Introduction to VITESS

V irtual
I nstrumentation
T ool for the
E uropean
S pallation
S ource

Klaus Lieutenant

History of VITESS

Idea of Ferenc Mezei to realize a package to simulate all kinds of neutron scattering instruments especially on neutron spallation sources because of the European spallation source (ESS)

Important dates

1998: Some existing programs put together, GUI added

1999: Release of VITESS 1.0
First complete instruments simulated

2000: SCANS collaboration started (followed by MCNSI in FP6)
(McStas, VITESS, …)

2001: Release of VITESS 2.0 containing polarisation, absolute flux values, improved GUI
Several ESS instruments simulated

2003: Decision: ESS will not be built in the near future
VITESS will be used for instruments on other sources

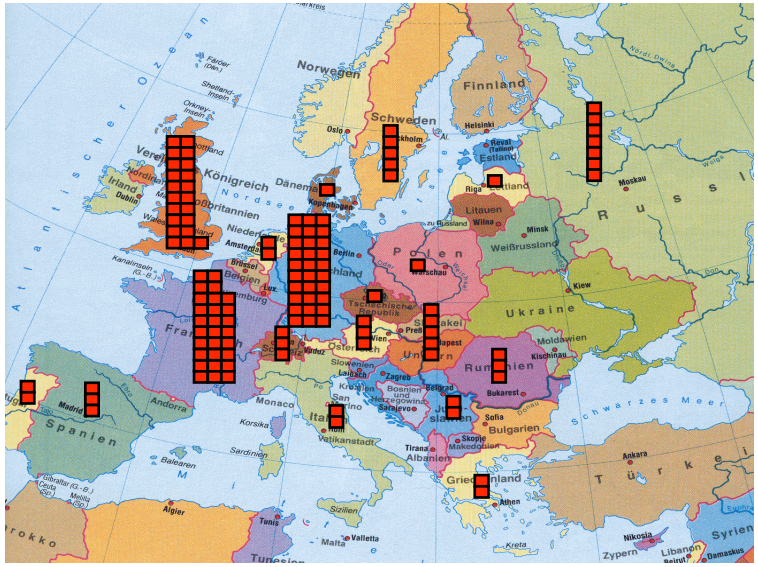
2005: VITESS group at HMI closed

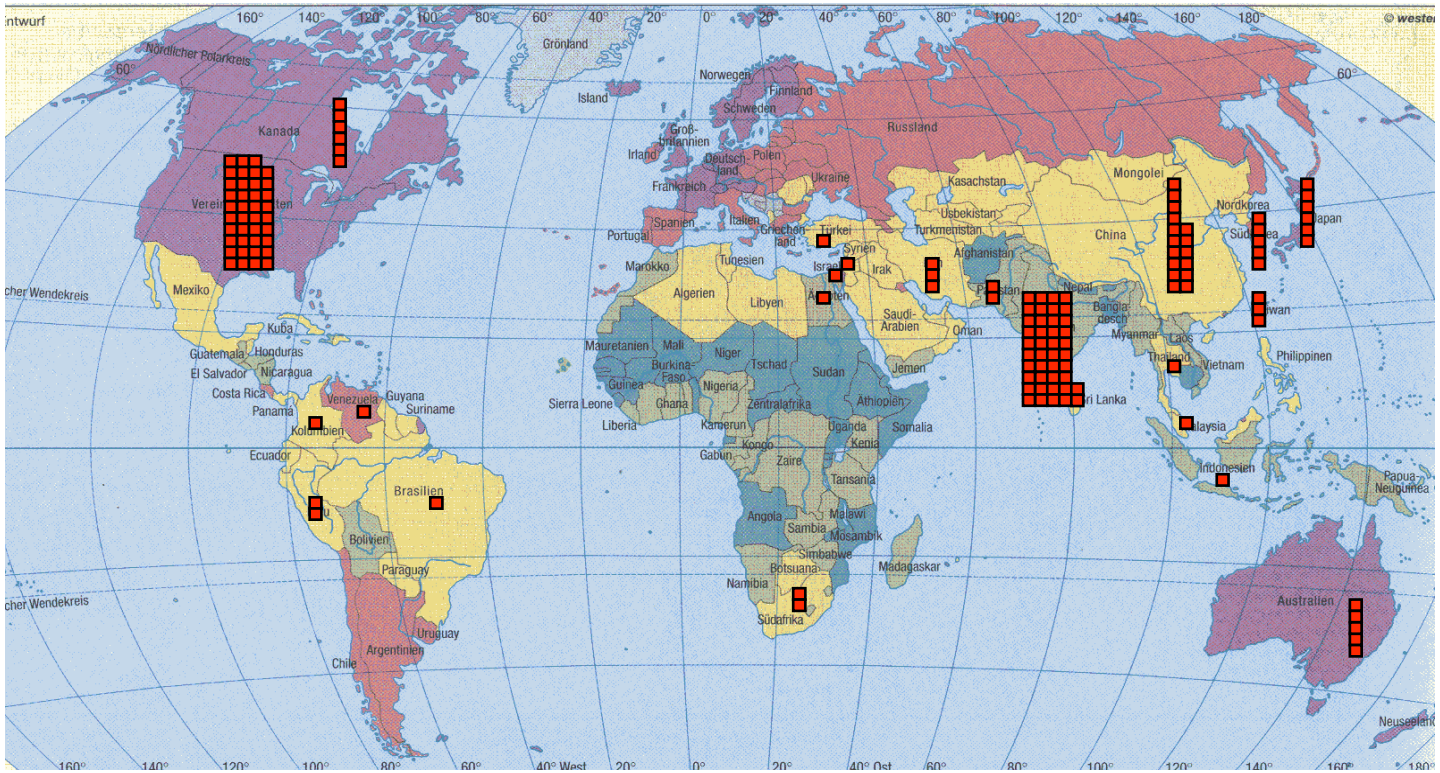
2006: VITESS released under GNU license
larger developer team, all working only partly on VITESS

2008: Version 2.8 released

2009: Latest version 2.9 released

Distribution of Downloads in 2003





Present Status

Staff

Michael Fromme (HMI; GUI and release of new versions)

Sergey Manoshin (JINR; development of new modules)

Klaus Lieutenant (IFE; module development)

Andreas Houben (RWTH Aachen, module development)

Phillip Bentley (ANSTO; optimisation routine)

