

# CYTRAN 2.1

W.A. Houlberg

August 16, 2004

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Public Routines</b>	<b>7</b>
<b>3</b>	<b>Private Routines</b>	<b>10</b>
<b>4</b>	<b>Files and Test Cases</b>	<b>11</b>
<b>5</b>	<b>Revision History</b>	<b>12</b>
<b>6</b>	<b>Limitations</b>	<b>12</b>
<b>7</b>	<b>Future Extensions</b>	<b>12</b>
<b>8</b>	<b>Code Variables</b>	<b>13</b>
<b>9</b>	<b>Code Diagrams</b>	<b>14</b>

# 1 Introduction

The **CYTRAN** module was developed in the early 1980's by S. Tamor as a fast approximation to the cyclotron radiation transport problem in tokamak plasmas, suitable for inclusion in time-dependent transport codes. In principle, it can also be applied to non-axisymmetric plasmas because it approximates the plasma geometry by a series of annular shells with an average magnetic field strength, density and temperature. However, the strong anisotropy of the cyclotron emission and reabsorption coupled with the toroidal gradients in the magnetic field could make the radiation transport stronger in non-axisymmetric plasmas, although this problem apparently has never been studied. The present version is written in Fortran 90/95, using features that are relatively easy to adapt to different Fortran compilers and to link to other languages.

Following is a description of the **CYTRAN** module, its availability, and its F90/95 features. The module is actually named **CYTRAN\_MOD** and the primary routine in the module is **CYTRAN**. However, in this documentation the module will be referred to by its root name **CYTRAN**. In subsequent sections we describe the routines in the module. Also included are sections on revision history, limitations and known problems, future extensions, and simplified code diagrams.

In this documentation **teletype** style identifies names of Fortran modules, routines, variables and other coding.

## Description of CYTRAN

The **CYTRAN** module calculates the radial profile of net cyclotron radiation loss/heating for toroidal plasmas. Because cychotron emission increases strongly with temperature, it is highly peaked in the plasma center. However, because of the relatively high reflectivity of walls a significant amount of reflected radiation is reabsorbed near the plasma boundary. This can lead to net heating in the outer plasma. Thus, in addition to the net loss of energy from the plasma, there is a strong element of non-local energy transport. The **CYTRAN** code was originally developed by Tamor [1, 2], and applied to system studies by Baxter and Tamor [3, 4]. It was benchmarked for standard aspect ratio elongated tokamak reactor plasmas against a more sophisticated treatment of the problem, **SNECTR**, that was also developed by Tamor [5]. In the late 1980's there was another set of benchmark calculations performed by Tamor for high field ignition machines, but only minimal documentation of the effort is available (copy of poster and short unpublished document with the same title) [6]. Although there appear to be no extant versions of **SNECTR**, the **CYTRAN** routines have survived in some predictive transport codes. This F90/95 module was generated from the original version in the **WHIST** code and recently benchmarked against the EC code for ITER parameters by Albajar, Bornatici and Engelmann [7]. In this documentation we describe the I/O from the module and give only a rough outline of the physics calculations; the reader interested in a more detailed physics description of the cychotron radiation transport problem is urged to consult the references.

## List of Public Routines

- **CYTRAN**

## Availability

This module is available with:

- Stand-alone test driver
- Test cases used in the development
- External documentation
- IDL graphic procedures for viewing results of the test cases

When all documentation and testing is complete enough for release, these should be obtained through the Module Library:

<http://w3.pppl.gov/rib/repositories/NTCC/catalog/>

For a beta release version, comments, suggestions, or other information, contact:

Address: Wayne A. Houlberg  
Fusion Energy Division  
Oak Ridge National Laboratory  
P.O. Box 2008  
Oak Ridge, TN 37831-6169

Phone: 865-574-1350  
Fax: 865-576-7926  
e-mail: [houlbergwa@ornl.gov](mailto:houlbergwa@ornl.gov)

