

# The Role of Bioreactors in Tissue Engineering

And Using Jupyter Notebooks to Teach the Fundamental Concepts of Bioreactors  
and Mass Transfer More Effectively

Mojtaba Barzegari

Biomechanics Section, Department of Mechanical Engineering

November 2018

# 0 Outline

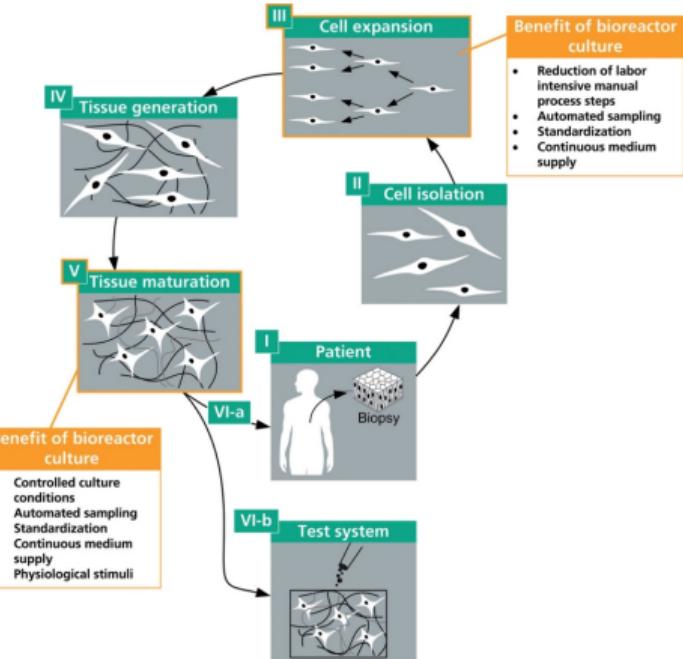
- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

# 1 Outline

- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

# 1 Tissue Engineering

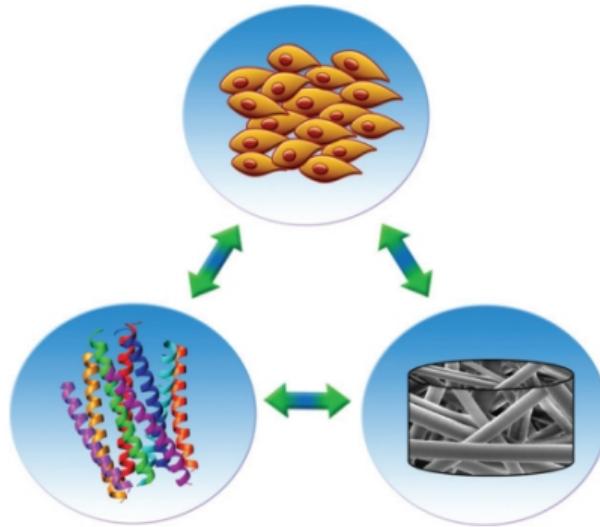
Tissue Engineering (TE) focuses on developing ideal tissue replacements that adequately mimic, or are functionally equivalent to, native tissues.



(Jan Hansmann et al., 2013)

# 1 Tissue Engineering

TE aims to develop methods and technologies to create tissue constructs *in vitro* that have tissue-specific **morphological, biological, chemical, and mechanical** properties, as well as **functions** similar to those found *in vivo*.



TE Paradigm: Cells, Scaffolds, and Signals

(E. J. Levorson, 2011)

# 1 TE Fundamental Challenges

Major obstacles to the generation of functional tissues:

- ▶ Limited understanding of physicochemical culture parameters
- ▶ High manufacturing costs

Solutions:

- ▶ Enabling reproducible and controlled changes of environmental factors
  - Technological means to reveal fundamental mechanisms
  - Improve the quality of engineered tissues
- ▶ Automating and standardizing tissue manufacture in controlled systems

(I. Martin et al., 2004)

## 2 Outline

- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

## 2 Bioreactors

Generally defined as devices in which biological and/or biochemical processes develop under closely monitored and tightly controlled environmental and operating conditions, such as:

- ▶ pH
- ▶ Temperature
- ▶ Pressure
- ▶ Nutrient supply
- ▶ Waste removal

## 2 Role of Bioreactors

*Ex vivo* engineering of 3D tissues:

- ▶ Cell seeding
- ▶ Nutrition supply
- ▶ Stimulation as a guide



Prometheus Perfusion Bioreactor

## 2 Cell Seeding

Cells should be seeded with the highest possible efficiency (**density** and **uniformity**).

