

EDGE2D/EIRENE code interface report

sven.wiesen@jet.efda.org

Abstract

This technical report summarizes a part of the code integration project which had the goal to couple the 3D neutral particle Monte-Carlo code EIRENE to the 2D plasma fluid code EDGE2D. EIRENE should complement the NIMBUS option (another 3D Monte-Carlo code for neutral dynamics for fusion plasmas) which is already in place and coupled to EDGE2D to provide neutral profiles. Additionally, neutral particle dynamics become more and more important especially in cases where the fusion plasma is heavily influenced by plasma-wall interactions and/or when the operational regime is strongly dependent on the neutral behaviour (eg. within detached plasma scenarios or due penetration of impurity neutrals into the core etc.).

Compared to NIMBUS, EIRENE has some additional capabilities: it communicates with an enhanced A&M database in particular for molecular processes, a simpler user-interface, can handle arbitrary geometry structures in 1D/2D/3D, has a time-dependent mode and some more sophisticated features (eg. non-linear processes like neutral-neutral interactions (viscosity) or photon transport (opacity) issues can be treated).

For a switching between NIMBUS and EIRENE it is important to ensure mutual compatibility of the physics model in both codes. This was done by mimicking NIMBUS behaviour with EIRENE as good as possible. Remaining differences and some new features which had to be implemented into EIRENE for a strict one-to-one comparison are discussed in this present report.

It has been attempted to keep the code interface as transparent as possible from the EDGE2D point-of-view. The experienced EDGE2D/NIMBUS user should be in a position to easily switch from an EDGE2D/NIMBUS to a new simulation with EDGE2D/EIRENE to derive SOL plasma fluid solutions including EIRENE features.

Contents

I	EDGE2D geometry and plasma data	3
Ia	EDGE2D plasma background and geometry	3
Ia.1	Output produced for use in EIRENE	6
Ia.2	EDGE2D plasma background preparation	10
Ib	Global analysis	12
Ib.1	total volume	12
Ib.2	total number of electrons	13
Ib.3	recombination rates	13
Ib.4	total particle fluxes on targets	14

II	Verification of target/surface fluxes & passing	16
	EIRENE data back to EDGE2D	16
IIa	EDGE2D target conditions	16
IIa.1	EDGE2D target conditions	16
IIa.2	Implementation of target conditions in EIRENE	17
IIa.3	Perpendicular fluxes	18
IIb	Passing EIRENE neutral flow fields back to EDGE2D, closing the loop	19
IIb.1	General EIRENE output file-format for fluid-codes	22
IIb.2	Macro-zones	25
III	Physical model, atomic and surface physics	26
IIIa	Identification and implementation of atomic, molecular and surface processes	26
IIIa.1	History of a test-particle in NIMBUS	26
IIIa.2	Volumetric processes	30
IIIa.3	Surface processes	32
IIIa.4	Implementation of NIMBUS standard surface reflection model into EIRENE	35
IIIa.5	Implementation of ADAS database into EIRENE	37
IIIb	Analysis of test cases using the same A&M physics model	38
IIIb.1	Molecules switched off	40
IIIb.2	Molecules switched on	45
IIIb.3	Carbon impurities	53
IV	Completion of the EDGE2D/EIRENE code package	55
IVa	Source term re-evaluation in EDGE2D including EIRENE atomic and molecular rates	55
IVb	Coupled EDGE2D/EIRENE runs	59
IVb.1	Molecules switched off, case (A)	59
IVb.2	Molecules switched on, case (A)	65
A	NIMBUS supplements	71
A.I	NIMBUS definition of macro-zones	71
A.II	NIMBUS volume tallies	72
A.III	NIMBUS surface tallies	73

I EDGE2D geometry and plasma data

The EDGE2D/EIRENE interface consists of the following parts:

1. An interfacing routine on the EDGE2D side which generates temporarily an output file including information about the geometry, plasma-profiles and any additional data needed by the interface. This file (namely `e2deir.dat`) is created by the new EDGE2D routine `linkeirene` which is an add-on to the `linkpg` routine which does a similar thing but for the NIMBUS Monte-Carlo code.
2. A F90 module RDE2D which reads the EDGE2D data file `e2deir.dat` and sets up the geometry structure including cell-indexing etc.
3. The main part of the general interface is EIRINT2D which creates standardised files to be read in by EIRENE including the plasma background distributed on a triangularised 2D geometry. An EIRENE input-file is also created partly from a template and partly dependent on the plasma-background (eg. impurities or no impurities) and geometry (eg. septum or no septum, puffs, pumps, etc.).
4. On the EIRENE side the output from the neutral simulation (spatial neutral profiles, density, temperature etc.) is written out to files which also have a standardised file-format. These files are created via user-defined subroutines within EIRENE (`outusr`).
5. Finally, the output of EIRENE (written out to a file `eirene.transfer`) is read by the new EDGE2D subroutine `linkeirene` again. To close the loop one also has to transfer the information about the applied neutral A&M model used in EIRENE to have the same on the EDGE2D side.

The central pieces of EIRINT2D are the modularised parts 2 to 4 which are working on standardised input/output file-formats. Therefore, EIRINT2D is seen to be general for various fluidcode-EIRENE couplings (ie. EDGE2D-EIRENE, OSM-EIRENE, B2-EIRENE etc.). Only the first and last parts of the interface depend on the fluid-code itself since they are directly connected to it.

Ia EDGE2D plasma background and geometry

The EDGE2D magnetic mesh (quadrangles) is taken from `e2deir.dat` and each quadrangle is split up into two triangles. The cell structure of EDGE2D (ii-jj) is similar to that of B2: a core region is enclosed by two private regions which provides two cuts and an adjacent sol region (see fig. 1).

For the void spaces (we have two of them: a sol-void (main-chamber) and a private-void) we have the following options:

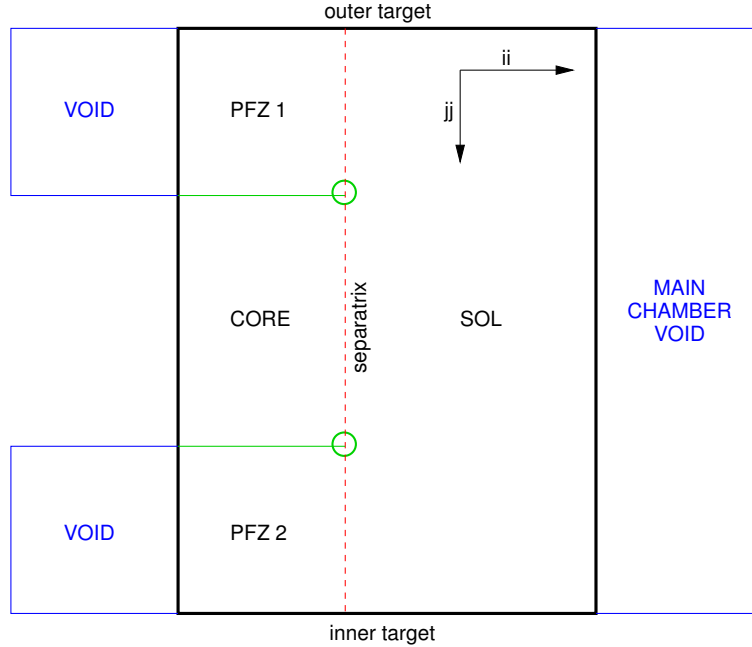


Figure 1: Schematic view of the EDGE2D (ii-jj) ring-row-geometry. The black line encloses the magnetic-mesh grid bounded by the inner and outer targets. The two encircled points at the separatrix are drawn together in physical space providing the X-point. Thus, the bottom of the private flux zone PFZ 1 is connected to the top of PFZ 2 (green lines). Additionally, the upper and lower core-boundaries are connected as well to give a periodic boundary for the jj-rows (green lines again). The main-chamber and the void-zones outside of the PFZs (blue) are discretised with *Triangle* or *tria/meshgen* finite-element generators. The void-space encircled by the inner core-ring is (usually) not triangularised.

- option 1, both void spaces are triangularized via the *Triangle* program which is called from EIRINT2D interface.
(see: <http://www.cs.cmu.edu/~quake/triangle.html>)
This is an external program called via the `system-command`.
- option 2, *tria/meshgen*: this is a program to triangularize the void spaces can also be used in module RDE2D. Here, the original *tria/meshgen*-code developed by B.Kueppers (FZ-Juelich) was rewritten to have a F90 module to make it more flexible. This should be the standard option.
- option 3, no void space discretization: EIRINT2D features a flag, which switches the void space triangularisation on or off. Instead, one big additional cell is created for the SOL and the PFZ void space. Followed particles crossing the magnetic mesh border are tracked to the next collision with a wall segment (i.e. vessel). On the other hand, particles coming from outside (i.e. main chamber or PFZ) are placed correctly on the magnetic mesh triangle at reentry.

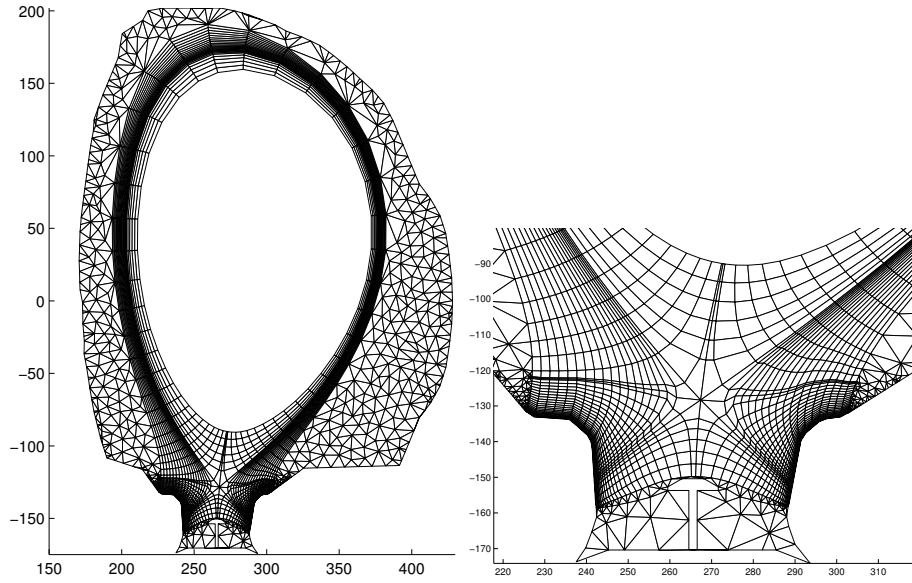


Figure 2: Example of a combined cell geometry of the magnetic-mesh grid (quadrangles) and void-space triangles generated by *Triangle* (JET shot no. 50401). The quadrangles will be split up into just 2 triangles before the geometry is transferred into EIRENE thus the triangularisation is quite coarse within the main-chamber void-region (to overcome this, additional *Steiner-points* (not shown here) between the void-regions and the magnetic-mesh boundary may be introduced and according to this the outer magnetic-mesh ring-cells must be split up into more than just two triangles. But this is not yet included within the current EDGE2D/EIRENE interface). There might be septum-structures (see private-void region) which can be handled by the triangle generators.

Triangle and *tria/meshgen* do checks if all polygon segments are connected to triangles etc. When using the *Triangle* program the user has not to consider the direction of the void-surrounding polygon segments. Here, it is also easy to put holes into the geometry (e.g. for a septum/baffles) by specifying additional coordinates inside the hole.

The produced triangles can be controlled via parameters (for example the angle between two triangle edges can be set to a minimum value to make the triangularized void-space more homogeneous).

At the moment, there is one drawback when EIRINT2D uses *Triangle*: although it does take care of additional so called *Steiner-points* which should be defined when the triangle size is smaller than an adjacent magnetic mesh boundary cell, but it does not make use of this feature up to now (the boundary cells would have to be split up in more than two triangles).

The module RDE2D was intended to use *tria/meshgen* (Option 2). In this case the additional Steiner points are already defined for the void spaces and the boundary cells of magnetic mesh are split up correctly. Holes in the mesh are automatically identified by the orientation of polygonal lines. The *tria/meshgen* option is extensively tested and e.g. used by SOLPS. (*tria/meshgen* is supported by FZ-Juelich

and ITER-IT, Germany)

For the time being *Triangle* is used rather than *tria/meshgen* because EIRINT2D produces an EIRENE input-file and mesh-data on the basis of *Triangle* output (EIRINT2D is based on the OSM-specific interfacing module and here *Triangle* has been used by default). Nevertheless, the *tria/meshgen* option should be introduced and should be used as the default option in the future to avoid double coding (single-sourcing approach).

Another option could be the use of the NIMBUS void-space discretization (split-up into so-called macro-zones, see appendix A.I). This is useful when one wants to compare EIRENE and NIMBUS at a higher accuracy.. But as long as no processes in the void space occur (neutral-neutral or neutral-photon), the benchmark results between EIRENE and NIMBUS are entirely independent of that issue (in fact, NIMBUS was never intended to simulate non-linear neutral-neutral dynamics and this will possibly not be the case in the future).

After processing the geometry the plasma-data for main-plasma species (ie. hydrogen) and impurities (eg. carbon) is distributed upon the triangularised magnetic-mesh grid.

Ia.1 Output produced for use in EIRENE

EIRINT2D produces four standardised output files (for the grid and plasma-profiles) and an input file (`eirene.input`) for EIRENE. In both triangularisation options 1 and 2 (*Triangle* and *tria/meshgen*) the same grid file format is assembled:

Mesh vertices `triangles.npco_char`:

```
nver
do i=1,nver
    i      ver(i,1:3)                      (I6,3F12.6)
enddo
```

There are `nver` vertices. The z-coordinate `ver(i,3)` is always set to zero (in the future this may change when in 3D applications more than one layer of plasma-grids are used or more sophisticated discretisation options (ie. tetrahedrons) may be used, not discussed in this context furthermore.). All coordinates are given in [*cm*].

(**Note:** the special upside-down flip in EDGE2D coordinates is already singled out here).

Mesh elements `triangles.elemente`:

```

ntri
do i=1,ntri
  i    tri(i)%ver(1:3)  dum1(1:3)  dum2(1:3)  dum3(1:2)
                                     (I6,3(4x,3I6),4x,2I6)
enddo

```

There are `ntri` triangles with vertice points `tri(i)%ver(1:3)` defined in `.npco_char`. The triangle vertices are counter-clockwise ordered. The `dum(:)` arrays are not used in EIRENE and only for EIRINT2D.

Neighbour relations `triangles.neighbors`:

```

ntri
do i=1,ntri
  i  (tri(i)%map(j) tri(i)%sid(j) tri(i)%sur(j),j=1,3)\
    dummy(1:4)
                                     (I6,4x,3(3I6,4x),2I6,2x,2I4)
enddo

```

`dummy(:)` are numbers which are not used by EIRENE.

Here the neighbouring information of each triangle is defined. `map(j)` and `sid(j)` give the number and side of the neighbouring triangle (connected to side `j` of triangle `i`), `sur(j)` gives (if $\neq 0$) the surface to which side `j` is connected to. The value of `sur(j)` points to a surface reflection model specified in the EIRENE input file (block 3a). Different surface models can be defined there. For the EDGE2D coupling the following 6 types of reflection zones/models are introduced:

1. core boundary, absorbing
2. inner target, 100% recycling or partly pumping
3. outer target, 100% recycling or partly pumping
4. SOL boundary, switch to void-space
5. PFZ boundary (outer half), switch to void-space
6. PFZ boundary (inner half), switch to void-space

The consistency of all volumes between fluid-code cells and the cells used within EIRENE is important for the overall global balance (eg. He-pumping) and for recombining divertor plasma in particular, if

$$n_i n_e \langle \sigma v \rangle_{rec} \approx n_0 n_e \langle \sigma v \rangle_{ion} \quad (1)$$

locally and globally. Differences between these two rates may result in oscillations in the source-terms when coupled runs are executed iteratively.

Plasma and target data `triangles.plasma` (old version, see also below):

```
*
* HEADER
*
ntri
do i=1,ntri
  i   Te(i)  Ti(i)  ne(i)  vx(i)  vy(i)  vz(i)  bx(i)\\
      by(i)  bz(i)  pot(i)  dummy, idummy(1:2)
      (I7,2E14.6,2x,1P,E14.6,2x,3E14.6,2x,5E12.4,0P,6x,2I4)
enddo
* TARGET DATA
ntardat
do i=1,ntardat
  tri(i)  side(i)  flux(i)  Tet(i)  Tit(i)  nit(i)  vpar(i)  \\
      m(i)    dum(i)  fe(i)    fi(i)    fsh(i)  fel(i)  \\
      vxt(i)  vyt(i)  vzt(i)  ii(i)   jj(i)
      (I7,I6,1P,E14.6,0P,2E14.6,1P,2E14.6,0P,F6.2,1P,      \\
      E10.2,0P,F6.2,1x,3(F6.2,1x),3E14.6,6x,2I4)
enddo
```

Plasma: Electron/ion-temperatures T_e, T_i are given in $[eV]$, electron density n_e in $[cm^{-3}]$, velocities (cartesian laboratory system) v_x, v_y, v_z in $[cm/s]$. Magnetic field vector b_x, b_y, b_z in $[T]$. El. potential pot in $[V]$. All values are assumed to be cell-averages in EIRENE (given at the centre of gravity of each cell).

Target: `ntardat` values (step-functions) are read in (corresponding to triangle target segments). `tri` is the triangle number defined in `.elemente`, `side` is the corresponding triangle side. `flux` is the particle flux in $[A]$ impinging on the target segment. `Tet, Tit` and `nit` are the electron temperature, ion temperature and ion density, respectively (actually, `nit` is derived from parallel flux density and parallel velocity at the target segment). `vpar` and `M` describing the parallel velocity and the corresponding Mach number at the target. `fe` is the electron energy flux multiplier at sheath entrance (se): $Q_e^{se} = f_e T_e^{se}$, `fi` is the thermal part of ion energy flux at sheath edge $Q_i^{se} = f_i T_i^{se}$ and `fsh` is the sheath multiplier (sheath potential $f_{sh} T_e^{se}$). `fel` is the total ion energy flux at sheath entrance. If the latter is zero, EIRENE derives f_{el} from f_i, f_e, M, T_i^{se} and T_e^{se} . `vxt, vyt, vzt` are the components of the parallel velocity $v_{||}$ of ions directly at the targets. `ii` and `jj` are the corresponding cell coordinates of the original EDGE2D magnetic grid. All target surface values are given directly at the target (no cell averages).

The new triangles.zplasma file format:

The new file-format should incorporate more than one plasma species (multispecies). Firstly, a general plasma data block independent of species should be written out (volume and surface data). Then, the main ion species should be written (presumably fuel ions, *H* or *D*), then all the other ion species (impurities). The block-format of main-ion-species and impurity-species is identical (again: first volume averaged then surface averaged data) and the usual EIRENE numbering is applied ($i_{pls} = 1 \dots N_{pls}$) which means that there are N_{pls} blocks (additionally to the MISC block). For EDGE2D that means, that the main-ion-species corresponds to $I_{pls}=1$ and the nzs impurity-species correspond to $I_{pls}=2 \dots nzs+1$ with $N_{pls} = nzs+1$. The number nzs gives the number of additional impurity species, thus $N_{pls} = nzs+1$.

Note: here, the number/postions of the asterisks *** is important since they define tags which are essentially for processing the file. Also the tags itself (ie. MISC, ION #01 etc.) must be in place as exactly as defined below:

```
* EIRENE PLASMA FILE FOR TRIANGULAR GRID
*
*** MISC PLASMA DATA
*
*
ntri
do i=1,ntri
    i   te(i)  ne(i)   bx(i) by(i) bz(i)  pot(i)  psi(i)
    (I7,7(1x,e14.7))
enddo
*** MISC TARGET DATA
*
*
ntardat
do i=1,ntardat
    i   tri(i) side(i) tet(i) fe(i) fsh(i)
    (3I7,3(1x,e14.7))
enddo
*** ION #01 PLASMA DATA
*
*
nmass nchar nchrg
do i=1,ntri
    i   ti(i)  ni(i)   vx(i)  vy(i)  vz(i)
    (I7,5(1x,e14.7))
enddo
*** ION #01 TARGET DATA
*
```

```
*
do i=1,ntardat
  i  flux(i)  tit(i)  nit(i) vxt(i) vyt(i) vzt(i)
      fi(i) fel(i) vpart(i) mach(i) usr(i)
      (I7,11(1x,e14.7))
enddo
*** ION #02 PLASMA DATA
*
*
etcetc...
```

Temperatures t_i, t_e etc. given in [eV], densities n_i, n_e etc. in [cm^{-3}], velocities v_x, v_y etc. in [$cm s^{-1}$], magnetic field b_x, b_y etc. in [T], fluxes $flux$ in [A], el. potential pot in [V], magn. flux psi normalised.

Like in the old .plasma file-format $flux, tit$ and nit are values for the flux-[A], temperature- [eV] and density- [cm^{-3}] step-functions defined at the surface, v_{xt}, v_{yt}, v_{zt} are the velocities at the target in [cm/s]. fel is the total ion energy flux at the sheath entrance and fi is again the thermal ion heat flux transmission factor.

usr is a user-defined step-function. It is used, for example, in EDGE2D if poloidal neutral puffs are to be simulated where the puffing rate is defined at the crossing border between SOL and the main-chamber. From this point of view it is an abuse of language to speak of *target fluxes*. Rather, one should speak about *surface fluxes* defined on poloidal or parallel surface segments.

This new file-format (which is now the standard) is called .zplasma to avoid confusion with the .plasma file.

Ia.2 EDGE2D plasma background preparation

The term 'multi-species' used oftenly within the EDGE2D context is sometimes a bit confusing. In EIRENE terms, multi-species means to have more than one plasma species (i.e. hydrogen ions plus additional plasma species like carbon ions (including all charge states) or deuterium/tritium ions). In EDGE2D terms, multi-species means to have more species than the usual main ion species (i.e. hydrogen) plus two impurity species, thus at least four plasma species. Here, in this context the EIRENE multi-species definition (at least two plasma species) is used throughout.