## Fortran 90/95 Features

This module uses Fortran 90/95 features that enhance portability, flexibility and efficiency. For the most part we have followed the recommendations in the document 'European Standards for Writing and Documenting Exchangeable Fortran 90 Code,' which is available on the Web at:

[http://www.met-office.gov.uk/research/nwp/numerical/fortran90/f90\\_standards](http://www.met-office.gov.uk/research/nwp/numerical/fortran90/f90_standards)

These are accomodated by:

- Use of **KIND** to declare the precision of all **REAL** variables:
  - Replaces **REAL\*4** and **REAL\*8**, which are deprecated in Fortran 90/95

- Set in a companion module, **SPEC\_KIND\_MOD**
- There are no **COMMON** blocks or any other coding features that are deprecated in Fortran 90/95
- **PRIVATE** features are used to minimize conflict with other parts of linked coding:
  - Data
  - Computational methods and numerical procedures
- Generic names are used for all intrinsic functions
- Use of optional I/O in the arguments to routines:
  - Reduces allocation and CPU for information that is not necessary for a specific application
  - Lets the routine use default assumptions
  - Allows extension of the application to add other optional I/O without changing existing calling routines that do not need the new information
  - For illustration, consider the hypothetical case of **SUBROUTINE GETINFO(a,b,c,X,Y,Z)** with required I/O **a, b, c**, and **OPTIONAL** I/O **X, Y, Z**. It can be called in various ways from another Fortran 90/95 routine:

```
CALL GETINFO(a1,b1,c1,x1,y1,z1)
CALL GETINFO(a1,b1,c1,Y=y1,X=x1,Z=z1)
CALL GETINFO(a1,b1,c1,Z=z1,X=x1,Y=y1)
CALL GETINFO(a1,b1,c1)
```

where we have appended a 1 to the variables in the calling routine and used upper case for the optional variable names in the called routine to clarify the use of optional arguments as keywords (this is only a stylistic choice, there is no case sensitivity in Fortran). The first three cases are entirely equivalent, while the last one does not communicate any of the optional I/O. These calls obey the rules that all required I/O appears first in the list of arguments and must be called in sequence. Optional variables can be called in sequence without using **GETINFO** names as keywords, but the **OPTIONAL** arguments may be accessed in any order if keywords are used.

- Dynamic and automatic allocation of local variables are used to reduce storage:
  - Dynamic allocation is invoked by an **ALLOCATE** statement that uses input information (variable that sets the number of elements to be included in array operations, or a check of the **SIZE** of an array when all elements are to be operated on), and always deallocated if it is not necessary to store the information for future calls
  - Automatic allocation is accommodated in the declarations at the beginning of a routine using input variables to set the dimension of arrays
- Use of free format
- Array syntax is used to simplify coding by removing many **DO** loops and to allow the compiler to generate more optimum coding, e.g.,
  - Initialization of a fourth rank array to zero with the precision **PARAMETER rspec** set in **SPEC\_KIND\_MOD**:

```
f = 0.0_rspec
f = 0.0
f(:, :, :, :) = 0
```

These case illustrate that the numeric values on the right are automatically converted to the appropriate **KIND** associated with the variable on the left.

- Initialization of all fourth rank elements to unity with the first three indices fixed at  $i = 1$ ,  $j = 3$ ,  $k = 5$ :

```
f(1,3,5,:) = 1.0_rspec
f(1,3,5,:) = 1.0
f(1,3,5,:) = 1
```

- A volume integral of the product of density and temperature over plasma zones 13–20 for plasma species 2 can be written as:

```
i1 = 13
i2 = 20
energy = SUM(den(2,i1:i2)*temp(2,i1:i2)*vol(i1:i2))
```

where the first rank of the density and temperature arrays indicates species and the second rank indicates radial grid