- ▶ Static cell seeding
  - Low seeding efficiencies
  - Non-uniform distributions of cells
  - Diffusion is the main source of transport
- ▶ Dynamic cell seeding
  - Using principle of convective transport
  - Using direct flow perfusion
  - Yielding more-uniformly seeded scaffolds

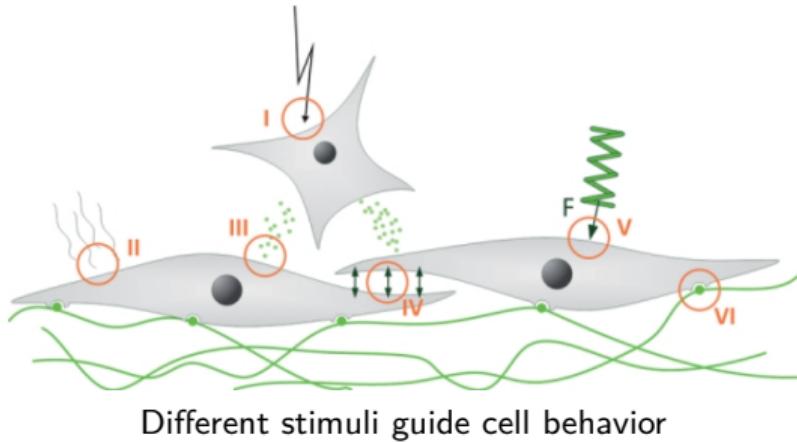
## 2 Nutrition Supply (Increase Mass Transfer)

- ▶ Perfusion of medium → increasing the mass transport of nutrients and oxygen
- ▶ The effects of direct perfusion depends on:
  - Medium flow-rate
  - The maturation stage of the constructs
- ▶ The optimal operation conditions of a bioreactor should not be determined through a trial-and-error approach ⇒ Quantitative Models

## 2 Stimulation for Guiding Tissue Structure

In a complex environment, a cell is exposed to

- i. electrical
- ii. electromagnetic
- iii. biochemical
- iv. cell-cell interactions
- v. mechanical forces
- vi. cell-matrix interactions

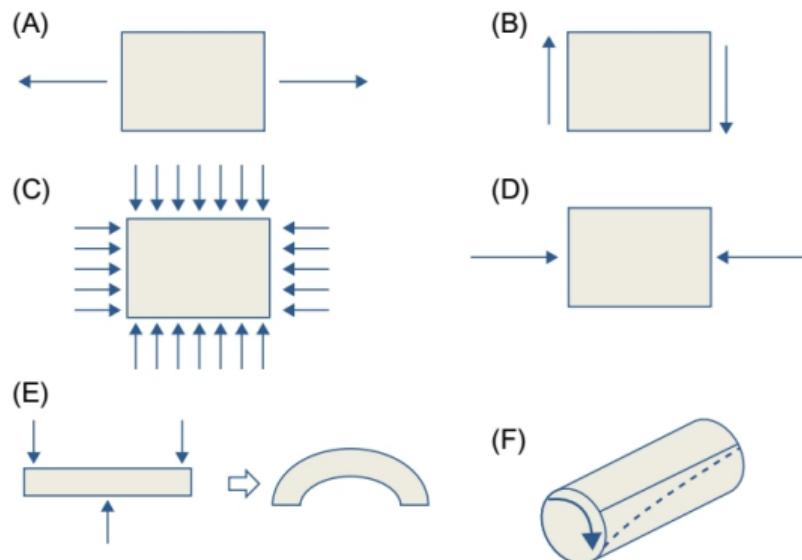


(J. Hansmann et al., 2013)

## 2 Mechanical Stimulation

Loading conditions used in bioreactors for tissue engineering:

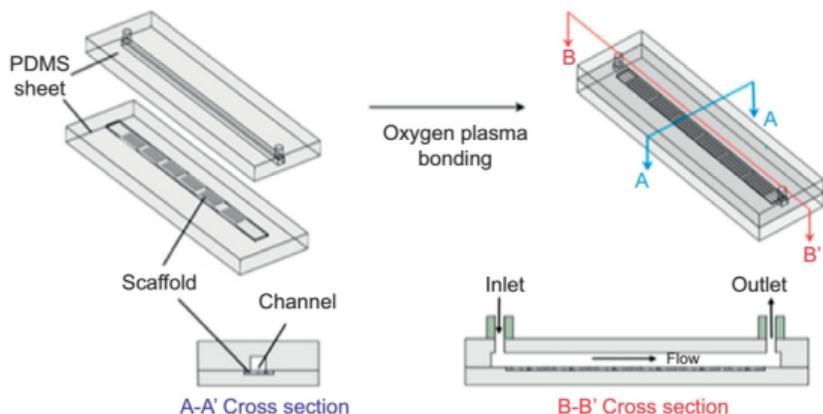
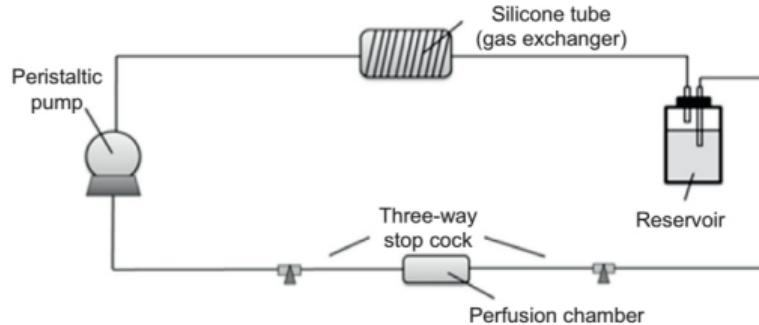
- A. Uniaxial stretch
- B. Shear stress
- C. Pressure
- D. Compression
- E. Bending
- F. Torsion



(K. J. Bløse et al., 2014)

## 2 Shear Stress

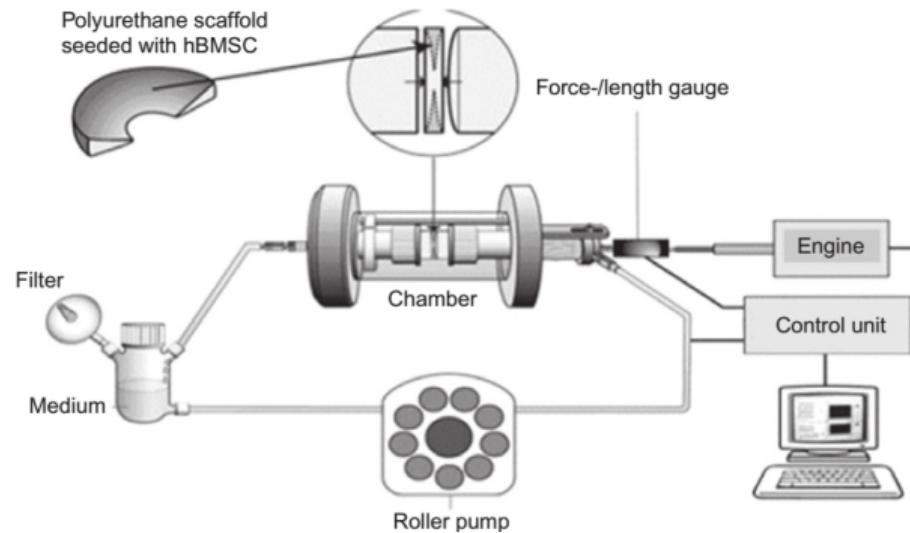
Schematic illustration of a bioreactor providing shear stress via parallel flow



(K. J. Blosea et al., 2014)

