2022 SYLLABUS for course PHYD57, Advanced Computational Methods in Physics
Lecturer: prof. Pawel Artymowicz (pawel@phys.utoronto.ca); please put PHYD57 in the subject line and make sure the address is as shown, otherwise mail may be misplaced and not answered.

Lectures: (L1-L12, 2 hrs with 10 min break) on Tuesdays 14:00-16:00
Tutorials (T1-T10) on Tuesdays, 17:00-18:00, on days listed below.
Meetings on zoom, login via Quercus.
Deadlines for 4 sets of assignments/projects are denoted A1-A4 at 2p.
Expect to see them posted 7-14 days before the deadline.

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>L4</td>
<td>T3</td>
<td>A1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>mid-term, L7</td>
</tr>
<tr>
<td>18</td>
<td>L2</td>
<td>L1</td>
<td>T1</td>
</tr>
<tr>
<td>25</td>
<td>L3</td>
<td>T2</td>
<td>A2</td>
</tr>
<tr>
<td>22</td>
<td>reading week</td>
<td>22</td>
<td>L10, T8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29</td>
</tr>
</tbody>
</table>

*) midterm in class, 1st hour of lecture 7 (14:05-15:00) on 1 March

Syllabus is subject to small changes. Please download the updates every week.

1 Structure and scope of the course
   Syllabus of PHYD57
   Numerical Comp. in Physical Sciences: History and Contemporary efforts

2 HPC: need for speed, why and how
   History and modernity: microprocessors, Unix, Linux, and Internet
   Intro to Linux (CentOS)
   Connectivity (ssh & sftp, traceroute & ping)
   Securing your system against break-ins via /etc/hosts.deny
   Basic Linux commands (cd, ls, ps, cd, & bg, fg, alias, setenv, output redirection to file via >, rm)
   Getting more info: manual pages (man), -h, --help, or -help modifiers
   Recommended compilers: GNU: gcc, gfortran; Intel: icc, ifort;
   PGI: pgcc, pgf95
   Simple program in C, Fortran95, Python, Matlab and IDL
   (Schoerghofer book p.30)
   More complex program, example of HPC:
   2nd order Laplace operator stencil for diffusion equation
   Speed comparison of C/F95 with Python & Numpy: why we learn HPC

3 C and Fortran 95 - compilers, basic usage
   Numerical puzzle of 711 - learning C and Fortran
   Kruskal counts and their connection to linked lists
   Coding Kruskal counts trick in Python and Fortran
   C: Language overview, compilers
   Integration with Python: calling C from Python examples of programs

4 More Fortran
   Examples of programs: Init. value problems for ODEs
   More C
   Parallel execution of programs on CPU and MIC
   OpenMP in Fortran and C
   Parallel implementations: diffusion and wave equation
   Modern computing (continued)

5 Multi-dimensional arrays in C vs. Fortran
   Bottlenecks: Computation vs. CPU-RAM bandwidth
   An example program in C and Fortran.
   Parallelization via OpenMP
   Automatic vectorization and compiler reports
   Segmentation faults due to limited stack

6 Calling C functions from Python

7 Computations on GPUs with CUDA
   Examples in C and Fortran
   N-body and other problems of computational physics
   Fourier transforms and FFT
   Bayesian methods of statistics: Markov chain Monte Carlo
   Establishing orbits of extrasolar planets
   Numerical Comp. in Physical Sci: Particle disks on MIC cluster
   Interaction of protoplanets with disks

8 Fluids by Eulerian vs. Lagrangian methods
   Optically thick disk calculation (IRI)
   Smoothed Particle Hydrodynamics: theory
   Discussion of projects

9 Bayesian methods of statistics: Markov chain Monte Carlo
   Establishing orbits of extrasolar planets
   Numerical Comp. in Physical Sci: Particle disks on MIC cluster
   Interaction of protoplanets with disks

10 M-body integration methods and implementations
   Introduction to MPI and SPH
   Fluids by Eulerian vs. Lagrangian methods
   Optically thick disk calculation (IRI)
   Smoothed Particle Hydrodynamics: theory
   Discussion of projects

11 Fluid computations on CPU and in CUDA C
   Planet in a 3d disk
   Optically thick disk calculation (IRI)
   Smoothed Particle Hydrodynamics: theory
   Discussion of projects

12 SPH Implementation. Linked lists, nearest neighbor search
   Can one simulate pandemic?
   Machine Learning, Artificial Intelligence, Neural Networks
   Optimum Search: Simplex Nader-Mead
   Why NNs work despite dimensionality curse
   Discussion of projects