An introduction to USpekPy package

Uncertainty estimation on protection quantities for x-rays using SpekPy and Monte Carlo techniques

Xandra Campo

Ionizing Radiation Metrology Laboratory (LMRI)
CIEMAT, Spain

June 2024

Table of Contents

Python usage poll

Wellcome to USpekPy!

What is USpekPy?

- Python package: Open source and GPLv3-licensed library compatible with Python 3 Go
- Goal: Compute mean radiation protection quantities for a simulated x-ray spectrum with uncertainties using Monte Carlo techniques
- Based on SpekPy: Python package for modelling the x-ray spectra from x-ray tubes Go

Main features of USpekPy

- Compute mean values of radiation protection quantities of a simulated x-ray spectrum: $\overline{E}$, K_{air} and $\overline{h_K}$
- Compute mean radiation protection quantities of a simulated x-ray spectrum with uncertainties using Monte Carlo techniques: first and second HVL for Al and Cu, $\overline{E}$, K_{air} and $\overline{h_K}$
- Perform batch simulation to compute mean values and uncertainties of radiation protection quantities for several simulated x-ray spectra

USpekPy in a nutshell

Status

Last version:	1.0.2
Release date:	Jun 2024
Maintenance:	Active

Links

Source code:	GitHub	Go
Documentation:	README @GitHub	Go
Contribute:	Issues @GitHub	Go

Testing

Tests:	Passing
Code coverage:	65%

Distribution

Distribution:	PyPI	Go
License:	GNU GPL v3.0	Go

Requirements

Python:	$\geq$3.8
Dependencies:	spekpy
	pandas
	openpyxl

Authors

Authors:	X. Campo & P. Avilés	
Email:	xandra.campo@ciemat.es	
	paz.aviles@ciemat.es	
Organization:	LMRI-Met @GitHub	Go

How to get support?

Package documentation

Check the documentation of USpekPy at GitHub ▸ Go

README GPL-3.0 license

USpekPy

Radiation protection quantities for x-rays with uncertainties

Date Jun 24 | Version 1.0.2 | Maintenance Active

Surce code GitHub | Documentation README | Contribute Issues | Organization LMRI-Met

Distribution PyPi | License GPLv3.0 | Tests Passing | CodeCov 65%

Table of Contents

Contact developers

Contact the developers of USpekPy via email:

Xandra Campo
xandra.campo@ciemat.es

Paz Avilés
paz.aviles@ciemat.es

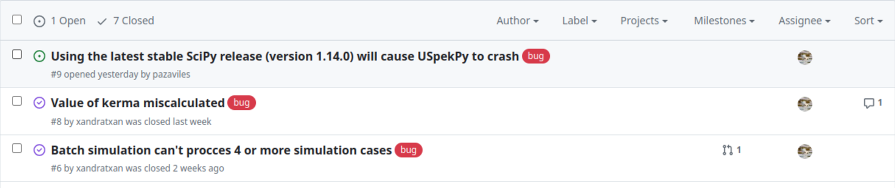
How to contribute to USpekPy?

What may be a contribution?

- Bug reports & fixes
- Documentation improvements
- Feature enhancements

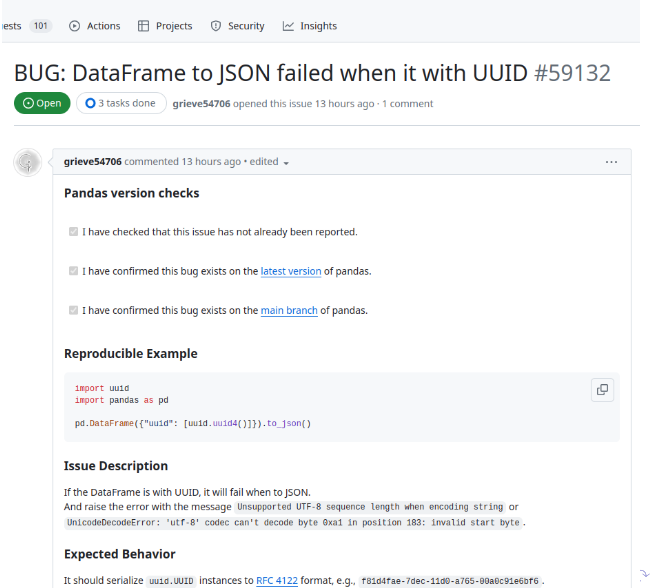
How to deliver a contribution?

- Issues page at GitHub (Recommended) Go
- Contact the developers via email



What to include in a contribution?

- Title
- Description
- Expected behavior
- Steps to reproduce
- Minimal, reproducible example
- Environment
- Error messages, logs
- Potential fix


ests 101 Actions Projects Security Insights

BUG: DataFrame to JSON failed when it with UUID #59132

Open 3 tasks done grieve54706 opened this issue 13 hours ago · 1 comment

grieve54706 commented 13 hours ago • edited

Pandas version checks

- [x] I have checked that this issue has not already been reported.
- [x] I have confirmed this bug exists on the latest version of pandas.
- [x] I have confirmed this bug exists on the main branch of pandas.

Reproducible Example

```
import uuid
import pandas as pd

pd.DataFrame({"uuid": [uuid.uuid4()]}).to_json()
```

Issue Description

If the DataFrame is with UUID, it will fail when to JSON.
And raise the error with the message `Unsupported UTF-8 sequence length when encoding string` or `UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa1 in position 183: invalid start byte`.

