . Run forecast model
                  Postproc . Inmodel postprocessing
              Main . . . . . Continue!
            Lp . . . . . Send file to printer

```

The following scripts are called in several places:

Archive	Interface to archive (CFS at ECMWF)
Boot	Create executables for the experiment
Lock	Protect files
Mkdir	Make directory recursively
Sendtm	Send file; calls Ftpput, if send is by ftp
Trace	Control debug output
Traps	Define signals to be trapped

```
Wrapup      Wrapup experiment, cycle, or failed script
ecfile.x    Access HIRLAM reference system files
```

The HIRLAM system under Unix

10.1.36

The script Postproc is called by the forecast model, to further process files created by the model. Actions taken by Postproc could be to move a file to the archive, or to send it to a different computer, or to plot it, etc. It will be clear that this script will have to be depend strongly on your computer configuration.

5.4.2. _H_i_r_l_a_m

The script Hirlam is the prefered interface to the HIRLAM system. Depending on its arguments it will start, resume, stop, or monitor an experiment.

5.4.3. _S_t_a_r_t

The script Start sets up the experiment and starts it. You must copy it to your own directory (HL_WD), edit and submit it. The prefered method to submit Start is to change directory to HL_WD and type Hirlam start. E.g., for the experiment A00:

```
cd ~/A00
Hirlam start
```

5.4.4. _E_N_V_I_R_O_N_M_E_N_T

By sumbitting Start, two files are created, with titles as given in \$SETENV and \$PROGRESS, by default \$HL_WD/ENVIRONMENT, and \$HL_WD/progress.log, resp. These contain the definition and the status of the experiment. The file \$SETENV is executed by each script, and thus passes information as provided by the user to the place where it is needed. The file progress.log passes information from one cycle to the next, on the progress the experiment has made so far.

5.4.5. _M_a_i_n

The script Main performs a few small admnistrative tasks, prepares the libraries that are used throughout and are experiment specific, by calling the script PrepareLibs. Then it submits Run (see below). The main reason for Main to exist is that it may be called by Run to continue an experiment in a new batch job.

5.4.6. _R_u_n

The script Run runs the experiment, by looping

over the cycles. A cycle consists of one or more of the following:

The HIRLAM system under Unix

10.1.37

Analysis: Analysis
Forecast: Forecast

The order in which you specify these is the order in which they are executed. It is your responsibility to do something sensible.

You specify the contents of a cycle in the environment variable CYCLE. It suffices to use the first two characters, e.g.

CYCLE=An,Fo

_5._5. _A_p_p_e_n_d_i_x: _Q_u_i_c_k _i_n_s_t_r_u_c_t_i_o_n_s

The following gives a short description on how to start an experiment.

Experiments are started by submission of a script, Start. A sample script is found in the ECMWF Unicos directory /ec/nkg/HL2_SCR. The script Start needs modifications. At least you have to change the experiment name. You may want to adjust environment variables to your own experiment.

The script Start is a short job. It merely creates \$SETENV, \$PROGRESS, and submits Main using the NQS utility 'qsub'. You may run Start in one of the following ways:

```
Interactively:      Type 'Hirlam start'  
Remotely at ECMWF: Send_to_ECMWF Start  
                  (where Send_to_ECMWF is a utility to send  
                   a job to ECMWF).
```

Each of these may require changes to the beginning of Start. For Interactive or Batch submission, the lines starting with '#QSUB' may have to be changed. For remote submission at ECMWF, you have to change the first line.

After the '#QSUB' lines there are some environment variables that you have to set:

EXP

The following have reasonable defaults for ECMWF:

```
HL_WD  
TEMP  
OUTDIR  
QUEUE  
OPTION (Further options to submit Main are not needed at ECMWF)
```

The following changes may also have to be made:

Time limit -lT must be sufficient for one cycle
Environment variables must be set properly; at least:
HL_PW FFILE MEMBERSTATE

After 'Start' has run, you find the files ENVIRONMENT and progress.log in your working directory HL_WD.

Also a batch job will have been submitted.

After a first experiment has run, say with

The HIRLAM system under Unix

10.1.39

experiment code A00, then you may reuse the input files it created by setting the following environment variables in the script Start of your second experiment to the code of your first experiment. So in the example:

```
CLEXP=A00 # for climatology files  
AEEXP=A00 # for the first analysis errors  
BDEXP=A00 # for the boundary data  
OBEXP=A00 # for the observations
```

You achieve the same effect, simpler but less flexible, by:

```
PAREXP=A00
```

```
_5._6. _A_p_p_e_n_d_i_x: _P_o_r_t_i_n_g _t_h_e _H_I_R_L_A_M _r_e_f_e_r_e_n_c_e  
_s_y_s_t_e_m _t_o  
_y_o_u_r _o_w_n _s_i_t_e
```

Manual of Bitmap

Author: gerard Cats, KNMI
Fri Aug 28 11:06:58 WET 1992

Bitmap: Create a bitmap specification file for use at ECMWF

The script Bitmap creates on its standard output a file containing the bitmap specification to be used for boundary zone extraction from ECMWF archives.

The script uses a number of variables to obtain the definition of both the ECMWF grid and the HIRLAM boundary zone. Those variables are read from the file \$SETENV, or, if they are not in that file, from the environment. They are:

1. To define the HIRLAM area:

NLON
NLAT
SOUTH
WEST
NORTH
EAST
POLON (of the South pole)
POLAT (of the South pole)
NBNDRY (width of the boundary zone, in gridpoints)

2. To define the ECMWF area, assumed to be regular lat/lon:

DXEC
DYEC
WESTEC
EASTEC
SOUTHEC
NORTHEC

The resulting bitmap specification should be directly suitable to control the extraction of ECMWF gridpoints. The specified bitmap should contain at least all points for a bi-linear interpolation. Usually, even enough points for a 16-point interpolation are obtained, but round-off errors may cause non-included gridpoints to be used with an extremely small weight. If non-included gridpoints show up as very large (missing data) values, expect problems here!

If the geographical North or South pole is within the HIRLAM boundary zone, the convergence of latitude lines causes each HIRLAM gridpoint to select its own 16 surrounding gridpoints. Precautions were taken to avoid very long lines in the specification file by including intermediate gridpoints, even if they are not required for a 16-point interpolation.

The script Bitmap reports on its standard error file how many ECMWF data points were selected (both absolutely and relatively to the original number of gridpoints). It should be possible to input the GRIB file produced at ECMWF directly into the horizontal interpolation program (see script Horint), after conversion to the ASIMOF format.

Hi Erland,
Ben sent you a short documentation of Boot. However, we
have more available. I suggest you add that to the
manual after the contribution from Ben. See after my
autograph.

R G s
e ard
G R

Gerard Cats, KNMI, PO Box 201, 3730 AE De Bilt, The Netherlands.
Tel. +3130206442, Fax +3130210407, e-mail cats@knmi.nl

Subject: 10.3.7.Boot

The following gives more detailed information on Boot. Normally, the user should not have to have knowledge of this, but if he wants to develop an alternative Boot, or if he wants the HIRLAM system in a more crafty way, he may find the information provided below useful.

HIRLAM DOCUMENTATION

Manual for the utility Boot. (rough draught)

Author: Ben Wijchers Schreur, KNMI, 910321.
Updated: Gerard Cats, KNMI, 940627.
Criticisms and other useful remarks are welcome.
Full path of the original: /ec/nkg/HL2_SCR/Boot

Boot is the HIRLAM construction and maintenance programme. It creates and runs the executables requested and the libraries in "ar"chive format needed or specified by the user or it "nupdate"s them if an nupdate input file for the libraries of a more recent date exists. Programmes and libraries are large files and they therefore reside on the unrestricted, but volatile disk space of the /tmp disk at the ECMWF. They are always at risk of being deleted to create space on the /tmp disk. Boot ensures that programmes and libraries exist on the /tmp disk when they are needed and that they are upto date.

Boot has been designed to operate in the HIRLAM environment, but it can be used as a stand-alone utility if the following environment variable has been set:

HL_PW: the user's HIRLAM password, allowing access to HIRLAM programmes and libraries.

When these parameters are not set the utility aborts. These parameters are set and exported in the standard UNIX way:

C shell:

```
setenv HL_PW mysecret
```

Bourne shell:

```
HL_PW=mysecret; export HL_PW
```

or they can be put in the utilities environment on invocation from the Bourne shell:

```
set -k
Boot HL_PW=my_secret [and the rest of the command line]
```

In a similar way the system parameter SHFLAGS may be set. This parameter can be used to set the Bourne shell invocation flags for Boundaries. For example setting SHFLAGS=-kx will cause environment variables to be passed to child

processes and it will cause shell commands to be echoed to standard error output before execution (if you are a fan of long listings).

Boot uses HIRLAM system parameters in its operation. It uses defaults for these parameters when they are not set:

System directories:

```
HL_SCR:    Scripts directory. Default HL_SCR=/ec/nkg/scripts.  
HL_UP:     Path of nupdate with a / appended. Default=". ".  
HL_LIB:    Directory on /tmp containing HIRLAM archives.  
           Default=/tmp/nkg/lib.  
HL_LEVEL:  Directory of the HIRLAM system on backup storage.  
           Default=/nkg/hirlam2/member.
```

Working directories:

```
HL_WD:     Working directory for (small) permanent files, e.g. update  
           correction sets, alternative fortran files, include files.  
           Default HL_WD='pwd'  
HL_DATA:   Working directory for (large) semi-permanent files, e.g.  
           executables, libraries. Default=$HL_WD.  
WD:        Scratch directory, deleted after the job. Default=$HL_WD.
```

Hard and soft systems:

```
COMPCENTRE: Centre at which programme is used. This variable helps to  
           set up directory structures and the storage and retrieval of  
           files. Currently implemented are ECMWF and KNMI.  
           Default=ECMWF.  
HOST:      Machine on which Boot runs. Implemented are CRAY and CONVEX.  
           Default=CRAY.
```

Debug options:

```
DEBUG:     When DEBUG=on the executable is run in the local symbolic  
           debugger. Default=off.  
PMD:      When PMD=on an extensive post mortem dump is produced should  
           the executable fail in execution. Default=off.  
COST:     When COST is specified to contain a utility (e.g. COST=time)  
           the executable will be run through 'COST' (e.g. COST=time  
           will produce the 'time' diagnostics of the execution). The  
           default is ""
```

The following parameters specify the executables and libraries to be created or updated by Boot and the way this is to be done:

```
PGM:      The name of the programme to be constructed, the extension .x is  
           appended to this name. This variable may be omitted when only the  
           construction of a library is required. No default.  
LIB:      Format: LIB=lib1[@updin1][,lib2[@updin2]][,....]. A comma separated  
           list of libraries (no extension, .a is assumed) with an optional  
           @ paired update correction set. Default=$PGM. If $LIB is not  
           defined, explicitly or implicitly, Boot will abort.  
APL:      Directory of the "ar"chived nupdate programme library from which  
           the library can be constructed. Default=$HL_LIB.  
DEFAULT:  Directory to be searched for default object libraries.  
           Default=$HL_LIB.  
ALT:      Directory containing alternative fortran source files and  
           precompiled objects. These take precedence. No default, optional.  
INCLUDE:  Directory to be searched for include files. No default, optional.  
           >>> Not implemented  
MAIN:     A fortran programme defining the entry point(s) in the libraries.  
           No default, optional.  
LDLIB:    External load libraries. Enclose multiple entries in quotes so that  
           $LDLIB may be read as a single string. No default, optional.
```

Construction of the libraries.

Boot starts the library construction and/or update by looking for the update correction set specified in \$LIB. If none is specified or if it can not be read Boot looks for the default update correction set \$HL_WD/'basename \$lib'.updin. Boot then looks for the library specified in \$LIB. When the library is not readable, Boot tries to read the default library \$DEFAULT/'basename \$lib'.a. When this default library is also unreadable Boot will fetch the library from backup storage (ecfile) or failing this it will prepare to do a full update on a corresponding "ar"chived nupdate programme library. If this retrieval process leads to a readable library and no update correction has been specified, either implicitly or explicitly, Boot assumes the library to be upto date and the library construction to be completed. Otherwise Boot checks the dates of the update correction set and the library and if it finds that the update correction set is newer it updates the library by a nupdate on the archived programme library \$APL/'basename \$lib'.apl, followed by a compilation of the updated modules and alternative fortran files \$ALT/*.f and finally a substitution of corresponding objects in the library by the compiled objects and alternative object files \$ALT/*.o. If the the debug option is set to DEBUG=on the fortran source files produced by nupdate are saved in the subdirectory \$\$ (process number) of the directory from which Boot is started. Boot repeats this process of library updates for all pairs of library and update correction set specified in \$LIB. Check the Boot script for the finer details of this contorted logic.

Construction of the executable.

Boot checks whether \$PGM.x exists and is upto date with respect to the libraries specified in \$LIB. If this is not the case Boot "make"s \$PGM.x. If a the variable \$MAIN has been specified the corresponding file is taken as the main programme defining the entry point(s) in the libraries. If not then a file main.f is constructed that defines the entry point in the libraries by a call to the subroutine with name 'basename \$PGM'. The main programme is compiled and loaded together with the specified HIRLAM libraries, any external libraries specified by the variable \$LDLIB and include files specified by \$INCLUDE.

Running the executable.

After the executable \$PGM.x has been constructed it is automatically run by Boot in one of three ways specified by the debug options. If DEBUG=on the programme is run by a symbolic debugger. If PMD=on the programme is run and if it fails an extensive post mortem dump is produced. When both DEBUG and PMD are off the programme is run normally. If COST is set, the programme is executed by the utility specified by COST. Positional parameters specified to Boot are passed to \$PGM.x. The programme is connected to standard input allowing redirection to pass through Boot. Standard output of \$PGM.x is redirected to an artificial output channel, number 3.

Bourne shell examples-----:

1) Simple construction of the HIRLAM grib library, no explicit update:

```
set -k #to allow the passing of $LIB and $HL_PW to Boot
Boot LIB=port HL_PW=mysecret
```

Upon completion Boot echoes the full path of the constructed library on output channel 3.

2) Run the forecast programme after an update using the alternative update correction set \$HL_WD/altgrdy.updin:

```
Boot PGM=hlprog \
LIB=grdy@altgrdy.updin.phys,prpo,grwl,util,vari,tsfs,port,gcod \
LDLIB=$SCILIB
```

- 3) A true-to-life example of the construction and execution of the programme vineta.x that interpolates boundaries from the ECMWF model to eta-levels. vineta.x reads a namelist from standard input, which is connected to the file vineta.stdin in this example; the standard output is appended to the file with title VINOUT (taken from the script Vineta):

```
Boot PGM=vineta LIB=prpo,phys,grwl,port,gcod \
LDLIB="$SCILIB" >> $VINOUT < vineta.stdin
```

Use has been made of the variable LDLIB to define external libraries.

- 4) A crafty example showing the passing of the positional parameters \$DTG, +, and \$bdoffset to the programme mandtg.x and a way to discard all output from Boot while retaining the output from mandtg.x on channel 3 through the use of a subshell, indicated by (), allowing this output to be re-directed to standard output and subsequently to be assigned to the variable \$bddtg:

```
bddtg='( Boot PGM=mandtg LIB=util\
$DTG + $bdoffset 1>/dev/null 2>&1 ) 3>&1'
```

10.2 Analyse, Forecast, Preps, Boundaries

The core scripts in the HIRLAM reference system are

Analyse:	Perform an analysis
Forecast:	Perform a forecast
Preps:	Create climatological input files
Boundaries:	Create files with lateral boundary conditions

Of these, a description follows below.

10.2.1 Analyse

HIRLAM system script: Analyse	=====
script:	Analyse
purpose:	to perform the analysis and all necessary pre- and postprocessing.
method:	analysis input files are created by an inline call to AnalyseInput. the analysis observation file is then constructed by a call to Prepob, if the file doesn't already exist. if the purpose of the current experiment was to create observations, Analyse is ready and exits. otherwise it continues with destaggering the first-guess \$IN. analysis coefficient files are retrieved, random access files are assigned, all other input files are linked to the appropriate fortran channels and namelists are constructed, before Boot-ing the analysis. after a succesful analysis the increments are staggered and added to the first-guess to create the analysis file. the analysis is postprocessed and verified if this was specified by the user in \$PP and \$VERIFY. finally the results of the analysis are archived.
variables:	IN - first-guess filename OUT - analysis output filename COF - comprehensive observation file title
author:	Gerard Cats, KNMI

Toon Moene, KNMI
Rita Standhaft, KNMI
Ben Wijchers Schreur, KNMI

10.2.2 Forecast

HIRLAM system script: Forecast

script:

Forecast

purpose:

to prepare and control the forecast in Prog

method:

the titles of the boundary files required are constructed from the information read from standard input. normally the standard input to Forecast will come from the file bdstrategy. these boundary files are interpolated to the forecast area by the script Nextbd which is called inline. next, the namelist controlling the forecast is constructed using parameters set by the user in Start. finally the forecast model is executed by a call to the script Prog.

variables:

IN - start data file title

author:

Gerard Cats, KNMI

Toon Moene, KNMI

Rita Standhaft, KNMI

Ben Wijchers Schreur, KNMI

Manual of forecast

Author: Gerard Cats, KNMI, 910405

Updated: Gerard Cats, KNMI, 940627

Fri Apr 5 12:47:47 WET 1991

Forecast: Organise the forecast

The script Forecast organises the forecast run . Environment variables control the initial data, the forecast length, etc.

The initial data are read from the file, of which the title is given by the environment variable IN (full path).

The boundary files to be used are constructed from a list, that is to be provided on standard input. This list is conveniently made with 'MakeStrategy'. The list is spooled until the date/time in the list matches that of the start of the forecast. Therefore, the same stdin file can be used for several cycles in an experiment. From stdin the number of boundary files is inferred.

The actual construction of the boundary file title and the vertical interpolation of the file is done by the inline script Nextbd.

The input files (initial data and boundaries) must be in grib file format.

The list of write-up times (WRITUPTIMES, in hours), is used to construct the namelist variable NWTIME, in seconds, and to count the number of history files. The history files will be archived through the script Postproc, which is called from inside the

forecast model.

The forecast model itself is called through the script Prog.

Parameters passed are:

NAM: The file containing the forecast namelist namrun

IN: The file with initial data (full path)

NUMH: The number of history files produced

NUMB: The number of boundary files to be used

10.2.3 Preps

HIRLAM system script: Preps