The plasma data coming out of EDGE2D is prepared onto the triangles in the following way: all scalar values of a EDGE2D quadrangle cell (i.e. temperatures, densities) are just equally shared to both corresponding triangles. EDGE2D provides plasma velocities on a coordinate system defined by the local magnetic flux

surface: parallel velocity $v_{||}$, radial velocity v_{ρ} , diamagnetic or transverse velocity v_d . To do a transformation into the cartesian lab-system one has to derive the magnetic field at the point at which $(v_{||}, v_{\rho}, v_d)$ is given. The magnetic EDGE2D grid is provided by GRID2D which creates a plasma grid in correspondence with fitted magnetic surfaces (i.e. EFIT) to provide a mostly orthogonal mesh-geometry. The magnetic field can then be derived back from the edges of a magnetic cell. For the EDGE2D cell (ii, jj)

$$\Delta r = r(ii, jj + 1) - r(ii, jj - 1) \quad (2)$$

$$\Delta z = z(ii, jj + 1) - z(ii, jj - 1) \quad (3)$$

$$\tan \alpha = \frac{\Delta z}{\Delta r} \quad (4)$$

$$B_x = s \cos \alpha \quad (5)$$

$$B_y = s \sin \alpha \quad (6)$$

$$B_z = \sqrt{1 - s^2} \cdot \text{sign} B_{\phi} \quad (7)$$

where $s = B_{\theta}/B_{total}$. With the poloidal velocity $v_{\theta} = v_{||}s + v_d(1 - s)$ the velocity can be transformed to cartesian coordinates:

$$v_x = v_{\theta}s \cos \alpha + v_{\rho} \sin \alpha \quad (8)$$

$$v_y = v_{\theta}s \sin \alpha - v_{\rho} \cos \alpha \quad (9)$$

$$v_z = v_{||}B_z - v_d s \quad (10)$$

Note: $v_{||}$ is usually defined on a staggered grid on the EDGE2D side.

The ion density at the targets can also be derived from the flux and velocity at the target:

$$n_i^{targ} = \frac{\Gamma_i^{targ}}{v_i^{targ}} \quad (11)$$

EDGE2D provides the integrated flux F in $[s^{-1}]$ over the full poloidal and toroidal extend of each target segment

$$F = \int \Gamma dA \quad (12)$$

To derive n_i at the target one has to derive Γ from F for each target segment. To account for a possible inclination of the velocity to the target. the following formula for the target surface which is actually seen by the velocity v_i^{targ} is used

$$A = 2\pi R_t \cdot l \cdot |\cos \alpha| \cdot s \quad (13)$$

where R_t is the radius at the center of each target segment and l is the length of that segment (in the poloidal cross-section). $\cos \alpha$ accounts for the inclination of the poloidal velocity component to the target segment surface normal. $s = B_{\phi}/B_{tot}$

accounts for the toroidal inclination.
Finally, the target ion density is given by

$$n_i^{targ} = \frac{F}{A \cdot v_i^{targ}}. \quad (14)$$

Ib Global analysis

As a first check to verify the correct transfer of plasma and geometry from EDGE2D into EIRENE the following measures have to be matched:

- total volume of computational grid
- total number of electrons
- total recombination rate (for particles, momentum and energy)
- total fluxes to the target

The following EDGE2D cases had been considered:

(A): gcor 50401v1 mar3104 seq#1
(no drifts, no imp., septum)
swiesen 50401 aug1505 seq#1 run3

(B): gcor 53549 dec1504 seq#1
(no drifts, no imp., septum)
swiesen 53549 aug1605 seq#1 run4

(C): gcor 90011v9 mar3104 seq#1
(no drifts, no imp., no septum, special JET-case for simulating ITER)
swiesen 90011 aug1605 seq#1 run5

Ib.1 total volume

case	EDGE2D w/o void	EIRENE w/o void	ratio
(A)	2.84756	2.84797	0.999856
(B)	2.50686	2.50599	1.000347
(C)	20.5818	20.5779	1.000190

values given in $10^7 [cm^{-3}]$. There is only a small difference in the total volume. This may for example depend on the calculation of the volume for each cell. In EDGE2D the centre of gravity of a quadrangle is used to calculate the volume (area of quad times $2\pi R$). In EIRENE the quad is split up into 2 triangles with

different centre of gravities. But this provides no error since both approaches are equivalent.

Ib.2 total number of electrons

case	EDGE2D	EIRENE	ratio
(A)	2.76275	2.76332	0.999794
(B)	4.67734	4.67622	1.000240
(C)	34.9419	34.9396	1.000066

values given in 10^{20} . The small mismatch is probably due to the small mismatch in the cell volumes Ib.1

Ib.3 recombination rates

First of all, a constant volume recombination rate coefficient $\langle\sigma v\rangle = 10^{-14}[cm^3/s]$ was used to check the consistency of the transfer of plasma data into EIRENE. Usually, ADAS is used to derive recombination rates in EDGE2D/NIMBUS (see IIIa.5). To ease the things, a rather simple formula was used for the recombination rate check (Gordeev et al.). The Gordeev-Formula implemented in EIRENE has a different form as in EDGE2D/NIMBUS:

EIRENE (Gordeev et al., Pis'ma Zh. Ehksp. Teor. Fiz. 25 (1977) p223),:
(Gordeev rate version I):

$$\langle\sigma v\rangle_{rec} = \frac{1.27 \cdot 10^{-13} \left(\frac{E_I}{T_e[eV]}\right)^{1.5}}{\left(\frac{E_I}{T_e[eV]} + 0.59\right)} \quad (15)$$

where $E_I = 13.6 eV$ is the hydrogen ionisation potential.

EDGE2D/NIMBUS (Gordeev et al., Jept. Lett. 25 No 3(1977) p204),
(Gordeev rate version II):

if $T_e < 4 keV$

$$\langle\sigma v\rangle_{rec} = \frac{1.48 \cdot 10^{-14}}{\sqrt{T_e[keV]}(1 + 43.3T_e[keV])} \quad (16)$$

else

$$\langle\sigma v\rangle_{rec} = \frac{3.58 \cdot 10^{-16}}{T_e[keV]^{1.388}} \quad (17)$$

Both Gordeev rates I and II are equal when the numerical factors are factored out and the $4 keV$ cutoff is not considered. However, the latter version of the Gordeev formula has been implemented into EIRENE for the sake of numerical accuracy.

Particle recombination loss rate $\int d^3r n_i n_e \langle\sigma v\rangle_{rec}$ in $[10^1 A]$:

case	EDGE2D	EIR (Gordeev II)	EIR (Gordeev I)
(A) const	0.76361	0.76392	0.76392
(A) Gordeev	6.1038	6.1074	6.1122
(B) Gordeev	0.96383	0.96386	0.96467
(C) Gordeev	38.459	38.421	38.454

Ion energy recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec} \frac{3}{2} T_i$ in $[10^3 W]$:

case	EDGE2D	EIR (Gordeev II)	EIR (Gordeev I)
(A) const	1.5927	1.5950	1.5950
(A) Gordeev	1.0571	1.0768	1.0778
(B) Gordeev	0.78802	0.79077	0.79160
(C) Gordeev	27.169	29.785	29.816

Electron energy recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec} (\frac{3}{2} + c) T_e$ in $[10^3 W]$:

case	EDGE2D	EIR (Gordeev II) $c = 0$	EIR (Gordeev I) $c = 0$	EIR (Gordeev I) $c \neq 0$
(A) const	1.5038	1.5043	1.5043	2.3566
(A) Gordeev	0.91555	0.91594	0.91680	1.1003
(B) Gordeev	0.66532	0.66531	0.66602	1.0059
(C) Gordeev	18.383	18.391	18.410	25.894

Here, a correction c was implemented in the current EIRENE version:

$$c = \frac{d \ln \langle \sigma v \rangle_{rec}}{d \ln T_e} = \frac{-0.5 \frac{E_I}{T_e} + 0.59}{\frac{E_I}{T_e} + 0.59} \quad (18)$$

which accounts for the electron velocity dependence of the recombination cross section.. For the comparison c was set to zero and shows good matching. If $c \neq 0$ significant difference can be seen.

Parallel ion-momentum recombination loss rate

$\int d^3r n_i n_e \langle \sigma v \rangle_{rec} m v_{i||}$ in $[10^{-17} A g cm/s]$:

case	EDGE2D	EIR (Gordeev II)	EIR (Gordeev I)
(A) const	0.76388	0.76431	0.76431
(A) Gordeev	7.1455	7.1498	7.1556
(B) Gordeev	1.1174	1.1179	1.1189
(C) Gordeev	291.06	290.99	291.25

Ib.4 total particle fluxes on targets

The neutral source is mainly due to the plasma recycling-flux at the targets. Therefore, the plasma fluxes impinging on the target must be exactly the same seen by EIRENE as like in NIMBUS:

total outer target flux in $[10^4 A]$:

case	EDGE2D	EIRENE
(A)	1.6104	1.6099
(B)	0.81048	0.81569
(C)	5.4401	5.4409

total inner target flux in $[10^4 A]$:

case	EDGE2D	EIRENE
(A)	1.3157	1.3155
(B)	1.0315	1.0290
(C)	3.6862	3.6850

II Verification of target/surface fluxes & passing EIRENE data back to EDGE2D

IIa

For the comparison of different neutral kinetic codes like EIRENE and NIMBUS a correct treatment of the boundary conditions is important. Especially, the fluxes impinging on the targets and their sheath boundary models must be identical. This section reviews the implemented target boundary conditions in EDGE2D and NIMBUS. The applied models may then be used to extend the models within EIRENE if they are not already implemented yet.

IIa.1 EDGE2D target conditions

The target boundary conditions are set via the routines `eqparty`, `eqmomy` and `eqtemy`. These routines set up the matrix equations for solving the particle, momentum and temperature equations, respectively, in the y direction (which means jj-row direction in EDGE2D convention) and are driven by the routine `eqy`:

```
spltup --> eqy --> eqparty/eqmomy/eqtemy
```

The target boundary conditions are also set in these routines. There are also routines setting up the equations in x- or i-direction: `eq . . x`, but there are, of course, no target b.c. Rather, there the perpendicular fluxes are set up, see below.

`eqparty`, particle b.c.:

The b.c. is set to be zero density-gradient if there is no temperature-gradient (i.e. no pressure-gradient). See also `eqtemy` temperature b.c., below.

`eqmomy`, momentum b.c.:

The default is sound speed at the targets. There is a flag `LNORBC` which switches to non-orthogonal b.c. (default: `LNORBC = .FALSE.`, orthogonal cell approximation applied, i.e. flux only along EDGE2D rings).

Different models can be applied for momentum b.c., set by flag `IVISC` (eg. Igithanov et al., see `data.f`)

`eqtemy`, energy/temperature b.c.:

The default is to have a zero parallel temperature gradient at the targets so that the kinetic part of the thermal ion energy flux onto the targets reduces to

$$\frac{5}{2}T_i n_i v_i = \frac{5}{2}T_i \Gamma_i \equiv f_i T_i \Gamma_i \quad (19)$$

Here, the ion heat flux transmission factor $f_i = 2.5$ is fixed and can be changed via a special variable `HTFI` in `data.f`.

Default: $f_i = 2.5$

Integrated temperature boundary conditions are used for electrons as default option, `LITBCY == .TRUE.`. The variable $f_e^* = \text{HTFE}$ is set in `data.f` (default. 4.5). Here, it is assumed that the sheath potential drops with $2.5T_e$ ($f_{sh} = 2.5$ already included in `HTFE`) which is correct for a hydrogen plasma (for deuterium a value of $2.8T_e$ would be more appropriate). The electrons are assumed to flow with a Maxwellian velocity distribution onto the target, hence $f_e = 2.0$.

Default: $f_e^* = 4.5 \longrightarrow f_e = 2.0, f_{sh} = 2.5$

Consistently with EDGE2D, the sheath potential is assumed to drop off with constant $2.5T_e$ (variable `SHP`)

IIa.2 Implementation of target conditions in EIRENE

Because EDGE2D solves a temperature equation and not an energy equation the b.c. at the targets must be altered to account for the additional convective $m/2v^2$ term lacking in the temperature equation. Therefore, to the $2.5T_i$ from EDGE2D one has to add $m_i/2M^2c_{si}^2$ with sound speed

$$c_{si} = \sqrt{\frac{T_e + \gamma T_i}{m_i}} \quad (20)$$

(γ adiabatic coefficient, here: $\gamma = 1$) which means for example with $M \equiv 1$: $0.5T_i + 0.5T_e$ (throughout this section index i means an arbitrary heavy particle species, also impurity ions). The sum gives then $3T_i + 0.5T_e$ or, more generally: $\alpha T_i + \beta T_e$.

Additionally we have the Maxwell-contribution from the electrons $2T_e\Gamma_e$ and the part from the sheath potential which is set to be constant in EDGE2D/NIMBUS ($2.5T_e\Gamma_e$, which is shared by the ions according to their electric fluxes $Z_i\Gamma_i$, $\Gamma_e = \sum_i Z_i\Gamma_i$).

Usually, EIRENE calculates the sheath potential in front of the targets by itself (routine `SHEATH`) and gives values around $2.8T_e$ for a Deuterium plasma. To consider a different sheath potential drop and to use the real heat flux factors provided by EDGE2D, the target data output file format had to be extended (see part I, file `.plasma`). Additional values for f_i , f_e^* and f_{sh} is now read in where one usually assumes that $f_{sh}T_e = (f_e^* - 2.0)T_e$ is the sheath voltage drop. But to keep the things as open as possible, f_{sh} is also read in.

Additional step-functions for each target segment have to be implemented for these transmission factors: `FISTEP(ipls)`, `FESTEP`, `FSHSTEP`.

The velocity step functions at the target segments are usually taken from the `VXIN` / `VYIN` / `VZIN` arrays in EIRENE. To match the velocity boundary conditions exactly (i.e. $M = 1$) the velocities directly at the targets must be provided by EDGE2D for EIRENE. Again, the output file format for the target conditions is extended by values `vxt`, `vyt`, `vzt` to have the cartesian velocity components of $v_{||}$ at each target segment (see part I). Mach numbers exactly unity are now seen by EIRENE.

For the impurities the target conditions are likewise implemented. There is the option to set the impurity ion-speed to the sonic value of the fuel ions in `eqmomy` (flag `LVZEVI` in `data.f`) which is the default.

IIa.3 Perpendicular fluxes

Similarly, like for the target fluxes NIMBUS can also treat perpendicular fluxes streaming between SOL and main-chamber or between PFZ and private void. In most cases the perpendicular flux is small compared to the parallel fluxes onto the targets and they might be disregarded. But sometimes there is a distinct amount of perpendicular flux into the main-chamber. In the same manner as step-functions are defined for the poloidal projection of the parallel fluxes (ie. target fluxes) the perpendicular fluxes can be expressed likewise.

NIMBUS treats the crossing fluxes in a certain way especially for the main-chamber wall interaction. On top of that poloidal neutral puffs are introduced as a specific fraction to the perpendicular plasma flux. EIRINT2D can create step-functions for perpendicular flows as well and passes them via the `.zplasma` file to EIRENE.

An additional issue is the definition of poloidal puffs in NIMBUS at the border to the main-chamber SOL: they are defined as a fraction of the perpendicular flux and treated via the NIMBUS subroutine `HOW` but with a different energy distribution than the plasma-flux hitting the main-chamber vessel, which is recycled there.

In fact, `HOW` decides dependent on the source-strength ratio if a neutral is created from the plasma flux leaving the magnetic-mesh grid or coming from the puff. In the latter case, the puff geometry is for example defined on the outermost ring of the magnetic-mesh grid. The starting-point of a neutral particle is located by sampling upon that outer magnetic-mesh ring. This new particle is followed then outwards up to the vessel wall (without registration its flight-path within the Monte-Carlo statistics) and a new cosine-like angular distribution is sampled inwards at the vessel-wall which provides the actual birth point of the neutral. The neutral energy is defined elsewhere (for the puffs) or sheath-physics is applied if the neutral is originating from the perpendicular plasma flux (with plasma parameters at the boundary between magnetic-mesh grid and the void region).

EIRENE was set up to mimic the NIMBUS behaviour as good as possible. But different ways had been chosen for emulation. Neutral puffs are implemented via

additional surfaces generated from a linear projection of the puffing ring-segments onto the vessel wall. Those additional surfaces act as additional strata (ie. neutral particle sources) completely disconnected from the perpendicular plasma flux. On the other hand the birthprofile of neutrals coming from the perpendicular plasma flux itself is located at the magnetic-mesh grid. The EIRENE way of separating neutral influxes from plasma fluxes and puffs seems to be more realistic but it is the question if this procedure resembles the NIMBUS algorithm sufficiently.

IIb Passing EIRENE neutral flow fields back to EDGE2D, closing the loop

For the transfer of neutral flow fields back into the EDGE2D environment the following arrays/tallies have their correspondences (another list of array-names for the actual NIMBUS tallies is given in appendices A.II and A.III):

surface tallies:

EDGE2D	units	EIRENE	units
HESCPD (ih) ZESCPD (iz) escaped neutral flux (fuel and impurities) measured in flux per toroidal length	$s^{-1}cm^{-1}$	POTAT(iatm)-PRFAAT(iatm) -PRFMAT(iatm) -PRFIAT(iatm) POTML(imol)-PRFAML(imol) -PRFMML(imol) -PRFIML(imol) POTIO(iion)-PRFAIO(iion) -PRFMIO(iion) -PRFIIO(iion)	A
HESCAL (ih) ZESCAL (iz) pumped neutral flux (albedoed fuel andimps) measured in flux per toroidal length	$s^{-1}cm^{-1}$	SPUMPAT(iatm) + SPUMPML(imol) + SPUMPIO(iion)	A
ZSPUT (iz) sputtered impurity flux per tor.length	$s^{-1}cm^{-1}$	SPTAT(iatm) + SPTML(imol) + SPTIO(iion) SPTPL(ipls)	A

ipls is an EIRENE (bulk) plasma species, *ih* is an EDGE2D main ion species, *iz* is an EDGE2D impurity species. The correspondence between EDGE2D species numbers *ih/iz* (for fuel ions/impurities) and EIRENE species indices *iatm, imol, iion* and *ipls* is given externally via a file `eirene.species` which is created by EIRINT2D.

EDGE2D interprets the neutral profiles in atomic units. For example tallies for molecular hydrogen (ie. POTML, PRFMML etc.) must be multiplied by factor two. To keep the things open, those multiplication factors can be defined also in `eirene.species`.

volume tallies:

EDGE2D	units	EIRENE	units
PROFSN_H(k, ih) PROFSZ(k, iz) ionisation source	$cm^{-3}s^{-1}$	PAPL(ipls, j) (atoms) PMPL(ipls, j) (molecs) PIPL(ipls, j) (ions)	Acm^{-3}
PROFMP(k) par. momentum src.	$gcm^{-2}s^{-2}$	COPV(1*NPLS+ipls, j) (at) COPV(2*NPLS+ipls, j) (mol) COPV(3*NPLS+ipls, j) (ion)	$Agcm/s/cm^3$
PROFQ(k) ion energy source	$ergcm^{-3}s^{-1}$	EAPL(j) (atoms) EMPL(j) (molecs) E IPL(j) (ions)	Wcm^{-3}
PROFQE(k) el. energy source	$ergcm^{-3}s^{-1}$	EAEL(j) (atoms) EMEL(j) (molecs) EIEL(j) (ions)	Wcm^{-3}
PROFA_H(k, ih) PROFZ(k, iz) neutral at. dens.	cm^{-3}	PDENA(iatm, j)	cm^{-3}
PROFM_H(k, ih) PROFMZ(k, iz) neutral mol. dens.	cm^{-3}	PDENM(imol, j)	cm^{-3}
VA_H(1:3, k, ih) VAZ(1:3, k, iz) at. veloc. comp.	cm^{-3}	VXDENA(iatm, j) /PDENA(iatm, j) VYDENA(iatm, j) /PDENA(iatm, j) VZDENA(iatm, j) /PDENA(iatm, j)	cm/s
VM_H(1:3, k, ih) VMZ(1:3, k, iz) mol. veloc. comp.	cm^{-3}	VXDENM(imol, j) /PDENM(imol, j) VYDENM(imol, j) /PDENM(imol, j) VZDENM(imol, j) /PDENM(imol, j)	cm/s
VNUTAP_h(k, ih) VNUTAPZ(k, iz) neutr. at. $v_{ }$	cms^{-1}	(VXDENA(iatm, j)*BXIN(j)+ VYDENA(iatm, j)*BYIN(j)+ VZDENA(iatm, j)*BZIN(j)) /PDENA(iatm, j)	$gcm^{-2}s^{-1}$
VNUTMP_h(k, ih) VNUTMPZ(k, iz) neutr. mol. $v_{ }$	cms^{-1}	(VXDENM(imol, j)*BXIN(j)+ VYDENM(imol, j)*BYIN(j)+ VZDENM(imol, j)*BZIN(j)) /PDENM(imol, j)	$gcm^{-2}s^{-1}$
ENEUTA_H(k, ih) neutr. at. temp. (kin. part included)	eV	EDENA(iatm, j) /PDENA(iatm, j)	eV
ENEUTM_H(k, ih) neutr. mol. temp. (kin. part included)	eV	EDENM(imol, j) /PDENM(imol, j)	eV

k is an EDGE2D cell number and j is a virtual EIRENE cell consisting of two combined EIRENE triangles which belong to cell k (hence the scalar values for neutral density and neutral energy density are simply volume averaged over the two triangles).

Note: the kinetic energy $mv^2/2$ which is to be subtracted from the energy (derived from the EDENx tallies) to derive a temperature must be included in ENEUTA, ENEUTM to be consistent with EDGE2D (it is subtracted there; also the degrees of freedom factor $3/2$ must be still included ENEUTx).

The calling sequence is like this:

```
spltup --> hsource --> hmc
               --> intfac
               --> linkeirene --> eirint2d
                               --> eirene
```

Here, the new subrotuine linkeirene is an add-on to the old linkpg subrou-tine calling NIMBUS (linkpg is still called to derive the vessel-geometry which is passed to EIRINT2D. To avoid this, linkpg will be split-up in the future.).

Once the neutral profiles are read in by the linkeirene call, the neutral density for atoms and molecules and the ionisation source are renormalised within intfac before they are passed back to routine hmc. For each cell k three factors are multiplicated onto the tallies:

1. $VOLP(N)/AREAP(N)*FNORM$, where
 $FNORM = (TOTSOU+STMOUT)/(HIONIZ+HESCPDT+HTMOUT)$

The first factor corrects small differences in the cell areas/volumes between EDGE2D (AREAP) and NIMBUS (VOLP).

The second factor FNORM cares for statistical errors coming from the MC-code. This is actually done by EIRENE already (routine GETSCL). Thus, this factor can be always set to unity. But to have a single-sourcing approach, one has to transfer then also the EIRENE values for HIONIZ (ionisation source, which is actually calculated in intfac via $PROFSN_H(k) \cdot VOLP(N)$, with the lack of a factor $2\pi R_0$) and HESCPDT (the total escape of the neutrals). The latter is provided by EIRENE through the outflux tallies (POTATI,POTMLI,POTIOI) and the reflection tallies (PRFAAI, PRFMMI, PRFIII). The sum of incident neutral fluxes minus the sum of reflected fluxes gives then HESCPDT.