Expected Behavior

It should serialize `uuid.UUID` instances to RFC 4122 format, e.g., `f81d4fae-7dec-11d0-a765-00a0c91e6bf6`.

How does USpekPy package work?

Compute mean radiation protection quantities

Information flow

For a single case, given an x-ray quality and an operational quantity at a specific irradiation angle:

Input

Value:

- Filter thickness
- Peak kilovoltage
- Anode angle
- $\frac{\mu_{tr}}{\rho}(E)$
- $h_K(E)$

Tool

SpekWrapper class

Output

Value:

- HVL(Al, Cu)
- $\overline{E}$
- K_{air}
- $\overline{h_K}$

Compute mean radiation protection quantities

Workflow

Workflow for $\overline{E}$, K_{air} and $\overline{h_K}$:

1. Compute x-ray spectrum (energy and fluence) using SpekPy
2. If necessary, interpolate $\frac{\mu_{tr}}{\rho}(E)$ and $h_K(E)$ to spectrum energies in logarithmic scale
3. If necessary, apply units conversion
4. Compute the integral quantity using the corresponding definition

$$\overline{E} = \frac{\int_0^\infty \phi(E) E dE}{\int_0^\infty \phi(E) dE}$$

$$K_{air} = \int_0^\infty \phi(E) \frac{\mu_{tr}}{\rho}(E) E dE$$

$$\overline{h_K} = \frac{\int_0^\infty \phi(E) \frac{\mu_{tr}}{\rho}(E) h_K(E) E dE}{\int_0^\infty \phi(E) \frac{\mu_{tr}}{\rho}(E) E dE}$$

HVLs are calculated using SpekPy methods.

Compute mean RP quantities with uncertainties

Information flow

For a single case, given an x-ray quality and an operational quantity at a specific irradiation angle:

Input

Value and uncertainty:

- Filter thickness
- Peak kilovoltage
- Anode angle
- $\frac{\mu_{tr}}{\rho}(E)$

Value:

- $h_k(E)$
- Number of iterations

Tool

USpek class

Output

Value and uncertainty:

- HVL(Al, Cu)
- $\overline{E}$
- K_{air}
- $\overline{h_K}$

Compute mean RP quantities with uncertainties

Workflow

A simulation is performed for the specified number of iterations.

1. For each iteration:
    1. Generate random values of the input variables considering their mean values, uncertainties and distributions.
    2. Compute mean values of the integral quantities using the SpekWrapper class
2. Once the iterations are completed:
   Compute statistical mean value and standard deviation of the integral quantities from the different values obtained in the iterations

Compute batch simulation for several x-ray spectra

Information flow

For a set of cases, each case for a given x-ray quality and operational quantity at an irradiation angle:

Input

For every case:
Value and uncertainty:

- Filter thickness
- Peak kilovoltage
- Anode angle
- $\frac{\mu_{tr}}{\rho}(E)$

Value:

- $h_k(E)$
- Number of iterations

Tool

batch_simulation function

Output

For every case,
value and uncertainty:

- HVL(Al, Cu)
- $\overline{E}$
- K_{air}
- $\overline{h_K}$

Compute batch simulation for several x-ray spectra

Workflow

1. The set of cases are provided in an input file, each case is a column in that file
2. For each case, a simulation is performed for the specified number of iterations using the USpek class
3. Results for the set of cases are returned in an output file, appending the result to the corresponding column of the input file

Units and uncertainties convention

- All the uncertainties are standard uncertainties ($k = 1$)
- The units of relative uncertainties are expressed as fraction of one

Quantity	Unit
Distance	mm
Voltage	kV
Angle	deg
Energy	keV
Fluence	$1/cm^2$
Mass energy transfer coefficients of air	cm^2/g
Air kerma	µGy
Mono-energetic K to H conversion coefficients	Sv/Gy

Verification of the code

- Integral quantity values: Compared with values provided by SpekPy (except $\overline{h_K}$)

- Integral quantity uncertainties: Compared with values provided by:
  - Previous script version of USpekPy developed by Paz Avilés
  - Results obtained by CMI

How to use USpekPy?

How to install USpekPy?

Steps to install USpekPy using PyCharm IDE:

1. Clone the seminar repository to your computer ▸ Seminar repository
2. Set up a virtual environment for the project
3. Install USpekPy
4. Fix SciPy dependency issue

Now you are ready to go!

▸ Detailed guide

Examples of USpekPy usage

Using USpekPy to compute integral quantities for x-ray spectra:

1. Values for a single case using data files
2. Values and uncertainties for a single case using data files
3. Values and uncertainties for several cases using CSV/Excel input file

▸ Detailed guide

Wrapping up: What's next?

Improvements on the horizon

- Bug: Fix SciPy dependency bug
- New feature: Add the contribution to the uncertainty of the variation of the mono-energetic air kerma-to-dose conversion coefficients
- Documentation: Improve package documentation (GitHub Wiki, GitHub Pages)
- Testing: Improve test code coverage

Let us know what you think

Complete our satisfaction survey about this seminar!

Help us make future seminars better.

▸ Satisfaction survey

Contribute to USpekPy package!

This sofware is for you. We want to make it fit better your necesities. Let us know if you find any issue or if you would like to have any new feature in future versions.

▸ USpekPy Issues page

xandra.campo@ciemat.es
paz.aviles@ciemat.es

Thank you very much!

We're grateful for your time and attention today.

We appreciate your interest in USpekPy.

Thank you for joining us.