1 Purpose

2 Scope

3 Definitions

4 Introduction

4.1 CXRS Diagnostics

4.1.1 CXRS for Plasma Measurements

The Charge Exchange Recombination Spectroscopy (CXRS) diagnostic measures line emissions of several low Z impurities in the plasma due to interaction with a neutral beam.

Measured parameters:

- Line's width – ion temperature:

$$kT_{\mathrm{ion}} = mc^2 \frac{\Delta\lambda^2_{\mathrm{Dopp}}}{\lambda_0^2} \tag{1}$$

- Line center's shift — toroidal and poloidal velocities:

$$v_{\mathrm{rot}} = c\frac{\Delta\lambda_{\mathrm{rot}}}{\lambda_0} \tag{2}$$

- Line's intensity — impurity's density and Z_{eff}:

$$n_{\mathrm{imp}} = \frac{4\pi \int I(\lambda)\, d\lambda}{n_{\mathrm{b}} Q^{\mathrm{eff}}_{\mathrm{CX}}(v_{\mathrm{b}})\, dl} \tag{3}$$

Ion	Transition	Wavelength, nm
BeIV	$6 \to 5$	465.8 nm
BeIV	$8 \to 6$	468.5 nm
HeII	$4 \to 3$	468.5 nm
ArXVIII	$16 \to 15$	522.5 nm
NeX	$11 \to 10$	524.9 nm
CVI	$8 \to 7$	529.1 nm
Hα		656.3 nm
MSE		659.1 nm

Table 1. Spectroscopic lines observed by CXRS

$$X^{Z+} + H^0 \to X^{(Z-1)+}(n_2) + H^+ \tag{4}$$

$$X^{(Z-1)+}(n_2) \rightarrow X^{(Z-1)+}(n_1) + h\nu \tag{5}$$

4.1.2 CXRS in ITER

Geometry

Equipment

IMAS Database

4.2 Development of the New Simulation Code

4.2.1 Existing Code

Simulation of Spectra (SOS) code by M. G. von Hellermann [1]

Features:

- Simulation takes into account many physical effects (halo effect, crossection effect, plume effect and others);
- Written in Matlab;
- Has Graphical User Interface (fig. 1).

`images/sos_interface.png`

Figure 1. SOS interface.

4.2.2 Motivation

Existing code (Simulation of Spectra - SOS) lacks some features:

- Simplified plasma, tokamak and diagnostic geometry
  (e.g. elliptical plasma, point emission and others);
- Does not take reflections into account;
- Cannot use data from IMAS directly;
- Requires Matlab license, hard to extend by new developers.

The goal was to create an open and extensible simulation code using Python.

Sub goals:

- Implement interaction with IMAS database (read and write);
- Use IMAS data to create a plasma and diagnostic beam with spatial distributions;
- Use a ray-tracing engine to simulate spectra, this includes how reflections affect simulated spectra;
- Ensure that emission models include all physics already captured by SOS.

4.3 Raysect and CHERAB

Raysect [2] is a ray-tracing framework for Python designed for scientific purposes.

- Supports scientific ray-tracing of spectra from physical light sources such as plasmas.
- Easily extensible, written with user customisation of materials and emissive sources in mind.
- Different observer types supported such as pinhole cameras and optical fibres.

CHERAB [3] is a Python library for forward modelling diagnostics based on spectroscopic plasma emission which provides physical models for Raysect. Provided models for Raysect:

- Tools for plasma and diagnostic beam simulations;
- Physical emission models (active charge exchange, bremsstrahlung and more).

5 For Users

5.1 Quick Example

```
cxrs simulate -s 134000 -r 30 -c config.xml
```

5.2 Setting Simulation Parameters

5.3 Using Command Line Interface

5.3.1 Gathering Information from IMAS

`info`

images/raysect_demo.png

Figure 2. Demonstration of Raysect features.

`composition`

5.3.2 Creating Simulation Environment

`create_env`

`create_config`

`populate`

5.3.3 Reading from and Writing to IMAS

`read_ids_config`

`write_ids_config`

5.3.4 Performing Simulation

6 For Developers

6.1 Project Structure

6.2 Reading data from IMAS

6.2.1 `equilibrium` IDS

6.2.2 `core_profiles` IDS

6.2.3 `edge_profiles` IDS

6.2.4 `charge_exchange` IDS

6.2.5 `nbi` IDS

6.2.6 Supplementary Functions

6.3 Setting the Wall

6.4 Observers

6.4.1 Base Class

6.4.2 Sightlines

6.4.3 Optics

6.4.4 Fibres

Separate Fibres

Fibre Bundle

6.4.5 Camera

6.4.6 Others

Scanner

Total Radiance

Spectrometer

6.5 Populating CHERAB Atomic Database

6.6 Utility Functions

6.6.1 Parsing XML Configuration File

6.6.2 Setting Emission Parameters

6.6.3 Math Functions

6.6.4 Others

References

[1] Manfred von Hellermann et al. "Simulation of Spectra Code (SOS) for ITER Active Beam Spectroscopy". In: *Atoms* 7.1 (Mar. 2019), p. 30. DOI: 10.3390/atoms7010030.

[2] Dr Alex Meakins and Matthew Carr. *raysect/source: v0.5.2 Release*. Version v0.5.2. Aug. 2018. DOI: 10.5281/zenodo.1341376. URL: https://doi.org/10.5281/zenodo.1341376.

[3] Dr Carine Giroud et al. *CHERAB Spectroscopy Modelling Framework*. Version v0.1.0. Mar. 2018. DOI: 10.5281/zenodo.1206142. URL: https://doi.org/10.5281/zenodo.1206142.