Geza Zsigmond (PSI; maintenance of his modules)

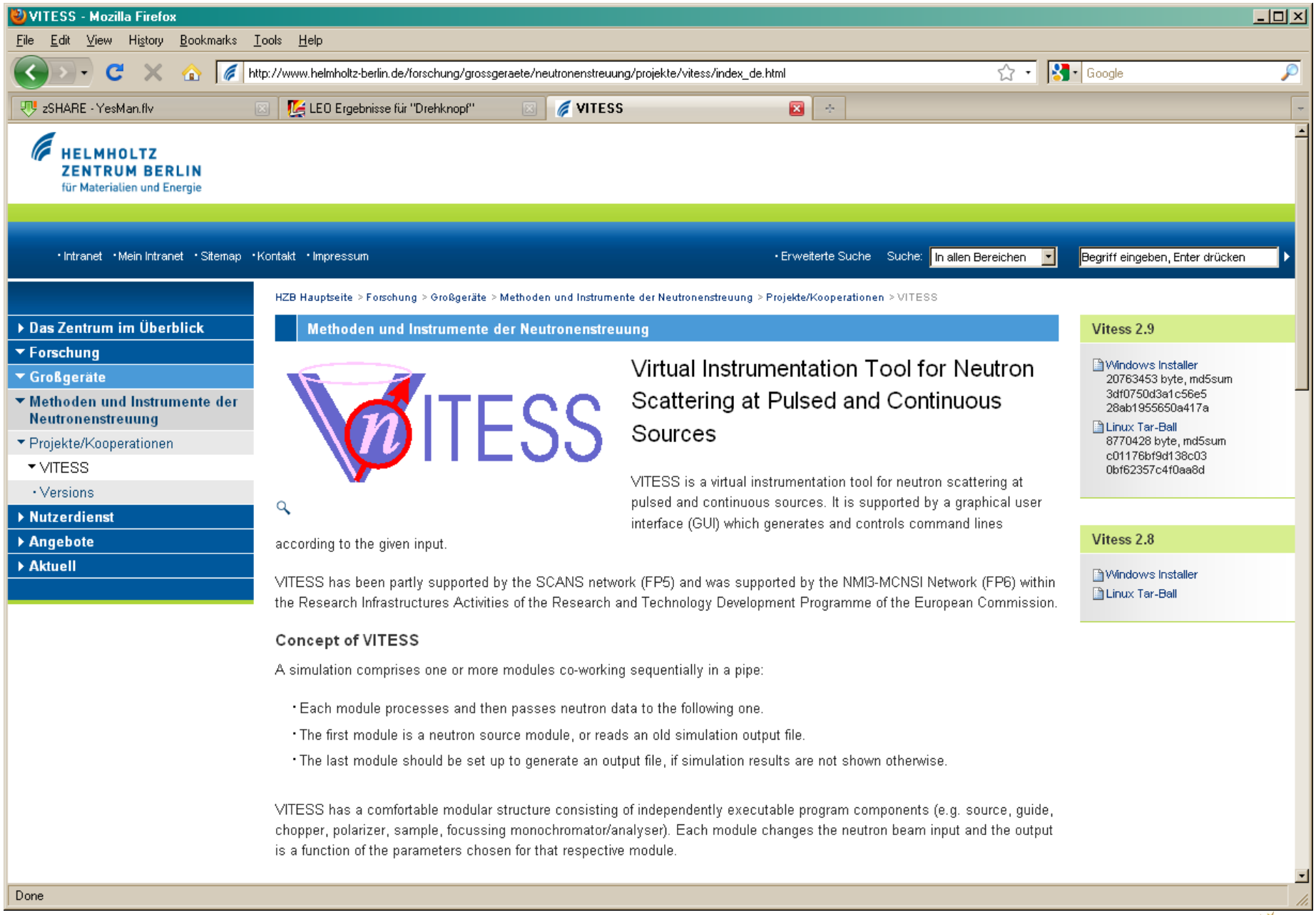
Program

Executables for Windows/DOS, Unix (SunOS: versions from 5.6, OSF1 V4.0) and Linux (versions from 2.0.35), Macintosh on demand