## Coding Style

The coding style in this and other related Fortran 90/95 transport simulation modules developed at ORNL is followed fairly strictly, so that once a few basic rules are learned, the code is much easier to decipher, modify and debug (for other users as well as the original developers). Among the major style features are:

- All variables are declared, including the **INTENT** of all I/O arguments to each routine
- All arguments to a routine are described in a comment block at the beginning of the routine, including their units enclosed in square brackets, [units]
- There is no I/O to external devices in general physics modules; all I/O with external resources is handled by routines that are designed specifically for that purpose
- Error handling uses an error flag (**iflag** < 0 for warnings, > 0 for errors, and = 0 for no problems) and message (**message**) system with no termination while inside the module
- Although Fortran is not case sensitive, upper case is used to emphasize all reserved words and procedures, i.e.,
  - System calls
  - Internal Fortran functions
  - **FUNCTION** and **SUBROUTINE** names
  - Names of optional arguments

otherwise all variables are in lower case

- There is extensive use of naming conventions to improve readability and facilitate debugging:
  - Routines in a module begin with the module root name (i.e., minus the **\_MOD** descriptor)
  - Underscores in a name generally indicate some characteristic of the variable that is often designated as a subscript or superscript in mathematical notation, description of the variable describing the rank of an array, etc.
  - Options begin with **k\_**

- Logical variables begin with **L**
- Comments, indentation, and blank lines are used to identify blocks of related coding
- There is only one exit to each routine, with any early exits sent to a line:

```
9999 continue
```

followed by any relevant deallocations or other closing activities before exiting

- All **DO** and **END** statements are identified, i.e.:

```
MODULE Y
CONTAINS
SUBROUTINE X
...
DO i=1,n !Over radial nodes
...
ENDDO !Over radial nodes
END SUBROUTINE X
END MODULE Y
```

- Indentation to identify code structure:

- **DO** loop constructions
- **IF** constructions

## 2 Public Routines

### CYTRAN

CYTRAN( ree, reo, roo, roe, nr\_r, bmod\_r, den\_r, te\_r, area\_rm, dvol\_r, psync\_r, K\_CYT\_RES)

#### General Description

**CYTRAN** calculates the radial profile of the net power lost or absorbed by the emission and reabsorption of cyclotron (synchrotron) radiation,  $P_{\text{sync}}(\rho)$ , by approximating the transport (emission, absorption and reflection) as a 1-D problem. See the references for a deeper discussion of the physical model and computational techniques. An optional input value has been added to allow the user to increase the number of frequency steps in the emission/absorption problem, because the original value of 100 typically gave significant glitches in the radial power profile.

#### Mathematical Description

Let  $f_{\text{abs}}$  designate the wall absorption coefficient for cyclotron radiation. In addition to this, **CYTRAN** also requires input information about the conversion between extraordinary (e) and ordinary (o) waves at the wall. Let  $f_{\alpha\beta}$  designate the net coefficient assigned to the reflection of e or o mode waves relative to the incident e or o mode intensity, where  $\alpha$  designates the reflected form and  $\beta$  designates the incident form. If we let  $f_{\text{self}}$  represent the self-reflection (of o to o and e to e), then the set of four reflection coefficients needed by **CYTRAN** are:

$$\begin{aligned} R_{ee} = R_{oo} &= f_{\text{self}}(1 - f_{\text{abs}}) \\ R_{oe} = R_{eo} &= (1 - f_{\text{self}})(1 - f_{\text{abs}}) \end{aligned} \tag{1}$$

Typical values are  $f_{\text{self}} = 0.95$  and  $f_{\text{abs}} = 0.7 - 0.95$ .