```
=====
script:          Preps
purpose:        to create climate files for an experiment
method:         programmes from the clim library are Boot-ed to create
                constant analysis error files and monthly climate files
                interpolated to the grid of the experiment
variables:      none
author:         Gerard Cats, KNMI
=====
```

10.2.4 Boundaries

HIRLAM system script: Boundaries

```
=====
script:          Boundaries
purpose:        this script creates boundaries from MARS
method:         for each boundary file required by the current cycle
                (as specified in the stdin file, usually this stdin is
                redirected from the file bdstrategy):
                1. extract from mars
                2. either send to memberstate (to simulate operational
                   data-dissemination) or interpolate by the script
                   Bdries (to make boundaries to a run at ECMWF)
                the switch SEND, set in the Start script by the user
                determines whether to send or to interpolate.
author:         Nils Gustaffson, SMHI
                Gerard Cats, KNMI
                Ben Wicher Schreurs, KNMI
=====
```

modifications:

Kalle Eerola, FMI

1. If the boundary file exists in \$BDDIR do nothing
2. If not ECMWF get the dissemination file from ECMWF
from archive \$BDDIR
3. If not ECMWF get the local orography file
4. Save the interpolated local file to \$BDDIR

HIRLAM DOCUMENTATION

Generating Boundaries using the Hirlam system.

Author: Ben Wijchers Schreur, KNMI, 910611..

Updated: Gerard Cats, KNMI, 940627.

Criticisms and other useful remarks are welcome.

The utility Boundaries extracts lateral boundaries for HIRLAM from ECMWF mars archives. The utility has been designed to operate in the HIRLAM environment and should be started from the Hirlam start-up job Start.

Generation of user requested boundaries-----

The first step in using any utility from the Hirlam system is to copy the Hirlam start-up job. For submission of a Hirlam job from a remote machine using the Remote Queuing System the start-up job /ec/nkg/HL2_SCR/Start should be used, for submission of a Hirlam job from the Cray using the Network Queuing System the QSUB parameter heading of Start should be modified to suit the NQS format.

Secondly this copy of the start-up job must be edited by the user to reflect his particulars and requests. For the generation of boundaries using the Hirlam system at least the following parameters must be edited:

```
EXP=bdy           :three letter code identifying the experiment
HL_PW=yours456   :secret Hirlam password for access to Hirlam software
FFILE=uahwicb    :generic name for batch output file. A numeric extension
                  ranging from 01 to 99 (cyclic) is appended as an
                  identifier for batch output of different forecast cycles
                  in a single job. The resulting filename is used for the
                  -f parameter of the ECMWF sendtm utility that forwards
                  output from each forecast cycle as it is completed.
MEMBERSTATE=nt1   :memberstate identifier required by sendtm.