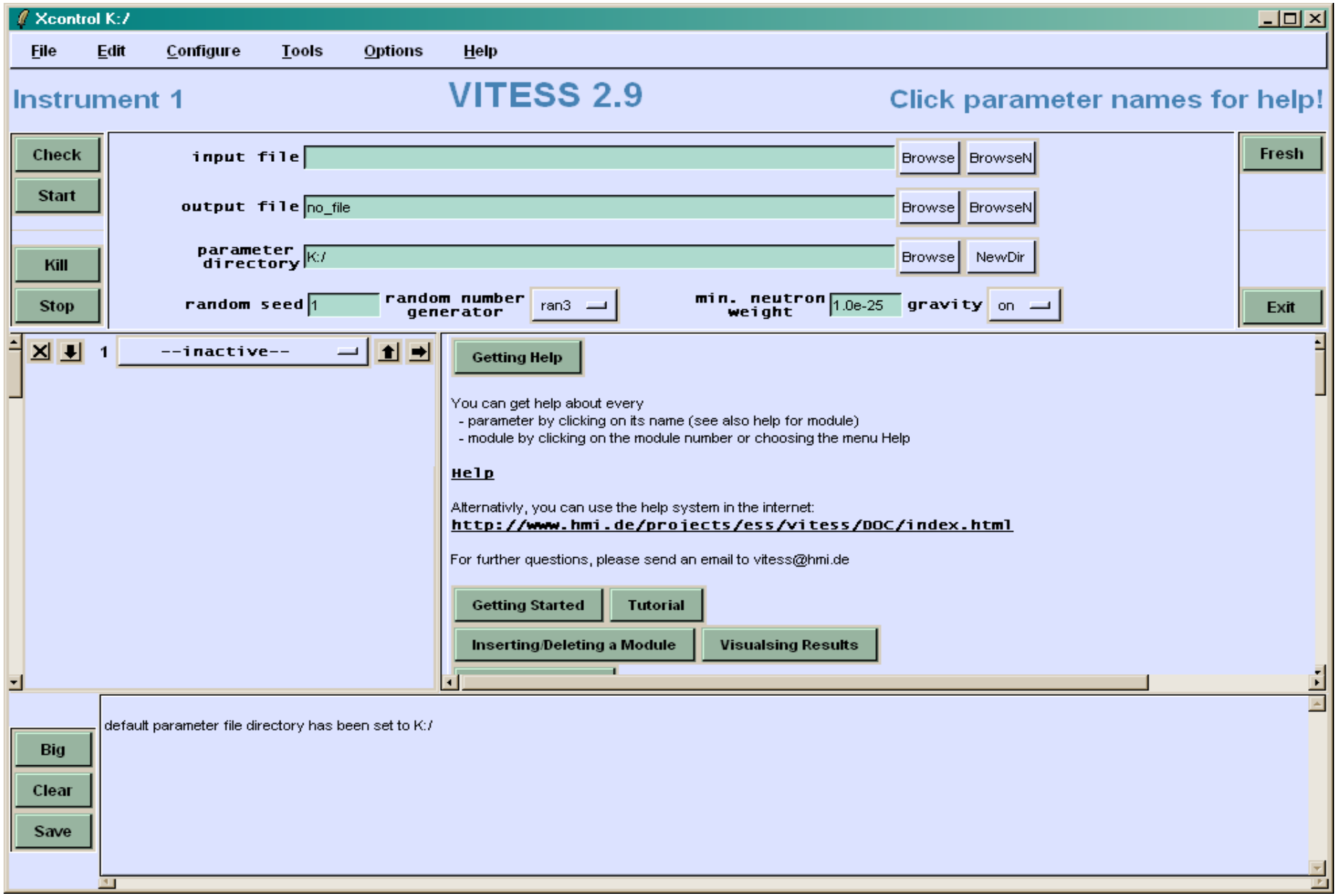
free of charge

Can be downloaded from internet address http://www.helmholtz-berlin.de/vitess/

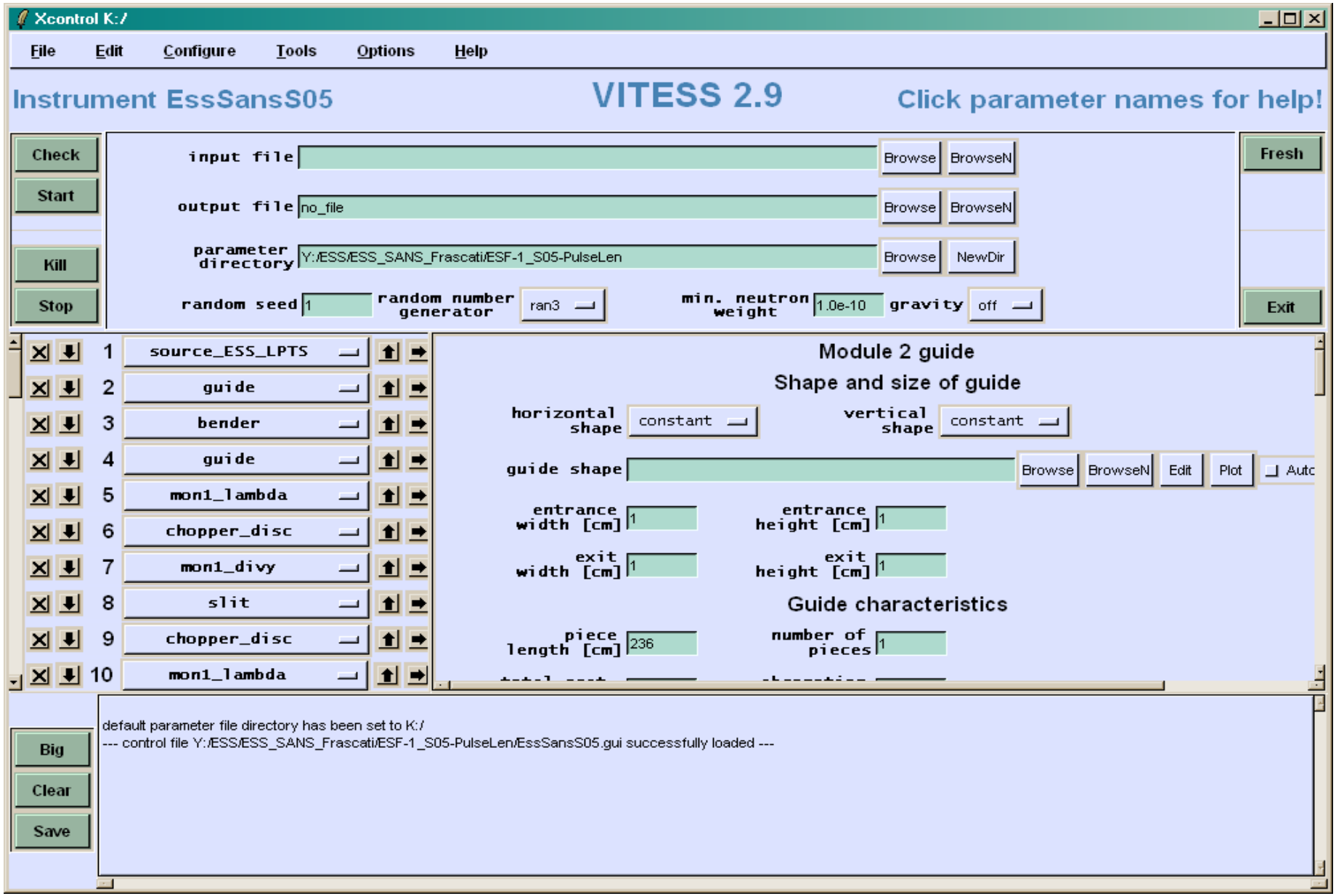
Home Page 'www.helmholtz-berlin.de/vitess'



