Appendix 1. UML class diagram

Observation

- freq : list(float)
- flux : list(float)
- error : list(float)
- t : float

+ addPoint(freq : float, flux : float, error : float)
+ setError(error : array[float])
+ setTime(t : float)
+ getFreq() : array[float]
+ getFlux() : array[float]
+ getError() : array[float]
+ getTime() : float
+ findMaxFluxIndex() : int
+ obtainSlope(index1 : int, index2 : int) : float
+ readEpochCSV(name_CSV : String)

ObservationSequence

- observationList : list(Observation)

+ addObservation(Observation)
+ readCSV(name_CSV : String)
+ writeCSV(name_CSV : String)
+ getNumObservations() : int
+ getObservation(index : int) : Observation

Result

+ t : float
+ p, pErr : float
+ st, stErr : float
+ vt, vtErr : float
+ rChiSq : float
+ quadError : float
+ stdErrorReg : float
+ errorCorrFactor : float
+ freqExpanded : list(float)
+ sFlux : list(float)
+ sm, smErr : float
+ vm, vmErr : float
+ r, rErr : float
+ b, bErr : float
+ a, aErr : float

+ setTime(t : float)
+ setFitting(p : float , st : float, vt : float)
+ setFittingErr(pErr : float , stErr : float, vtErr : float)
+ setFittingGoodness(rChiSq : float, quadError : float, stdErrorReg : float, errorCorrFactor : float)
+ setTheoreticalCurve(freqExpanded : array[float], sFlux : array[float])
+ setMax(sm : float, vm : float)
+ setMaxErr(smErr : float, vmErr : float)
+ setPhysParams(r : float, b : float, a : float)
+ setPhysParamsErr(rErr : float, bErr : float, aErr : float)
+ getTime() : float
+ getFitting() : Dict[str,float]
+ getFittingErr() : Dict[str,float]
+ getFittingGoodness() : Dict[str,float]
+ getTheoreticalCurve() : Dict[str,float]
+ getMax() : Dict[str,float]
+ getMaxErr() : Dict[str,float]
+ getPhysParams() : Dict[str,float]
+ getPhysParamsErr() : Dict[str,float]

ResultSequence

- resultList : list(Results)
- pList : list(float)
- pErrList : list(float)
- pAvg : float
- pAvgErr : float
- stList : list(float)
- stErrList : list(float)
- vtList : list(float)
- vtErrList : list(float)
- rList, rErrList : list(float)
- rSlope, rConst : float
- bList, bErrList : list(float)
- bSlope, bConst : float
- aList, aErrList : list(float)
- aSlope, aConst : float

+ addResult(Result)
+ setTimeList(tList : list(float))
+ setPList(pList : list(float))
+ setPErrList(pErrList : list(float))
+ setPAvg(pAvg : float, pAvgErr : float)
+ setStList(stList : list(float))
+ setStErrList(stErrList : list(float))
+ setVtList(vtList : list(float))
+ setVtErrList(vtErrList : list(float))
+ setREvol(rList : list(float), rErrList : list(float), rSlope : float, rConst : float)
+ setB(bList : list(float), bErrList, : list(float), bSlope : float, bConst : float)
+ setA(aList : list(float), aErrList : list(float), aSlope : float, aConst : float)
+ getPAvg() : list(float)
+ getPList() : list(float)
+ getPErrList() : list(float)
+ getStList() : list(float)
+ getStErrList() : list(float)
+ getVtList() : list(float)
+ getVtErrList() : list(float)
+ getResult(index : int) : Result
+ getNumResults() : int
+ getTimeList() : list(float)
+ getREvol() : Tuple[list(float), list(float), float, float]
+ getBEvol() : Tuple[list(float), list(float), float, float]
+ getAEvol() : Tuple[list(float), list(float), float, float]