## 2 Pressure

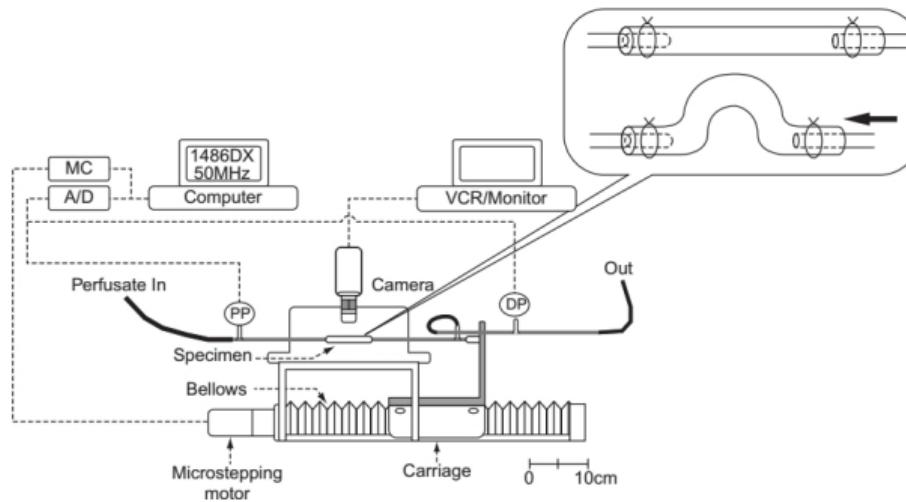
Schematic illustration of a cyclic pressure bioreactor.



(K. J. Blosea et al., 2014)

## 2 Bending

Schematic illustration of a cyclic bending bioreactor.

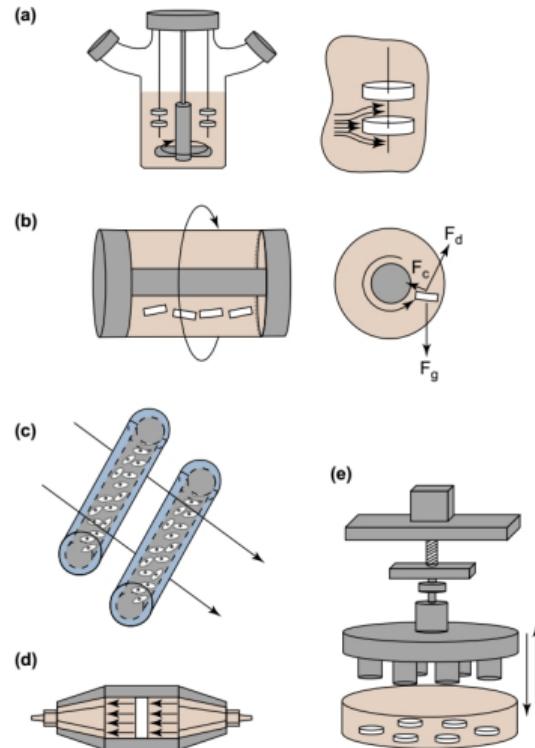


(K. J. Bløse et al., 2014)

## 2 Different Types of Bioreactors

- a. Spinner flask
- b. Rotating-wall vessels
- c. Hollow-fiber
- d. Direct perfusion
- e. Controlled mechanical forces

(I. Martin et al., 2004)



## 2 Features of Different Categories

- ▶ Spinner flask bioreactors
  - Dynamic cell seeding
  - Less control on the level of shear stress
  - Manual waste removal
  - Non-uniform velocity field
- ▶ Rotating-wall bioreactors
  - High rate of mass transfer
  - Relatively low shear stress conditions can be generated
- ▶ Perfusion bioreactors
  - Taking advantage of convective fluid flow but avoiding high levels of shear stress
  - Effective seeding process and nutrient supply
  - Automatic waste removal

## 2 Scaffolds

Scaffolds fill critical roles in regenerative medicine:

- 1 Provide anatomic fill and shape in tissue defects
- 2 Provide temporary function in anatomic defects while tissue regenerates
- 3 Enhance tissue regeneration through material-cell interaction and delivery of biologics, including cells, proteins, and/or genes

(S. J. Hollister, 2011)