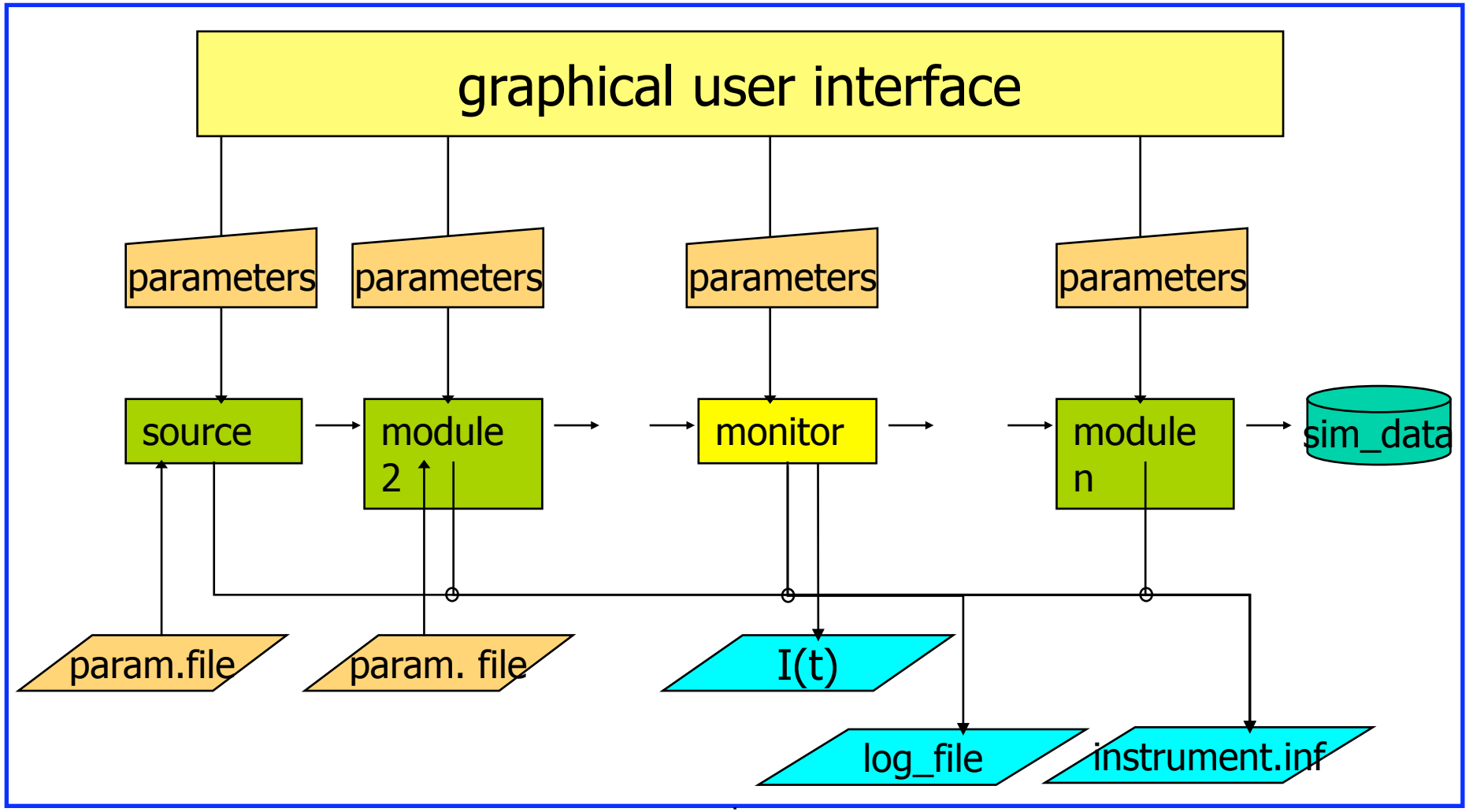
GUI after program start



GUI after loading an instrument



Concept of VITESS



Parameter set transferred

Right-handed system

x: along the beamline
y: to the left
z: (vertically) up

ID

criterion 'ray tracing'

'colour'

Time of flight t [ms]

wavelength λ [Å]

count rate p [n/s]

location of neutron x [cm]

location of neutron y [cm]

location of neutron y [cm]