The plasma is approximated as a series of  $n_\rho$  nested radial elements, labelled by  $i$ , each with a characteristic volume,  $\Delta V_i$ , with inner and outer surface areas represented by  $A_i$  and  $A_{i+1}$ , respectively. The innermost surface area is at the magnetic axis with  $A_1 = 0$  and the outermost area is the plasma surface at  $n_\rho + 1$ . The volume and area elements are used in converting between energy fluxes and volumetric sources/sinks for a conservative scheme in arbitrarily shaped toroidal plasmas. The cyclotron transport problem in **CYTRAN** uses characteristic electron densities and temperatures for each zone,  $n_e$  and  $T_e$  respectively, plus an average magnetic field intensity,  $\bar{B}$ , in a 1-D model of an intrinsically 3-D problem. Thus, the scheme can be applied, in principle, to non-axisymmetric plasmas (stellarators) as well as axisymmetric plasmas (tokamaks). However, since the emission is strongly anisotropic with respect to the magnetic field, the stronger curvature parallel to the local magnetic field in non-axisymmetric plasmas may lead to a stronger loss than in axisymmetric plasmas.

There have been no studies to indicate the best weighting for the average magnetic field strength for a surface; in benchmark tests we have used  $\bar{B} = \langle |B| \rangle$  or  $\bar{B} = \langle B^2 \rangle^{1/2}$ , but because the emission scales as  $B^{5/2}$  it can be argued that  $\bar{B} = \langle |B|^{5/2} \rangle^{2/5}$  may be a better choice.

A frequency step size for the emission and absorption calculations is set from the range of  $\bar{B}$  (apparently using empirical values for the reference frequency limits for which no documentation has been found). Expressed in terms of the electron cyclotron frequency, these are:

$$\omega_{\min} = 2.843\omega_{ce} \quad (2)$$

$$\omega_{\max} = 13.645\omega_{ce} \quad (3)$$

$$\Delta\omega = \frac{0.08}{k_{\text{res}}}(\omega_{\max} - \omega_{\min}) = \frac{0.864}{k_{\text{res}}}\omega_{ce} \quad (4)$$

where  $k_{\text{res}}$  has been added in this version of **CYTRAN** to give the user some control. Note that the value of  $\omega_{\max}$  is only used to calculate  $\Delta\omega$ , while the emission and absorption is then calculated over the frequency interval

$$\omega \in [\omega_{\min}, \omega_{\min} + 8(\omega_{\max} - \omega_{\min})] \quad (5)$$

by taking  $100k_{\text{res}}$  steps in frequency starting at  $\omega_{\min}$ . In testing the code it was found that when the frequency range was divided into 100 intervals as originally coded ( $k_{\text{res}} = 1$ ), it yielded fairly large glitches in the net power profile. Tests indicated that the problem was not governed by the frequency range or number of radial grid points, but by  $\Delta\omega$ . A factor of  $k_{\text{res}} \approx 10$  is required in order to keep the fluctuations to less than a couple percent.

A characteristic radial dimension for each plasma zone is calculated from the volumes and surfaces areas for use in the reabsorption calculation.

The heart of the calculation is a loop over frequencies, with a call to **CYTRAN\_OPACITY** to calculate the opacity factors for ordinary and extraordinary mode radiation. The frequency loop contains loops over the plasma radial elements for emission and absorption at each frequency.

The only output is the net power per unit volume in each radial cell,  $P_{\text{sync}}$  (in keV/m<sup>3</sup>/s, with a negative value representing a loss). The total power loss from the plasma is then:

$$\dot{W}_{\text{sync}}(w) = 1.602 \times 10^{-16} \sum_{i=1}^{n_p} P_{\text{sync},i} \Delta V_i \quad (6)$$

## I/O

```
!Input:
! ree      -refl of X mode from incident X mode [0-1]
! reo      -refl of X mode from incident 0 mode [0-1]
! roo      -refl of 0 mode from incident 0 mode [0-1]
! roe      -refl of 0 mode from incident X mode [0-1]
! nr_r     -no. of radial plasma cells [-]
! bmod_r(nr_r)  -<|B|> [T]
! den_r(nr_r)  -electron density in cell i [/m**3]
! te_r(nr_r)   -electron temperature in cell i [keV]
! area_rm(nr_r+1) -surface area at inner boundary of cell [m**2]
!           -plasma outer surface area is at nr_r+1
! dvol_r(nr_r) -cell volume [m**3]
!Optional input:
! K_CYT_RES  -frequency resolution level
!           =1 delta(omega)=default of 100 frequency intervals
```

```
! >1 100*K_CYT_RES frequency intervals
! 10 recommended to keep graininess < a couple %
!Output:
! psync_r(nr_r) -net power source/loss in cell i [keV/m**3/s]
```