## 2 Roles of Scaffolds - 4F

- ▶ **Form:** Fill anatomic defect shape
- ▶ **Fixation:** The mode of surgical fixation of the scaffold
- ▶ **Function:** Defining temporary function
- ▶ **Formation:** Enhance tissue regeneration

(S. J. Hollister, 2011)

## 2 Hierarchical Computational Scaffold Design

To compute scaffold structure-function relationships, one should solve appropriate field equations at the architecture level:

- ▶ Effective elasticity → stress equilibrium
- ▶ Effective permeability → creeping Stokes fluid flow
- ▶ Effective diffusivity → local diffusivity

Finite Element Method (FEM)  
Computational Fluid Dynamics (CFD)  
Finite Difference Method (FDM)

(S. J. Hollister, 2011)

## 2 Effective Elasticity

Microstructural elasticity equation:

$$\frac{\partial}{\partial x_j} E_{ijkl} \frac{\partial \chi_k^{pq}}{\partial x_l} = \frac{\partial}{\partial x_j} E_{ijpq}$$

→ integration over the microstructure:

$$E_{ijkl}^{eff} = \frac{1}{|V_{micro}|} \int_{V_{micro}} E_{ijpq} \left( \delta_{kp} \delta_{lq} - \frac{\partial \chi_p^{kl}}{\partial x_q} \right) dV_{micro}$$

(S. J. Hollister, 2011)

## 2 Effective diffusivity

Diffusion equation:

$$\frac{\partial}{\partial x_i} D_{ij} \frac{\partial \chi^p}{\partial x_j} = \frac{\partial}{\partial x_i} D_{ip}$$

→ integration over the microstructure:

$$D_{ij}^{eff} = \frac{1}{|V_{micro}|} \int_{V_{micro}} D_{ip} \left( \delta_{jp} - \frac{\partial \chi^j}{\partial x_q} \right) dV_{micro}$$

(S. J. Hollister, 2011)

## 2 Effective permeability

Pressure gradient equation:

$$\frac{\partial p^k}{\partial x_i} - \frac{\partial}{\partial x_j} \left( \frac{\partial v_i^{0k}}{x_j} \right) = e_i^k$$

→ integration over the microstructure:

$$K_{ik}^{eff} = \frac{1}{\mu |V_{micro}|} \int_{V_{micro}} v_i^{0k} dV_{micro}$$

(S. J. Hollister, 2011)

### 3 Outline

- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

### 3 Design Parameters of Flow Perfusion Bioreactors

- ▶ Direct Perfusion of Scaffolds
- ▶ Media Flow Pathway
- ▶ Maintenance of Sterility
- ▶ Environmental Control
  - Nutrient supply
  - Waste removal
  - Gas exchange

(E. J. Levorson et al., 2011)

### 3 Building a Simple Perfusion Bioreactor

#### **Objective:**

Design a perfusion bioreactor to employ circulatory flow throughout the system to provide

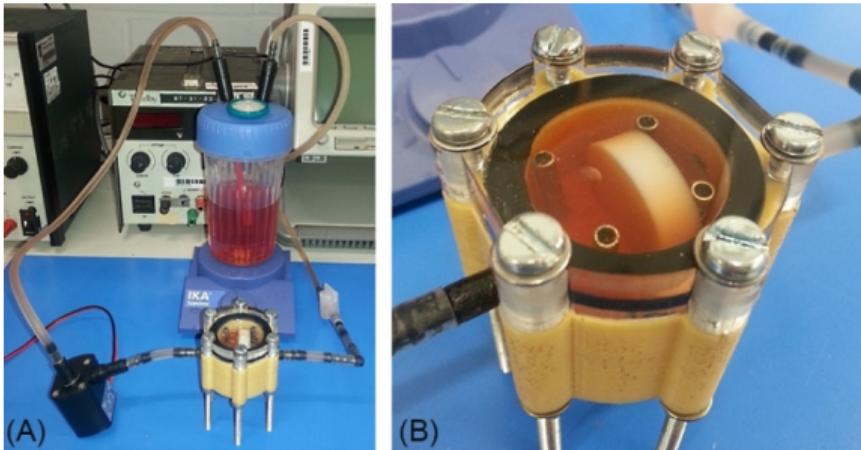
- ▶ Sufficient nutrient transfer
- ▶ Removal of waste products
- ▶ Application of mechanical stimuli