2. $RNIMB/RMESH(K)/FLXREF$, where FLXREF is the sum over the total plasma outflux $FLXOUT_H(k,ih)$, the external controlled influx (e.g. puff-

ing) $\text{HEXTRL_H}(k,ih)$ and the total recombination source $\text{SRECOH} \cdot \text{RNIMB}$.

3. $\text{RMESH}(K) \cdot \text{AREAP}(N) / \text{DVOL}(K)$, which renormalise locally the difference of the volumes between the EDGE2D polygons and the volume $\text{DVOL} = H \cdot \text{DRHO} \cdot \text{DTHETA}$ which comes from the metric $dV = h_\varrho h_\theta d\varrho d\theta 2\pi$

Note: all values for particle fluxes (i.e. FLXREF , TOTPRO) are divided by 2π in intfac and hmc . The real outflux is given by $\text{FLXOUT} \cdot \text{TWOPI}$.

Iib.1 General EIRENE output file-format for fluid-codes

The output file format of `eirene.transfer` is presented below. All tallies got their names according to the EIRENE manual [4], except `pspl`, `espl` and `esel`, which are the sums over `pXpl`, `eXpl` and `eXel`, respectively, where X stands for a, m, i and ph.):

```

*
ntri nrad                (3(1x,i6))
*
nstordr                 (1x,i6)
*
natm nmol nion nphot    (4(1x,i6))
*
npls mplsti mplsv       (3(1x,i6))
*
nlimps                 (1x,i6)
* BULK SPECIES
do ipls = 1,npls
  * vol.av.
  do i = 1,ntri
    i                                \
    papl(ipls,i) pmpl(ipls,i) pipl(ipls,i) \
    pphpl(ipls,i)                    \
    pspl(ipls,i)                      \
    msplx(ipls,i)                     \
    msply(ipls,i)                     \
    msplz(ipls,i)                     \
    mspl(ipls,i)                      \
    eapl(ipls,i) empl(ipls,i) eipl(ipls,i) \
    ephpl(ipls,i)                    \
    espl(i)                          \
    eael(ipls,i) emel(ipls,i) eiel(ipls,i) \
    ephel(ipls,i)                     \

```

```
        esel(i)
        (i6,30(1x,e14.7))
    enddo
    * srf.av.
    do i = 1,nlimps
        i potpl(ipls,i) eotpl(ipls,i) sptpl(ipls,i)
        (i6,20(1x,e14.7))
    enddo
enddo
* ATOMIC SPECIES
do iatm=1,natm
    * vol.av.
    do i = 1,ntri
        i, pdena(iatm,i),
        vxdena(iatm,i) vydena(iatm,i) vzdena(iatm,i) \\
        vparaa(iatm,i) edena(iatm,i) \\
        (i6,20(1x,e14.7))
    enddo
    * srf.av.
    do i=1,nlimps
        i, potat(iatm,i) \\
        prfaat(iatm,i) prfmat(iatm,i) prfiat(iatm,i) \\
        prfphat(iatm,i) prfpat(iatm,i) \\
        eotat(iatm,i) \\
        erfaat(iatm,i) erfmat(iatm,i) erfiat(iatm,i) \\
        erfphat(iatm,i) erfpat(iatm,i) \\
        sptat(iatm,i)
        (i6,20(1x,e14.7))
    enddo
enddo
* MOLECULAR SPECIES
do imol=1,nmol
    * vol.av.
    do i = 1,ntri
        i, pdenm(imol,i),
        vxdenm(imol,i) vydenm(imol,i) vzdenm(imol,i) \\
        vparam(imol,i) edenm(imol,i) \\
        (i6,20(1x,e14.7))
    enddo
    * srf.av.
    do i=1,nlimps
        i, potml(imol,i) \\
        prfamml(imol,i) prfmml(imol,i) prfiml(imol,i) \\
        prfphml(imol,i) prfpml(imol,i) \\
    enddo
enddo
```

```
        eotml(imol,i)                                \\
        erfaml(imol,i) erfmmml(imol,i) erfiml(imol,i) \\
        erfphml(imol,i) erfpml(imol,i)              \\
        sptml(imol,i)
        (i6,20(1x,e14.7))
    enddo
enddo
* TEST-ION SPECIES
do iion=1,nion
    * vol.av.
    do i = 1,ntri
        i, pdeni(iion,i),                                \\
        vxdeni(iion,i) vydeni(iion,i) vzdeni(iion,i) \\
        vparai(iion,i) edeni(iion,ir)                  \\
        (i6,20(1x,e14.7))
    enddo
    * srf.av.
    do i=1,nlimps
        i, potio(iion,i)                                \\
        prfaio(iion,i) prfmio(iion,i) prfiio(iion,i) \\
        prfphio(iion,i) prfpio(iion,i)                \\
        eotio(iion,i)                                \\
        erfaio(iion,i) erfmio(iion,i) erfiio(iion,i) \\
        erfphio(iion,i) erfpio(iion,i)                \\
        sptio(iion,i)
        (i6,20(1x,e14.7))
    enddo
enddo
* PHOTON SPECIES
do iphot=1,nphot
    * vol.av.
    do i = 1,ntri
        i, pdenph(iphot,i),                                \\
        vxdenph(iphot,i) vydenph(iphot,i)                \\
        vzdenph(iphot,i)                                \\
        vparaph(iphot,i) edenph(iphot,ir)                \\
        (i6,20(1x,e14.7))
    enddo
    * srf.av.
    do i=1,nlimps
        i, potpht(iphot,i)                                \\
        prfapht(iphot,i) prfmpht(iphot,i)                \\
        prfipht(iphot,i)                                \\
        prfphpht(iphot,i) prfppht(iphot,i)                \\
    enddo
enddo
```



```
        eotpht(iphot,i)                                \\
        erfapht(iphot,i) erfmpht(iphot,i)             \\
        erfipht(iphot,i)                               \\
        erfphpht(iphot,i) erfppht(iphot,i)            \\
        sptpht(iphot,i)
        (i6,20(1x,e14.7))
    enddo
enddo
* MISC DATA
do i = 1,ntri
    i ncltal(i) dummy dummy vol(i)                   \\
    voltal(i)                                           \\
    dein(i) tein(i) bxin(i) byin(i) bzin(i)          \\
    bfin(i)
    (i6,20(1x,e14.6))
enddo
```

IIb.2 Macro-zones

The whole EDGE2D region (SOL, PFZ, main) and the void-regions are sliced up into so called macro-zones (see appdenix A.I). EDGE2D provides within its output file `print` information about densities, volume, ionisation rates, recombination rates etc. in terms of macro-zones. The idea is to adapt this scheme and provide also from the EIRENE side information already sliced up into such defined macro-zones. This is not done yet (especially the void-regions are complicated when data defined on triangles is transferred back onto macro-zones) and therefore, direct comparisons of neutral profiles between EIRENE and NIMBUS in the void-regions are not possible for the while.

III Physical model, atomic and surface physics

This section reviews the atomic and molecular physics models and their volumetric and surface processes implemented in EDGE2D/NIMBUS. The differences to the EIRENE physical model are shown and compared to NIMBUS. The question which additional atomic rates and surface processes should be included in EIRENE and/or NIMBUS for a direct comparison is discussed. Test runs of specific cases are carried out. Remaining differences are documented.

IIIa Identification and implementation of atomic, molecular and surface processes

For a one-to-one comparison of EIRENE and NIMBUS the same physical model (i.e. same atomic and molecular rates as well the same surface processes) for the sources and sinks must be used. Therefore, the upcoming differences of the implementation have to be discussed and documented. If it is possible to have the same physical model on both sides, EIRENE should give the same results as NIMBUS. Before getting deeper into the physical details, a short discussion of the history of a Monte-Carlo test-particle in NIMBUS is given.

IIIa.1 History of a test-particle in NIMBUS

There are only a few main routines in NIMBUS which are responsible for the following of a test-particle. Each particle is started from the routine `STARTN`. Here it is decided whether a recombination (volume-)source from background ions or a recycling (surface-)source from the target creates a neutral particle.

For the volumetric source, position and energy of the neutral is sampled via predefined source-strengths (like in EIRENE). Here, particles are sampled isotropically plus the part of a shifted Maxwellian velocity distribution. The type of the particle is selected (in rotational order) rather than sampled which is the case in EIRENE (see NIMBUS subroutine `WHICH` with `ibirth=0`).

The surface source is a bit more complex. Firstly, the segment at the target is sampled. Then, routine `HOW` is called to give the position, direction and energy of the new neutral particle at the target. A routine `WALINT` is called to define the recycle model.

Routine `RUN` then tracks the test-particle. Actually, `RUN` calls `STARTN` for a specified number of histories (but correlation sampling is also possible). The collision rates for volumetric processes with other particles (ions, electrons, impurity ions) are provided by routines like `IOSIG`, `CXSIG` `HMSIG` etc, which are discussed below. The rates are pre-calculated and saved in specific arrays (i.e. done by routines like `PREIOZ` which calls `IOSIG` etc.).

The post-collisional character of the test-particle (energy, velocity, type) for elastic

or charge-exchange processes is given by routines ELAS or CHARGX, respectively.. If the test-particle hits a surface (vessel-wall or target) routine WALINT is called (w/ `ibirth=1`). Here, chemical or physical sputtering models are applied or a simple reflection can be carried out or pumps can be included via given albedoes for each vessel/wall-segment or predefined pump-structures.

If the particle hits a transparent-surface splitting or Russian roulette schemes may be applied.

The test-particle is finished up, when it was either ionised or sputtered into a plasma-type particle (bulk) or is simply absorbed by a surface.

A (far too crude) flow-diagram of the test-particle history tracking in NIMBUS (next page):

```

RUN --> STARTN --> WHICH
      either --> sampling from predef. source
              strengths, (pre. RECSIG)
      or --> WALINT (via HOW)

      --> vol. processes --> XST --> XSI    (pre IOSIG)
                          --> XSCX0 (pre CXSIG)
                          --> XSEL0

                          --> DIFFUS --> XSCX --> HXHP
                                  --> ELXS --> HHP
                                  --> HEHC
                                  --> HEHM
                                  --> H2HP

      either --> CHARGX
              (charge exch. CX)

      or --> ELAS --> COLES (pre HMSIG)
              (elast.)

      or --> COLAM
              (mol.diss.)

      --> surf. processes --> WALINT
                          either --> TRIMOD
                                  TRIM dbase

                          or --> std. refl. model
                                  (Eckstein et.al)

                          or --> SPUTER

                          or --> albedoes (pumps)

```

survival biassing

An EIRENE mothballed feature was reactivated for mimicking NIMBUS Monte-Carlo results. Usually, a neutral atom is followed as long as it is not absorbed, that means ionised via electron impact. NIMBUS does not absorb that just created ion into the bulk. Rather, the weight of the neutral is reduced and the test-particle is followed further on.

A similar (but more general) option for such a non-analog method was implemented in EIRENE in the past and is now reactivated again (EIRENE input file block9, flag `WMINV`). Here, the user can specify a minimum weight threshold.

When the particle weight falls below that threshold it is absorbed into the bulk finally.

With this *survival biassing* feature (sometimes also termed as *expected survival* or *absorption suppression*) it is possible to have small but non-zero atomic and molecular outfluxes from the edge into the core region. NIMBUS treats this outgoing neutral flux as to be reflected as ions coming from the core, thus, the outgoing neutral flux should be the same when mimicking NIMBUS with EIRENE.

IIIa.2 Volumetric processes

Different atomic and molecular rates can be used. They are switched via the `IXS(:)` array in `data.f`.

ionisation model for hydrogen atoms, `IXS(1) =`

- 1 NRL
- 2 AURORA
- 3 NIMBUS
- 4 NIMBUS single stage
- 5 ADAS (default)

(see also routine `IOSIG`)

NIMBUS uses ADAS for the default ionisation rate coefficient $\langle \sigma v \rangle_i$. This is not the case in EIRENE, where usually the rates from a collisional radiative model (CRM, Sawada/Fujimoto [6]) are taken. Thus, one has either to implement the CRM rate into NIMBUS or ADAS into EIRENE. The latter would then be a new feature within EIRENE and is therefore a favoured option (see also sec. IIIa.5)

charge exchange model, `IXS(2) =`

- 0 Greenland/Riviere
- 1 Janev (default)

(see also routine `CXSIG`)

NIMBUS uses the microscopic cross-section σ derived from Janev's book (reaction ref. 3.1.8). In contrast to EIRENE (which usually uses the same reaction-fit reference), where precalculated polynomial-fits are used to provide the rate coefficient $\langle \sigma v \rangle_{CX}$, NIMBUS evaluates the averaging integral numerically over a Maxwellian distribution (which may be shifted) multiplied by the relative velocity to derive the rate coefficient $\langle \sigma v_{rel} \rangle_{CX}$. The integrals for the Maxwellian is given via precalculated parametric fits.

On the other hand, EIRENE transforms first to a stationary Maxwellian ($\vec{v}'_0 = \vec{v}_0 - \vec{v}_{flow}$) before it evaluates the rate coefficient. This should be identical to NIMBUS.

molecular processes, `IXS(3) =`

- 0 Jones
- 1 Janev (default)

(see also routine `HMSIG`)

Same molecular rate coefficients are use in NIMBUS and EIRENE:

- (fast) dissociation (ref. 2.2.5) $H_2 + e \rightarrow H + H + e$
- mol. ionisation (ref. 2.2.9) $H_2 + e \rightarrow H_2^+ + 2e$ (followed by $H_2^+ \rightarrow H^+ + H$)
- dissoc. ionisation (ref. 2.2.10) $H_2 + e \rightarrow H + H^+ + 2e$

In EIRENE H_2^+ can be tracked explicitly whereas in NIMBUS the molecular ion break-up into an atom and an ion is assumed to be fast. Effectively $H + H$ is produced from H_2 in 2.2.9. To accomplish this also in EIRENE the inputfile (block4b) is modified to have also that direct channel without the H_2^+ diversion. NIMBUS and EIRENE use exactly the same polynomial fits for the Janev rates. (In the Jones model an additional slow dissociation rate is applied).

impurity ionisation model, $IXS(4) =$
 -1 ADAS(EDGE2D), BELL(NIMBUS)
 0 BELL
 1 ADAS (default)
 2 LOTZ

NIMBUS uses ADAS, should be implemented in EIRENE also (see comment on hydrogen ionisation).

hydrogen recombination model, $IXS(5) =$
 0 none
 1 Gordeev (see eqn. 16).
 2 Janev [5]:

$$\langle \sigma v \rangle_{rec} = 3.29 \cdot 10^{-14} \frac{\beta^{3/2}}{\beta + 0.35} \quad (21)$$

where $\beta = 13.6/T_e$.

3 NRL
 4 ADAS (default)

Default option is ADAS. In sect. Ib.3 the Gordeev rate was used to match the recombination source rates for the hydrogen neutrals. Anyway, ADAS should be an additional option (see sec. IIIa.5).

elastic collisions

Elastic processes can be switched on via variable $ISEHHE$ (default: switched off). Scatterings between H/H^+ , H_2/H^+ and He/H^+ can be carried out.

linearisation scheme, $IXS(6) =$
 0 no
 1 yes (default)

The linearisation is done by calculating numerically the derivatives around the temperature $T_i \pm 0.01T_i$ for the charge-exchange process and $T_e \pm 0.01T_e$ for all other processes. A bilinear variation including the plasma density is not foreseen.

The linearisation scheme has to be accomplished also for the EIRENE rates. For details see IV.

IIIa.3 Surface processes

wall interaction: When routine `WALINT` is called, different wall interaction models can be used. The TRIM database support is installed into NIMBUS. But unfortunately it is not used regularly. If TRIM could be used, EIRENE and NIMBUS would use the same reflection model (TRIM is the default in EIRENE).

By default now, NIMBUS uses a so called standard wall interaction model, which is basically based on Eckstein (1984). The latter is now also implemented in EIRENE. See also description in IIIa.4

physical sputtering: NIMBUS has only one standard option for physical sputtering which is the Roth/Bohdansky/Eckstein model (SPUTER93). The following formula gives the sputtering yield if the impinging energy E_0 in [eV] is larger than the threshold energy E_{th} :

$$Y_{phys,93} = Q_p (1 - a^{2/3}) (1 - a)^2 \frac{\frac{1}{2} \ln(1 + 1.2288b)}{b + 0.1728\sqrt{b} + 0.008b^{0.1504}} \quad (22)$$

where $a = E_{th}/E_0$, $b = E_0/E_{TF}$. Q_p , E_{th} and E_{TF} are tabulated. This formula is as exactly implemented in EIRENE as in NIMBUS. But in contrast to NIMBUS, EIRENE has an additional angular dependence of the sputtering yield included (Yamamura fit):

$$Y_{phys,93}^{EIRENE} = Y_{phys,93} \cdot (\cos \theta)^{-f} \exp \left(fc - \frac{fc}{\cos \theta} \right) \quad (23)$$

where θ is the angle between particle velocity and surface normal, $f = 2$ and $c = 0.26$.

For a one-to-one comparison between NIMBUS and EIRENE, this angular dependence must be switched off in EIRENE.

The energy distribution of physically sputtered particles is for NIMBUS and EIRENE given by the Thompson formula:

$$F(E, E_b) = C \frac{E}{(E - E_b)^3} \quad (24)$$

if the impinging energy is below a threshold $E < E_{max}$. E_b is the binding energy of the sputtered particle and the constant C is defined as

$$C = \frac{E_b}{0.9 (1 + E_{max}/E_b)^{-2} - (1 + E_{max}/E_b)^{-1} + \frac{1}{2}} \quad (25)$$

Whereas in EIRENE an algorithm for direct sampling of the Thompsons formula is implemented to derive the energy E_{phys} of the sputtered particle, NIMBUS uses rather a rejection technique.

chemical sputtering: NIMBUS has several options for chemical sputtering implemented (switched by `IYCHEM`). In the following formula list wall temperatures T_w and energies E are given in [eV], fluxes Γ in [$cm^{-2}s^{-1}$].

- IYCHEM=1:Garcia-Rosales (EPS, 1994) for $D \rightarrow C$

$$Y_{chem} = Y_{chem}^{th} + Y_{chem}^{ath} \quad (26)$$

$$Y_{chem}^{th} = \frac{6 \cdot 10^{19} \exp(-T_w^{-1})}{1 \cdot 10^{15} + 3 \cdot 10^{27} \exp(-2T_w^{-1})} \cdot \frac{2 + 200 \cdot Y_{phys,GR}}{\frac{\max(10^{16}, \Gamma)}{10^{16}}} \quad (27)$$

$$Y_{chem}^{ath} = \frac{0.05 \exp(E_0 + 10^{-3}(20 - T_w^{-1}))}{\left(1 + \exp\left(\frac{E_0 - 150}{25}\right)\right) \left(1 + \exp\left(\frac{T_w - 740}{25}\right)\right)} \quad (28)$$

which depends on the Garcia-Rosales physical sputtering yield $Y_{phys,GR}$ defined as

$$Y_{phys,GR} = 0.1 \cdot s \cdot (1 - x_{th}^{-2/3})(1 - x_{th}^{-1})^2 \quad (29)$$

where $x_{th} = E_0/E_{th}$ is to the threshold energy normalised projectile energy (ie. for carbon $E_{th} = 28 \text{ eV}$) and s is the stopping power defined as

$$s = \frac{1}{2} \ln(1 + 1.2288x_f) \left(x_f + 0.1728\sqrt{x_f} + 0.008x_f^{0.1504} \right)^{-1} \quad (30)$$

with the to the Fermi energy normalised projectile energy $x_f = E_0/E_f$ (ie. for carbon $E_f = 447.0 \text{ eV}$).

$Y_{phys,GR}$ is zero for $x_{th} < 1$.

- IYCHEM=2: Pospieszczyk (EPS, 1995)

$$Y_{chem} = 0.04254 \cdot \left(\frac{\max(5 \cdot 10^{18}, \Gamma)}{5 \cdot 10^{18}} \right)^{-0.477} \quad (31)$$

- IYCHEM=3:Vietzke (Phys. Proc. Inter. Fusion)

$$Y_{chem} = 0.0215 \left(\frac{\max(10^{14}, \Gamma)}{10^{16}} \right)^{-0.1} \quad (32)$$

- IYCHEM=4:Haasz (J. Nuc. Mat., 1996)

The chemical sputtering yield is derived via a 3th order polynomial fit:

$$Y_{chem} = \sum_{i=0}^4 A_i(T_w) \log^{i-1}(E_0) \quad (33)$$

- IYCHEM=5:Garcia-Rosales (Roth, 1996)

$$Y_{chem} = Y_{surf} + Y_{therm} \cdot (1 + 125 \cdot Y_{phys,GR}) \quad (34)$$

consisting of a part for the surface process:

$$Y_{surf} = 0.1s \cdot A \frac{\left(1 - E_0^{-2/3}\right) \left(1 - E_0^{-1}\right)^2}{1 + \exp\left(\frac{\min(90, E_0) - 90}{50}\right)} \quad (35)$$

which is non-zero for $E_0 > 1 \text{ eV}$ and thermal activation part

$$Y_{therm} = 0.033A \frac{\exp(-1.7T_w^{-1})}{2 \cdot 10^{-32}\Gamma + \exp(-1.7T_w^{-1})}. \quad (36)$$

$Y_{phys,GR}$ and s are given as in IYCHEM=1. A is defined as

$$A = \frac{2 \cdot 10^{-32}\Gamma + \exp(-1.7T_w^{-1})}{10^{-32}\Gamma + \left(1 + 2 \cdot 10^{29}\Gamma^{-1} \exp(-1.8T_w^{-1})\right)} \times \frac{\exp(-1.7T_w^{-1})}{1 + 10^{13} \exp(-2.45/T_w^{-1})} \quad (37)$$

- IYCHEM=6:Haasz (J. Nuc. Mat., 1997) w/o flux dependence
Same as IYCHEM=3 but with new values for fit-coefficients $A_i(T_w)$.

- IYCHEM=7:Haasz (J. Nuc. Mat., 1997) w/ flux dependence
Same as IYCHEM=6 but with additional flux dependence:

$$Y_{chem} = \left(\frac{\max(3 \cdot 10^{14}, \Gamma)}{3 \cdot 10^{14}} \right)^{-\alpha} \sum_{i=0}^4 A_i(T_w) \log^{i-1}(E_0) \quad (38)$$

The parameter α can be specified via the f2d input (ALFCH7).

- IYCHEM=8:Roth (19th PSI, 1998)
An extended version of IYCHEM=5.

