MATH920: Continuum mechanics and machine learning

An exploration of the synthesis of continuum mechanics with stochastic machine learning

Course syllabus

Times	MF 2:30-3:20PM, Phillips 385
Office hours	MF 12:30-2:00PM, and by email appointment, Chapman 451
Instructor	Sorin Mitran

1 Motivation

This directed reading course investigates the theory of continuum mechanics from both a classical and contemporary perspective. Classical continuum mechanics is usually presented using the tools of differential calculus and provides a complete description for linear media with no memory effects, as exemplified most prominently by the Cauchy elasticity equations. While sufficient for the purposes of traditional mechanical engineering relying on small deformation of crystalline metals, vast classes of important materials encountered in plastics processing, paper processing, non-Newtonian flow or biological materials are only awkwardly described through the mathematical apparatus of partial differential equations (PDEs) mainly due to four causes:

Nonlinearity. Many materials of practical contemporary interest exhibit intrinsic nonlinear behavior, typically arising from a *mesoscopic* underlying structure instead of a *microscopic* one. Whereas the mechanical behavior of metallic monocrystals arises from electrostatic interaction acting at the length scale of the crystalline lattice of $\ell \sim 10^{-10}$ m along all directions, cytoskeleta exhibit an underlying structure with an average actin filament length of $a_1 \sim 10^{-8}$ m with a radius of $r_1 \sim 10^{-9}$ m, while muscle tissue fibers are of length $a_2 \sim 10^{-2}$ m with a radius of $r_2 \sim 10^{-5}$ m. When averaging out, or *homogenizing*, the behavior of the constituent elements of the medium to a macroscopic length scale of practical interest, say $L \sim 10^{-2}$ m, the large number of interactions in materials with microscopic structure often leads to isotropic and linear partial differential equations, but anisotropic and nonlinear behavior is obtained for materials with mesoscopic structure.

Active behavior. In classical continuum mechanics the underlying microscopic structure is typically considered to be fixed in time, a good approximation of the slow rate of chemical reactions (e.g., oxidation of iron) within the range of materials of interest. In contrast, conformational changes in flowing polymers or cellular dephosphorylation of ATP into ADP leads to markedly different mechanical behavior in viscoelastic flow or cellular motility. Such materials are said to be *active*, and typically exhibit very complex and incomplete mathematical descriptions within the framework of differential equation theory.

Stochastic behavior. Active materials tend to rearrange their mesoscopic structure, typically under the influence of random excitation (i.e., thermal bath) from the surrounding medium. This leads to the need for stochastic processes to describe the response of the medium to application of external forces.

Memory effects. The reorganization of mesoscopic structures due to chemical activity or external forces often occurs at time scales longer than those of observation, such that the prior history of the medium influences the observed behavior. Such memory effects can be modeled by differential equations of fractional order (i.e., integro-differential equations), and when combined with random thermal forcing require the consideration of non-Markovian stochastic processes.

The classical approach to address the above features of complex materials has been to posit algebraic or differential constitutive relations between deformation and stress, and investigate the validity of the hypothesized models. An alternative suggested by current progress in machine learning is to extract the constitutive relations directly from data.

2 Course topics

TEN. Tensor algebra and calculus

KIN. Kinematics of point masses, rigid bodies, deformable bodies

CON. Conservation laws (mass, momentum, energy) applied to continua

ELS. Elasticity, reversible small-amplitude deformations of a solid

PLS. Plasticity, irreversible deformation of a solid

NSF. Navier-Stokes flow of a Newtonian fluid

VEF. Viscoelastic flow

NET. Fiber networks

ACT. Active media

LSQ. Review of least squares approximation, seen as a particular example of machine learning

ANN. Review of artificial neural networks, seen as a particular example of approximation theory

DNN. Deep neural network models of constitutive equations:

- **DNN-ELS.** Classical elasticity theory recovered from a deep neural model of linear deformation
- **DNN-PLS.** Deep neural model of plastic deformation
- **DNN-NSF.** Navier-Stokes equations recovered from deep neural model of Newtonian flow
- **DNN-VEF.** Deep neural models for viscoelastic flow
- **DNN-NET.** Deep neural models of fiber networks

SNN. Stochastic neural network models of active media

- **SNN-VEF.** Polymeric fluids with changing connectivity, reptation.
- **SNN-NET.** Fiber networks with changing connectivity
- **SNN-ACT.** Fiber networks with active elements

3 Course materials

3.1 Bibliography

There is no single course text. Topics are drawn from the following sources that are available in electronic form from the course repository or UNC library.