(J. Rosser et al., 2018)

### 3 Building a Simple Perfusion Bioreactor

#### Components:

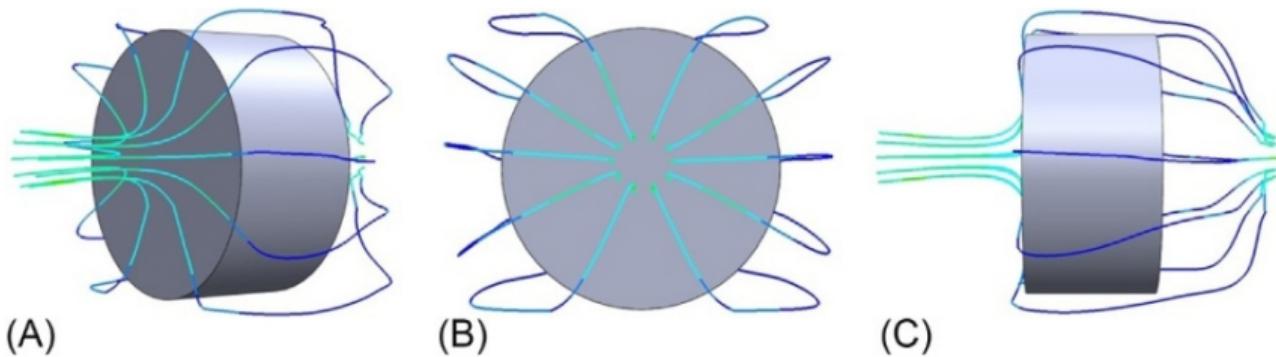
- 1 a culture encasing bioreactor
- 2 a vessel containing the oxygenated nutrient-rich medium
- 3 a pump capable of generating flow throughout the system



(J. Rosser et al., 2018)

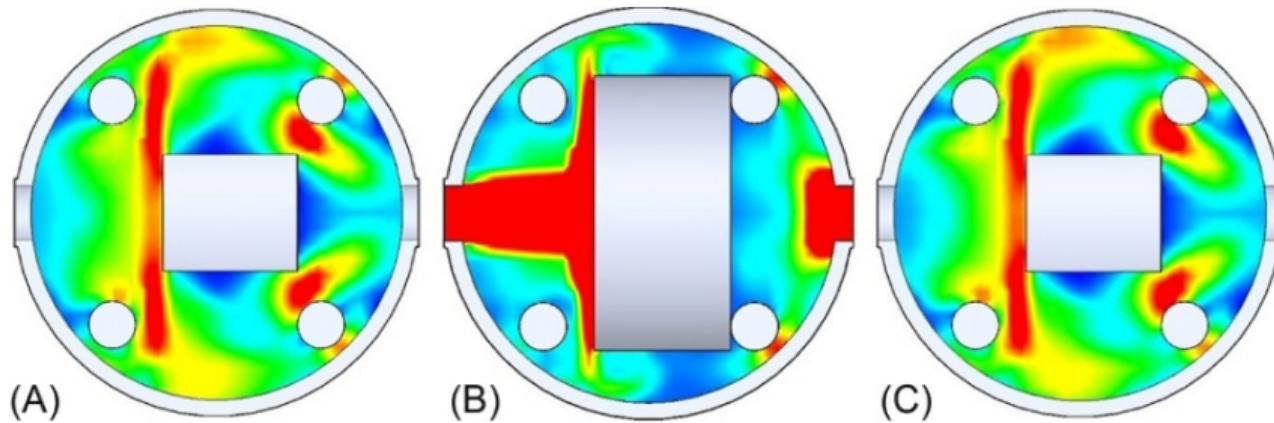
### 3 Taking Advantage of CFD

Using **CFD** to simulate fluid flow to obtain the optimum *internal shape*, *inlet/outlet positioning*, and *scaffold fixture positioning*.