The chemical sputtering yield can be enhanced (seperately from the physical sputtering yield), specified via f2d ($a_{chem} = \text{AYCHEM}$, default: $a_{chem} = 1$):

$$Y_{chem}^{enhanced} = a_{chem} \cdot Y_{chem} \quad (39)$$

Additionally, a yield enhancement can be applied in NIMBUS (specified in f2d: $a = \text{AYIZ}$, $b = \text{BYIZ}$) for the total yield $Y_{tot} = Y_{phys} + Y_{chem}$ (only for H/D/T-projectiles):

$$Y_{tot}^{enhanced} = a \cdot Y_{tot} + b \quad (40)$$

where the default is $a = 1$ and $b = 0$.

The energy of the sputtered particle is a combination of the energy originating from the physical suppetring process (ie. the Thompson formula) E_{phys} and from the chemical sputtering which is specified in f2d: $E_{chem} = \text{EYCHEM}$. If $Y_{phys} > 0$ then

$$E_{sput} = E_{phys} + \frac{Y_{chem}}{Y_{phys}} E_{chem} \quad (41)$$

else

$$E_{sput} = E_{chem}. \quad (42)$$

The angular distribution of the sputtered particles is cosine-like.

EIRENE supports currently the following chemical sputtering models:

- MODCHM=1: const value, defined in EIRENE input-file
- MODCHM=2: Roth/Pacher, PSI 1998 (weak flux-dependence)
- MODCHM=3: Roth/Pacher, (strong flux-dependence)
- MODCHM=4: Roth/Pacher, 2003 (new version of flux-dependence)
- MODCHM=5: N/A
- MODCHM=6: Haasz-Davis formula (1998), w/o flux dependence
- MODCHM=7: Haasz-Davis formula (1998), w/ flux dependence
- MODCHM=8: N/A
- MODCHM=9: user supplied

In contrast to NIMBUS, EIRENE does not combine chemical and physical sputtering yields. Instead it follows a particle either originating from physical ($Y_{phys} > Y_{chem}$) or chemical sputter-process ($Y_{chem} > Y_{phys}$). Thus it does not create a combined energy E_{sput} as well. When physical sputtering has occurred $E_{sput} = E_{phys}$ and in the case of chemical sputtering EIRENE uses the wall temperature for the sputtered particle energy $E_{sput} = E_{chem} = \frac{3}{2}T_w$. Like in NIMBUS, the angular distribution is cosine-like.

For the one-to-one comparison of the sputtering processes between NIMBUS and EIRENE (see below) the combination of physical sputtering and the Haasz formulas (IYCHEM=6, MODCHM=6) without flux-dependence for chemical sputtering has been used which are implemented with the same polynomial fit-coefficients. There is only a slight difference in the yields for projectile energies E_0 below 10 eV in the EIRENE case due the additional Davis approximation (see fig. 3).

pumping: Pumps are implemented as so called albedoes on predefined pump-structures in NIMBUS or on given ranges of rings and rows (eg. target-surfaces). EIRENE defines arbitrary pumping surfaces or segments as reflecting surfaces with or without albedoes (defined as additional surfaces).

The NIMBUS pump-structures are not implemented into EIRENE. Only pumps at the targets can be used at the moment.

IIIa.4 Implementation of NIMBUS standard surface reflection model into EIRENE

Due to the fact that the TRIM database for surface interaction models (which is the standard in EIRENE) is not the default in NIMBUS, the so called standard surface reflection model according to Eckstein (1984) is implemented in EIRENE

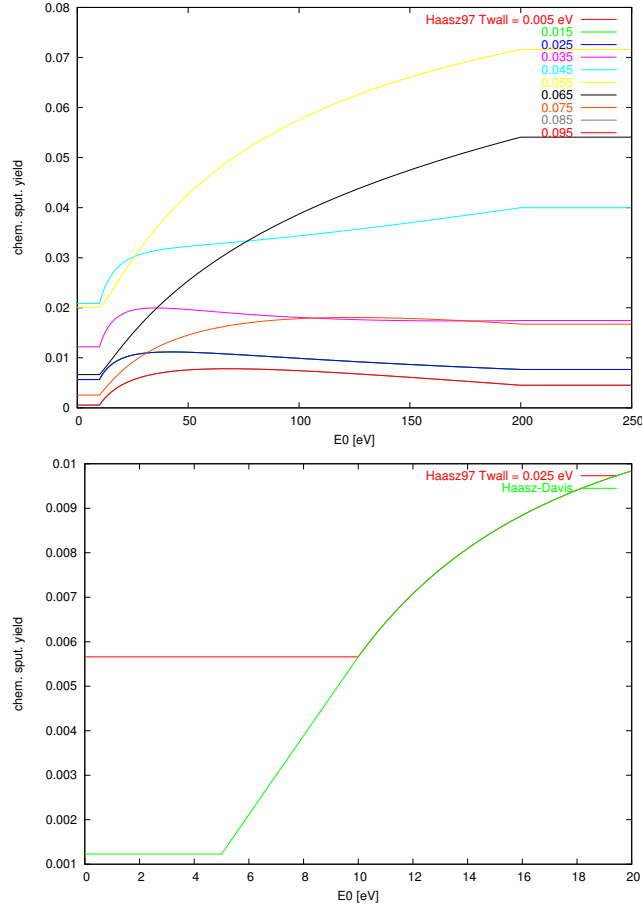


Figure 3: top: chemical sputtering yields after the Haasz97 formula w/o flux dependence (NIMBUS: IYCHEM=6) for different wall temperatures T_w , bottom: comparison to Haasz-Davis formula (EIRENE: MODCHM=6) for room temperature ($T_w \approx 0.025$ eV). The latter has somewhat lower yields for $E_0 < 10$ eV compared to the original Haasz97 cutoff.

to mimic the wall interaction model from NIMBUS. An additional F90 module `mod_eckstein` is introduced and the responsible subroutine `reflec.f` is extended. The additional option `ILREF=4` for this new surface model is introduced which can be switched in the EIRENE inputfile (block2/block3 or via a general definition of a `SURFMOD` in block6).

The Eckstein model implementation in NIMBUS is a bit messy, thus, the formulas for the particle and energy reflection coefficients (R_n and R_e resp. with $0 < R_n, R_e < 1$) have been extracted from the source (NIMBUS routine `WALINT`) and are reprogrammed for the use within `mod_eckstein`.

In the Eckstein model, special formulas for R_n and R_e for projectiles of the kind H , D , DT (effective atom of mass 2.5), He^+ and He^{++} onto wall materials of

the kind *Fe*, *Ni*, *C*, *W*, *Au*, *Cu* are provided whereas for all other materials some default formulas are given.

In WALINT it is decided via sampling from the R_n distribution if a backscattering (reflection) or a re-emission (after sticking at the wall) occurs. In the former case the new energy of the reflected particle is given by

$$E_{new} = E_{old} \frac{R_e}{R_n} \quad (43)$$

and the sort of particle stays the same.

In the latter case of re-emission the particle species may change (provided by a matrix RECMAT for each wall segment) which is set to the identity matrix by default (i.e. the particle type doesn't change). The re-emitted atoms or molecules get an energy either given by the molecular dissociation energy (provided by the array TABLE) or is set to a prescribed constant energy.

In all wall interaction processes it is assumed, that the angular distribution of reflected/re-emitted particles is cosine-like and is sampled thereof.

IIIa.5 Implementation of ADAS database into EIRENE

To have the same physical model it is important to use the same ionisation and recombination rate coefficients in both MC-codes. The ADAS database is extensively used in NIMBUS and therefore ADAS is now implemented into EIRENE, too.

To use ADAS in EIRENE, one has to supply the tag-string ADAS_{yy} in the reaction rate definition block 4 in the EIRENE input-file (similar to the tags AMJUEL, HYDHEL, etc...). Here, _{yy} stands for the year of the ADAS dataset to be used. If that specified year is not existing in the database, the default year 89 is used.

Up to know, only so called H. 4 processes for ionisation and recombination can be used (which means double parametric rate coefficients dependent on T_e and n_e).

The ADAS database must be stored in a directory tree ADAS/ which usually can be set up as a symbolic link to the real ADAS path (i.e. at JET with the command

```
ln -s /home/adas/adas ADAS
```

within the EIRENE run directory.

Only slight changes had to be made within the EIRENE code for the (rudimentary) ADAS implementation:

- an additional module `mod_adas.f90` is provided, which manages and reads the ADAS database. Only one public subroutine can be used from outside the module which drives the database (`subroutine adas`), which is essentially the same subroutine as used by NIMBUS.
- `comxs.f`: an additional array `iadas(:)` is provided which gives the ADAS datafile year `yy` (see above). If `iadas(kk)` is zero, then `kk` is an ordinary (non-default, non-adas) process.

- `slreac.f`: can now handle the ADASyy tag and extracts the datafile year yy into `iadas(kk)`
- `xstei.f`, section 2.D the ionisation rate coefficient can now be read from the ADAS database (if `iadas /= 0`). The array `TABDS1` is filled up here.
- `xsectp.f`, section 2.D: the recombination rate coefficients can be read from ADAS database (if `iadas /= 0`), the array `TABRC1` is filled up. Here, also the electron energy loss coefficient `EELRC1` is set up (including the correction from eq. 18). Hence, section 4 in `xsectp` is skipped, if `iadas /= 0`.

The NIMBUS formula for the electron energy loss for atomic ionisation is also included into EIRENE (via routine `XSTEI`) and an additional flag is introduced `ISCDE=40000`):

$$E_{loss} = 17.5 + \left(5 + c \frac{37.5}{T_e} \right) \log_{10} \left(\frac{10^{15}}{n_e} \right) \quad (44)$$

with $T_e[eV]$ and $n_e[cm^{-3}]$. $c = 1 + 0.2511/T_e$ is a correction factor if $n_e < 10^{15} cm^{-3}$ (after Hotston, Nov. 1985, updated Feb.1987), otherwise c is set to unity.

To check the consistency between EDGE2D/NIMBUS and EIRENE the recombination rates (i.e. the neutral source, see sect. Ib.3) and also the outcoming neutral profiles (i.e. neutral densities, temperatures and plasma particle, energy and momentum sources) are compared.

IIIb Analysis of test cases using the same A&M physics model

(A): `swiesen 50401 dec1205 seq#1`

This case (originally case `50401aug2305seq#1`) is run from scratch, which means that EDGE2D/NIMBUS was run just from the very beginning with molecules been switched off throughout. NIMBUS was called every 10th fluid-timestep. The run ends converged at $t = 50 ms$ internal fluidcode-time. A septum is located at the center of the divertor. Drifts and impurities have been switched off.

(B): `swiesen 53549 mar1006 seq#1`

This case (originally case `53549aug2305seq#1`) is run from scratch. NIMBUS was called every 10th fluid-timestep. The run ends converged at $t = 4000 ms$ internal fluidcode-time. There is no septum, drifts and impurities have been switched off.

The following tables show integrated rate values for the neutral sources (due volume recombination) going from EDGE2D into EIRENE/NIMBUS. This shows

only the correct transfer of plasma data into EIRENE and functioning of access to ADAS database from the EIRENE side.

Particle recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec}$ in $[10^2 A]$:

case	NIMBUS	EIRENE
(A) ADAS96	1.4059	1.4085
(B) ADAS96	4.8017	4.8065

Ion energy recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec} \frac{3}{2} T_i$ in $[10^2 W]$:

case	EDGE2D	EIRENE
(A) ADAS96	6.1369	6.2483
(B) ADAS96	30.621	30.768

Electron energy recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec} (\frac{3}{2} + c) T_e$ in $[10^2 W]$, $c = 0$:

case	EDGE2D	EIRENE
(A) ADAS96	5.4692	5.4776
(B) ADAS96	28.249	28.279

Parallel ion-momentum recombination loss rate $\int d^3r n_i n_e \langle \sigma v \rangle_{rec} m v_{i||}$ in $[10^{-17} A g cm/s]$:

case	EDGE2D	EIRENE
(A) ADAS96	5.6256	5.5863
(B) ADAS96	7.3586	7.3236

To verify the results of EIRENE runs compared to NIMBUS on the same EDGE2D plasma background (ie. mimicking NIMBUS with EIRENE) just atoms are simulated and the molecules (thermally re-emitted from the wall/targets) are switched off (NIMBUS: set `IAEMIS=1`, EIRENE: just emit atoms in Eckstein model (by setting a flag in `mod_eckstein`, switchable via a file `iaemis.tag` generated by the `EIRINT2D` interface). If this is successful, the molecules are switched on again and the same tests are carried out. Afterwards, some special cases where impurities are included (ie. carbon) are also compared (here, the sputtering mechanisms etc. can be tested and compared.)

Global and local tests are carried out. Where the global numbers are providing just a starting point of the real NIMBUS mimicking, the local analysis gives more insight on variables like densities, temperatures, fluxes etc. Thus, for a spatial resolved analysis (here spatial analysis means both, surface and volumetric distribution of the variables) a few MATLAB scripts have been set up to plot neutral profiles either from NIMBUS or EIRENE output on the mesh.

From the EDGE2D routine `hmc` a new output routine `hmcout` is called to print out the volumetric neutral profiles coming from NIMBUS or EIRENE for the use with MATLAB (on a quadrangle mesh, the EIRENE triangles are already combined into quadrangles via the `linkeirene` interfacing routine).

For the fluxes impinging on surfaces routine `linkpg` and `nimbus` was extended to print out NIMBUS target/wall fluxes, whereas on the EIRENE side, the surface fluxes are printed out directly via a slightly extended version of routine `outflx`.

IIIb.1 Molecules switched off

Global analysis

The global values in the following table are actually seen by EDGE2D routine `hmc`. All values correspond to integrations or averages of variables within the magnetic grid, the void-regions are left out here (20000 test-particles had been followed in each case):

w/o molecules	case	NIMBUS	EIRENE
tot. plasma source [$10^4 A$]	(A) (B)	2.7604 6.5257	2.7485 6.2033
tot. ion energy src (abs.) [$10^5 W$]	(A) (B)	5.8741 30.702	6.2407 12.679
tot. el. energy src [$10^5 W$]	(A) (B)	-8.1459 -16.300	-8.1090 -15.371
tot. par. momentum src (abs.) [$10^6 gcm s^{-1}$]	(A) (B)	1.2290 2.1344	1.2452 2.0833
vol. averaged atom dens. [$10^{11} cm^{-3}$]	(A) (B)	4.3911 1.1789	4.5138 1.0971
vol. averaged atom energy dens. [$10^{11} eV/cm^3$]	(A) (B)	6.4661 5.6808	6.6971 3.5255
vol. avg. atom par. velocity dens. (abs) [$10^{16} 1/cm^2 s$]	(A) (B)	3.7563 1.3277	3.6734 1.2915

The following table gives the global balances of the fluxes seen by `intfac` and the derived FNORM factor (global balance scaling factor, see section IIb). TOTSOU is the total neutral source (recycling and recombination), HIONIZ is the total ionisation and HESCPD corresponds to net neutral net flow onto the surfaces (neutral collisions with wall minus reflected/re-emitted part):

Global balances (<code>intfac</code>) w/o molecules	case	NIMBUS	EIRENE
TOTSOU [$10^4 A$]	(A) (B)	2.8383 6.9254	2.8238 6.6047
HIONIZ [$10^4 A$]	(A) (B)	2.8145 6.9104	2.8044 6.5958
HESCPD [$10^4 A$]	(A) (B)	0.0182 0.0228	0.0173 0.0013
derived FNORM	(A) (B)	1.0020 0.9989	1.0008 1.0012

Local analysis for case (A) without molecules

Figure 4 shows the *D*-atom neutral density, temperature and pressure profiles for

case (A) with the same colour scaling. 20000 test-particles have been calculated either in NIMBUS and in EIRENE. Apparently, there is almost no difference in the profiles.

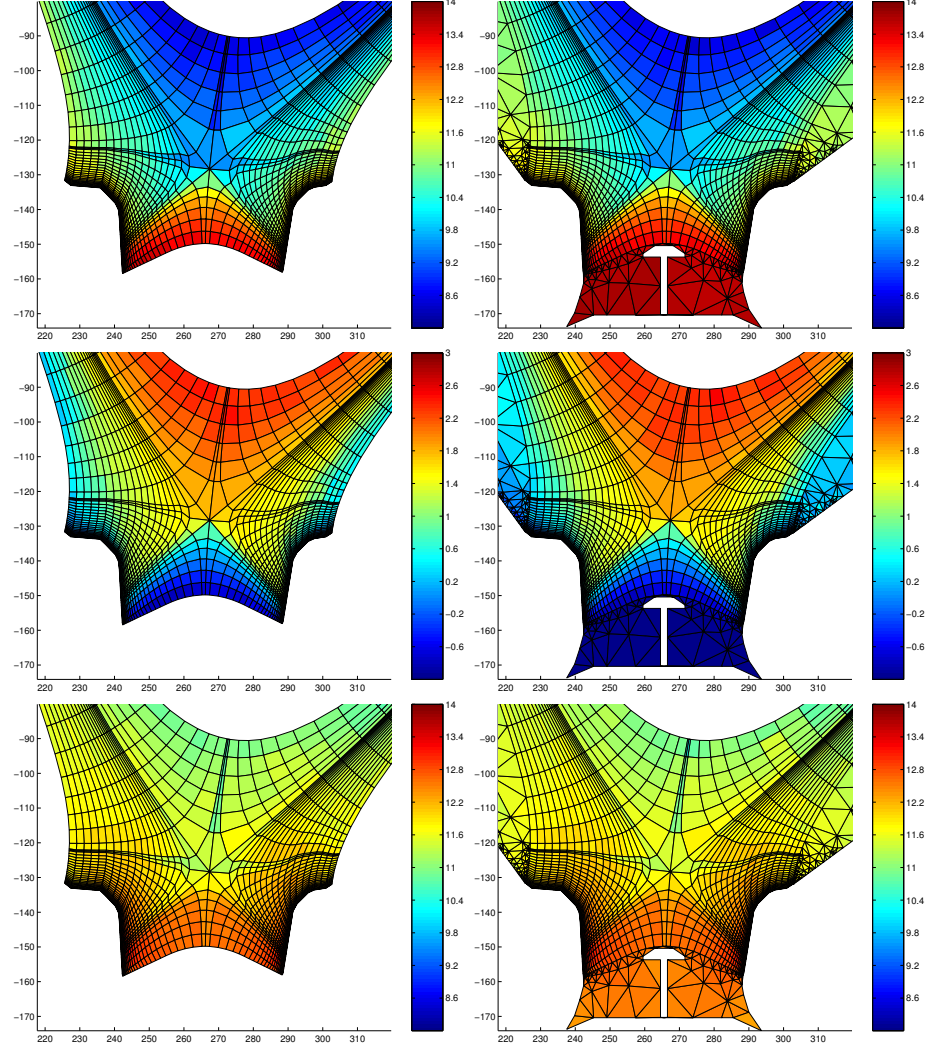


Figure 4: From top to bottom: profiles for density $\log[cm^{-3}]$, temperature $\log[eV]$ and pressure $\log[eV/cm^3]$ for *D*-atoms (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, identical colour scale, molecules switched off

The figures in 5 show the plasma particle source, the ion energy source, electron energy source and the ion momentum source respectively. The ion energy and momentum sources are quite noisy (due to the fact that large numbers are subtracted from each other when scoring onto the Monte-Carlo tallies is done) and there is a little increase in the PFZ for the ion energy source. But it seems that EIRENE

mimics NIMBUS quite well.

Surface fluxes, case (A) without molecules

Figure 6 compares the neutral particle influx and outflux to the vessel-walls (including targets) for case (A) (with no molecular re-emission). The vessel wall is stretched out and provides the abszissa in these graphs.

The neutral incoming flux in NIMBUS is quite well reproduced by EIRENE. Even the large neutral flux in the vicinity of the separatrix (large peaks) compares good. The reflected neutral seems to be different: the large peak values differ by a factor 2-3. But when the difference of in- and outgoing fluxes are plotted (bottom figure in 6) EIRENE provides clearly zero netflux on vessel and target surfaces. On the other hand, NIMBUS gives obviously a too large reflected flux. Due to the fact that there are no additional pumps or leaks included in the NIMBUS simulation the reason for this behaviuor is not clear.

The following table shows integrated values along the vessel and targets. Whereas in EIRENE the net outflux is almost zero (small fluctuations at the separatrix may come due to noise) NIMBUS shows the discrepancy. The NIMBUS surface tally WACOL (8) for the influx and for the outflux the sum of tally WACOL (1) and WACOL (2) for the backscattered and re-emitted particles respectively had been used (see for a list of NIMBUS surface tallies app. A.III).

