

The N&W codes

These codes can be used to simulate the one dimensional radiative transfer of polarized radiation through a medium permeated by a magnetic field. We base these simulations on the formalisms by Deguchi & Watson 1990 (D&W90), and apply them for three different numerical solutions to the D&W90 formalism

- In the N&W90 method, the maser polarization model of D&W90 is applied for one frequency. Circular polarization can not be computed. Linear polarization can be computed in both the Stokes-U and -Q parameters.

- In the N&W92 method, the maser polarization model of D&W90 is applied under the limiting condition that the magnetic precession rate is way bigger than the rate of stimulated emission. Therefore, the Stokes-U component of the radiation can be ignored, and only diagonal elements of the density-population matrices need to be regarded. It is in this respect that this variant of the D&W90 models similar to the G&F95 models. Circular polarization can be computed. Multiple hyperfine sub-transition can be included.

- In the N&W94 method, we find the most extensive variant of the D&W90 models. In the N&W94 models, one accounts for off-diagonal elements in the population-densities, the Stokes-U component of the radiation field, and multiple frequency bins along the maser-line. Anisotropic pumping can be introduced and multiple hyperfine sub-transitions cannot be included.

	N&W 90	N&W 92	N&W 94
Linear polarization	yes	only Stokes Q	yes
Circular polarization	no	yes	yes
limiting condition	no	$R \ll g \Omega$	no
hyperfine subtransitions	no	yes	yes
anisotropic pumping	yes	no	yes
incident polarized seed radiation	yes	no	yes

This document contains information on the compilation, input files and output of the N&W codes. In the last pages of this document, a general description is given of the architecture of the program and a rough description is given of the different subroutines. For questions or suggestions, please contact lankhaar@chalmers.se .

Compilation

The code can be compiled using a standard FORTRAN90 compiler like gfortran or ifort. We make use of the matrix inversion subroutines 'dgrf' and 'dgetri' from the LAPACK libraries. Also, one can invoke the OpenMP looping in the N&W94 models, but this is not strictly necessary. Inside each N&W map, there are the compile.sh scripts that contain the compilation commands.

For more information on or to download LAPACK: <http://www.netlib.org/lapack/>

The OpenMP libraries can be downloaded here: <http://openmp.org/>

Code input

After having compiled the programs, it is time to simulate the first maser polarization propagation. Apart from the executable, for a successful run, one should define the input of the maser in the file 'in_watned.dat', and keep the file in the same map as the executable. Input files for the three N&W models are slightly different. See the next three pages for a detailed description.

Inside the maps, there is also a map 'input_files' included, where input files for some known maser transitions have been pre-prepared.

Example input N&W90

input file Si0 1-0

1 0 0.75d0 0.75d0 3.011E-6 10000.d0 0.02d0 5.d0 5.d0 0.01d0 0.01d0

43.42376E9

1000.d0 15.d0

0.1 5E9 144499 1.d0

0 0.d0 0.d0 1.d0

1 0.1d0

File description

(empty line)

j1 j2 g1 g2 Aij p1 δ Γ 1 Γ 2 ϵ 1 ϵ 2

(empty line)

v

B θ

Tb ds maxstep $\Delta\Omega$

anis a1 a2 a3

inci inc_deg

j1: angular momentum upper state

j2: angular momentum lower state

g1: g-factor upper level (s-1 / mG)

g2: g-factor lower level (s-1 / mG)

Aij: Einstein A coefficient transition (s-1)

p1: Pumping number

δ : Pumping efficiency

Γ 1: Decay rate upper level

Γ 2: Decay rate lower level

ϵ 1: anisotropy degree upper level

ϵ 2: anisotropy degree lower level

v: frequency transition

B: magnetic field strength

θ : magnetic field angle with propagation

Tb: Temperature seed radiation

ds: propagation step (cm)

maxstep: maximum number of steps before termination

$\Delta\Omega$: Solid angle (only use 1sr)

anis: Anisotropic pumping switch. If 1, program includes anisotropic pumping

a1,a2,a3: anisotropic pumping direction (x,y,z)

inci: Polarized incident radiation switch. If 1, program includes pol. inci. radiation

inc_deg: inc_deg = U/l — degree of polarization in the inci. radiation

Example input N&W92

input file water 22 GHz
3

5 4 -33375.709d0 -0.79d0 -1.34d0 1.798E-9 2E8 0.02d0 1.d0 1.d0
6 5 0.d0 3.71d0 4.12d0 1.806E-9 2E8 0.02d0 1.d0 1.d0
7 6 43017.581d0 6.51d0 7.24d0 1.860E-9 2E8 0.02d0 1.d0 1.d0

