InSilicoTEM v0.92 reference manual

Miloš Vulović

October, 2013

1. Quick start

This software is an implementation of algorithms described in "Image formation modeling in cryo-electron microscopy" M.Vulović, R.B.G. Ravelli, L.J. van Vliet, A. J. Koster, I. Lazić, U. Lücken, H. Rullgård, O. Öktem, and B. Rieger, *Journal of Structural Biology,* Volume 183, Issue 1, 2013, Pages 19–32

The algorithm to simulate cryo-electron micrographs has been implemented in DIPImage (Version 2.5.1), a publicly available software toolbox (www.diplib.org) for Matlab (The MathWorks, Inc.). DIPimage, TOM toolbox (Nickell et al., JSB 2005) and Matlab (including the Bioinformatics toolbox) need to be installed on the system.

The following input parameters are required for the main function `simTEM`:
1) Interaction potential volume of the specimen
2) List of physical and processing parameters supplied via a structure `params2`

Please have a look at the file **InSilicoTEM.m** which shows how to set all the parameters and calls the functions in the correct order to simulate a TEM image.

2. How to use the program

InSilicoTEM is intended to simulate micrographs by taking into account specimen, microscope and detector characteristics.
The program is run by executing the script **InSilicoTEM.m** which calls the functions **parsePar** (parses all the parameters)**, loadSamples** (loads or generates the interaction potential of individual particle)**, generateFullVolume** (constructs the full specimen volume) and **simTEM** (simulates electron specimen interaction, propagation through the optics and detector influence).

Appropriate parameters must be specified in the **InSilicoTEM.m** script. The parameters are organized in the structure `params` and split into 7 fields:
1. Processing parameters (`params.proc`)
2. Specimen parameters (`params.spec`)
3. Parameters describing electron-specimen interaction (`params.inter`)
4. Microscope aberrations and source incoherency (`params.mic`)
5. Acquisition settings (`params.acquis`)
6. Parameters of the detector (`params.cam`)
7. Parameters specifying what to display (`params.disp`)

For more details about the parameters consult Table1. For description of the functions type **help** followed by the function name in Matlab.

Table1. Description of the input parameters. The fields in blue are dependent on another entry.

Parameters (structure) `params.`	Field	Description
`proc.`	N	Image size is N x N [pix]
	partNum	Number of particles in the volume. It will be limited in the case it exceeds the maximum possible number of particles within the field of view. In case that the number of required particles is larger than available in the subfolder 'Particles', the interaction potential of the remaining particles will be generated.
	geom	Specify orientation and translation of particles in 'PartList.m' (=1) or generate them randomly (=0)
	cores	Number of matlab pools to be open for 'parfor' loop (parallelization). This works with `params.inter.type` ={'pa', 'wpoa', 'pa+wpoa', 'tpga'} (see below)
`spec.`	source	Describes how to obtain the interaction potential (IP). Options: 'map', 'pdb', or 'amorph'
	pdbin	The PDBid from which IP was constructed. In case that pdb is not available in the subfolder 'PDB', it will be downloaded from RCSB website. (Required if params.source = 'pdb')
	mapsample	The name of the file (mrc) from which the potential map is read. It should contain the info about voxel size with following convention '_VoxSize%02.2f' (Required if params.source = 'map')
	potcontribution	Type of the IP. Options: 'iasa' or 'iasa+pb'. In the case of 'iasa+pb' the software packages *pdb2pqr* and *APBS* must be installed in order to calculate IP via Poisson-Boltzmann approach.
	motblur	Motion blur [Å]. Additional damping of the scattering amplitudes that could be related to the beam-induced motion.
	thick	Thickness of the specimen [m].
	imagpot	Flag of the amplitude contrast and imaginary part of the potential (imagpot). Options: (=0, none) (=1 imagpot=constant*IP) (=2 ice plasmons considered) (=3 plasmons of ice and protein considered)
`inter.`	type	Model of electron-specimen interaction. Options: 'pa' (projection), 'wpoa' (weak-phase), 'pa+wpoa', 'tpga'(thick-phase grating) or 'ms'(multislice) *See Vulovic et al., Ultramicroscopy Vol. 136, 2014, pp 61-66*
	msdz	Approximate thickness of the slice for multislice [m] (Required if params.inter.type = 'ms')
`mic.`	Cs	Spherical aberration of the objective lens in [m]