- *Machine Learning and Continuum Mechanics*, S. Mitran
- *Continuum Mechanics: Foundations and Applications of Mechanics*, C.S. Jog
- *Continuum Mechanics*, Andrus Koppel and Jaak Oja
- *Elements of continuum mechanics*, R.C. Batra
- *Deep Learning*, I. Goodfellow and Y. Bengio and A. Courville

Additional material is available in the course material repository (`/biblio` subdirectory)

3.2 Slides

Lessons	Topics
Lesson01	TEN, KIN, CON
Lesson02	ELS
Lesson03	DNN-ELS
Lesson04	PLS
Lesson05	DNN-PLS
Lesson06	NSF
Lesson07	DNN-NSF
Lesson08	VEF
Lesson09	DNN-VEF
Lesson10	NET
Lesson11	DNN-NET
Lesson12	SNN-NET
Lesson13	ACT
Lesson14	SNN-ACT

3.3 Data sets

The following codes generate data sets for various continuum modles

Nr.	Topic	Problems	Solutions
1	LSQ, TensorFlow	Homework01	
2	ELS	Homework02	
3	PLS	Homework03	
4	NSF	Homework04	
5	VEF	Homework05	
6	NET	Homework06	
7	ACT	Homework07	

4 Computational resources

Though the course concentrates on concepts within continuum mechanics and machine learning, the computational implementation of these concepts is essential to an appreciation of the utility of the considered approaches. Templates are provided for all computational applications, typically comprising:

1. Generation of data for constitutive relations;
2. Definition of deep neural networks that approximate the constitutive relation data;
3. Numerical methods (finite element, finite volume) that solve the classical formulations of continuum mechanics, e.g., Cauchy elasticity equations, Navier-Stokes flow equations.

4.1 SciComp@UNC Linux environment

This course uses a customized Linux environment named SciComp@UNC available to students as a virtual machine in which all course software is preinstalled, and course applications are preconfigured. Download Virtual Box and the SciComp@UNC virtual machine image.

Various software tools for carrying out and documenting practical scientific computation will be successively introduced:

- TeXmacs: editing of documents containing live computation
- SciPy: scientific Python environment
- TensorFlow: machine learning platform accessible from Python
- Mathematica: system for numerical, symbolic, and graphical computation with DNN support
- Gnu compilers (Fortran, C++): high-performance compiled code development

- Julia: a high-performance interactive environment
- Paraview: data visualization
- BEARCLAW: a package for solving PDEs using finite volumes
- FreeFEM++: a package for solving PDEs using finite elements

The Mathematica commercial package is accessible to students while connected to the campus network (either directly or remotely through the UNC VPN server).

4.2 Course material repository

Course materials (lecture notes, workbooks, homework, examination examples) are stored in a repository that is accessed through the subversion utility, available on all major operating systems. The URL of the material is http://mitran-lab.amath.unc.edu/courses/MATH768.

The initial svn checkout is made using commands:

```
mkdir ~/courses
cd ~/courses
svn co svn://mitran-lab.amath.unc.edu/courses/MATH768
```

On SciComp@UNC the initial checkout can be carried out through the terminal commands:

```
cd ~/courses
make MATH768
```

Update the course materials before each lecture by:

```
cd ~/courses/MATH768
svn update
```

Links to course materials will also be posted to this site, but the most up-to-date version is that from the subversion repository, so carry out the svn update procedure prior to each lecture.