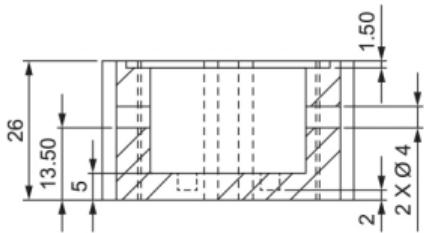
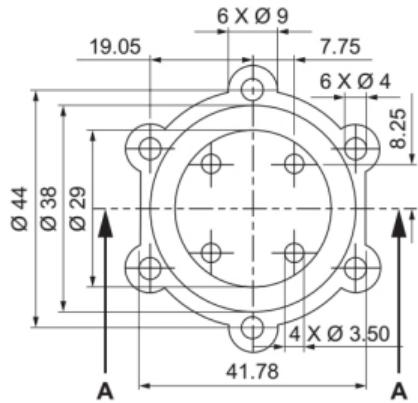
Flow trajectories around the vertical scaffold; (A) isometric, (B) front, and (C) side views

### 3 Optimizing Inlet and Outlet



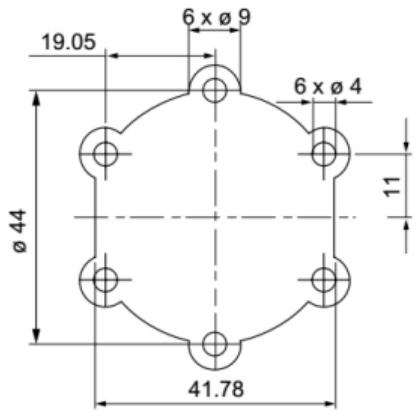
(A) Lower, (B) middle, and (C) upper velocity contours with application-wide scaffold supports

### 3 Bioreactor Base Design



Bioreactor base dimensions (mm) and fully constructed bioreactor unit

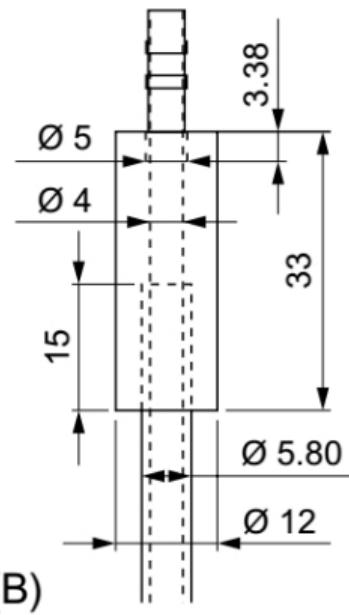
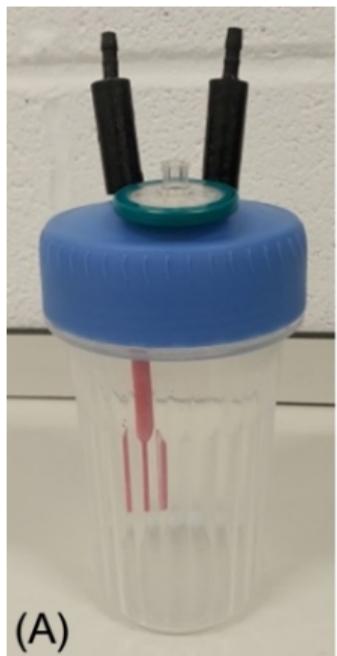
### 3 Bioreactor Lid Design