NLON=22          :number of longitudes in the grid
NLAT=21          :number of latitudes
NLEV=16          :number of model levels

SOUTH=-13.5      :latitude of southern boundary in shifted pole coordinates
WEST=-2.5        :longitude of western boundary in shifted pole coordinates
NORTH=-3.5       :latitude of northern boundary in shifted pole coordinates
EAST=9.0         :longitude of eastern boundary in shifted pole coordinates
POLAT=-30.0      :latitude of shifted south pole
POLON=0.0        :longitude of shifted pole

DTGBEG=90022400  :date/time group YYMMDDHH, normally the start date/time of
                  of the first forecast cycle, used here as an identifier
                  for the requested set of boundaries, it should be
                  identical to the DTG identifier in the boundary strategy
                  file (see below.)

CYCLE=           :only boundary generation is requested in this experiment
```

Take care that parameters DTGBEG, the date/time group for the beginning of the experiment, DTGEND, the date/time group for the end of the experiment, and LL, the length of the forecast, provide the information matching the use you are going to make of the generated boundary files.

The third step to be taken is the specification of the boundaries that are to be generated. The user has to create a file bdstrategy in the Hirlam working directory HL_WD (HL_WD defaults to \$HOME/\$EXP when not defined by the user) containing entries for every boundary requested. The first line specifies an identifier, format DTG\$DTG, for the requested set of boundaries, with \$DTG corresponding to \$DTGBEG defined above. The second line of the strategy file should be a mask defining the parameter types in the corresponding columns of the subsequent boundary entries. The parameters in this mask should include:

```

TYPE :type of archived file. AN for analysis, IA for initialised
      analysis, FC for forecast.
DATE :boundary date YYMMDD.
HH   :boundary time in hours GMT.
BDLL :forecast length of boundary in hours.

```

Optional parameters that can be included in the boundary strategy file are:

```

RELYMD :ecfile option for the save command. Default value
        RELYMD=365 ( -r is always automatically prepended to force
        the user to set a release date). When several options are
        specified the compound should again be enclosed in quotes.

```

In addition all parameters set by the start-up job can be redefined for individual boundaries by including them in the strategy file. Add for instance SOUTH, WEST, NORTH, EAST to redefine the area or add NLON, NLAT to change the resolution.

A typical minimum strategy file for the extraction of three different boundaries would thus contain the lines:

```

DTG90022400
TYPE DATE HH BDLL
AN 900224 00 00
IA 900224 00 00
FC 900223 18 06

```

A more esoteric example (the order of the parameters is not important):

```

DTG90022400
TYPE DATE HH BDLL NLON NLAT SOUTH WEST NORTH EAST RELYMD
AN 900224 00 00 22 21 -13.5 -2.5 -3.5 9.0 "365 u/nkg/r/-/s"
FC 900224 00 06 22 21 -13.5 -2.5 -3.5 9.0 365
IA 900224 00 00 110 100 -24.0 -23.5 25.5 31.0 365

```

It is perhaps simplest to generate the strategy file by including a cat command (catenate) in the Main part of the start-up job just before the Run command.

```

cat <<end-of-strategy >$HL_WD/bdstrategy
DTG90022400
TYPE DATE HH BDLL
AN 900224 00 00
IA 900224 00 00
FC 900223 18 06
end-of-strategy

```

Run

Automatic generation of boundaries for Hirlam forecasts-----

When the file \$HL_WD/bdstrategy has not been provided explicitly by the user, the Hirlam system will create its own strategy file using the script MakeStrategy. This script may be provided by the user and should determine which boundaries are required. A default script is contained within the Hirlam system, /ec/nkg/HL2_SCR/MakeStrategy. This script may be used as an example for the construction of other strategy generators. The default script MakeStrategy decides which boundaries are required by a particular forecast cycle, specified by the parameters:

```

DTG :the date/time group YYMMDDHH of the start of the forecast
LL  :the forecast length in hours

```

Assuming all boundaries to be of TYPE=AN, and therefore BDLL=00, it constructs the file bdstrategy accordingly. This default MakeStrategy also includes the strategies "most_recent", i.e. take an analysis when available and a forecast in other cases, and "operational", take the most recent files

available at the time of the forecast, simulating the strategy "most_recent" at a time in the past. The strategy 'available' will check the available files and choose from them with an algorithm that is suitable for operational use in member states; if it is assumed that the ECMWF mars archives are upto date, the strategy 'available' is useless at ECMWF, hence it is not implemented there; on the other hand, for operational use of the HIRLAM system, the strategy 'available' is the recommended one, because it handles a wide variety of possible problems with the data-dissemination products form ECMWF. Finally, there is the strategy 'parallel', to use the same boundaries as another experiment; specify that experiment through the environment variable PAREXP.

=====