flight direction $v_x/|\underline{v}| = \cos \alpha$

flight direction $v_y/|\underline{v}| = \cos \beta$

flight direction $v_z/|\underline{v}| = \cos \gamma$

Spin P_x

Spin P_y

Spin P_z

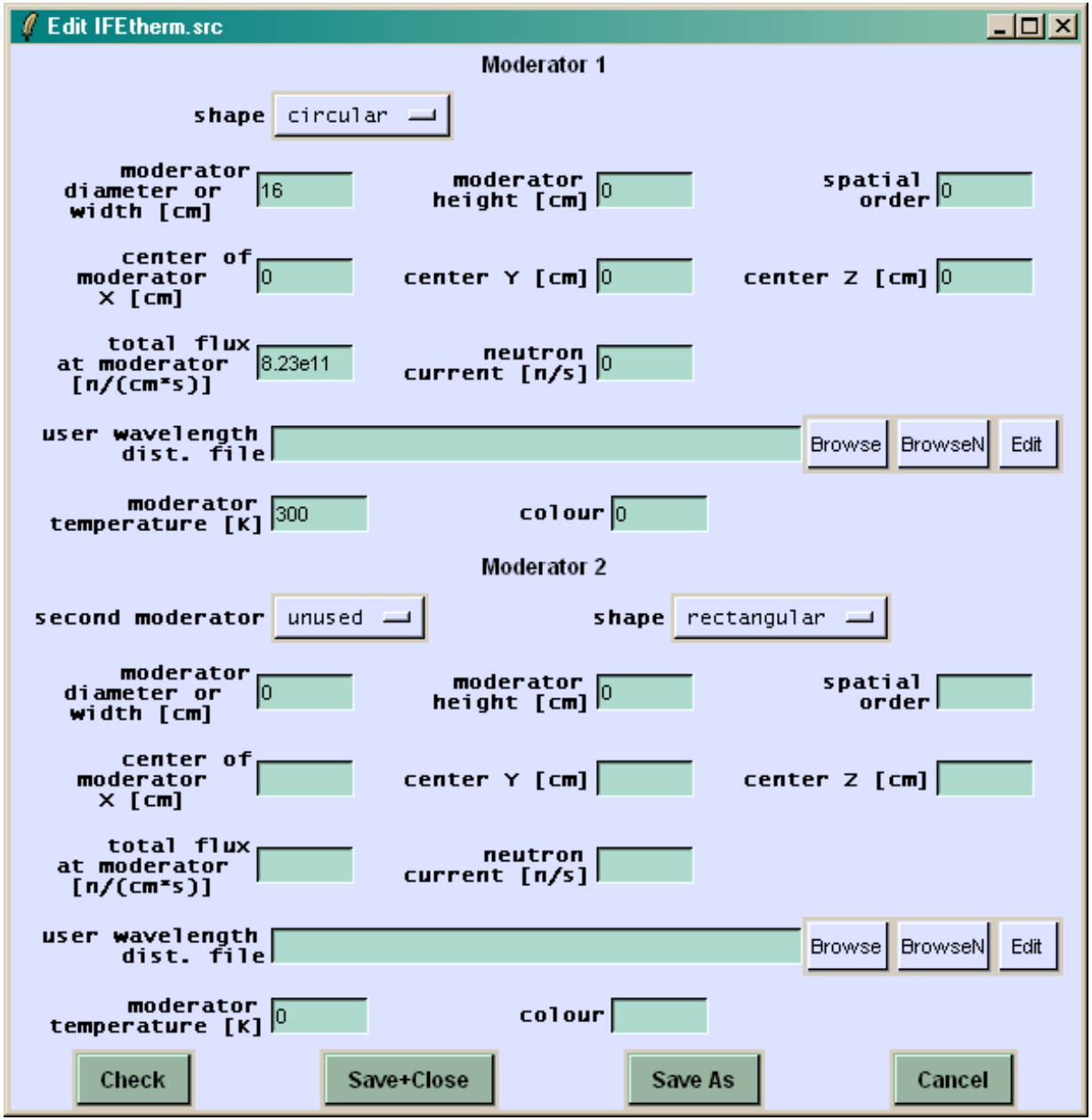
Modules representing Hardware

Sources	reactor	SPSS	LPSS		
Space + Windows	space	slit	window/ beamstop	multiple windows	grid
Choppers	disk	Fermi str.	Fermi cur		
Mono/Ana	flat	focus	foc.user		
Modules f. Polaris.	^{3}He-pol.	coil flip.	prec.field	sesansfield	pol. mirror
	SM-polar.	grad.flip.	rot. field	res.Drabkin	
Samples	elast.isot.	powder	SANS	reflecto.	
	inelastic	sngl.crys.	S(Q)	environm	
Collimat.	coll.simpl	coll.soller	coll.radial		
	guide	bender	vel.select	SM-ensmbl	detector

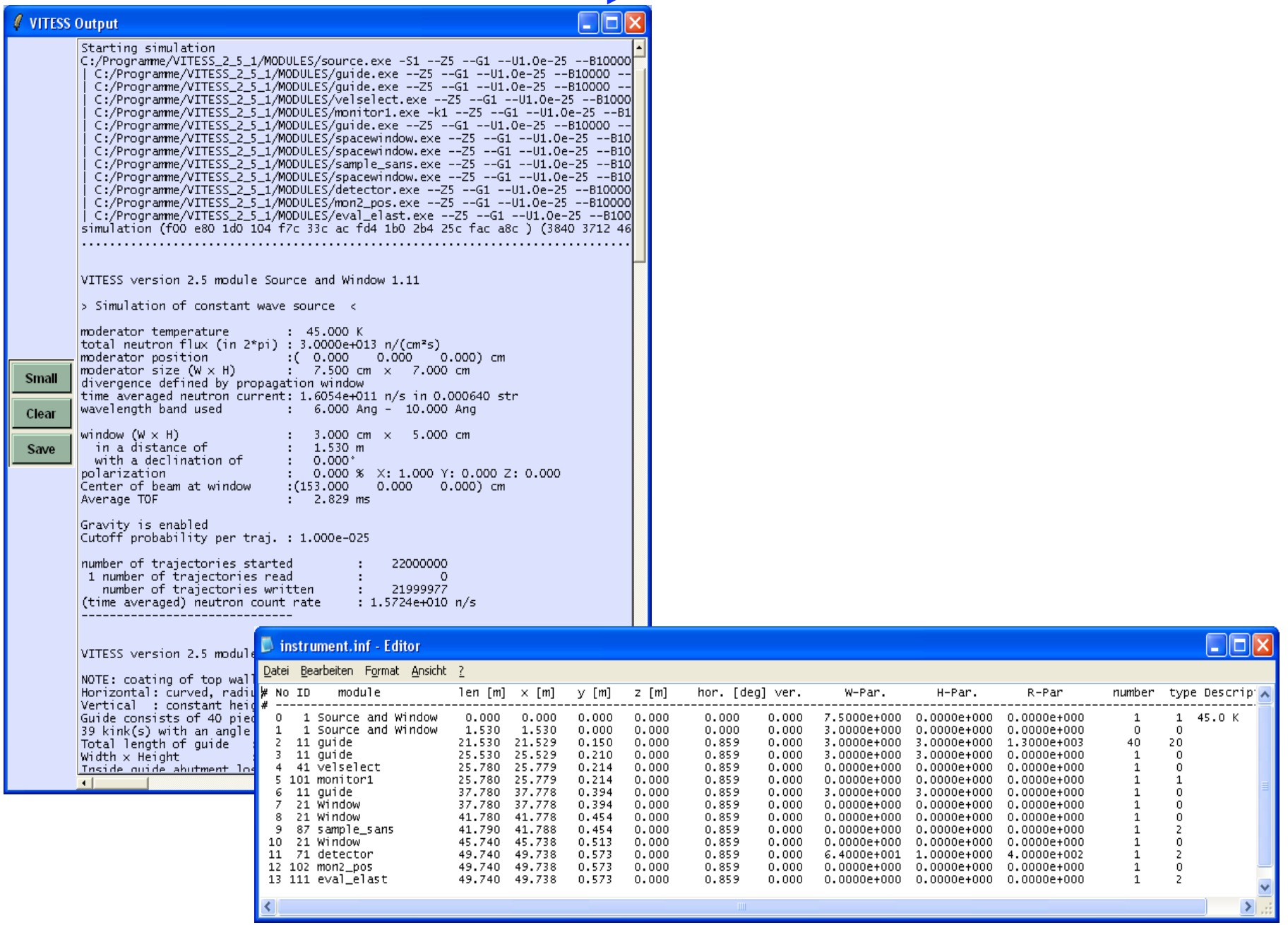
Modules for Data Evaluation

Monitor 1	Time	Wavelen	y-/z-pos	div-y/-z		
Monitor 2	y-pos z-pos	div-y div-z	TOF λ	λ div_y/-z	y div-y	z div-z
MonPol 1	Time	Wavelen	y-/z-pos	div_y/-z		
MonPol 2	y-pos z-pos					
Evaluate	Elastic	Inelast				
	Frame					
	Writeout					
	Visualize					