The following table gives integrated values along the outstretched vessel-wall for the ingoing/outgoing fluxes:

	NIMBUS		EIRENE
incoming flux [$10^4 A$] NIMBUS tally WACOL (8)	7.7540	incoming flux [A] EIRENE tally POTAT	6.5737
outgoing flux [$10^4 A$] NIMBUS tally WACOL (1) +WACOL (2)	6.4219	outgoing flux [A] EIRENE tally PRFAAT	6.5866
net flux [$10^4 A$]	1.3321		-0.012919

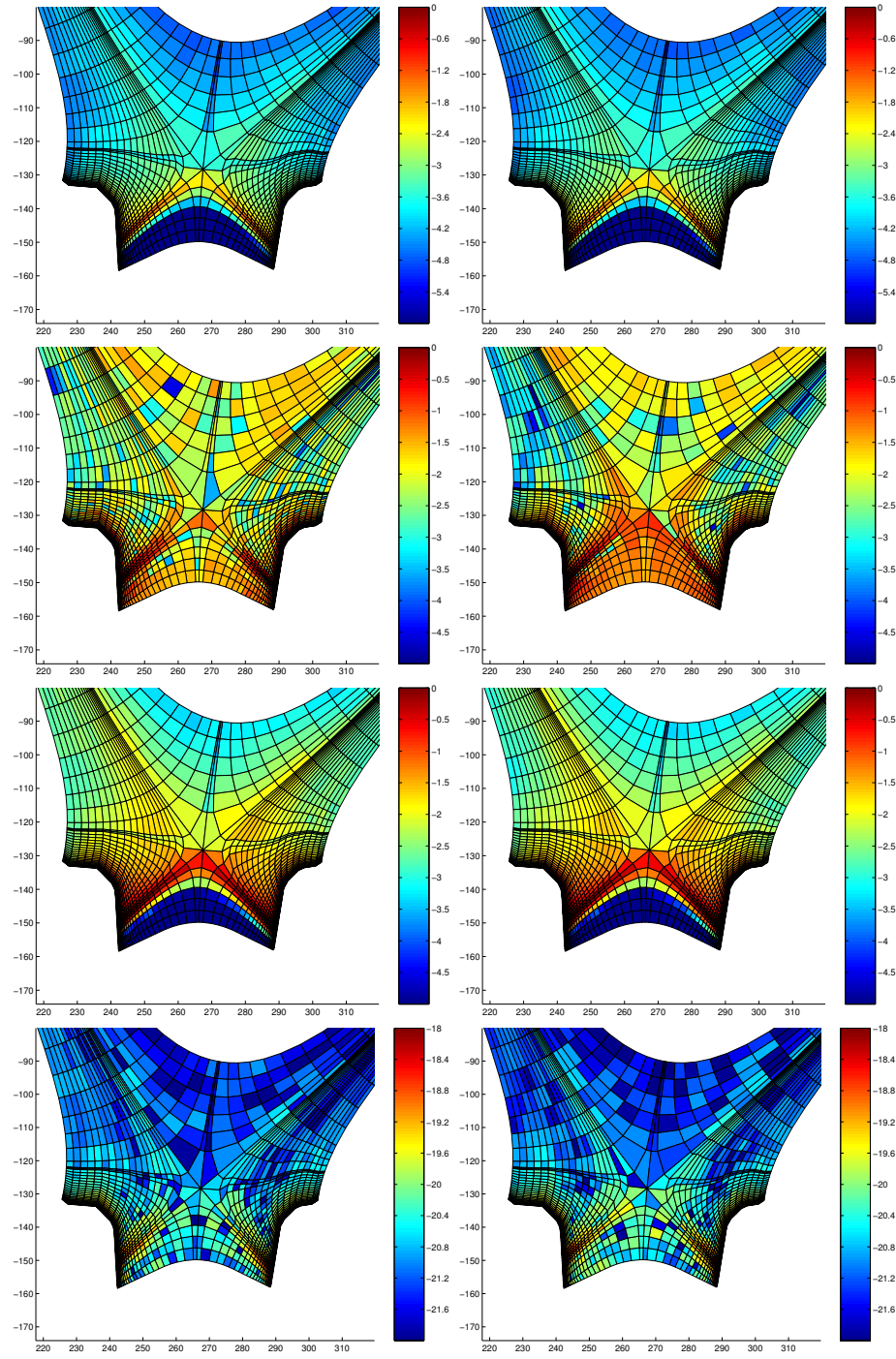


Figure 5: From top to bottom: source profiles for plasma particles $\log[A/\text{cm}^3]$, ion energy $\log[W/\text{cm}^3]$, electron energy $\log[W/\text{cm}^3]$ and parallel momentum $\log[g/\text{cm}^2\text{s}]$ (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, values given in $\log[\text{cm}^{-3}]$, identical colour scale, molecules switched off

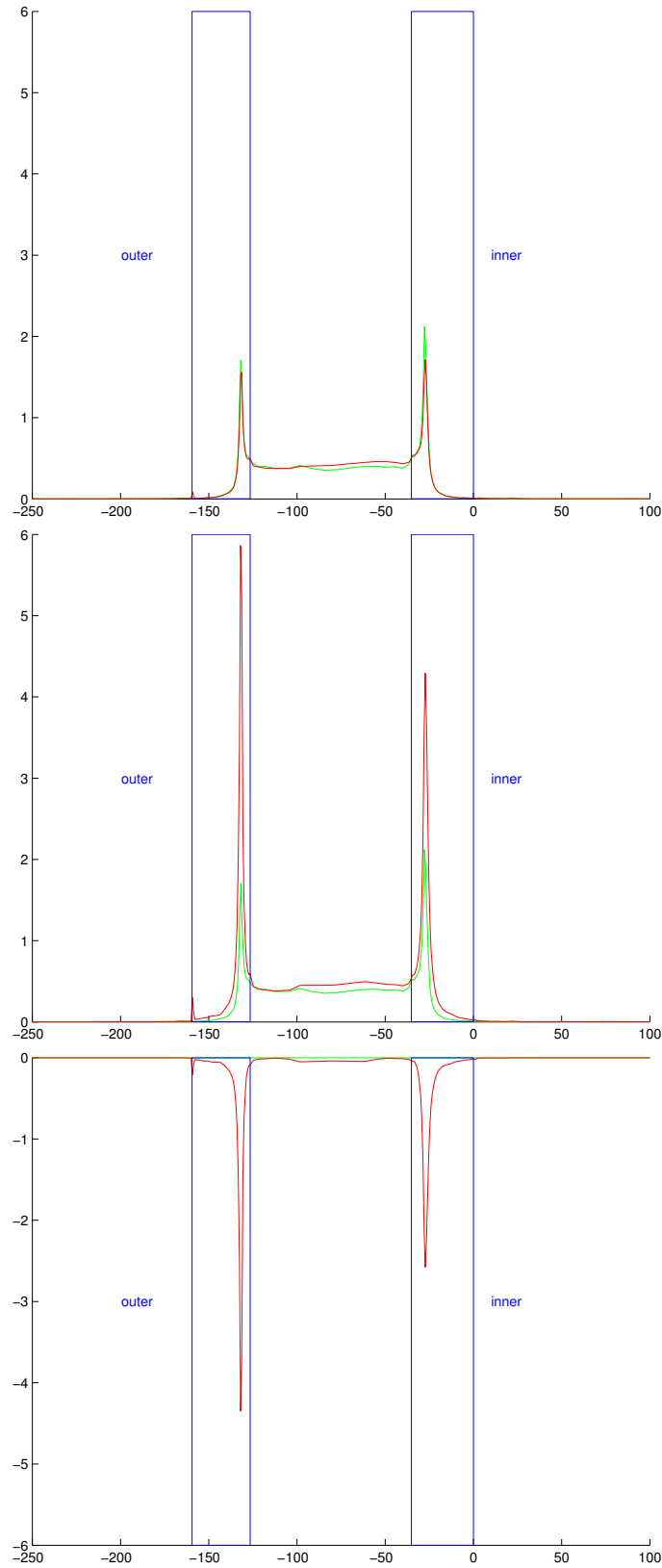


Figure 6: Top figure: Neutral particle flux in $[Acm^{-2}]$ hitting the stretched vessel-wall including targets (blue boxes). NIMBUS (red) and EIRENE (green). Middle: The same but for reflected neutral particle flux. Bottom: difference between incoming and outgoing neutral flux. (case A, no molecular re-emission)

IIIb.2 Molecules switched on

The plasma streams along the fieldlines onto the targets and recombines there. But normally not just atoms are recycled. Molecules are reemitted as well with energies according to the wall temperature and molecular ionisation and dissociation processes occur.

In EIRENE short living molecular ions like D_2^+ can be traced after a molecular ionisation process and the following processes are usually included in the simulation (the reference number according to Janev's book is given in brackets):

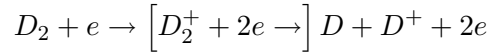
- dissociative ionisation (ref. 2.2.11):

$$D_2^+ + e \rightarrow D^+ + D^+$$
- dissociation (ref. 2.2.12):

$$D_2^+ + e \rightarrow D + D^+$$
- dissociative recombination (ref. 2.2.14):

$$D_2^+ + e \rightarrow D + D$$

The rate coefficient 2.2.11 is small compared to the other ones. The dissociation 2.2.12 becomes important for $T_e > 15 \text{ eV}$ and the dissoc. recombination 2.2.14 for $T_e < 10 \text{ eV}$. However, neither of the three processes is included in NIMBUS. Instead the reaction path of the molecular ionisation process which actually produces intermediate D_2^+ ions is reduced to:



where the molecular ionisation rate coefficient 2.2.9 is used for the combined process (assuming that T_e is high enough). For lower temperatures 2.2.14 provides a channel where two neutral atoms are produced but this is ignored within NIMBUS.

Thus, we end up with the following table of molecular processes which are actually used in NIMBUS. The inputfile of EIRENE has been changed accordingly:

- (fast) dissociation (ref. 2.2.5) $D_2 + e \rightarrow D[3 \text{ eV}] + D[3 \text{ eV}] + e$
 $E_{loss} = 10.4 \text{ eV}$
- mol. ionisation (ref. 2.2.9) $D_2 + e \rightarrow D[3 \text{ eV}] + D^+$
 $E_{loss} = 17.8 \text{ eV}$
- dissoc. ionisation (ref. 2.2.10) $D_2 + e \rightarrow D[0.3 \text{ eV}] + D^+ + 2e$
 $E_{loss} = 5.0 \text{ eV}$

The neutral energy of the atoms produced are given in brackets and E_{loss} is the electron energy loss for the process. Both, the molecular ionisation and the dissociative ionisation process (which is small compared to the former) are summed up and referred to as *total molecular ionisation* in the forthcoming text (because their

product species are the same, though they are produced with different energies). Molecular charge exchange or elastic processes for molecules are not used within the NIMBUS default set of reactions (although, they can be handled with some difficulty). Therefore in EIRENE these processes are also leaved out.

There is a different understanding in NIMBUS/EIRENE regarding the definition of the wall temperature. Whereas in EIRENE molecules are emitted with $1.5E_{wall}$ NIMBUS uses directly E_{wall} as the energy for the molecule. This has to be considered in EIRENE, i.e. when the wall temperature is set to 0.074 eV ($300\text{ }^{\circ}\text{C}$) (which is the default in NIMBUS), then the EIRENE inputfile must have a wall temperature $0.074/1.5 \approx 0.049\text{ eV}$.

Another difference between NIMBUS/EIRENE is the following: all molecular tallies in NIMBUS are counting per nuclei and not per molecule. That means for example that the molecular density tally `PROFM` is scaled as *nuclei per cm^3* rather as *molecules per cm^3* . That means when transferring molecular data from EIRENE to EDGE2D, the molecular density has to be multiplied by a factor of two (two nuclei for the molecule) in `linkeirene` before it is passed to routine `hmc`.

Global analysis

The following table shows data for cases (A) and (B) seen by routine `hmc`. Displayed values are integrations over the magnetic mesh leaving out the void regions:

w/molecules	case	NIMBUS	EIRENE
tot. plasma particle source [10^4 A]	(A) (B)	2.7772 6.6558	2.7636 6.3317
tot. ion energy src [10^5 W]	(A) (B)	3.5419 18.372	3.7106 7.9217
tot. el. energy src [10^5 W]	(A) (B)	8.7565 17.608	8.7072 16.757
tot. par. momentum src [10^5 gcm s^{-1}]	(A) (B)	6.2347 10.931	6.0051 10.165
vol. averaged atom dens. [10^{11} cm^{-3}]	(A) (B)	0.7244 0.4224	0.7689 0.4129
vol. averaged atom energy dens. [10^{11} eV/cm^3]	(A) (B)	4.4423 3.9341	4.6666 2.7124
vol. averaged atom par. velocity dens. [$10^{16}\text{ 1/cm}^2\text{ s}$]	(A) (B)	1.6056 0.7151	1.5835 0.7123
vol. averaged mol. dens. [10^{11} cm^{-3}]	(A) (B)	1.8420 0.7474	1.8439 0.6808
vol. averaged mol. energy dens. [10^{11} eV/cm^3]	(A) (B)	0.1365 0.0554	0.1366 0.0504
vol. averaged mol. par. velocity dens. [$10^{16}\text{ 1/cm}^2\text{ s}$]	(A) (B)	0.1053 4.9254	0.1150 4.8028

Again, as in the case where molecules had been switched off, there are differences for case (B) in the total ion energy source and atomic energy density.

The following table shows the global balances of the fluxes seen by `intfac`:

Global balances (<code>intfac</code>) w/molecules	case	NIMBUS	EIRENE
TOTSOU	(A)	2.8383	2.8283
[$10^4 A$]	(B)	6.9254	6.6047
HIONIZ	(A)	2.8344	2.8087
[$10^4 A$]	(B)	6.8697	6.5489
HESCPD	(A)	0.0140	0.0129
[$10^4 A$]	(B)	0.0557	0.0011
derived FNORM	(A)	0.9964	1.0008
	(B)	0.9969	1.0115

The total source strength for case (B) is differing.

Local analysis for case (A) including molecules

Figure 7 shows profiles for the neutral density, temperature and pressure for atoms (case A) in logarithmic scale in the cases where the molecules had been switched on again. The void space regions below the PFZ and main-chamber are not plotted in the NIMBUS plot.

Surface fluxes, case (A) including molecules

Figure 11 shows the EIRENE/NIMBUS comparison between incoming and outgoing neutral fluxes for neutrals (case A). Here, the molecular re-emission was switched on. The neutral vessel fluxes in the private void region (ie. between the targets) seem to be a little bit larger in the EIRENE case. Like for the pure atomic case with EIRENE the difference of ingoing to outgoing neutral flux is almost zero everywhere which is not the case for NIMBUS. Here again, the reflected neutral flux is too large to be comparable to the incoming flux especially at the strike-points. The question where this unphysical NIMBUS behaviour originates is yet unanswered.

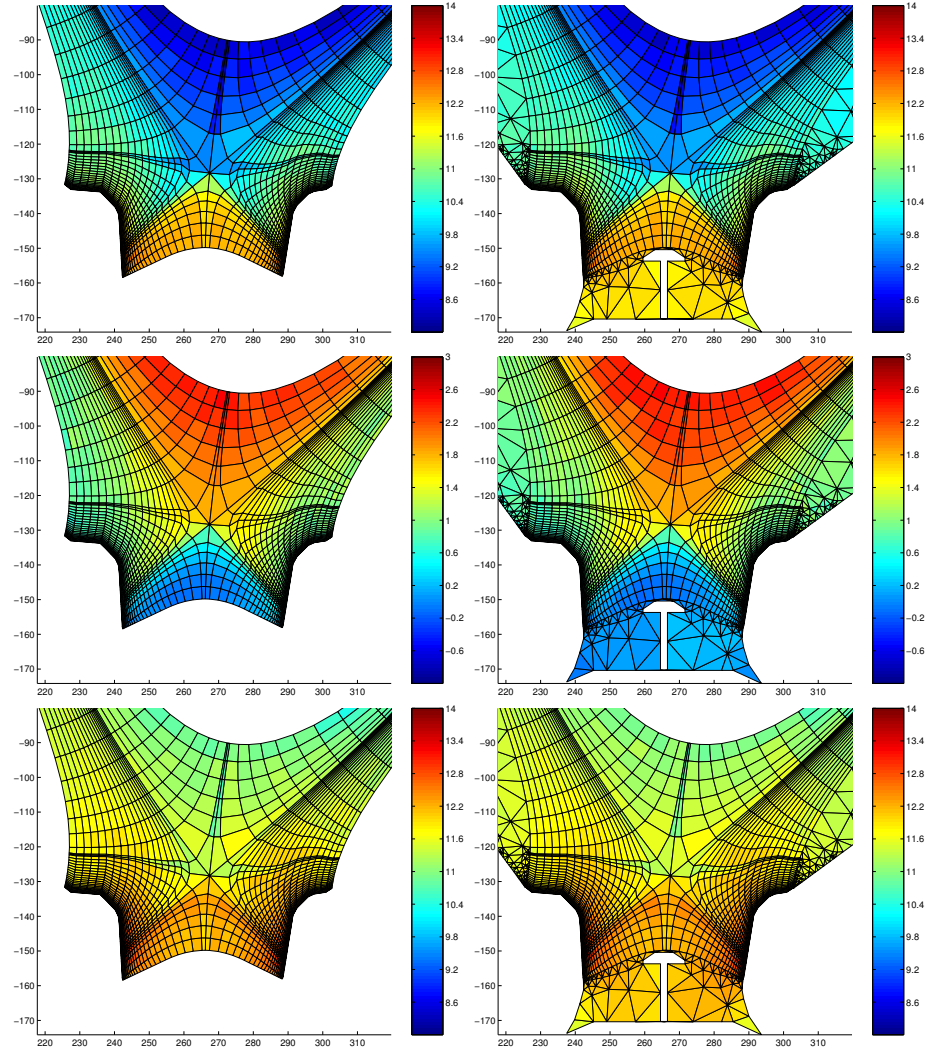


Figure 7: From top to bottom: profiles for density $\log[cm^{-3}]$, temperature $\log[eV]$ and pressure $\log[eV/cm^3]$ for D -atoms (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, identical colour scale.

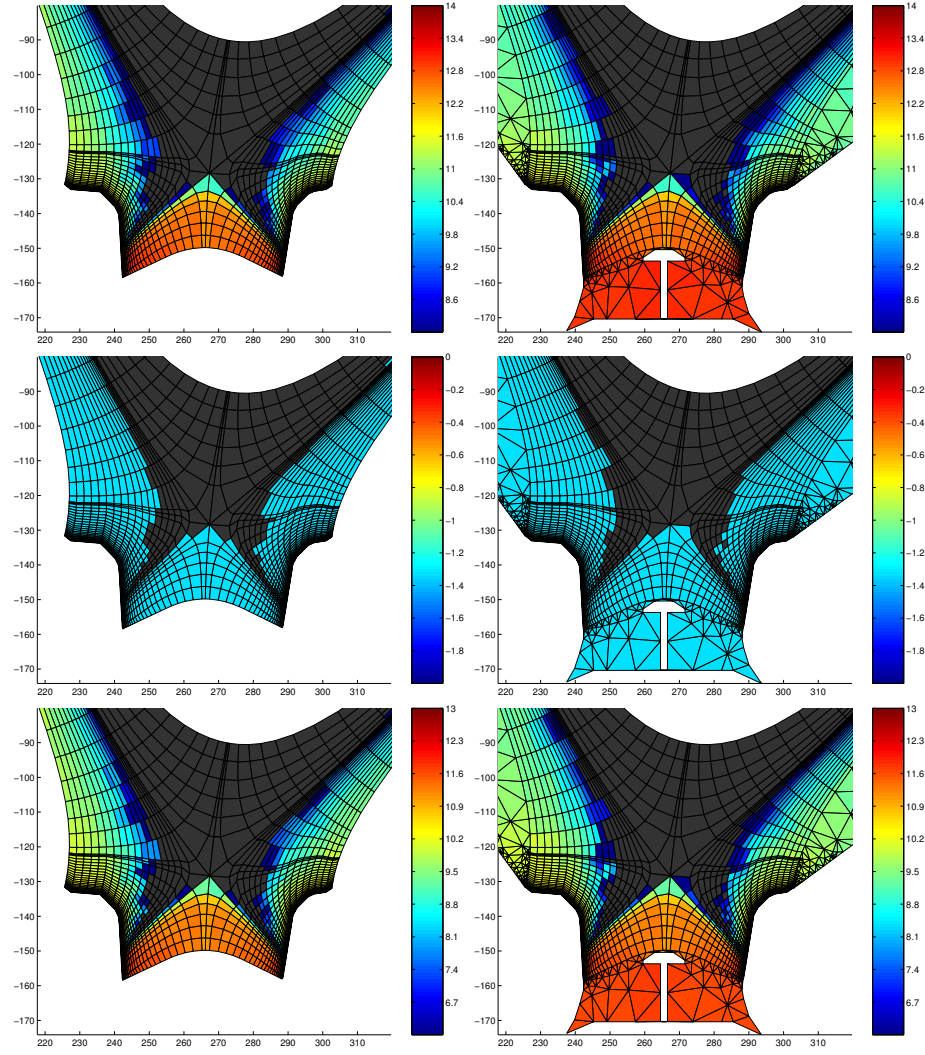


Figure 8: From top to bottom: profiles for density $\log[cm^{-3}]$, temperature $\log[eV]$ and pressure $\log[eV/cm^3]$ for D_2 -molecules (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, identical colour scale.

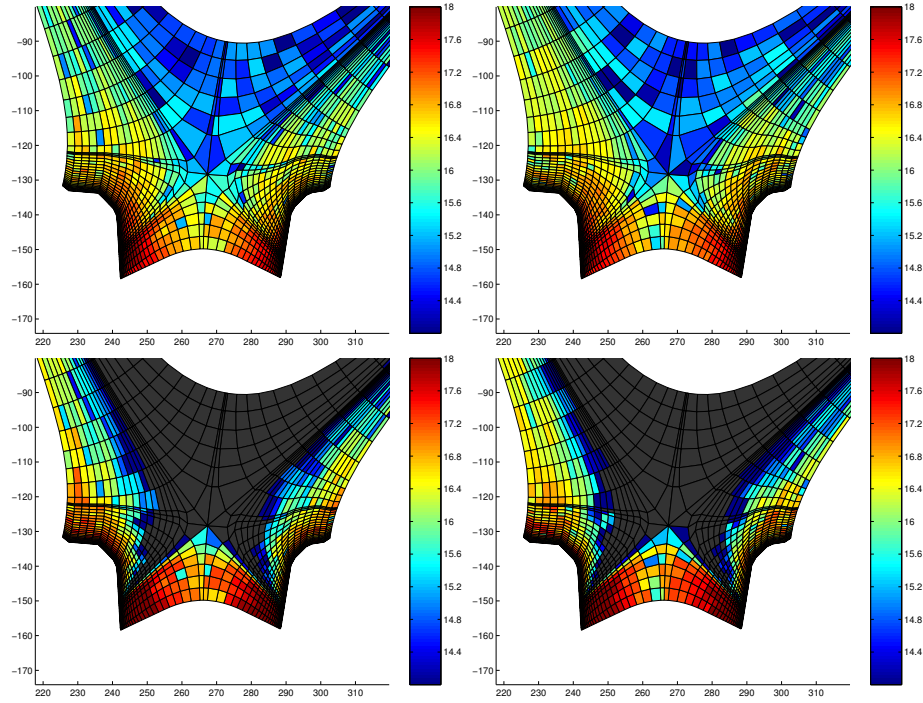


Figure 9: Parallel velocity profiles for atoms (top) and molecules (bottom) (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, values given in $\log[1/cm^2 s]$, identical colour scale. There is no data for the void region due to the fact that there is no magnetic field defined in the simulation.