```
script:          As2puregb
purpose:        Convert a file in ASIMOF format to pure GRIB
method:         run the program as2pur
variables:
    IN   - input ASIMOF-GRIB file
    OUT  - output pureGRIB file
    PROTOCOL - transmission protocol; default is ftp
              Used on CRAY only
    SEND_FORMAT - resulting file format; default is
                  pure GRIB, but if PROTOCOL is not ftp,
                  then default is COS-BLOCKED
                  Used on CRAY only
              On non-CRAY, the resulting file is natively blocked.
author:         Gerard Cats, KNMI
```

=====

=====

```
script:          Mail
purpose:        send mail to HIRLAM users
method:         invoke mailx with recipients as specified on command
                line. The first line of stdin to this script is used as
                subject header.
variables:
    all positional parameters are interpreted as recipients
    with the addition that 'all' means all HIRLAM users and
    'contacts' all national contact points. 'SM' stands for
    the system manager. Individuals are to be referenced by
    their e-mail addresses, obtainable from 'List_users'.
    If more positional parameters are given, they will be
    used incrementally, but if one of them is 'all' it
    will override all others.
author:         Gerard Cats, KNMI
```

=====

=====

```
script:      News
purpose:    to extract items from the news file(s)
syntax:     News [letter] [yyymmdd]
            News old
method:     In the first form ( News [letter] [yyymmdd] ):
            list all news items added on or after yyymmdd,
            in particular, News 0 lists all news items.
            Default for yyymmdd is the date you ran News
            before, unless 'letter' was specified, in which case
            default is since the last newsletter was produced.
            Persistent news will always be printed, unless 'letter'
            was set.
```

```
In the second form ( News old ):
```

```
author:     Gerard Cats, KNMI
```

```
=====
```

```
=====
script:      Scan_ecfile
purpose:    list contents of an ecfile directory
method:     The ecfile directory contents are listed with the list
            option -d for 'descendants'. If the option -r is used
            on the command line, all subdirectories are scanned
            recursively, and the string 'rec_string' is prepended
            to the directory or file name. The environment variable
            HL_PW must be set to gain access to the ecfile files
            of the HIRLAM reference system; use the option -nopw
            to access ecfile directories or files that do not have
            password protection. Ecfile list options may be requested
            by the option -o followed by an option string.
```

```
usage:      Scan_ecfile [-r rec_string] [-nopw] \
                  [-o ecfile_list_options ] directory/ies
author:     Gerard Cats, KNMI
```

```
=====
```

```
=====
script:      Man
purpose:    extract manpage(s)
method:     scan through manpage sections for requested manpages;
            extract and list them
variables:   all positional parameters are interpreted as manpage
            titles.
author:     Gerard Cats, KNMI
```

```
=====
=====
script:          Doc
purpose:         to extract hirlam documentation
method:          depending on the type of documentation proper actions are
                 taken to extract the requested documentation.
                 name of requested documentation. If no positional
                 parameters are given, Doc lists allowable documentation
                 requests. To use Doc, the environment variable $HL_PW
                 must be set to your hirlam password; and for some
                 requests, the hirlam script directory must be in your
                 path.
author:          Gerard Cats, KNMI
```

```
=====
=====
script:          Cleanup
purpose:         to cleanup traces of experiment
method:          identify the experiment name either from environment
                 variable 'EXP' or from the current working directory.
                 then remove files from $TEMP/$EXP (or $HL_DATA, should
                 that be given) and from CFS /$USER/hirlam/$EXP (or
                 $HL_EXP, should that be given).
                 in this preliminary version, by default analysis files
                 are removed; this is necessary if an experiment is to be
                 rerun the analysis, because normally a completed analysis
                 will be skipped when rerunning the experiment.
                 other files can be removed by specifying the initial
                 characters of their names. files will only be removed
                 if at least one digit follows the initial characters
                 that you specify, but fc and cc will be removed if you
                 specify 'cl' (with 'cl' all climate files will disappear)
                 By specifying 'ALL' somewhere on the command line all
                 files and directories except the main directory HL_WD
                 will be removed. This option takes full effect only if
                 you apply it twice, with a separation of at least one
                 day. In the meantime you can rescue files from CFS by
                 calling ecf file with the function 'rescue'.
variables:
           all positional parameters are interpreted as file types
           to be removed. Default is 'an'
           useful are:
             ae    remove analysis errors
             an    remove analysis
             bd    remove boundary strategy files
             bl    remove black lists
             bo    remove boundaries (same as 'ma mi mf')
             cl    remove all files produced by 'Preps'
             fc    remove forecasts
             in    remove initialised fields
             ma    remove boundaries from ECMWF analyses
             mf    remove boundaries from ECMWF forecasts
             mi    remove boundaries from ECMWF initialised fields
```