### 3 Private Routines

#### CYTRAN\_OPACITY

CYTRAN\_OPACITY( nr\_r, te\_r, bmod\_r, wfreq, alphae, alphao)

#### General Description

CYTRAN\_OPACITY calculates the x-mode and o-mode opacities at the wave frequency,  $\omega$ .

#### Mathematical Description

The opacity factors are empirical fits developed by Tamor for the SNECTR code, expressed in terms of a factor dependent on the electron temperature (in keV):

$$f_{Te} = \frac{1}{4 + T_e + 25/T_e} \quad (7)$$

The extra-ordinary and ordinary opacity factors are given by:

$$L_e = \max \left[ 0, 0.045 + \left( \frac{\omega}{\omega_{ce}} - 2 \right) f_{Te} \right] \quad (8)$$

$$\log_{10} \left( \frac{\omega^2}{\omega_{ce}^2} \alpha_e \right) = 1.45 - 7.8 L_e^{1/2} \quad (9)$$

$$L_o = \max \left[ 0, 0.180 + \left( \frac{\omega}{\omega_{ce}} - 1 \right) f_{Te} \right] \quad (10)$$

$$\log_{10} \left( \frac{\omega^2}{\omega_{ce}^2} \alpha_o \right) = 2.45 - 8.58 L_o^{1/2} \quad (11)$$

#### I/O

```
!Input:
! nr_r           -no. of radial cells [-]
! te_r(nr_r)    -electron temperature [keV]
! bmod_r(nr_r)  -<|B|> [T]
! wfreq         -wave angular frequency [radians/s]
!Output:
! alphae(nr_r)  -extra-ordinary mode opacity
! alphao(nr_r)  -ordinary mode opacity
```

## 4 Files and Test Cases

From the tarred/zipped files you should get the directory structure:

```
\cytran      -main directory
  \bin       -binaries
  \dat       -data
  \doc       -documentation
  \results   -results
  \src       -source code
```

In the `\src` subdirectory are the Fortran 90 source routines:

```
cytran_dr.f90      Driver for CYTRAN
cytran_mod.f90     CYTRAN module
spec_kind_mod.f90 Precision specification module
x_mod.f90          Cross-section and reaction rate module
setup_ajax.f90     Routines to set up AJAX
ajax_mod.f90       Plasma MHD equilibrium interface module
linear1_mod.f90    1D linear interpolation module
spline1_mod.f90    1D spline interpolation module
write_mod.f90      Formatted output module
```

Also in the `\src` subdirectory is the **makefile**. The **makefile** requires an environment variable, **CPU**, that can be set in any one of three ways for the system you are using:

1. Set externally before executing make, e.g.,  

```
setenv CPU SOL
```
2. Set internally by inserting a line in the **makefile** before first usage, e.g.,  

```
CPU=SOL
```
3. Set when executing the **makefile**, e.g.,  

```
make CPU=SOL cytran
```

Incorporated options are: DEC, HP, IBM, J90, NAG, PGI, SGI, and SOL. Options for running the **makefile** are:

```
make          -compile and load {\com CYTRAN}
make cytran   -compile and load {\com CYTRAN}
make test     -run with default namelist file
make fire     -run FIRE test case
make ignitor  -run IGNITOR test case
make iter     -run ITER test case
make all      -run all test cases
make clean    -remove all .o and .mod files
```

There are three test cases – ITER, FIRE, and IGNITOR – in the \dat directory. Each test case generates three output files:

<b>1d_cytran.dat</b>	1D profiles for IDL postprocessing
<b>sum_cytran.dat</b>	Summary of results
<b>msg_cytran.dat</b>	Error, warning and other messages

Sample output files for each test case have a case name appended and either **\_4** or **\_8** designating the precision set in **spec\_kind\_mod.f90**, which can be edited to change the precision.