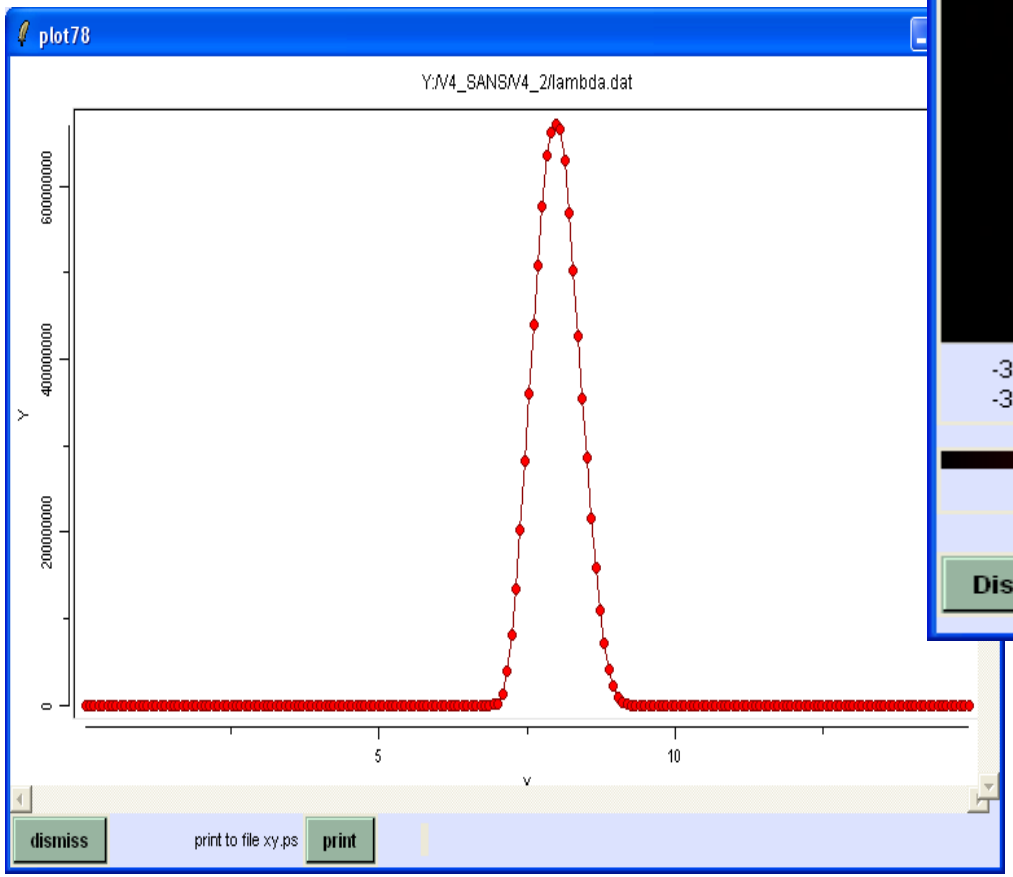
Input Data: Parameter File

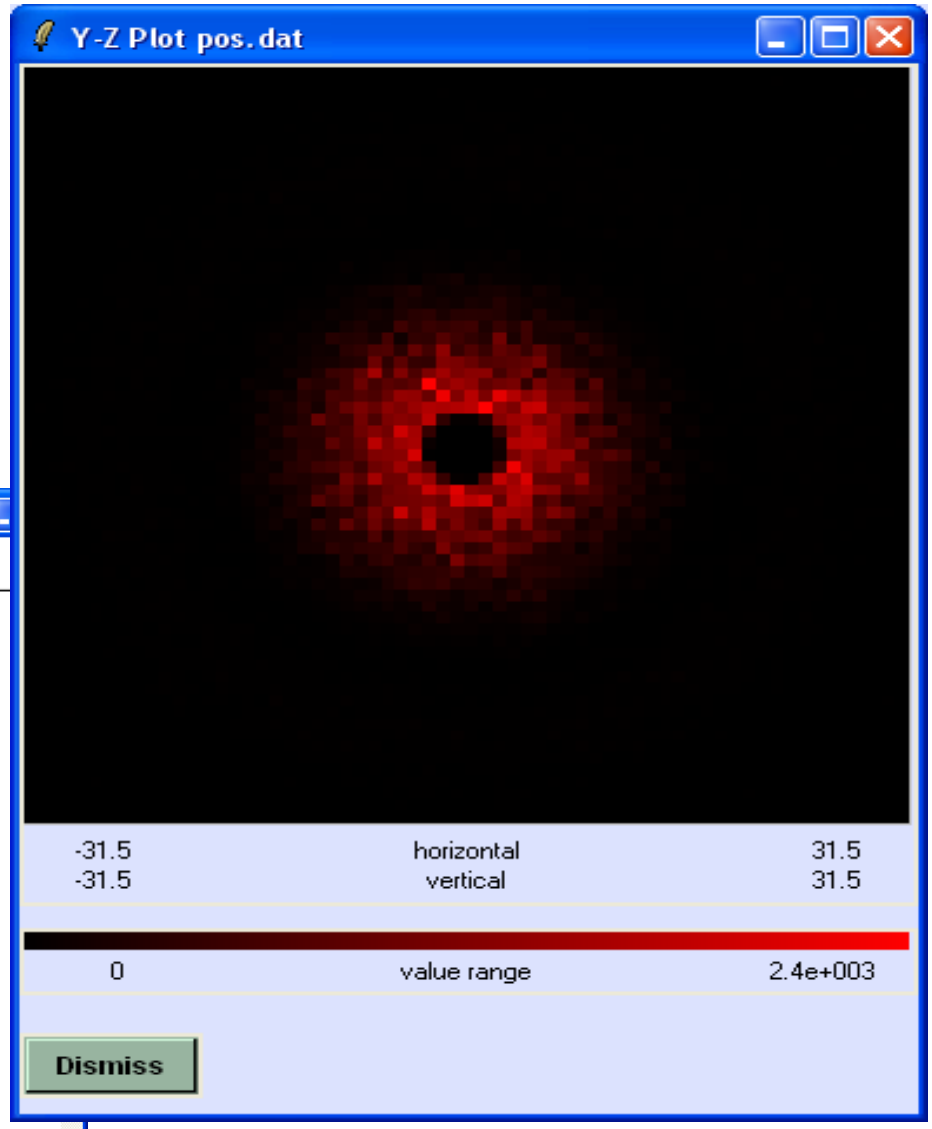


Output Data: Log File and Instrument Data

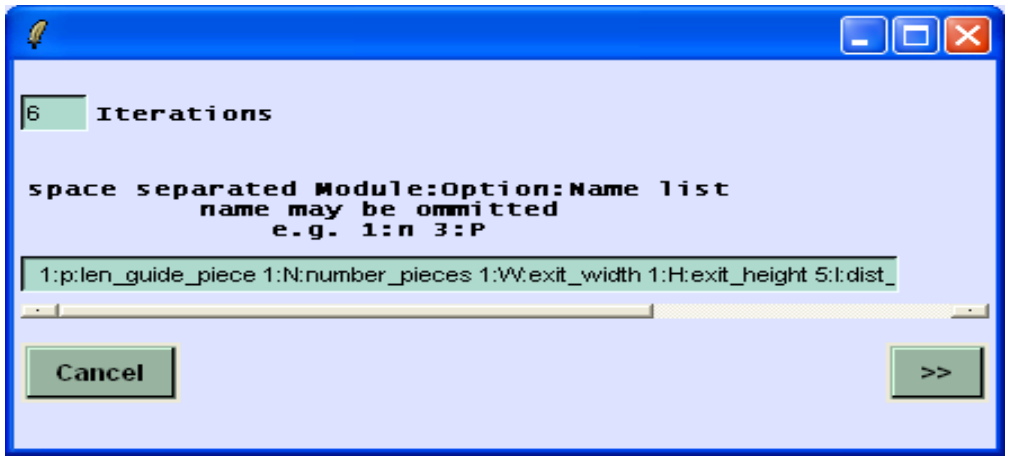


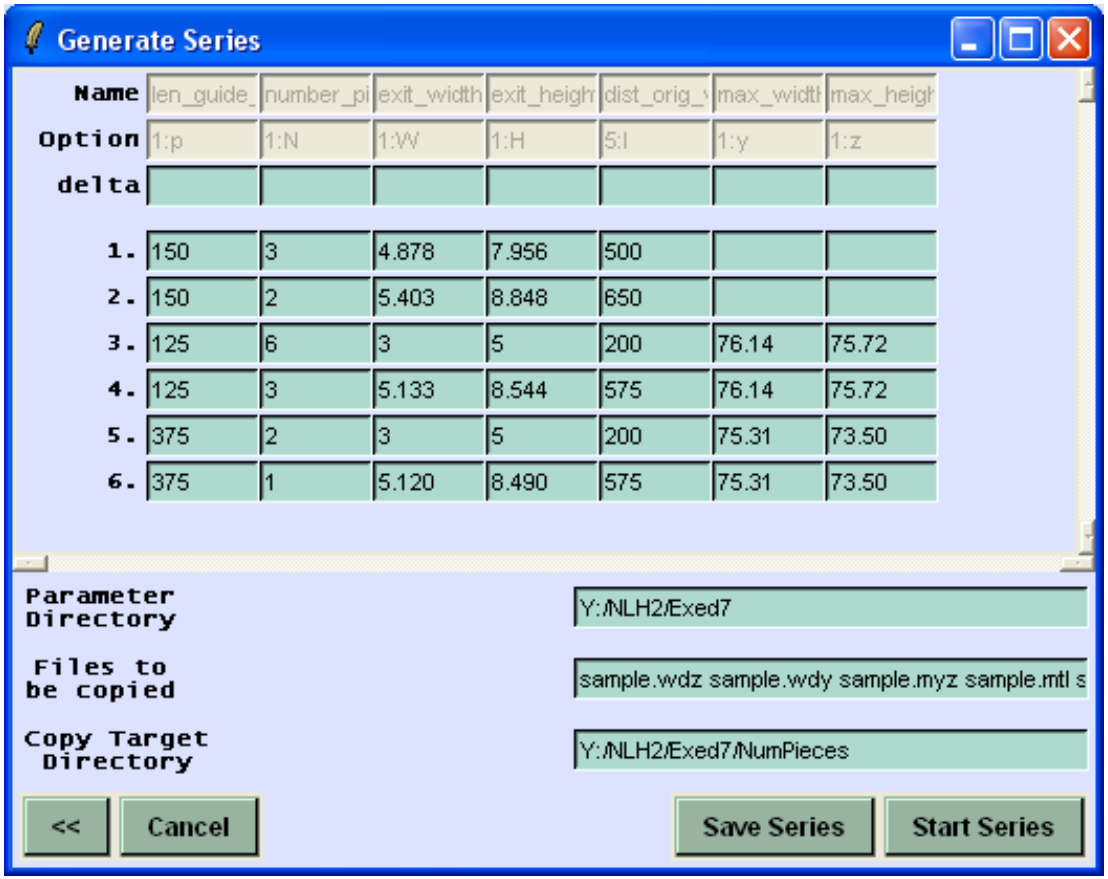
Graphical output from Monitors



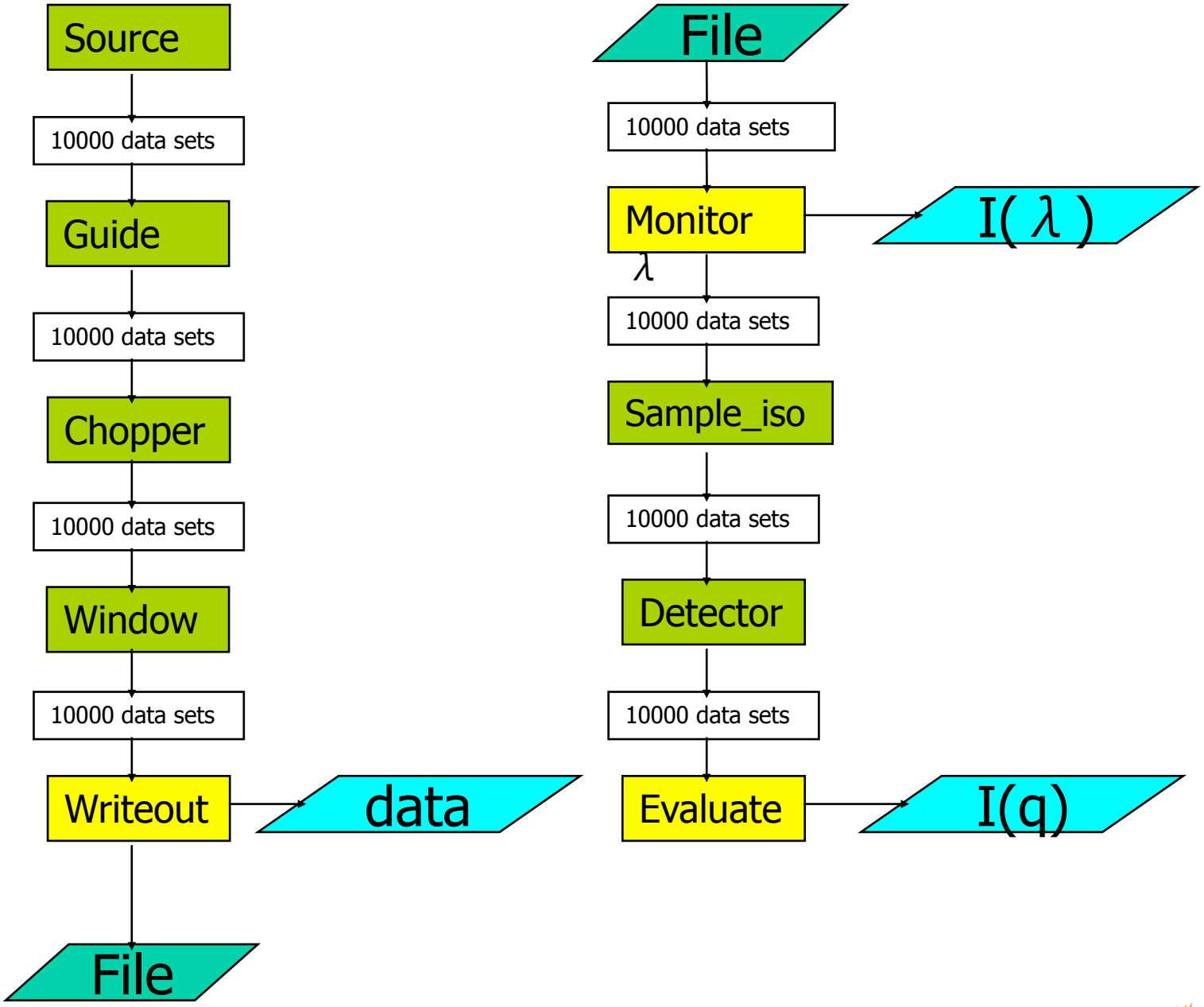


Generate Series





Splitting of the Simulation



Tools

- Most are used to calculate input data

AsciiToBinary	GenerMirrorFile	ChopPhases
DefineDirection	GenerSurfFile	ChopSystem
Direct View	StdDeviation	CrysAnalySpec
GenerateBatch	LatticeDist.	

- Others to visualize output

DistTimePlot	VisualOutput

Parallel Computing

Computer Grids

