Rarefied gas flow in gas injection systems (GIS)

The description of rarefied precursor gas flow in gas injection systems requires taking into account that the gas is a discrete system composed of individual molecules, being in constant, random motion. The gas flow is a superimposed ordered motion of molecules. However, at the molecular level, there is no distinction between the random and ordered motion. Probabilistic Monte Carlo simulations are the frameworks which allow to mathematically model molecular and transient flows.

The simulation of molecular and to some extend of transient flow through arbitrary shaped tubing components is ideally suited for a probabilistic simulation called test-particle Monte Carlo (TPMC) method which was first introduced by Davis [D. H. Davis, *J. Appl. Phys.*, 31(7): 1169, 1960]. A large number of serially generated molecule trajectories are computed which collectively predict the flow problem, i.e. the flow-field describing the macroscopic flow properties in the tube, e.g. number density, angular distributions, flux on a surface inserted into the flow, e.g. spatial flux distributions. The **GIS Simulator** program is a three-dimensional implementation of the TPMC method for rarefied gas flow through a circular tube which connects a large vessel containing the gas and a vacuum chamber. More details are given in [V. Friedli. *Focused electron- and ion-beam induced processes: In situ monitoring, analysis and modeling*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2008 (a pdf version is freely available at http://library.epfl.ch/theses/?nr=4036)].

How to use GIS Simulator

The graphical user interface "GISsimulator.exe"[1] developed in GUIDE, the GUI design environment provided with Matlab™, calls the core simulation "PrecursorDistribution_XcY.exe" which is written in C++. The input parameters read by the application are passed via the "parameter.txt" file located in the "SimulationParameters" directory. Do not change the folder structure and keep the files: "GetConfiguration_1c5.exe", "GISsimulator.exe", "GISsimulator.ctf", "PrecursorDistribution_1c5.exe", "About.jpg", "HelpConfig.jpg", "UseRestrictions.jpg" in the same root directory.

The output data is written to several .txt files in the "Data" directory in the format specified below. The files are readily edited and adapted to any post processing program. The Matlab™ script "GISsimulator_DataProcessing.m" serves as an example of how to read the .txt files and plot some of the output data. This script saves the Matlab workspace in "<..>Config<..>_workspace.mat" which can later be imported using the load command for further data processing.

Alternatively, simulations can be performed by editing "parameter.txt" manually and executing the program "PrecursorDistribution_XcY.exe" directly.

[1] If you do not have MATLAB installed on the target machine and you want to run "GISsimulator.exe" which was created by the MATLAB Compiler product, you still need to install the MATLAB Compiler Runtime (MCR) library (Version 7.6) on the target machine. You can install the MCR by running "MCRInstaller.exe". You have to install the MCR only once.

PrecursorDistribution_1c5.exe : Output files

Simulation summary and progress information:

<..> Simulation_Summary.txt

This file summarizes the simulation parameters. It is used to load a previously performed simulation run.

<..> Progress.txt

Progress information of the latest performed simulation run.

Configuration description:

<..>Configuration.txt

Keypoints which define the tube – substrate configurations in the x_s-z_s plane

Column 1	The x_s coordinate
Column 2	The y_s coordinate (not used)
Column 3	The z_s coordinate

The impinging precursor flux sampling output files:

<..>Image_Config<..>.txt

Molecule count $n_{i,j}$ within the sampling field on the substrate: rows along x_s, columns along y_s

<.. >Profiles_Config<..>.txt

Column headings:

`Position`	The x_s coordinate in the sampling field at $y_s = 0$.
`Counts`	Impingement counter on pixel i,j : $n_{i,j}$
`CountsCoeff`	Impingement coefficient (probability 0..1) on pixel i,j: $n_{i,j}/n$, where n are the total transmitted molecules (TR)
`Sterr`	Standard error on the impingement coefficient. See PhD thesis V. Friedli p.89.
`SterrRel`	Relative standard error in %. See PhD thesis V. Friedli p.89.
`FluxCoeff`	Flux coefficient (probability 0..1): $n_{i,j}/n*A/A_{i,j}$, where A and $A_{i,j}$ are the cross-section of the tube and the area of a pixel, respectively. The local flux impinging on pixel i,j is $J_{i,j}$ = `FluxCoeff`$*J_{tot}$, where J_{tot} is the total flux leaving the tube.

The “pressure” distribution sampling output files:

<..> PressureDistribution.txt

Column headings:

`Position`	The z coordinate along the tube axis divided by equispaced sampling planes.
`TRflux`	Counter of molecules which have crossed a sampling plane in the direction towards the tube exit.
`BSflux`	Counter of molecules which have crossed a sampling plane in the direction towards the tube entry.
`Netflux`	`TRflux - BSflux`
`BSfluxNorm`	`BSflux` normed by the number `BS` of backscattered molecules.
(`MFPsamplingNorm`)	
(`MFPsectionCount`)	
(`TubeCollCounter`)	

(CollisionCounter)

The angular distribution sampling output files:

<..> AngularDistributionHistogram.txt

Column headings:
(Histogram data in terms of zenith angles (0..180°) and azimuth angles (0..360°))

Angle	The angle in degree
ZenithLMC, AzimuthLMC	For molecules which leave the tube exit after having undergone an intermolecular collision (“Last Molecule Collision”)
ZenithLTC, AzimuthLTC	Molecules which leave the tube exit after having undergone a wall collision (“Last Tube Collision”)
ZenithDFT, AzimuthDFT	For molecules which leave the tube exit without having undergone an intermolecular or wall collision (“Direct Flight Through”)
ZenithAll, AzimuthAll	For all molecules leaving the tube.

The results statistics output files:

<..> Statistics_Tube.txt

Counters for molecules inside the tube:

TR	Molecules which transmit through the tube
TRH	Molecules which transmit through the access hole
BS	Molecules which leave the tube through the tube entry (backscattered)
UT	Molecules which are taken up by the tube wall.
TC	Tube collisions
MC	Intermolecular collisions
MolNum	TR+TRH+BS+UT
TR_LMC	Molecules which leave the tube exit after having undergone an intermolecular collision (“Last Molecule Collision”)
TR_LTC	Molecules which leave the tube exit after having undergone a wall collision (“Last Tube Collision”)
TR_DFT	Molecules which leave the tube exit without having undergone an intermolecular or wall collision (“Direct Flight Through”)

<..> Statistics_Substrate_Config <..> .txt

Counters for molecules having been transmitted through the tube:

LIS	Molecules which are “Lost In Space”, not impinging on the substrate
LOS	Molecules which are “Lost On Substrate”, not impinging in the sampling field
DB	Molecules which re-enter into the tube after (cosine-)desorption from the substrate plane.

<..> Statistics_export_Config <..> .txt

Used in Matlab script.

Format: TR TRH BS UT TC MC MolNum LIS LOS TR_LMC TR_LTC TR_DFT DB