22.2350798E9
50.d0 45.d0
60 0.6d0 2.d0 18.01528d0
0.1d0 5E9 122499
1.d0 13.d0
1

File description

nF

(empty line)

f11	f12	dv1	g11	g12	A1ij	p11	δ_1	Γ_{11}	Γ_{12}
f21	f22	dv2	g21	g22	A2ij	p21	δ_2	Γ_{21}	Γ_{22}
f31	f32	dv3	g31	g32	A3ij	p31	δ_3	Γ_{31}	Γ_{32}

(empty line)

v

B θ

nw vth dv mass

Tb ds maxstep

$\Delta\Omega$ maxTb

out

nF: number of hyperfine transitions

fij: angular momentum state ij ($j = 1/2$ upper/lower, $i =$ hyperfine transition)

gij: g-factor ($s-1 / mG$)

Aij: Einstein A coefficient transition ($s-1$)

pi1: Pumping number

δ_i : Pumping efficiency

Γ_{i1} : Decay rate upper level

Γ_{i2} : Decay rate lower level

v: frequency transition

B: magnetic field strength

θ : magnetic field angle with propagation

nw: number of channels (-nw:nw, so total number of channels is $2*nw+1$)

vth: kinetic temperature maser

dv: $2*dv =$ width of considered velocity/frequency region

mass: molecule mass in Dalton

Tb: Temperature seed radiation

ds: propagation step (cm)

maxstep: maximum number of steps before termination

$\Delta\Omega$: Solid angle (only use 1sr)

maxTb: stop simulation at luminosity of $10^{\{maxTb\}}$

out: if out = 1, then print full Stokes profiles. Otherwise only output in ned_props.dat

N.B. nF has to correspond with the number of lines describing the hyperfine subtransitions!!!

Example input N&W94

```
input file Si0 1-0
1

1 0 0.d0 0.75d0 0.75d0 3.011E-6 2E7 0.02d0 5.d0 5.d0 0.02d0 0.02d0

43.42376E9
100.d0 0.d0
40 44.0849d0 0.1d0
5E9 1500 1.d0 15.d0
0 1.d0 0.d0 0.d0
0 0.5d0 0.d0 0.d0 0.d0
0 1.5d0 4.d0
1
```

File description

```
nF
(empty line)
f11 f12 dv1 g11 g12 A1ij      p11 δ1  Γ11 Γ12 ε11 ε12
(empty line)
v
B    θ
nw  mass    Tb
ds  maxstep ΔΩ maxTb
anis a1 a2 a3
inci inc_deg
channel vth dv
out
```

nF: number of hyperfine transitions (has to correspond with the #lines describing subtransitions!!!)

fij: angular momentum state ij (j = 1/2 upper/lower , i = hyperfine transition)

gij: g-factor (s-1 / mG)

Aij: Einstein A coefficient transition (s-1)

pi1: Pumping number

δi: Pumping efficiency

Γi1: Decay rate upper level

Γi2: Decay rate lower level

ε1: anisotropy degree upper level

ε2: anisotropy degree lower level

v: frequency transition

B: magnetic field strength

θ: magnetic field angle with propagation

nw: number of channels (-nw:nw, so total number of channels is 2*nw+1)

mass: molecule mass in Dalton

Tb: Temperature seed radiation

ds: propagation step (cm)

maxstep: maximum number of steps before termination

ΔΩ: Solid angle (only use 1sr)

anis: Anisotropic pumping switch. If 1, program includes anisotropic pumping

a1,a2,a3: anisotropic pumping direction (x,y,z)

inci: Polarized incident radiation switch. If 1, program includes pol. inci. radiation

inc_deg: inc_deg = U/l — degree of polarization in the inci. radiation

channel: if channel = 1, choose the manual channel parameters:

vth: kinetic temperature maser

dv: 2*dv = width of considered velocity/frequency region

maxTb: stop simulation at luminosity of 10^{maxTb}

out: if out = 1 , then print full Stokes profiles. Otherwise only output in ned_props.dat

Running the code

Having compiled the code and written the input file with information on the maser region, we set out to run a maser polarization simulation. Compilation yields the executables:

```
watned_90.x  
watned_92.x  
watned_94.x  
which are run simply by  
./watned_90.x  
./watned_92.x  
./watned_94.x
```

N.B. the file 'in_watned.dat' has to be present and rightly defined (see above) in the map of the executable.