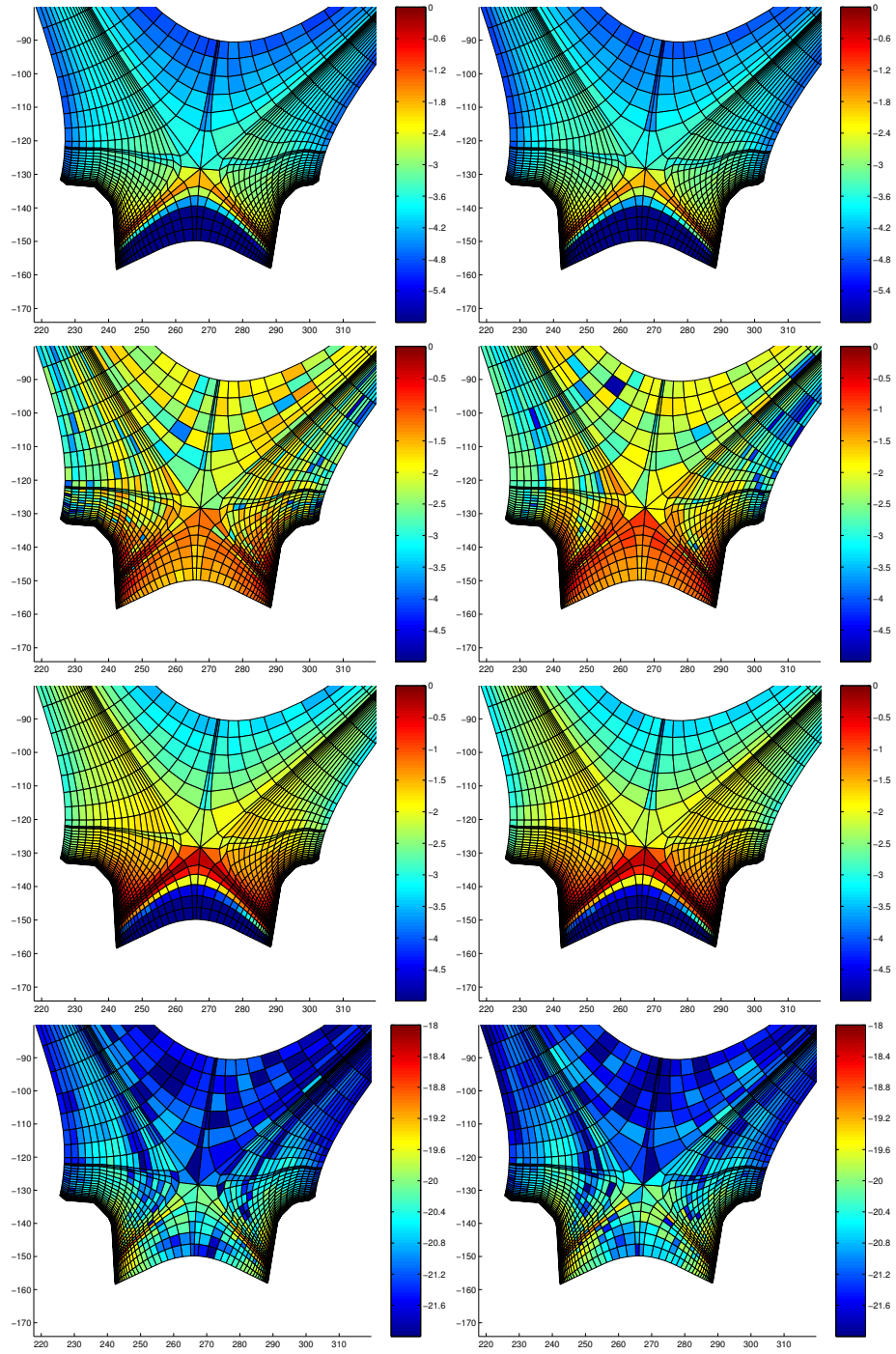


Figure 10: From top to bottom: source profiles for plasma particles $\log[A/cm^3]$, ion energy $\log[W/cm^3]$, electron energy $\log[W/cm^3]$ and parallel momentum $\log[g/cm^2s]$ (case A) as seen from routine hmc. left: NIMBUS, right: EIRENE, values given in $\log[cm^{-3}]$, identical colour scale.

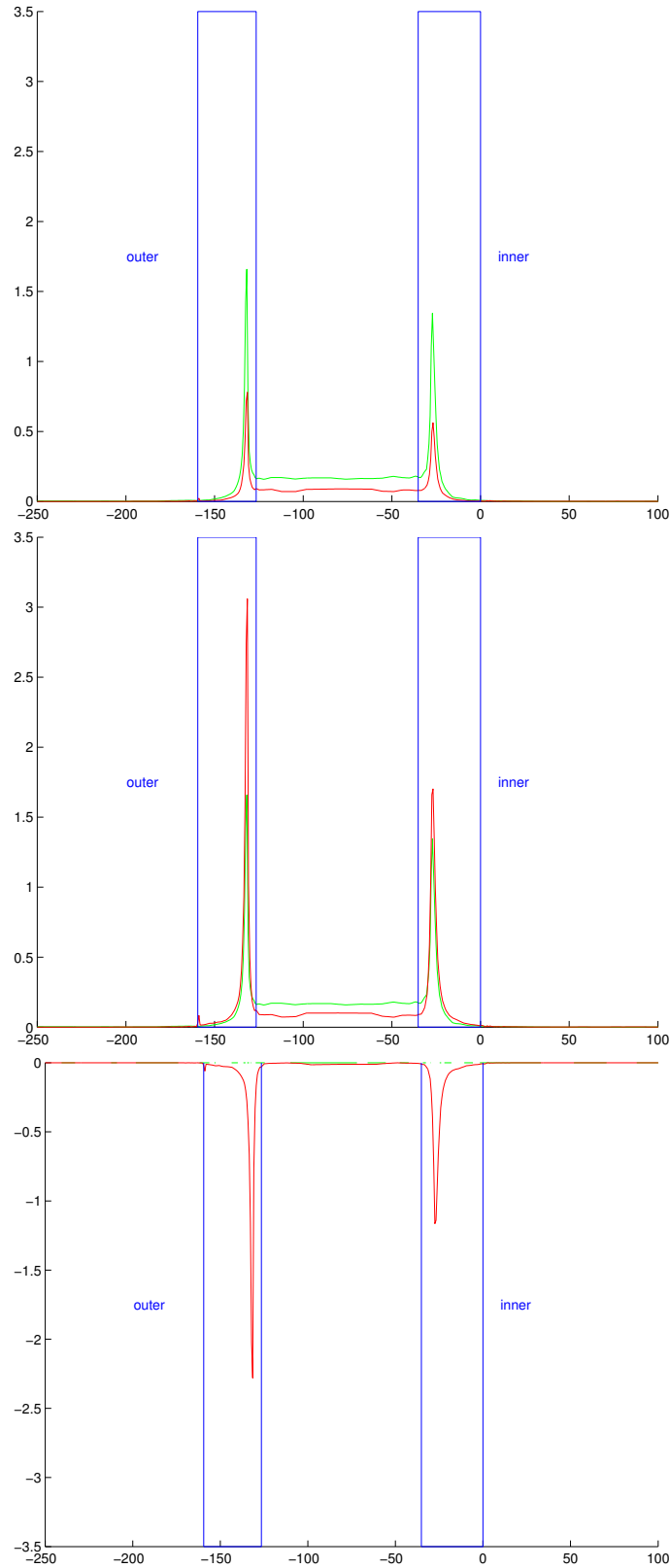


Figure 11: Top figure: Neutral particle flux in $[Acm^{-2}]$ hitting the stretched vessel-wall including targets (blue boxes). NIMBUS (red) and EIRENE (green). Middle: The same but for reflected neutral particle flux. Bottom: difference between incoming and outgoing neutral flux. (case A, molecular re-emission)

IIIb.3 Carbon impurities

A special JET case has been selected to compare neutral carbon impurity transport:

(D): swiesen 53140 jun2106 seq#1

In each of the simulated cases 20000 particle histories had been followed.

only physical sputtering (SPUTER93):

The following table gives global results for case (D) when only physical sputtering is switched on (SPUTER93). Whereas in NIMBUS no angular dependence on the impact angle is accounted for, results from both cases (with and without angular dependence) with EIRENE are shown:

	NIMBUS	EIRENE w/o ang.dep	EIRENE w/ ang.dep
<i>C</i> plasma source [$10^{-6} A$]	1.672	1.498	2.004
<i>C</i> neutral density [$10^7 cm^{-3}$]	3.029	2.547	3.756
<i>C</i> neutral energy dens. [$10^8 eV cm^{-3}$]	3.415	4.521	6.552

physical & chemical sputtering (Haasz):

In addition to the physical sputtering model the Haasz-model for chemical sputtering is switched on for the results in the following table:

	NIMBUS (Haasz)	EIRENE (Haasz-Davis)
<i>C</i> plasma source [$10^{-5} A$]	1.0330	1.0421
<i>C</i> neutral density [$10^8 cm^{-3}$]	6.1501	5.8070
<i>C</i> neutral energy dens. [$10^8 eV cm^{-3}$]	4.667	4.7628

(here, the angular dependence of the physical sputtering yield is switched off).

Despite the fact, that EIRENE uses a slightly modified version of the Haasz-model for $E_0 < 10 eV$ the results are matching quite well. Figures 12 to 14 show the 2D profiles of the carbon plasma source, the neutral particle and energy density for case (D).

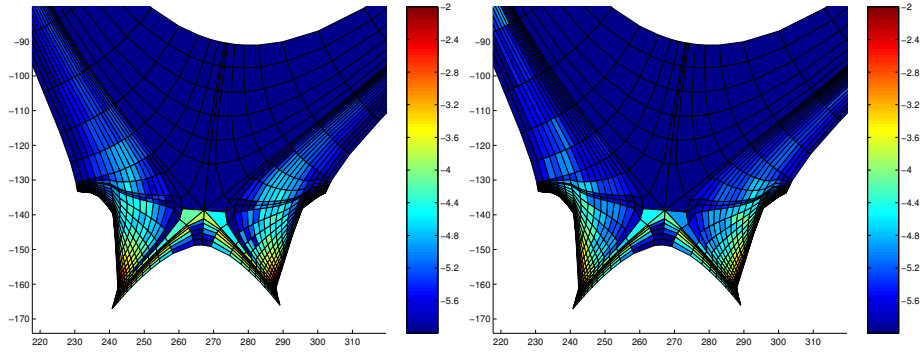


Figure 12: Carbon plasma particle source in $\log[A/cm^3]$, left: NIMBUS, right: EIRENE, same color scale.

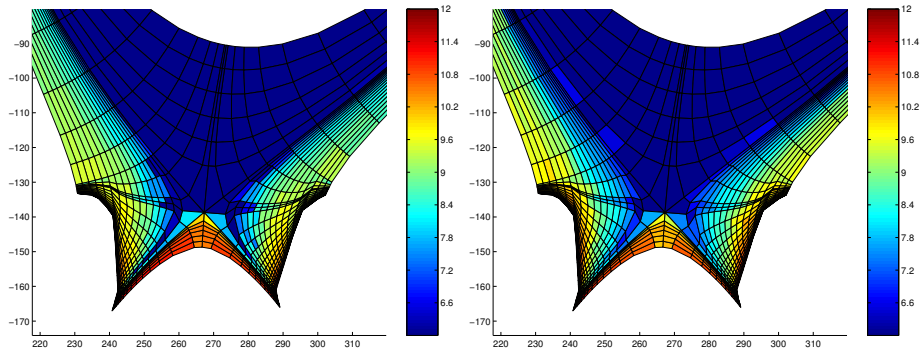


Figure 13: Carbon neutral density in $\log[cm^{-3}]$, left: NIMBUS, right: EIRENE, same color scale.

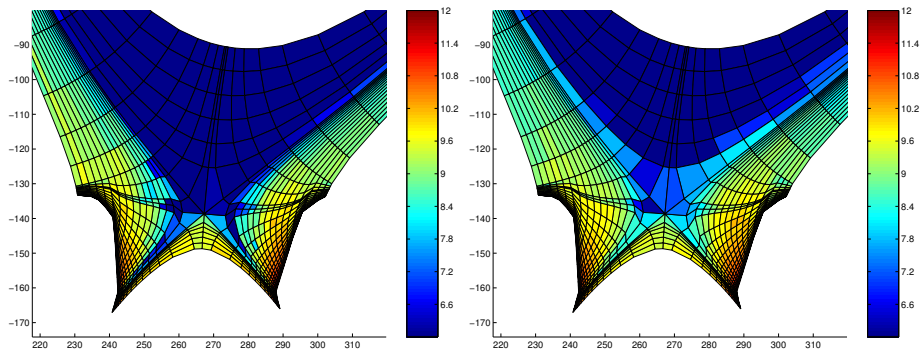


Figure 14: Carbon neutral energy density in $\log[eV/cm^3]$, left: NIMBUS, right: EIRENE, same color scale.

IV Completion of the EDGE2D/EIRENE code package

As a final step before running coupled EDGE2D/EIRENE simulations the EDGE2D specific source-term re-evaluation scheme has to be applied also for the EIRENE rates. Thus the EDGE2D/EIRENE interface has to transfer the used EIRENE rate coefficients and derive derivatives numerically respective to temperature (the density dependence is not regarded within this linearisation scheme).

Finally, when this would have been done, EDGE2D and EIRENE can be used iteratively. Coupled EDGE2D/NIMBUS and EDGE2D/EIRENE runs for specific cases are then needed to be examined and compared. Remaining differences have to be documented.

IVa Source term re-evaluation in EDGE2D including EIRENE atomic and molecular rates

Additional EDGE2D routines `xss_new` (and `zxss_new` for impurities) have been implemented and are called from `hmc` (instead calling `xss/verb+zxss+`) to derive rate coefficients and their derivatives and these are then put into the right arrays afterwards. Now, the EIRENE A&M database can be used, if the EIRENE input file `eirene.input` does exist (ie. right after an EIRENE call). The input-file provides information about the actual rate-coefficients which had been used in the neutral simulation and this is passed to EDGE2D.

For doing this, a Fortran90 module was implemented to read out the EIRENE input-file and access the A&M database which is used by EIRENE (AMJUEL, H2VIBR etc, see [5]) directly. Thus, the full EIRENE capabilities for atomic cross-sections and rate-coefficients can be used also on the EDGE2D side (as long as EDGE2D is capable to handle those).

If the EIRENE input-file `eirene.input` is not existing, then the default A&M database (via `xss / zxss`, see sect. IIIa.2) for volumetric processes is taken.

The source-terms in the plasma fluid equations due to neutral interactions are linearized in EDGE2D to provide a method for not calling NIMBUS (or EIRENE) every time step which would lead to much higher computation times. This linearisation is realised by deriving the derivatives of the rate coefficients $\frac{d}{dT}\langle\sigma v\rangle$ with respect to the temperature.

The derivative operator d/dT is treated numerically in EDGE2D via a 2nd order difference scheme:

$$\frac{d}{dT}\langle\sigma v\rangle = \frac{\langle\sigma v\rangle [T(1+\xi)] - \langle\sigma v\rangle [T(1-\xi)]}{2T\xi} \quad (45)$$

where ξ is a small parameter ($\xi = 0.01$ by default in NIMBUS).

Some rate coefficients $\langle\sigma v\rangle$ (like the ionisation and recombination coefficients) are represented as fits dependent on T_e and/or n_e . Some rate coefficients (like for charge-exchange) are calculated directly from the cross section σ :

$$\langle\sigma v_{rel}\rangle = \int_0^\infty f(v_b)\sigma(v)(v - v_b) dv_b \quad (46)$$

where $f(v_b)$ is the background velocity distribution function (target particles, e.g. an ion or electron) and v is the projectile velocity (of the neutral). This formula is evaluated in NIMBUS especially for the charge exchange rate where the integral over the velocity distribution of the background ions is calculated numerically which makes the actual rate coefficient during a test-particle flight indeed dependent on the neutral velocity. In EIRENE that integral is pre-evaluated and a double-fit procedure dependent in the ion temperature and the relative velocity of the neutral projectile is applied.

The following table compares the source-term re-evaluation (globally integrated rate-coefficients and their derivatives over the whole magnetic-mesh grid) between EIRENE and NIMBUS. Parameters like neutral temperature, energy etc are local and taken from MC-simulation.

ADAS rates have been used to derive the ionisation rates (the recombination source values $\int n_i n_e \langle\sigma v\rangle_{rec}$ are not shown because they are already presented in sect. IIIa.5).:

no molecules	NIMBUS	EIRENE
total ionisation [$10^{-3}A$] (ADAS 96) $\int n_H n_e \langle\sigma v\rangle_{ion} d^3r$		
case (A)	0.9938	0.9890
case (B)	2.7533	2.6339
total ionisation derivative [$10^8 A/eV$] $\int n_H n_e \frac{d}{dT_e} \langle\sigma v\rangle_{ion} d^3r$		
case (A)	1.4534	1.4748
case (B)	2.8756	2.8664
total charge exchange [$10^{-3}A$] (Janev 3.1.8) $\int n_H n_i \langle\sigma v\rangle_{CX} d^3r$		
case (A)	7.1836	7.3985
case (B)	10.456	10.014
total charge exch. derivative [$10^9 A/eV$] $\int n_H n_i \frac{d}{dT_i} \langle\sigma v\rangle_{CX} d^3r$		
case (A)	2.1850	2.2600
case (B)	1.5709	1.4920

Exactly the same NIMBUS routine CXSIG was implemented into EIRENE to derive almost equal results for the CX rates and its derivatives (but there is in fact

no big difference when using the default EIRENE's default CX routine, not shown here). Input for `CXSIG` is the neutral energy derived from neutral velocity components

$$E_H = \max \left(\frac{1}{2} A_H v_H^2 \cdot 1.04394 \cdot 10^{-12}, 0.001 \right) \quad (47)$$

where v_H is derived from $v_x = \text{VXDENA}/\text{PDENA}$, $v_y = \text{VYDENA}/\text{PDENA}$ etc.. and $1.04394 \cdot 10^{-12} \approx 1.667 \cdot 10^{-24} / 1.602 \cdot 10^{-12}$

Routine `CXSIG` then transforms the neutral energy back again to velocity components:

$$v_{H,(x,y,z)} = \sqrt{\frac{E_H}{A_H}} 1.3841 \cdot 10^6 e_{(x,y,z)} \quad (48)$$

with directions $e_x = v_x/|v|$ etc..

The ion-background velocity is derived from its local Mach-number M_i

$$v_i = \sqrt{\frac{T_e + T_i}{A_i}} 10^6 \cdot M_i \quad (49)$$

where $10^6 \approx 0.9787e6 = \sqrt{1 \text{ amu}/1 \text{ eV}}$.

The following table shows the results with molecules switched on:

with molecules	NIMBUS	EIRENE
total at. ionisation [$10^{-3}A$] (ADAS96) $\int n_H n_e \langle \sigma v \rangle_{ion} d^3r$		
case (A)	0.7785	0.7790
case (B)	2.0577	2.0103
total at. ionisation derivative [$10^8 A/eV$] $\int n_H n_e \frac{d}{dT_e} \langle \sigma v \rangle_{ion} d^3r$		
case (A)	1.0235	1.0387
case (B)	2.2166	2.2223
total at. charge exchange [$10^{-3}A$] (Janev 3.1.8) $\int n_H n_i \langle \sigma v \rangle_{CX} d^3r$		
case (A)	3.1160	3.2072
case (B)	6.0823	5.9806
total at. charge exch. derivative [$10^8 A/eV$] $\int n_H n_i \frac{d}{dT_i} \langle \sigma v \rangle_{CX} d^3r$		
case (A)	3.2708	3.5364
case (B)	5.8686	5.7743
total mol. ionisation [$10^{-3}A$] (Janev 2.2.9 + 2.2.10) $\int n_{H_2} n_e \langle \sigma v \rangle_{mol.ion} d^3r$		
case (A)	0.2164	0.2116
case (B)	0.6986	0.6237
total mol. ionisation derivative [$10^8 A/eV$] $\int n_{H_2} n_e \frac{d}{dT_e} \langle \sigma v \rangle_{mol.ion} d^3r$		
case (A)	0.4709	0.4648
case (B)	0.8927	0.8625
total mol. dissociation [$10^{-3}A$] (Janev 2.2.5) $\int n_{H_2} n_e \langle \sigma v \rangle_{mol.diss} d^3r$		
case (A)	0.4904	0.4844
case (B)	0.9470	0.9071
total mol. dissociaton derivative [$10^8 A/eV$] $\int n_{H_2} n_e \frac{d}{dT_e} \langle \sigma v \rangle_{mol.diss} d^3r$		
case (A)	1.4212	1.4197
case (B)	2.0805	1.9588

IVb Coupled EDGE2D/EIRENE runs

IVb.1 Molecules switched off, case (A)

On the following pages plots of time-traces are shown for plasma density, temperatures and fluxes (and some other characteristics) at different places (ie. outer mid-plane, inner and outer target, core).

For the steady-state simulation of case (A) where the molecular emission had been switched off, EDGE2D/NIMBUS was run up to convergence of the time-traces (here until 50 *ms* internal fluid-code time). From than point on, EDGE2D/EIRENE was restarted for some additional fluid time (additional 5 *ms*) and no remarkable changes in the time-traces should be visible.

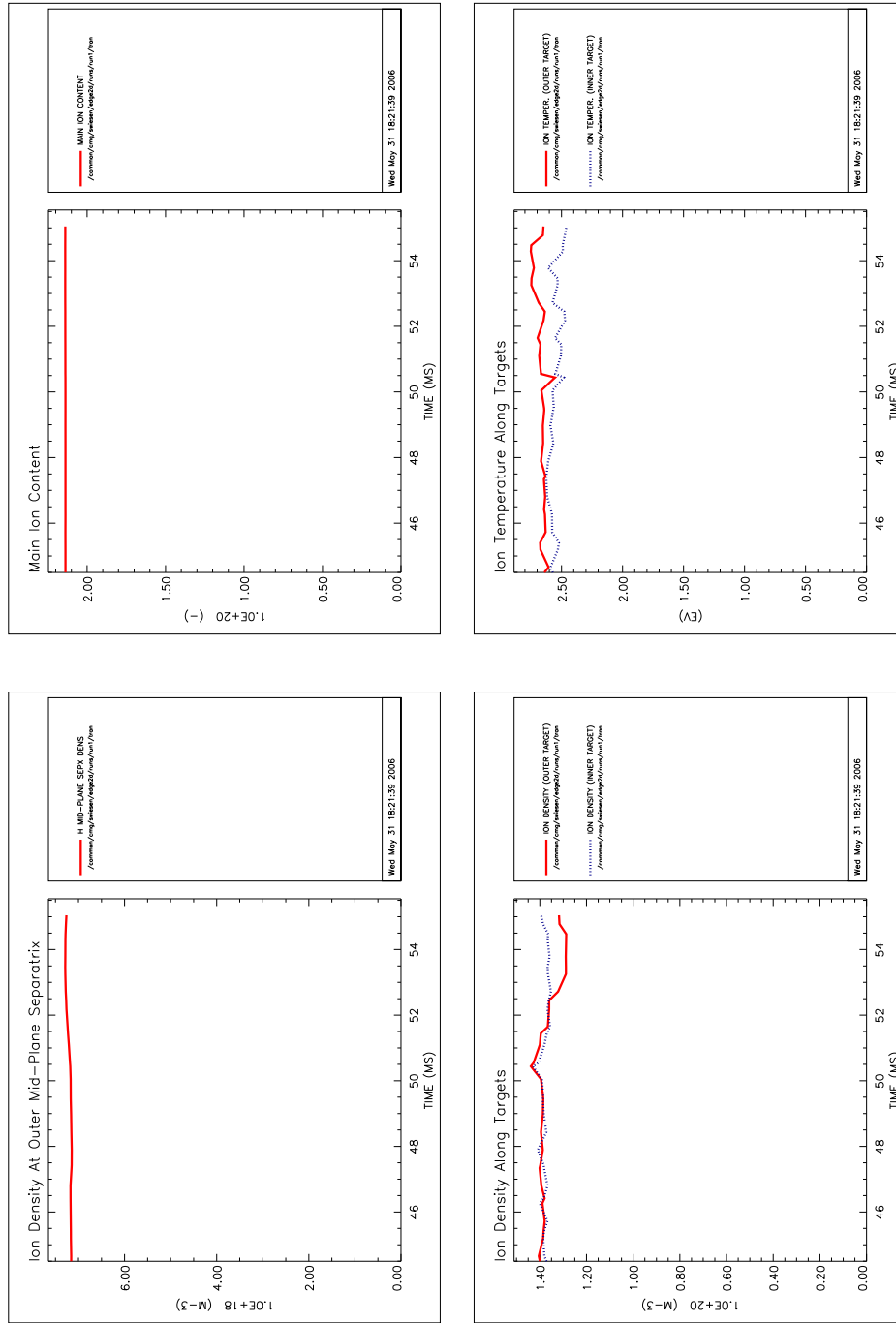
Indeed, in all time traces there is no big change when switching to EIRENE. But some small differences are obvious. There is an asymmetry between the inner and the outer targets. As small differences in the the post-collision routines could be singled out (ie. the NIMBUS version of the CX post-collision routine was implemented in EIRENE as well) and all other processes and reaction rates are the same the difference must originate from the geometry.

