Inverse Problems: Systematic Integration of Data with Models under Uncertainty
2018 Gene Golub SIAM Summer School

Lab sessions

1. Software

We will use these libraries for the hands-on interactive learning exercises that complement the morning lectures:

- hIPPYlib (Inverse Problems with Python libraries) implements state-of-the-art scalable algorithms for PDE-based deterministic and Bayesian inverse problems. It builds on FEniCS (a parallel finite element element library) for the discretization of the PDEs and on PETSc for scalable and efficient linear algebra operations and solvers. hIPPYlib's documentation and tutorial examples can be found at https://hippylib.github.io.

- MUQ (MIT Uncertainty Quantification) provides tools for exact sampling of non-Gaussian posteriors, approximating computationally intensive forward models, implementing integral covariance operators, characterizing predictive uncertainties, and defining the Bayesian models required for these tasks. MUQ's documentation and tutorial examples can be found at http://muq.mit.edu/.

We will also make use of the following software:

- Python 3.5, a high-level programming language
  https://docs.python.org/3/
- Numpy, Scipy, and Matplotlib, three Python packages that offer similar functionality to Matlab
  http://www.numpy.org/; https://www.scipy.org/, and http://matplotlib.org/
- Jupyter notebooks, a convenient way to write, run and document Python code using your web browser
  http://jupyter.readthedocs.io/en/latest/index.html
- Docker, a software containerization platform that provides the simplest way to run our software stack on your computer
  https://www.docker.com/
- FEniCS, the parallel finite element element library used by hIPPYlib
  https://fenicsproject.org/

2. Cloud resources

We will use cloud-based interactive tutorials that mix instruction and theory with editable and runnable code. You can run the codes presented in the hands-on workshop through your web browser. This will allow you to test our software and experiment with inverse problem algorithms quite easily, without running into installation issues or version discrepancies. Cloud computing resources for the summer school were made available by the Extreme Science and Engineering Discovery Environment (XSEDE) Jetstream through allocation TG-DMS180009, which is supported by National Science Foundation grant number ACI-1548562.

Cloud resources will be available for the entire duration of the summer school, and for a few weeks after the school itself to allow you experimenting with our software on your own. Log in information will be posted on our slack page. Please do not exchange the user info.

3. Local installation

As a precaution, we also suggest you to download and install Docker on your laptop so that you can also run the interactive tutorials locally in case of poor internet connection or issues with the cloud resources.

- **MacOS El Capitan 10.11 or above**
  1. Download `Docker for Mac` from https://download.docker.com/mac/stable/Docker.dmg.
  2. Open a new terminal shell.
  3. Type the following command

     ```
     docker pull mparno/muq-hippylib
     ```

  4. For additional resources, see the `Docker for Mac` tutorial https://docs.docker.com/docker-for-mac/
- **MacOS Yosemite 10.10 or below**
  1. Download `Docker Toolbox` from https://download.docker.com/mac/stable/DockerToolbox.pkg
  2. Double click on `Docker Quickstart Terminal`. A Linux-like shell will open.
  3. Type the following command

     ```
     docker pull mparno/muq-hippylib
     ```

  4. For additional resources, see https://docs.docker.com/toolbox/toolbox_install_mac/
- **Windows 7 or above**
  1. Download `Docker Toolbox` from https://download.docker.com/win/stable/DockerToolbox.exe
  2. Double click on `Docker Quickstart Terminal`. A Linux-like shell will open.
  3. Type the following command

     ```
     docker pull mparno/muq-hippylib
     ```

  4. For additional resources, see https://docs.docker.com/toolbox/toolbox_install_windows/

4. FEniCS overview

FEniCS is a powerful, open-source suite of tools for automated solution of PDEs using finite elements. Part of the power for FEniCS is the ease with which one can create FE solvers by describing PDEs using weak forms in nearly-mathematical notation. The FEniCS project can be found at: http://fenicsproject.org/ and comes with an extensive documentation and examples.

FEniCS includes a number of powerful features that include:
- Automated solution of variational problems;
- Automated error control and adaptivity;
- An extensive library of finite elements;
- High performance linear algebra through backends to such libraries as PETSc and Trilinos;
- Visualization via a simple interactive plotting function, as well as output in VTK format;
- Python and C++ interfaces;
- Extensive documentation (see for instance: https://fenicsproject.org/documentation/).

4.1. FEniCS resources

The documentation for FEniCS is extensive. Resources include:

- **The FEniCS Tutorial.** The book *Solving PDEs in Python: The FEniCS Tutorial Volume I* is the is the perfect guide for new users. The tutorial explains fundamental concepts of the finite element method, FEniCS programming, and demonstrates how to quickly solve a number of PDEs.

  The PDF version of the book can be downloaded (legally and for free) from https://fenicsproject.org/pub/tutorial/pdf/fenics-tutorial-vol1.pdf.

  Python scripts for all the examples described in the tutorial can be found at https://github.com/hplgit/fenics-tutorial/tree/master/pub/python/vol1

- **FEniCS Demos.** These documented demonstration programs are a great way to learn the different features in FEniCS. They come already packaged in FEniCS when you install it and are available on-line at:
  http://fenics.readthedocs.io/projects/dolfin/en/2017.2.0/demos.html.

- **Quick Programmer's References.** Some of the classes and functions in DOLFIN are more frequently used than others. The Python implementations are described in https://fenicsproject.org/docs/dolfin/2017.2.0/python/quick_reference.html.

- **Complete Programmer's References.** If you need more details on a particular class or function, you can also consult the Complete Programmer's References at https://fenicsproject.org/docs/dolfin/2017.2.0/python/programmers-reference/index.html .

- **Getting Help.** See: https://fenicsproject.org/community/

Other resources, although a little outdated and not fully compatible with the latest versions of FEniCS, include:

- **The FEniCS Book:** All 732 pages of the FEniCS book (*Automated Solution of Differential Equations by the Finite Element Method*) can be downloaded (legally!) from here: http://launchpad.net/fenics-book/trunk/final/+download/fenics-book-2011-10-27-final.pdf. This is the comprehensive reference to FEniCS, along with many examples of the applications of FEniCS to problems in science and engineering. You will notice that the first chapter of the book is just the FEniCS Tutorial (with some minor editorial differences).

- **The FEniCS Manual.** This is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method, and documentation of DOLFIN and UFL: http://launchpad.net/fenics-book/trunk/final/+download/fenics-manual-2011-10-31.pdf. Since it's an excerpt from the FEniCS Book, you probably won't need it.