	a_i	Illumination aperture [rad]
	C_c	Chromatic aberration [m]
	deltaE	Energy spread of the source [eV]
	diam_obj	Diameter of objective aperture [m]
	foc	Focal distance [m]
	PPflag	Phase plate flag (=0 no phase plate) (=1 phase plate)
	PP_Phase	Phase plate phase shift [rad] (Required if params.mic.PPflag=1)
	PP_qcuton	Cut-on frequency of the pahse plate [1/m]. (Required if params.mic.PPflag=1)
acquis.	pixsize	Pixel size in the specimen plane [m].
	df	Defocus value [m]. Note: undefocus df>0; overfocus df <0. Note that also that it is possible to simulate tilt , defocus and dose series but only one at the time with following priority order: tilt, defocus, dose.
	ast	Astigmatism [m]
	astangle	Astigmatism angle [rad]
	Voltage	Acceleration voltage of the electron source in [V]
	tilt	Tilt geometry. Specify the range [min:step:max]/180*pi. Put 0 for untilted image.
	dose_on_sample	Integrated flux - dose [e-/$Å^2$]
cam.	type	Type of the detector. Options: 'custom', 'Eagle4k', 'US4000', 'US1000GIF', 'FalconI', 'perfect' (MTF 64% at Nq), 'ideal' (MTF =1)
	bin	Hardware binning
	GenMTFasEMG	(=1 simulates MTF as exponentially modified Gaussian (EMG) without accurate characterization of the detector. MTF is looks more similar to the exponential function for higher voltages and more similar to Gaussian for lower voltages. (=0 you must specify mtf and dqe which should be of a size 256 x 256 pix) (Required if params.cam.type='custom')
	DQEflag	Flag for dqe (=0 means NTF = MTF)
disp	generateWhat	Options: 'im' (noisy image), 'exitw'(exit wave), 'imNoiseless' (noiseless image)

	ctf	Flag to display 1D CTF and to provide ctf in the output structure
	mtfdqe	Flag to display MTF and DQE and to provide them in output structure
Extra parameters (to be changed in parsePar.m)		
cam (lines 32-37 in parsePar.m)	nnps_empirical	Flag (=1 simulate nnps (normalized noise power spectrum) as mtf^exponent in case dqe is not available). Default=1. (Required if params.cam.type='custom' and params.cam.DQEflag=1)
	mtf_exp4nnps	in case dqe is not available define how much better nnps should be compared to the mtf ($0 <= mtf\_exp4nnps <= 1$) Default =3/8 (Required if cam.type ='custom' and cam.DQEflag=1 and cam.nnps_empirical=1)
	dqe0	DQE at zero frequency (Default =0.4) (Required if params.cam.type='custom')
	readn	Readout noise in [ADUs] (Default =3) (Required if params.cam.type='custom')
	darkn	Dark current noise in [ADUs] (Default =0.7) (Required if params.cam.type='custom')
spec. (lines 80-81 in parsePar.m)	voxsize	Voxel size of the initial potential map further resampled to the final pixel size. Default:1. Larger values will speed up the calculations but produce less accurate results.
	imagpot2specm	Type of the specimen to calculate the plasmon damping Options: 'amorIce' (default), 'amorC' and 'graphite' (Required if params.amplconsem = 2 or 3)
	imagpot1Q	Amplitude contrast represented as an imaginary potential which is a fraction of real potential. Typically 0.07 (7 %) for vitreous ice (default) and 0.1 (10 %) for amorphous carbon. (Required if params.amplconsem = 1)

Note:

- Some examples of MTFs, Potential maps, Particle maps and PDBS can be found in subfolders.
- in order to calculate the potential contribution due to the dielectric and ionic properties of the solvent, please install pdb2pqr (http://www.poissonboltzmann.org/pdb2pqr) and APBS (http://www.poissonboltzmann.org/apbs). After calculating potentials via APBS use the functions in subfolder apbsTools.

3. Conditions of use

The Quantitative Imaging Group of the TU Delft has developed and is the owner of the image processing routines distributed in this package, hereafter called SOFTWARE.

The SOFTWARE is free for non-commercial use by students and staff in universities or non-profit research institutes.

Redistribution of SOFTWARE or parts thereof in any form is not permitted.

This SOFTWARE is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.