Code output

The output of the code is neatly summarized in the file 'ned_props.dat'. Because of the different nature of each of the N&W variants, the output is written to ned_props.dat slightly differently:

N&W90: three columns:

R pQ pa

R: rate of stimulated emission

pQ: linear polarization fraction: $pQ = (Q^2 + U^2)/I$

pa: polarization angle: $pa = \text{atan}(U/Q)$

N&W92: five columns:

Tb $\Delta\Omega$ R vL pQ pV

Tb $\Delta\Omega$: maser luminosity

R: rate of stimulated emission

vL: full width at half maximum

pQ: linear polarization fraction: $pQ = (Q^2 + U^2)/I$

pV: circular polarization fraction: $pV = (V_{\text{max}} - V_{\text{min}}) / I$

N&W94: six columns:

Tb $\Delta\Omega$ R vL pQ pa pV

Tb $\Delta\Omega$: maser luminosity

R: rate of stimulated emission

vL: full width at half maximum

pQ: linear polarization fraction: $pQ = (Q^2 + U^2)/I$

pa: polarization angle: $pa = \text{atan}(U/Q)$

pV: circular polarization fraction: $pV = (V_{\text{max}} - V_{\text{min}}) / I$

In both N&W92 and N&W94, multiple velocity channels are included in the simulations. Thus, also all of the Stokes parameters can be outputted for all their velocity-channels. Analyzing the Stokes parameters in this way, will give a better insight in the polarization characteristics, especially for the complicated spectra resulting from transitions with multiple hyperfine components. One needs to invoke Stokes-output in the input-file (see Code input). If invoked, output will be given in 'I.dat', 'Q.dat', 'U.dat' (only in N&W94) and 'V.dat'. The columns correspond to the different velocity-channels (in total -nw:nw channels) and rows correspond to propagation steps.

Architecture of the code

In the following, I give a rough description of the N&W94 simulation program. Similar, but simplified, architecture makes up the N&W90 and N&W92 programs.

1. Initialization

input file is read in. From the input file, we initialize a radiation field a (velocity dependent) pumping matrix and all the relevant information on the molecular transitions is divided over different arrays.

relevant subroutines:

`initiate` —input consists of the information from the input files. In this subroutine, the transition energies are computed, the anisotropic pumping matrix is built (`anis_pump_nw` and `twovec_euler`) and the radiation field is initiated from as a black-body radiation field.

`gen_delta` —the delta operators (Eqs. (9)) are computed for the relevant transitions. To compute the Clebsch-Gordan coefficient, we use the NED subroutine.

2. Propagation

The propagation is a two-step process. First, the density-matrix is computed from the incoming radiation field (Eq. 7). With the newly acquired density-matrix, one computes the propagation coefficients (Eqs. 14). With the propagation coefficients, the incoming radiation field is propagated (Eq. 13), and inserted into the next propagation step. Before going to the next propagation step, the radiation field will be read out for its relevant information which will be written to the output files.

relevant subroutines:

`int_stokes` —The stokes parameters are inputted in this subroutine. The numerical integration of Eq. (11) is fulfilled for a range of γ_{ab} -functions.

`stateq` —The integrated stokes parameters, along with the delta operators and pumping matrix are inputted. The subroutine will set up the matrix M from Eq. (35) (using a lot of subroutines that compute particular matrix-elements) and consequently invert it, using `dgetrf` and `dgetri` routines from the LAPACK libraries. The matrix multiplication with the vectorized lambda-matrix yields the density-matrix.

`int_densities`—The density-matrix elements are integrated over the relevant gamma-functions.

`trans_el` —Inputting the integrated densities, we construct the propagation-coefficients

`prop_stokes`—inputting the propagation coefficients, we propagate the radiation field using a matrix-exponential.

3. Finalization

When the program recognizes that $\log(R/g\Omega) > 3.1$ the propagation will be stopped.