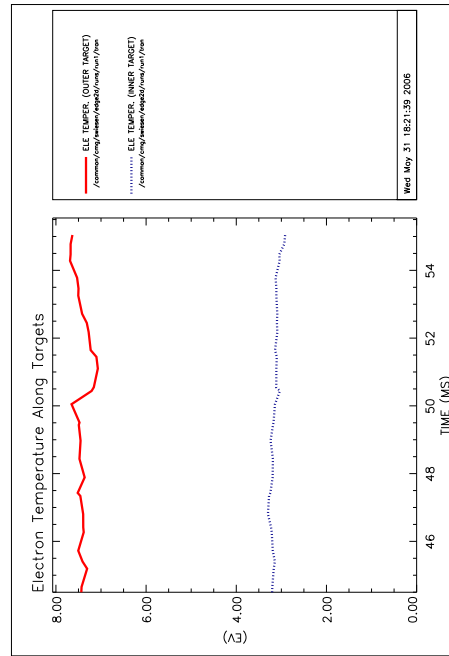
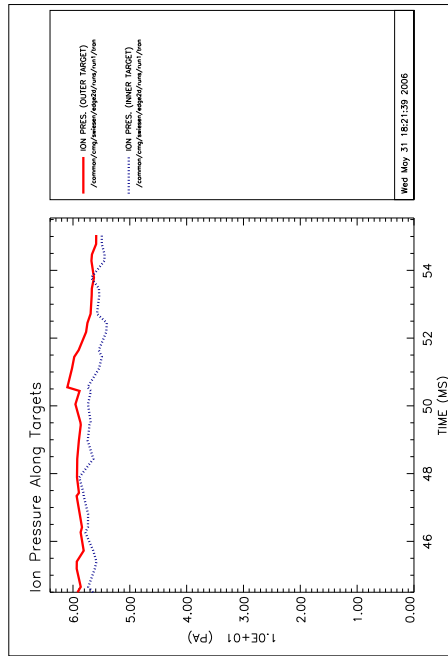
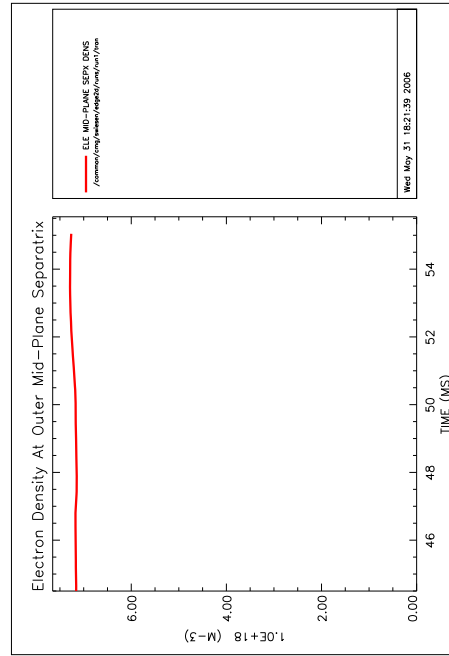
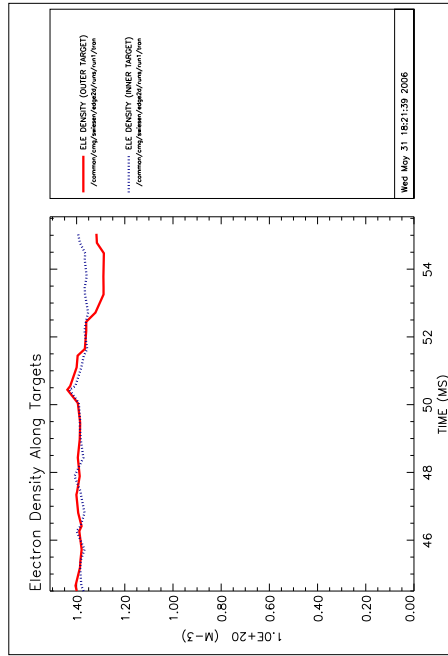
As it was shown in section Ib.1 the 2D cell-volumes (measured in cm^2) were at least for the magnetic mesh grid the same. But one striking difference between NIMBUS and EIRENE is the fact that NIMBUS is tracking its neutral test-particles within a cylindrical geometry rather than within a real toroidal geometry. In fact, to care for toroidal geometry `intfac` scales the cell-values for density, plasma sources etc in a post-process via correction factors R/R_0 .

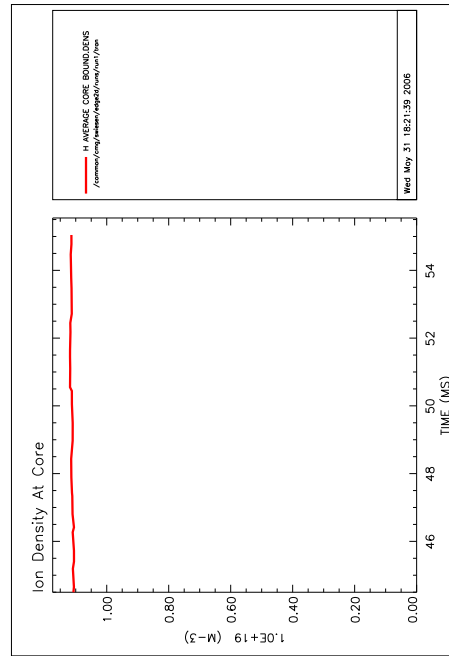
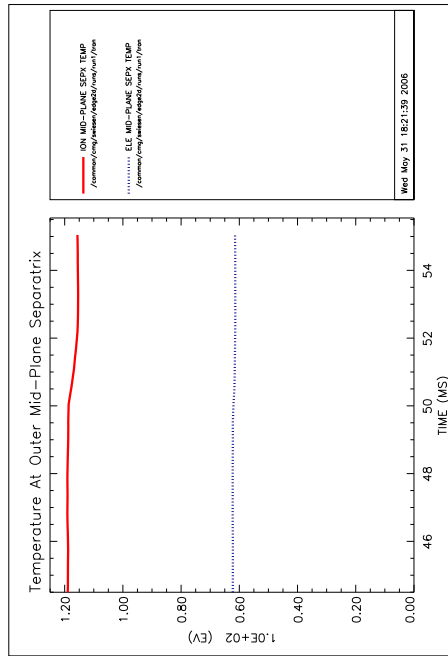
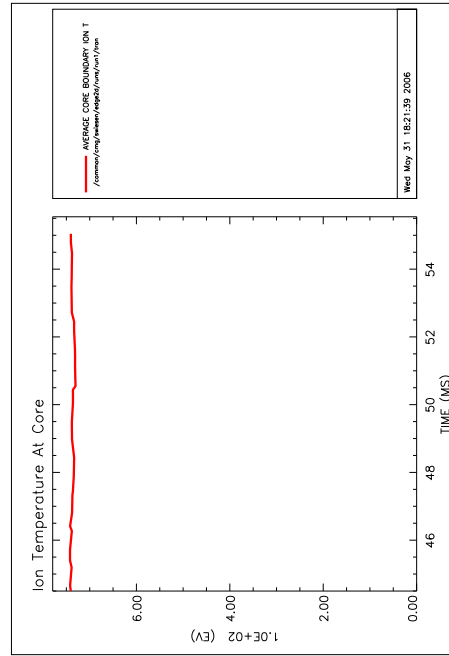
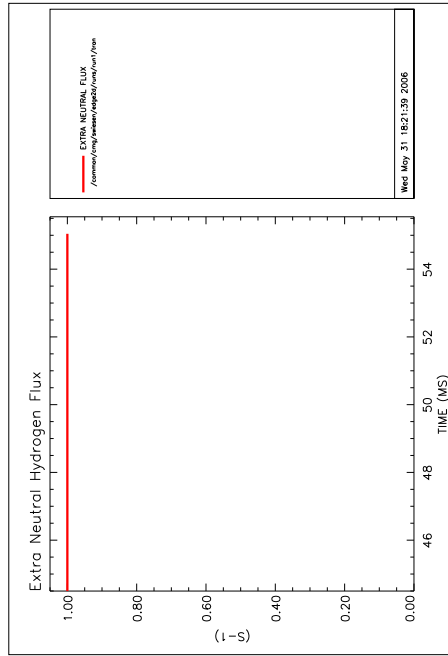
EIRENE calculates real toroidal geometry and no re-scaling with R/R_0 is done and it is thought to be that this is the reason for the inner-outer asymmetry.

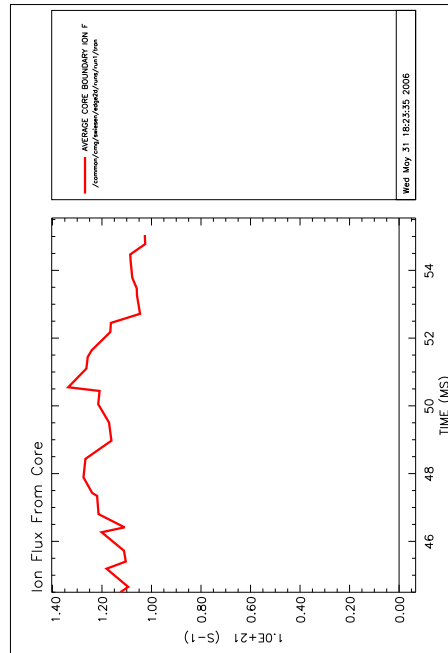
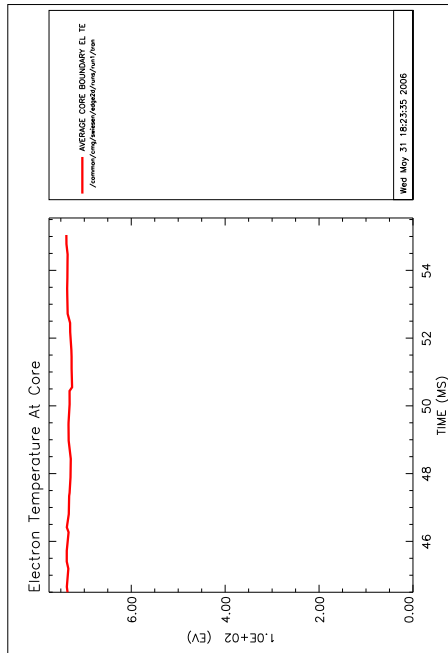
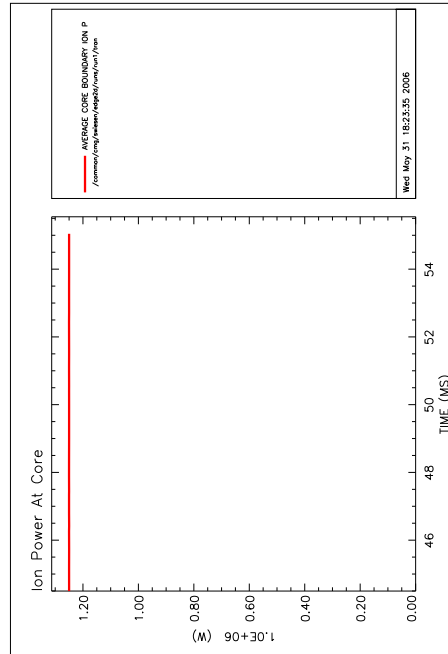
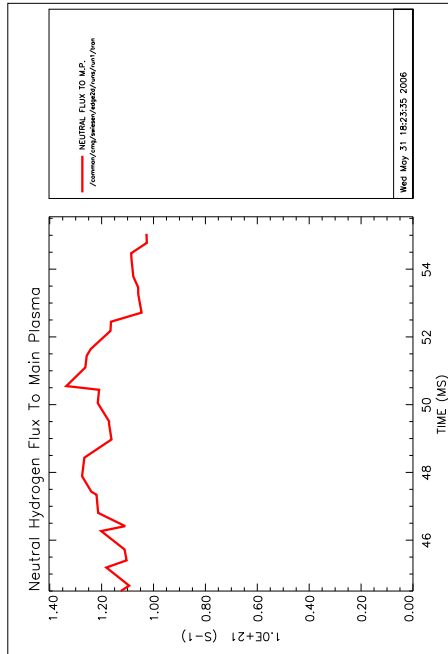
For further investigation EIRENE could mimic the cylindrical approximation like NIMBUS but this is not foreseen within this project.

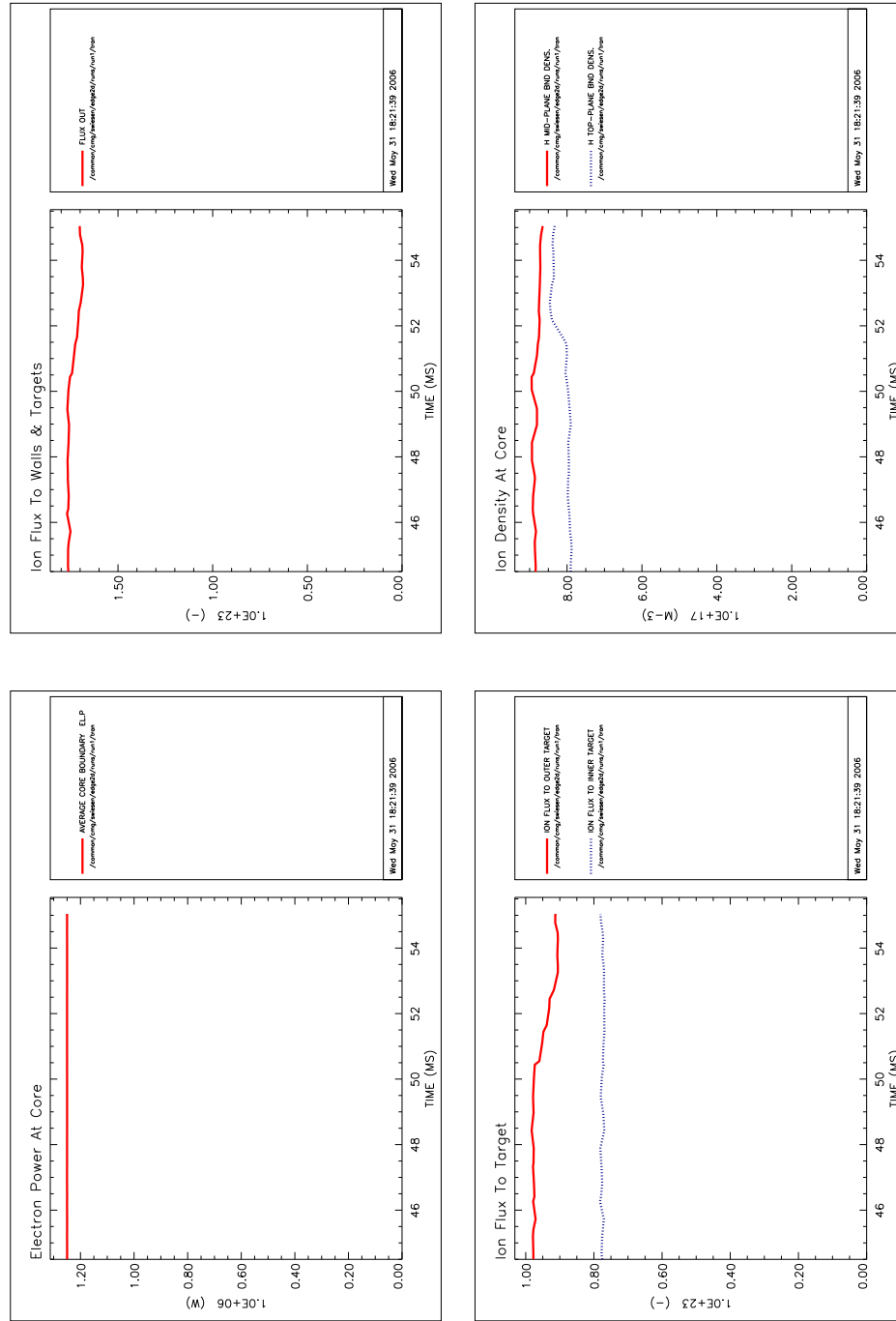
On the other hand the overall time-traces also for densities and temperatures etc at the outer-midplane or at the core are continuing the NIMBUS time-traces very well.









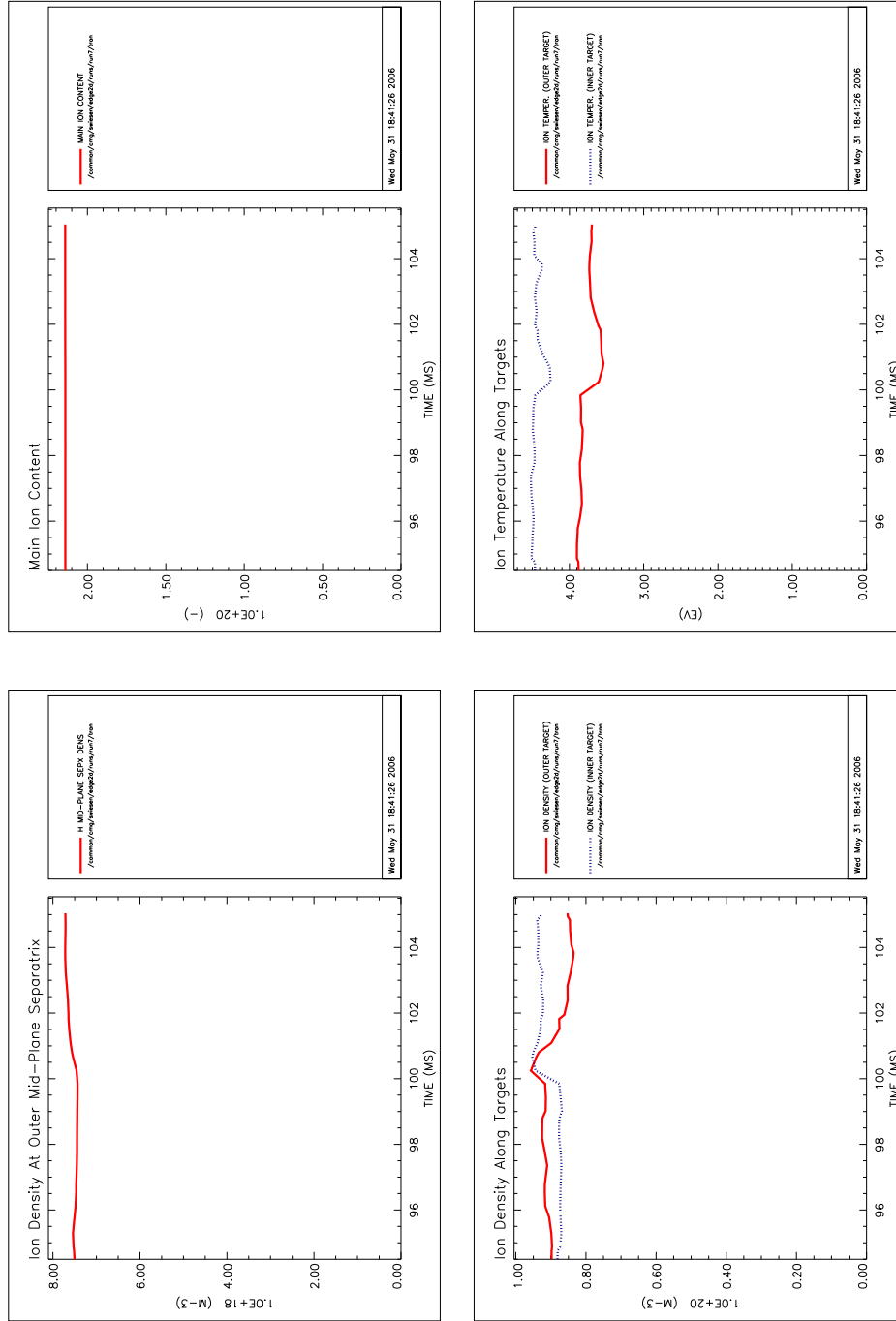


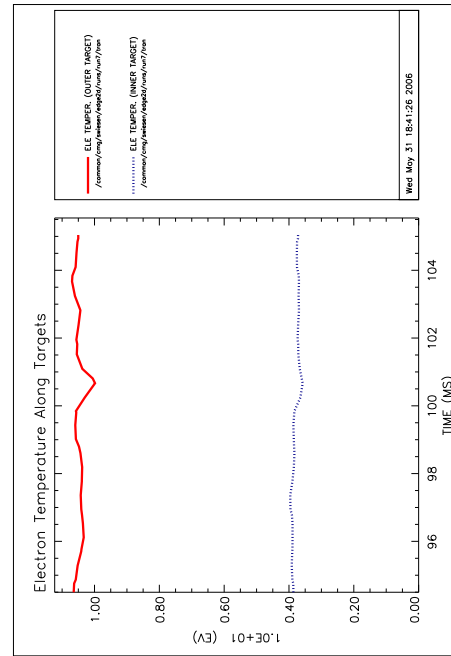
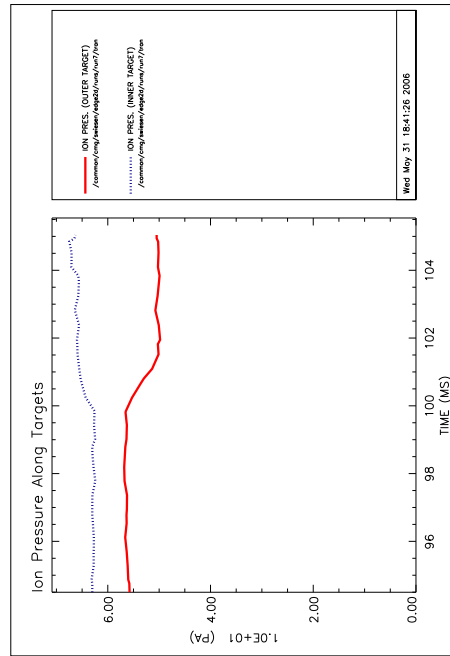
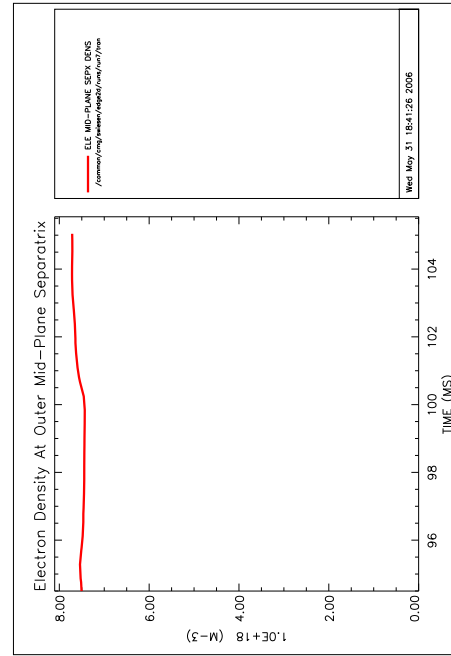
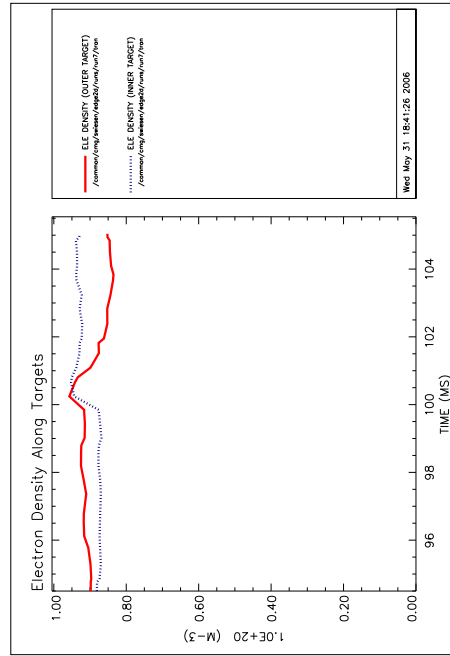
IVb.2 Molecules switched on, case (A)