File|‘Save as Grid Command’

saves the command line in a form that it can be used on computer clusters

Multi-core Processors

Ongoing development

Tests made for

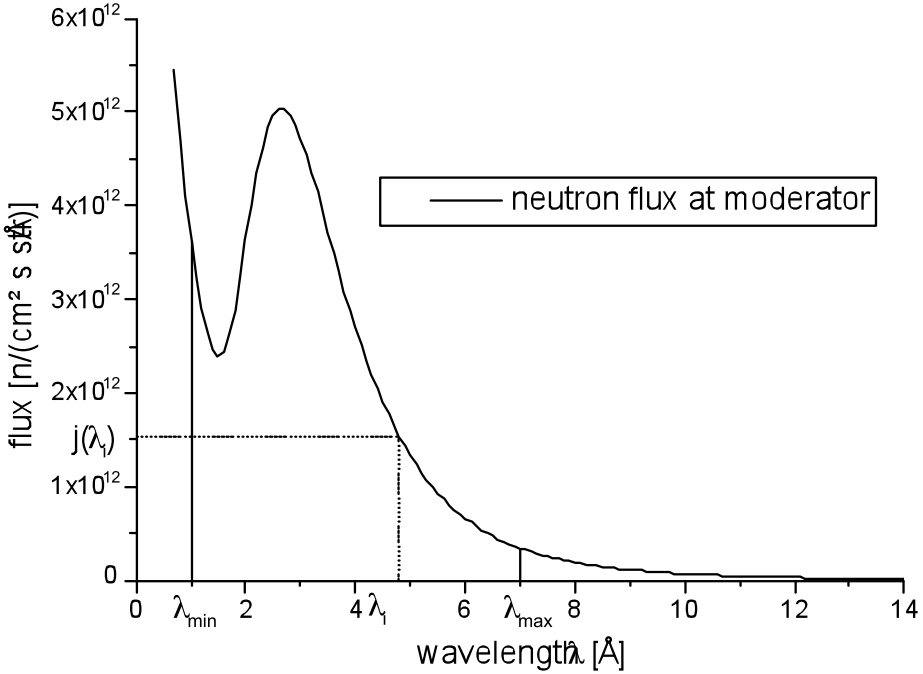
- Guide
- Fermi chopper
- supermirror ensemble
- polariser supermirror

Support

Address to Michael Fromme (fromme@helmholtz-berlin.de)

Thank you for your attention !

Absolute Flux Values



number of trajectories: 2000000

min. wavelength [Å]	1.0	min. time [ms]		max. divergence x <-> y [deg]	0.5
max. wavelength [Å]	7.0	max. time [ms]		max. divergence x <-> z [deg]	0.5

- $I_{CW} = \int j(\lambda)\, d\lambda$

  $\approx (\lambda_{max}-\lambda_{min})/N \sum_i j(\lambda_i)$

- $I_{SS} \approx (\lambda_{max}-\lambda_{min})(t_{max}-t_{min})/N \sum j(\lambda_i,t_i)$

- Each trajectory represents a package of a certain number of neutrons with the same starting conditions
- By statistical processes like reflection, the number of neutrons in the package decreases, while the number trajectories remains unchanged
- If the neutron package does not pass a module regularly, the trajectory is taken out of consideration
- A neutron count rate can be calculated from the number of neutrons in a package
- Summing of the count rates of all packages gives the neutron count rate at any point of the instrument