Report

- _htmlListItem(name : String, value : String) : String
- _htmlEpoch(Result, epochIndex : int, figureType : String, sizeSpec : String) : String
- _htmlHeaderItem(item : String) : String
- _htmlTableItem(value : float) : String
- _htmlTableItemWithError(value : float, error : float) : String
+ infoEpoch(Result)
+ createHTMLReport(ResultSequence, file_name : String, figureType : String)
+ createPDFReport(ResultSequence, file_name : String)

SNGraph

+ plotEpoch(Observation, Result)
+ plotSequence(ObservationSequence, ResultSequence)
+ plotTimeEvol(ResultSequence)
+ saveEpoch(Observation, Result, filename : String)
+ saveSequence(ObservationSequence, ResultSequence)
+ saveTimeEvol(ResultSequence)

AbstractFit

- model : AbstractModel

+ init(model : Model)
+ binarySearch(Observation, soughtParam : String, FitParameters, **modelParams : Dict[String,float]) : float
+ paramSearch(Observation) : Dict[String, float]
+ fitEpoch(Observation) : Result
+ sequenceFitting(ObservationSequence) : ResultSequence
+ calculatePhysParams(d : float, Result) : Result

AbstractModel

+ fluxModel(freq : float, **params : Dict[String, float]) : float
+ fluxModelArray(freq : array[float], **params : Dict[String, float]) : array[float]
+ chiSq(freq : float, flux : float, error : float , **params : Dict[String, float]) : float
+ chiSqSum(freq : array[float], flux : array[float], error : array[float] , **params : Dict[String, float]) : array[float]
+ rChiSq(freq : float, flux : float, error : float , **params : Dict[String, float]) : float
+ rChiSqSum(freq : array[float], flux : array[float], error : array[float] , **params : Dict[String, float]) : array[float]
+ quadErr(freq : float, flux : float, **params : Dict[String, float]) : float
+ quadErrSum(freq : array[float], flux : array[float] , **params : Dict[String, float]) : array[float]
+ chiSqDiff(freq : array[float], flux : array[float], error : array[float], step : float, param : String, **params : Dict[String, float]) : float
+ SNr(sm : float, vm : float, stdErrReg : float) : float
+ SNb(sm : float, vm : float, stdErrReg : float) : float
+ SNa(sm : float, vm : float, stdErrReg : float, t : float) : float

KraussModel (AbstractModel)

+ fluxModel(freq : float, **params : Dict[String, float]) : float

SNFit (AbstractFit)

+ paramSearch(Observation) : Dict[String, float]

FitParameters

- initialGuess : float
- preGuess : float
- postGuess : float
- preFactor : float
- postFactor : float
- min : float default = - inf
- max : float default = + inf

TimeEvol

+ timeEvol(ResultSequence) : ResultSequence

SNConfig

- numRowsHeader : float default = 9
- enableErrCorr : Boolean default = True
- enableFixedP : Boolean default = False
- fixedP : float default = 3
- fitAccuracy : float default = 0.0001
- alpha : float default = 1
- fillFactor : float default = 0.5
- eB : float default = 0.1
- figurePath : String default = 'figures/'
- reportPath : String default = 'reports/'
- figureWidth : int default = 600
- figureHeight : int default = 400
- plotEpochAxeX : Tuple[float, float] default = [1,100]
- plotEpochAxeY : Tuple[float, float] default = [0.1,15]

GlobalData

- title : String
- authors : String
- organization : String
- eventName : String
- eventStartDate : String
- eventDistance : float
- eventDistanceErr : float
- measuringEquipment : Strin

FolderDoesNotExistError(Exception)

EqualIndexError(Exception)

BinarySearchNoConvergenceError(Exception)

NotImplementedError(Exception)

Appendix 2. Corrections to Pacholczyk (1970)

As a complement to the package RadioPy includes a Python-implementation of the appendix 2 from Pacholczyk's (1970) book [5]. Its purpose is to facilitate the user an easier implementation of other models. Nevertheless during such implementations we found two errata in the text. The present Appendix intends to correct those errors.