## Documentation

This documentation was generated in LaTeX. Contact me ([houlbergwa@ornl.gov](mailto:houlbergwa@ornl.gov)) for the LaTeX source files.

## 5 Revision History

**CYTRAN 2.0** is a modernization an earlier version to an F90/95 module. This version allows the user to decrease the size of the frequency step size in order to obtain smoother radiation profiles.

In **CYTRAN 2.1** all source code was converted to free format. This allowed comments on I/O to incorporated in the declarations rather than in comment blocks at the beginning of each routine.

## 6 Limitations

None.

## 7 Future Extensions

None planned.

## 8 Code Variables

<b>fabs</b>	$f_{\text{abs}}$	fraction of radiation absorbed by the wall
<b>fself</b>	$f_{\text{self}}$	fraction of X (O) mode reflected as X (O) mode
<b>ree</b>	$R_{ee}$	reflection of X mode from incident X mode
<b>reo</b>	$R_{eo}$	reflection of X mode from incident O mode
<b>roe</b>	$R_{oe}$	reflection of O mode from incident X mode
<b>roo</b>	$R_{oo}$	reflection of O mode from incident O mode
<b>den_r</b>	$n_e(\rho)$	electron density profile
<b>te_r</b>	$T_e(\rho)$	electron temperature profile
<b>bmod_r</b>	$\bar{B}(\rho)$	average magnetic field profile
<b>area_rm</b>	$A(\rho)$	surface areas
<b>dvol_r</b>	$\Delta V(\rho)$	volume elements
<b>nr_r</b>	$n_\rho$	number of volume elements
<b>psync_r</b>	$P_{\text{sync}}(\rho)$	cycotron power loss/source profile
<b>alphae</b>	$\alpha_e$	opacity for X mode waves
<b>alphao</b>	$\alpha_o$	opacity for O mode waves
<b>wfreq</b>	$\omega$	wave angular frequency

## 9 Code Diagrams

An \* before the names of routines each module indicate that they are **PUBLIC** and can be accessed directly by the user.

CYTRAN\_MOD

```
*CYTRAN
  |--CYTRAN_OPACITY
```

## References

- [1] S. Tamor, "Studies of Emission and Transport of Synchrotron Radiation in Tokamaks," SAIC Report SAI-023-81-110LJ/LAPS-71 (1981).
- [2] S. Tamor, "A Simple Fast Routine for Computation of Energy Transport by Synchrotron Radiation in Tokamaks and Similar Geometries," SAIC Report SAI-023-81-189LJ/LAPS-72 (1981).
- [3] D.C. Baxter, S. Tamor, "Profile Effects of Synchrotron Radiation in Deuterium Fuelled Tokamaks," SAIC Report SAI-023-81-215LJ/LAPS-74 (1981).
- [4] D.C. Baxter, S. Tamor, *Nucl. Technol./Fusion* **3** (1983) 181.
- [5] S. Tamor, "Calculation of Energy Transport by Cyclotron Radiation in Fusion Plasmas," *Nucl. Technol./Fusion* **3** (1983) 293.
- [6] S. Tamor, R. Carrera, E. Montalvo, M.N. Rosenbluth, "Cyclotron Radiation and Thermal Stability in the IGNITEX Experiment," *Bull. Am. Phys. Soc.* **33** (1988) 1970.
- [7] F. Albajar, M. Bornatici, F. Engelmann, "Electron Cyclotron Radiative Transfer in Fusion Plasmas," *Nucl. Fusion* **42** (2002) 670.