```
ob      remove observations
ts      remove time-series ('meteograms')
ALL    remove all and also all directories except HL_WD
```

author:
 Gerard Cats, KNMI

```
=====
```

script:
 Add_sstice_obs
purpose:
 to append new SST and ice observations to a database
method:
 this script runs only at ECMWF. the database file is
 extracted from the archives by ecfile. the observations
 are read from standard input, most conveniently
 redirected from a data file, and appended to the
 database file. this file is then written back to the
 archive by ecfile.
variables:
 one positional parameter is used:
 \$1 - basename of the database file, in the format:
 [is]oYYMM0000[FNS], where i indicates ice, s SST
 observations and F Finland, N Norway, S Sweden.
 YY is the year, MM the month of observations.
author:
 Gerard Cats, KNMI

```
=====
```

script:
 List_users
purpose:
 to list registered HIRLAM users and their addresses
method:
 if no positional parameters are given, produce the full
 list. else select only those lines from the list that
 match the strings provided as parameters.
variables:
 all positional parameters are interpreted as search
 strings
author:
 this script was automatically generated by "Allusers"
 on Mon Jun 27 13:09:16 GMT 1994
 "Allusers" was written by Gerard Cats

```
=====
```

Manual of Run

Author: gerard Cats, KNMI, 920403
Updated: Gerard Cats, KNMI, 940627

Run: Run the experiment

Run is the script which actually runs the experiment, as defined

by the environment variables set by the calling script (usually Main), or as set in the file of which the title is in the environment variable SETENV (usually \$HL_WD/ENVIRONMENT).

After checking the status of the boundary strategy file, Run executes a loop over the cycles to be executed within one job.

The first task within the loop is to create the the boundary strategy file for this cycle, if needed. Then the boundary files, as specified in this boundary strategy file are created by a call to the script Boundaries.

Run uses the environment varialbe LATEST to define the file that was the last one created in the cycle. So LATEST will point to the analysis after Analyse, etc. Initially, LATEST must be set to the file to start the cycle from. This can be done by explicitly providing it, but the normal action is to let the script Fg determine it. So if LATEST was not set, Fg will be called to write the appropriate title to the file called LATEST, which then is read by Run to fill the variable with the appropriate file title.

The normal way to set LATEST and the start date/time of the experiment is through specification in the file of which the title is in the environment variable PROGRESS (default is \$HL_WD/progress.log).

Run performs the cycle(s) by scanning the environment variable CYCLE. It executes analysis and/or forecast construction and observation creation in the order specified.

After each cycle, Run cleans up the results by executing 'Wrapup cycle'. Then the date/time is incremented by executing mandtg.x. The new date/time, and the new file title LATEST are written to the file PROGRESS, so as to be available for continuation of the experiment.

If the end of the experiment is not reached, Run will either do another cycle or submit Main to initiate a new nqs/qsub cycle. By default, the latter is done if the experiment runs in batch mode, but the default can be overridden to specify 'MORE_CYCLES=yes', in which case the next cycle will be run within the same instance of Run. Interactive runs will do all the cycles within one single execution of Run. If a new job is submitted, the log of started HIRLAM jobs is updated; if a new cycle is run from within the same instance of Run, the files \$SETENV and \$PROGRESS are re-executed inline, so that the environment in which Run executes is adjusted for the new cycle.

If the completed cycle was the last one within this experiment, Run will clean up by executing 'Wrapup experiment'.

```
=====
script:          PrepareLibs
purpose:        to construct libraries that are general to the HIRLAM
method:         nupdate corrections sets, including all the necessary
                area and resolution dependent modifications, for the
                general HIRLAM libraries are included in this script.
```