1 Incorrect expression

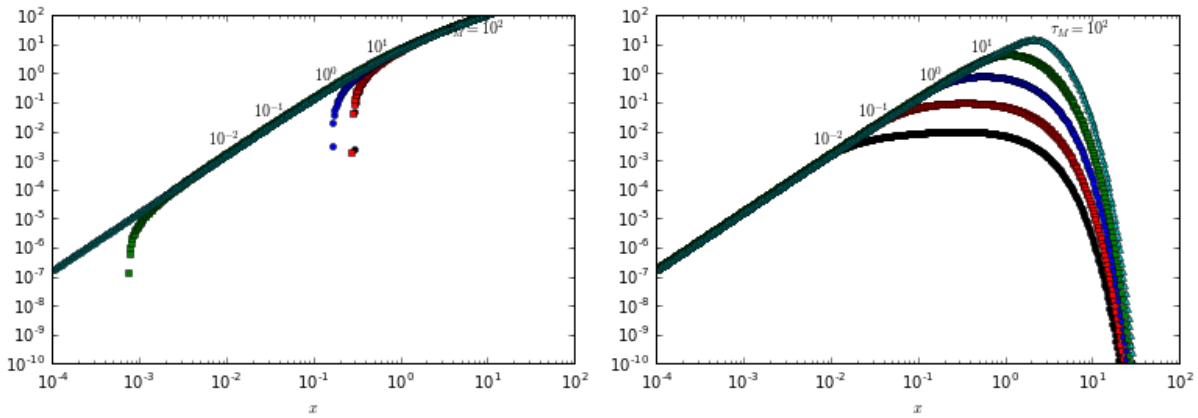
At page 95 an expression for function $J(\tau_m, x)$ is given (Eq. 3.48):

$$J(\tau_m, x) = S(x)\left[1 - e^{-\tau_m} \cdot \frac{K(x)}{K(0.29)}\right] \tag{A}$$

At appendix 2, page 226 there is another expression for the same function stated as:

$$J(\tau_m, x) = \frac{S(x)}{S(0.29)} \cdot \left[1 - e^{-\tau_m \cdot \left[\frac{K(x)}{K(0.29)}\right]}\right] \tag{B}$$

In order to distinguish whether expression is correct we plotted those expressions for an array of frequencies x and with several values of τ_m: $\{0.01, 0.1, 1, 10, 100\}$. The frequencies were taken within the interval x: $\{1 \cdot 10^{-4}, 50\}$ which is the interval given at Pacholczyk's appendix 2, Table 1.



(A) (B)

Looking at the figures we can assure that the correct expression is (B) by comparing with Pacholczyk's Fig. 3.7 and the values of appendix 2, Table 2.

In conclusion; Eq. 3.48 for function $J(\tau_m, x)$ is incorrect. Correct expression can be found at Appendix 2, page 226.

2 Incorrect values for function c_{10}

Pacholczyk's Appendix 2 gives an expression for function c_{10} as:

$$c_{10} = \int_0^\infty x^{\frac{2}{3}(p-1)} \cdot F(x)\, dx$$

Values for different values of p -p: $\{0.5, 6.0\}$- are given at Table 7 (page 232). The values for function c_{10} are excessively high. We found different values through numerical calculation using the expression stated above.

$\boldsymbol{p}$	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
$\boldsymbol{c_{10}}$	1.83	1.61	1.62	1.79	2.13	2.70	3.60	5.02	7.27	10.92	16.92	27.00

We assure those are the correct values as we proved the results by calculating c_{10} for $p = 1$ and x: $\{1 \cdot 10^{-4}, 50\}$ through the Riemann integral. For the limits of integration it is considered that $x = \infty$ can be approximated as $x = 50$ since $F(1 \cdot 10^{-4}) = 9.96 \cdot 10^{-2}$ and $F(50) = 1.74 \cdot 10^{-21}$.

Riemann integration with those conditions lead to a result of:

$c_{10} = 1.60408106868$

Proving our results to be better than the ones shown at Pacholczyk's Appendix 2, Table 7 – that states $c_{10} = 20.2$ with same conditions-.