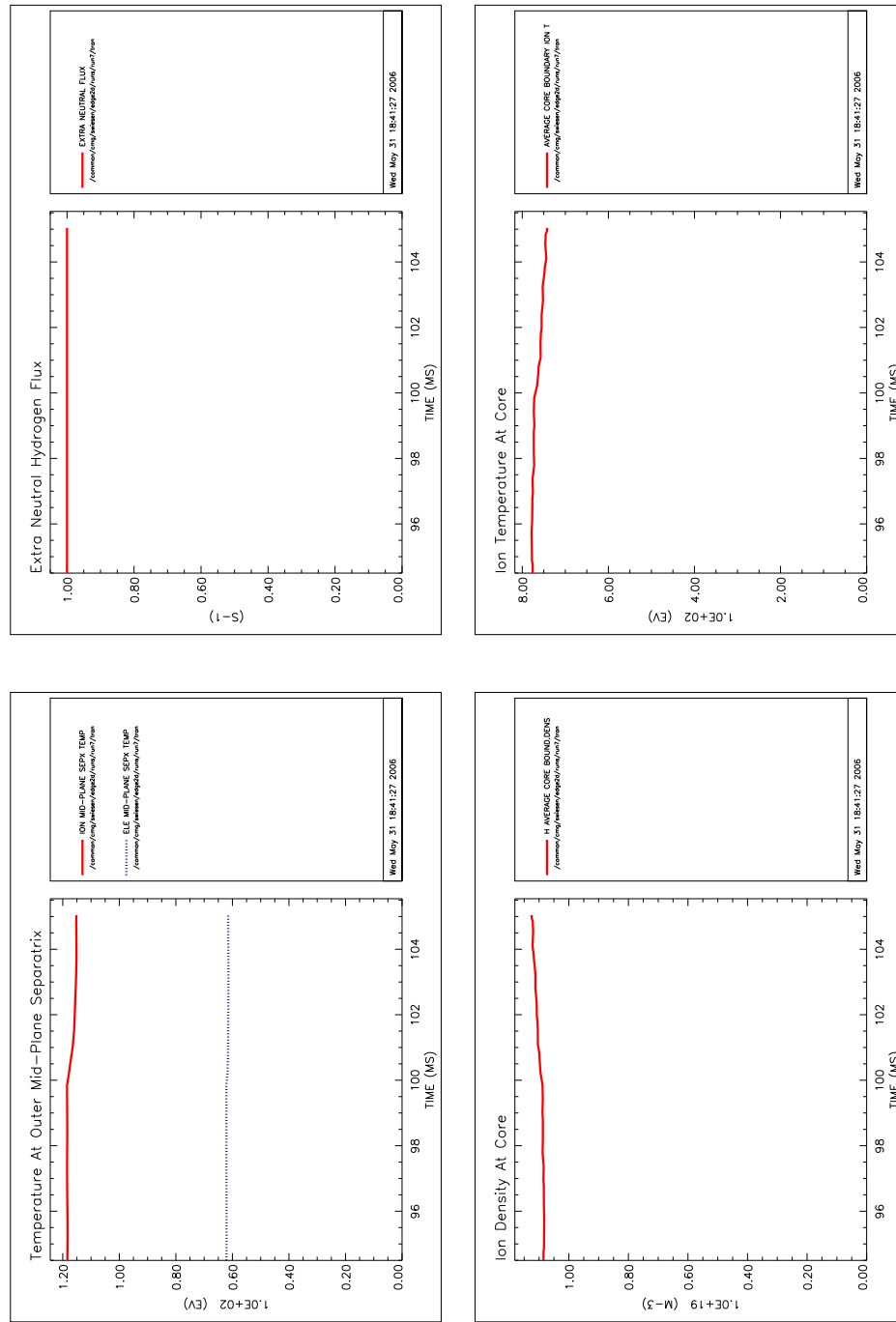
The same time-trace check when switching from EDGE2D / NIMBUS to EDGE2D / EIRENE is done for case (A) including the re-emission of molecules (see the next five pages). Here, it was switched to EDGE2D / EIRENE for additional 5 *ms* at 100 *ms* internal fluid-time after EDGE2D/NIMBUS was run until convergence.

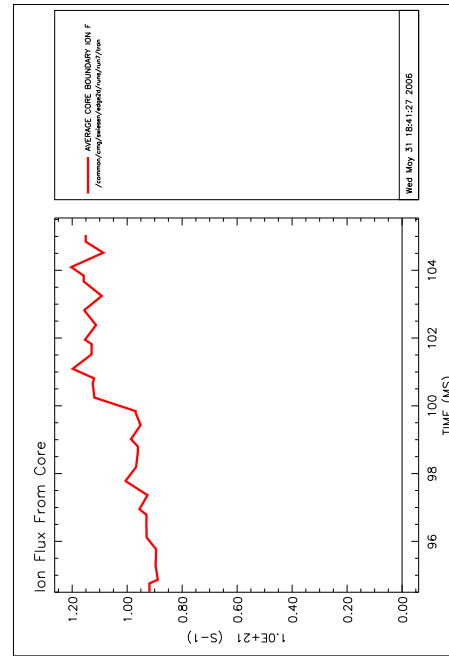
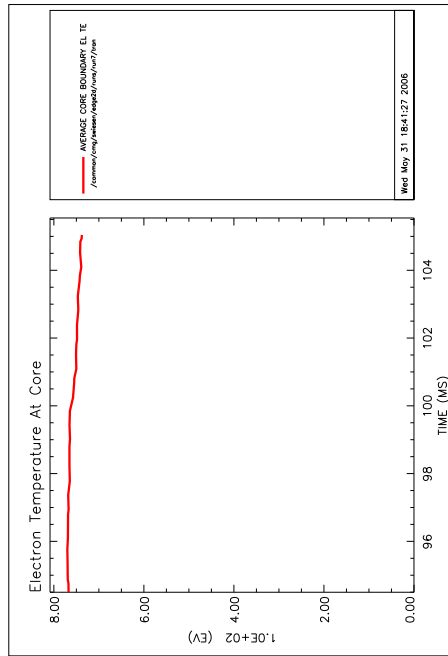
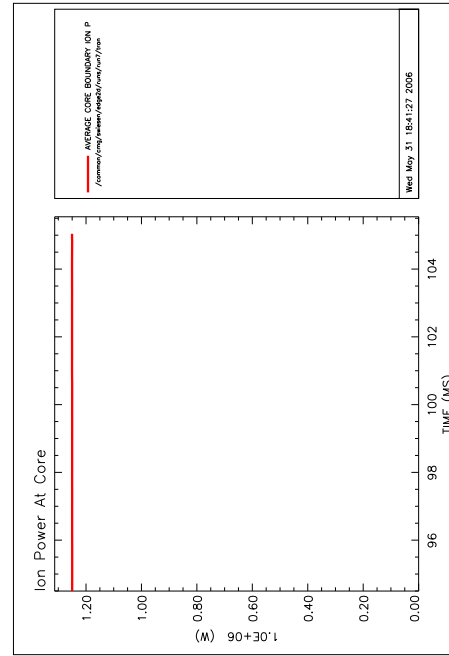
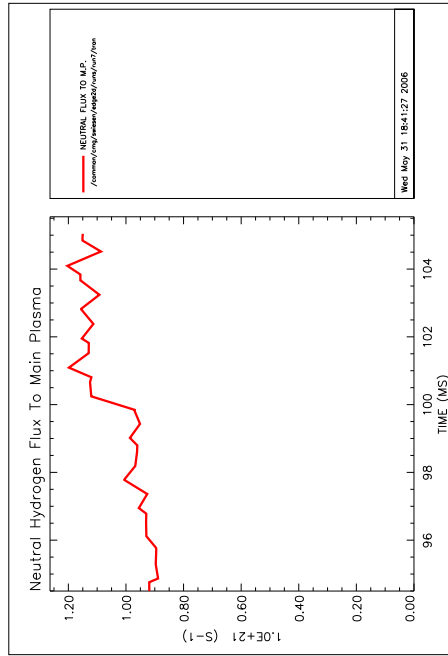
Again, asymmetries between inner and outer targets can be found for the time-traces especially for the plasma density and therefore also for the pressure. It is suggested that they originate for the same reasons as in the atom-only case (toroidal vs. cylindrical approximation). On the other hand, temperature traces at the targets are looking very well.

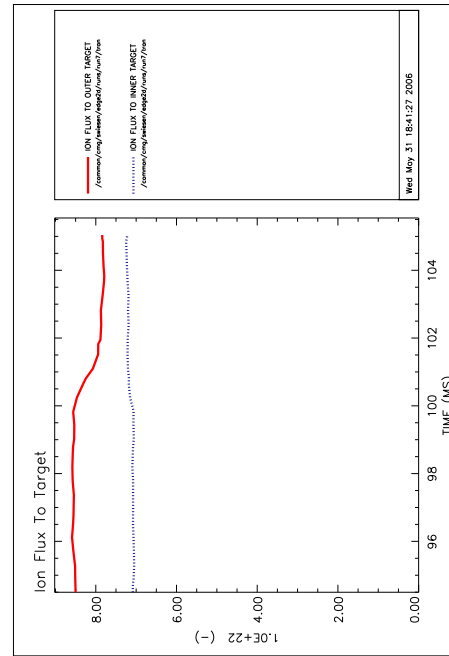
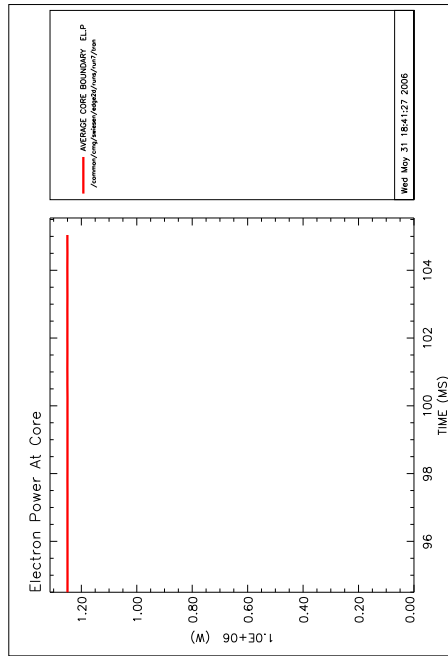
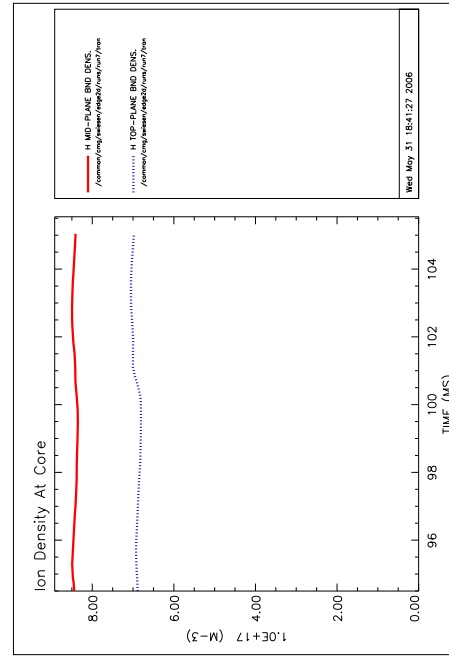
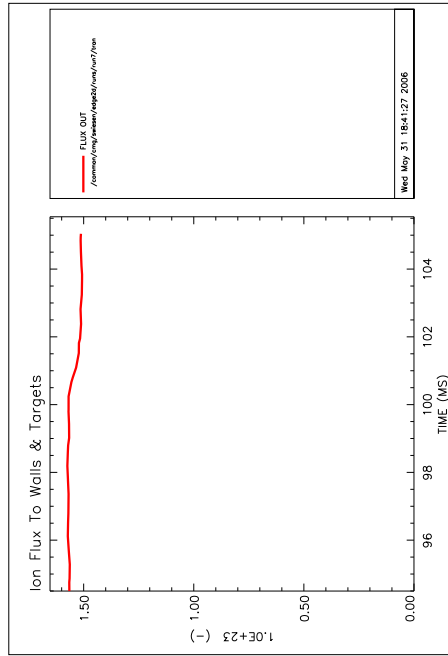
The overall time-trace behaviour also for the midplane or the core are matching quite good.











A NIMBUS supplements

A.I NIMBUS definition of macro-zones

Figure 15 shows the macro-zones which do belong to the combined magnetic-grid plus the void zones which is the actual EIRENE computation grid.

Note: In NIMBUS, void-macro-zones 17 and 18 just above the targets do not contain results from the simulation. The reason is not clear. EIRENE indeed calculates in these regions the neutral profiles.

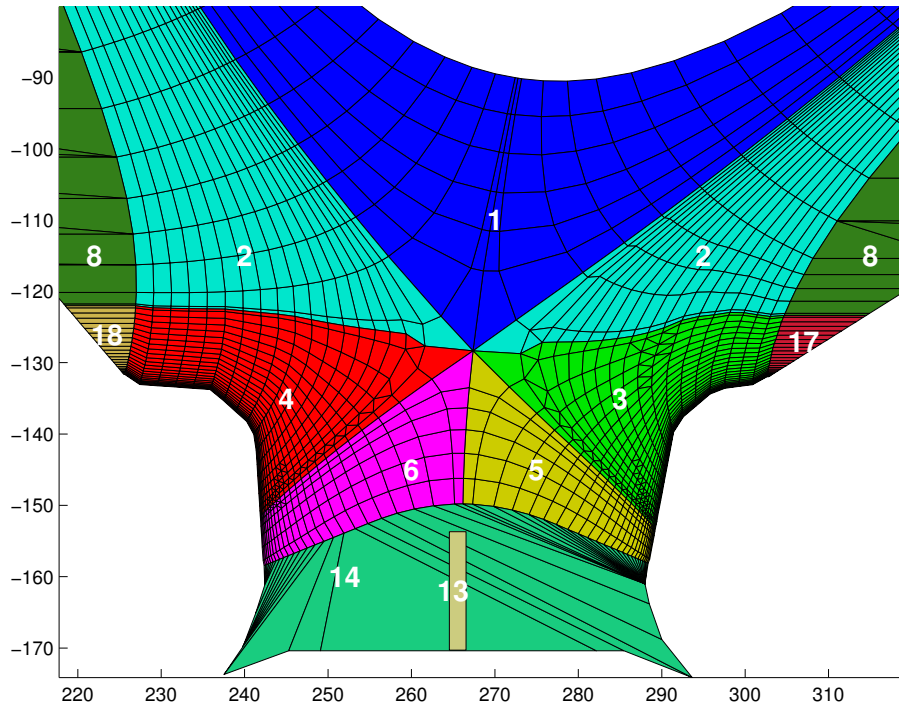


Figure 15: Definition of NIMBUS macro-zones. Only macro-zones which have a correspondence to the EIRENE grid are shown, case (A)

Figure 16 shows the NIMBUS polygons which do not belong to the combined grid shown in fig. 15. The regions are extending very far out and are needed for the combinatorial description of the volume in NIMBUS. Interestingly for the shown geometry, case (A), there are two small gaps between the targets and macro-zones 17 and 18. This may provide a discrepancy in comparison to EIRENE because NIMBUS may have a complete different reflection model for these leaks (flag ALBLK in NIMBUS)

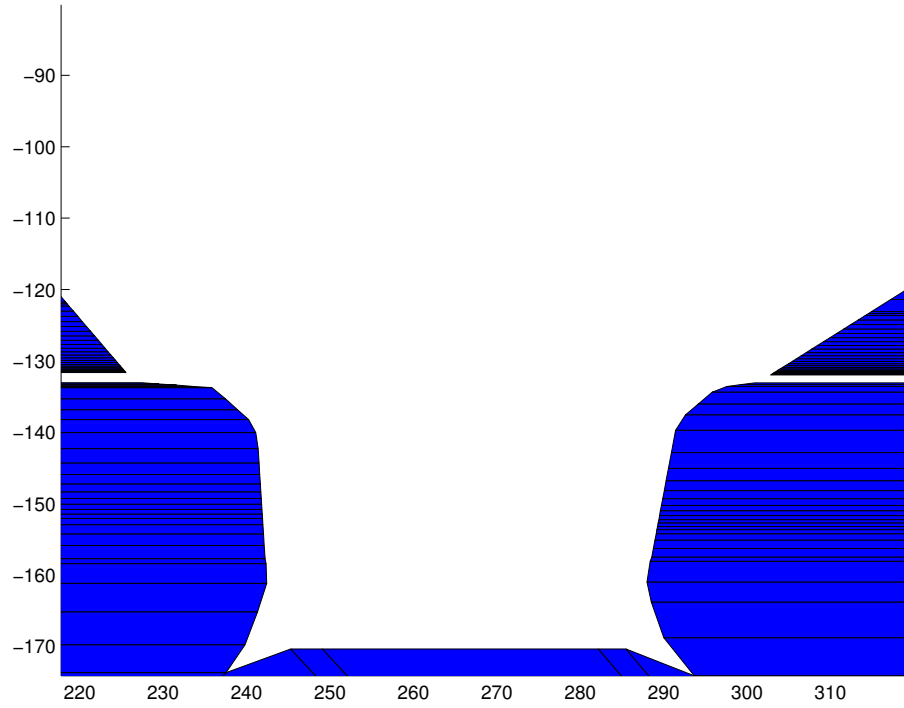


Figure 16: Definition of NIMBUS macro-zones. Only the outerlying macro-zones are plotted which do not belong to the EIRENE grid. There are two small gaps just above the targets, case (A)

A.II NIMBUS volume tallies

All NIMBUS tallies are tracklength estimated:

ENFLUX(k,izo)	neutral flux density
ENNE(izo,k)	neutral density
EAVR(izo,k,1)	avg. neutral energy
EAVR(izo,k,2)	avg. neutral energy
	relative to ion velocity
VELOC(izo,k,1)	neutral velocity x-comp.
VELOC(izo,k,2)	neutral velocity y-comp.
VELOC(izo,k,3)	neutral velocity z-comp.
ABSOR(izo,k)	ionised (absorbed) neutrals
CEXED(izo,k)	charge exchanged neutrals
DISSOC(izo,k)	dissociated neutrals
ELAST(izo,k)	elastic collisions
ELOST(k,izo)	electron energy loss from
ERECYC(k,izo,1)	ion energy gain
ERECYC(k,izo,2)	ion energy loss
PRECYC(k,izo,1)	ion momentum net-gain, x-comp.
PRECYC(k,izo,2)	ion momentum net-gain, y-comp.
PRECYC(k,izo,3)	ion momentum net-gain, z-comp.

A.III NIMBUS surface tallies

WACOL(k, izo, 1)	backscattered neutrals
WACOL(k, izo, 2)	re-emitted neutrals (molecules: half weight)
WACOL(k, izo, 3)	albedoed neutrals (pumps)
WACOL(k, izo, 4)	sputtered ions
WACOL(k, izo, 5)	collisions w/ walls (ions+neutrals)
WACOL(k, izo, 6)	sputtered neutrals
WACOL(k, izo, 7)	neutral collisions w/ walls (total, incl pumps)
WACOL(k, izo, 8)	neutral collisions w/ walls (w/o pumps)
WACOL(k, izo, 9)	ion collisions w/ walls
WACOL(k, izo, 10)	sputtered particles into main-chamber
WACOL(k, izo, 11)	redeposited particles from main-chamber (no div. gaps)
WACOL(k, izo, 12)	collision energy of neutrals onto walls
WACOL(k, izo, 13)	re-emission after desorption (implanted part)

References

- [1] Taroni A et al., Contrib. Plasma Phys. **32**, 438 (1992)
- [2] Simonini R, Corrigan G, Radford G J, Spence J and Taroni A, Contrib. Plasma Phys. **34** 2/3, 368–373 (1994)
- [3] Radford G J, Chankin A V, Corrigan G, Simonini R, Spence J and Taroni A, Contrib. Plasma Phys. **36** 2/3, 187–191 (1996)
- [4] EIRENE documentation, see <http://www.eirene.de>
- [5] Janev R K et al., Elementary Processes in Hydrogen-Helium Plasmas, Springer (1987)
- [6] Sawada K, Fujimoto T, J. Appl. Phys. **78** (5), 2913 (1995)