MIRART

Weighting Function Implementation

Franz Schreier

DLR — Remote Sensing Technology Institute
Oberpfaffenhofen, D–82234 Weßling

Slide 1

1. Overview on MIRART
2. Automatic Differentiation
3. Problems and Solutions

MIRART

Modular InfraRed Atmospheric Radiative Transfer

Slide 2

Line shapes: Voigt, VanVleck⊗Doppler, Lorentz

Line data: HiTran, HiTemp, Geisa, Jpl, ...

Continua: H_2O (CKD), CO_2 (Ridgway/GenLn2), N_2, O_2, "dry air" (Liebe)

Geometries: Limb, uplooking, downlooking (refraction optional)

Instruments:
- — line shape: FTS (optionally apodized), Heterodyne, Fabry–Perot
- — field-of-view: Box, Gauss, Trapez

Verification: Amil2da, Irtmw01, Piramhyd, ...

Weighting functions: Automatic differentation (molecular concentration, temperature)

Numerics — LbL: Voigt=$\Re$(cerf), wavenumber grid adjusted to level (p, T) and molecule

Numerics — path integrals: numeric quadrature schemes

Implementation: Fortran 77 with BLAS and SLATEC libraries

Interfaces: "classical" input file, Python scripts, web interface (`http://vl.nz.dlr.de`)

Weighting Functions

Weighting functions — Jacobians — derivatives wrt. gas densities/VMR, temperature, ...
required for retrieval applications (nonlinear least squares) and sensitivity analysis

Approaches

Slide 3

- Finite difference approximations:
  - time consuming
  - step size selection:
    (cancellation errors for small steps and truncation errors for large steps)
- Hand coded analytical derivatives:
  - boring and error prone,
  - no "automatic update" of derivative code if forward model is improved
- Symbolic differentation:
  - impracticable: no lbl ir rad transfer code in MATHEMATICA, MAPLE, ...
  - generally leads to huge expressions
- Automatic differentation

Weighting Functions

Automatic Differentation

Slide 4

- Even large codes are essentially formulated in terms of elementary mathematical operations (sums, products, powers) and elementary functions
- (In contrast to integration) differentiation is based on simple recipes such as the chain rule
- Differentiation rules can be performed automatically by some kind of precompiler

⋆ Automatic differentation tools available for Fortran, C, ...

⋆ Automatic differentation generates exact derivatives

→

ADIFOR

Automatic Differentation

ADIFOR

Slide 5

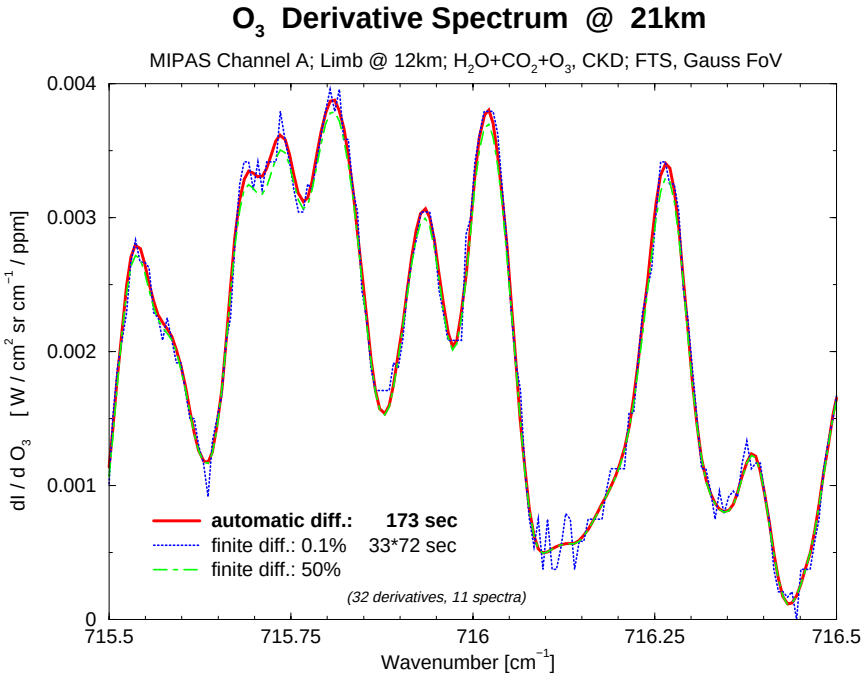
1. Top–level subroutine,
   independent ('x') and dependent ('y') variable(s),
   max. number of independent variables
2. Composition file (relevant routines and "main")
3. Generate derivative code
4. Adjust routine calling the top–level subroutine,
   initialize seed matrix
5. Compile and link (incl. ADIFOR libraries)

```
AD_TOP        = schwarzschild_ils_fov
AD_IVARS      = Density
AD_DVARS      = Radiance
AD_PMAX       = 50
```

```
dummy_main.f
schwarzschild_ils_fov.f
schwarzschild.f
convolve_ils.f
convolve_fov.f
..............
```

```
..............
y = a*sin(b*x+c)
g_y = a*b*cos(b*x+c)
..............
```

Slide 6



Automatic Differentation

Problems & Solutions

- ADIFOR significantly increases memory requirements:
  For each (intermediate or final) variable a gradient variable is generated:

  `REAL SCALAR` $\longrightarrow$ `REAL g_SCALAR(g_p)`

  `REAL VECTOR(N)` $\longrightarrow$ `REAL g_VECTOR(g_p,N)`

Slide 7

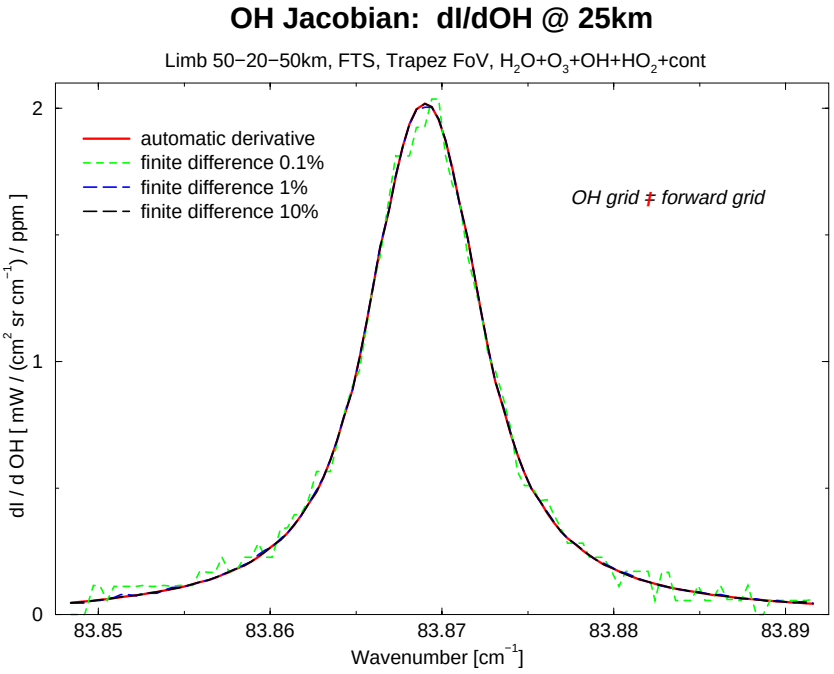
  ▷ compact storage scheme for cross sections

  ▷ individual executables depending on task:

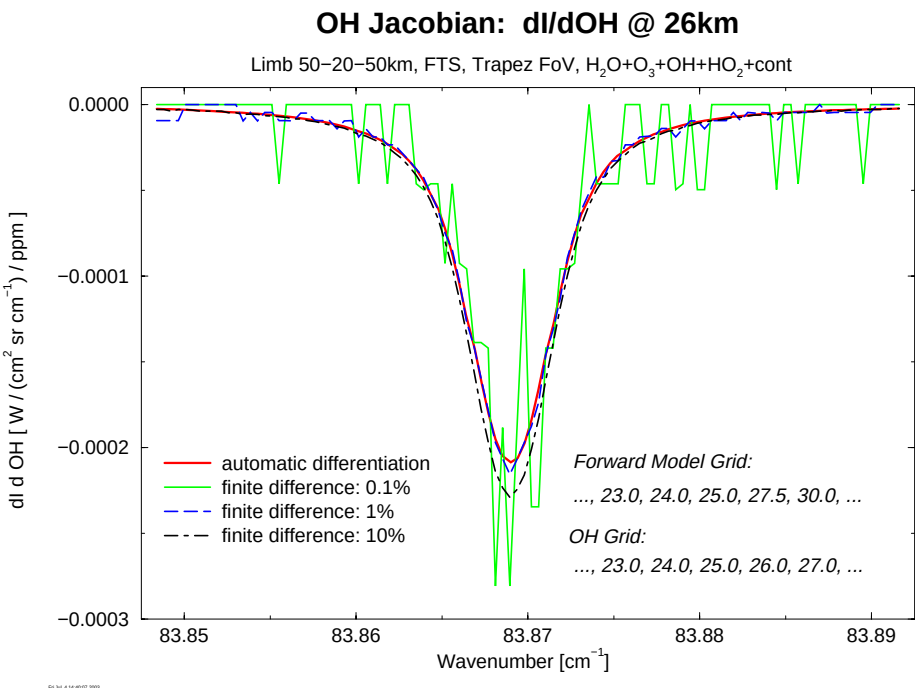
`SQUIRRL`	—	pure forward code	
`ColDRAT`	—	column derivatives	(`g_p=MxGas`)
`MADRAT`	—	molecular profile derivatives	(`g_p=MxLevel`)
`FATRAT`	—	molecular or temperature profile derivatives	(`g_p=MxLevel`)

- Retrieval grid $\neq$ forward grid
  ▷ interpolate to forward grid "inside derivative code"
- Temperature derivatives: $n = p/kT$ violated in old implementation

Slide 8



Slide 9


OH Jacobian: dI/dOH @ 26km
Limb 50−20−50km, FTS, Trapez FoV, H2O+O3+OH+HO2+cont
dI d OH [W / (cm2 sr cm−1) / ppm]
0.0000
−0.0001
−0.0002
−0.0003
83.85
83.86
83.87
83.88
83.89
Wavenumber [cm−1]
automatic differentiation
finite difference: 0.1%
finite difference: 1%
finite difference: 10%
Forward Model Grid:
..., 23.0, 24.0, 25.0, 27.5, 30.0, ...
OH Grid:
..., 23.0, 24.0, 25.0, 26.0, 27.0, ...