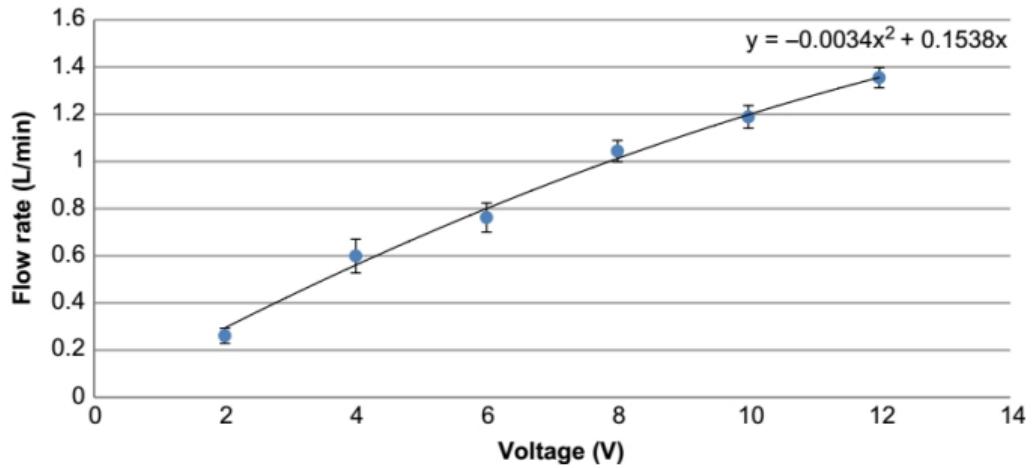
Bioreactor lid dimensions (mm) and finished bioreactor base component

### 3 Medium Vessel Design

Medium vessel dimensions (mm) and assembly



### 3 Pomp Design



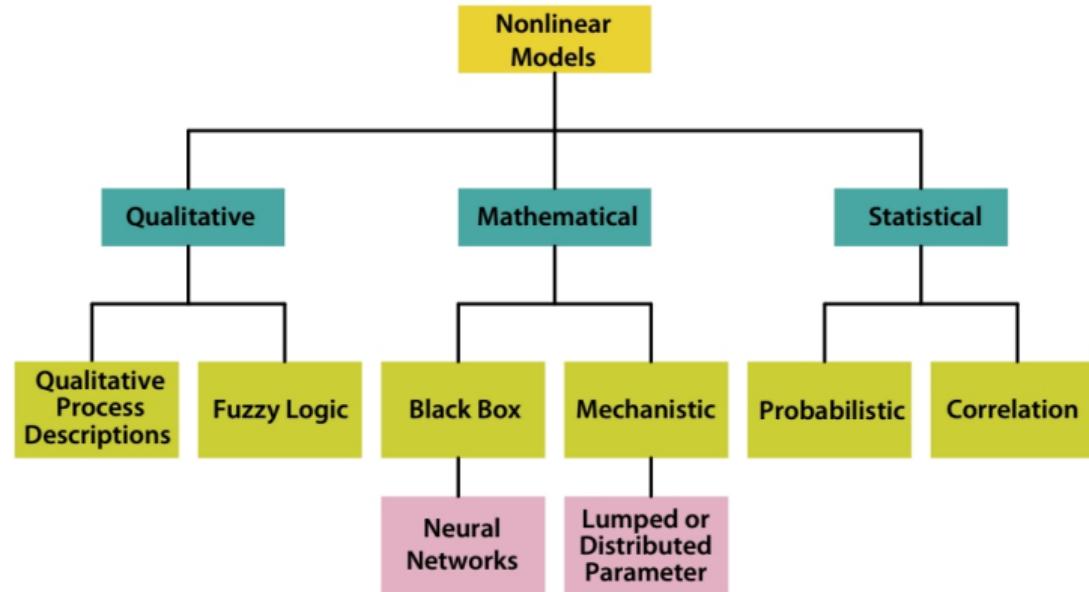
A 12V 4.8W 0.8A brushless DC pump and resulting flow rate values with specified voltages applied.

## 4 Outline

- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

## 4 Bioreactor Modeling

Classification of nonlinear model forms used in bioreactor modeling



(C. Julien et al., 2007)

## 4 Mathematical Modeling of Bioreactors

Mathematical modeling and simulation is essential to overcome the design challenges of Bioreactors.

### Examples of Mathematical Models

- ▶ Black Box Models:
  - Artificial Neural Networks
  - Machine Learning-based Control
- ▶ Mechanistic (Conservative) Models
  - ODE - Ordinary Differential Equations
  - PDE - Partial Differential Equations

## 4 Quantitative Models of Perfusion Bioreactors

Investigating 2 Examples of Bioreactor Models:

- ▶ **Example 1 (Simple Model):**

Modeling of the Flow within Scaffolds in Perfusion Bioreactors  
(X. Yan et al., 2011)

- ▶ **Example 2 (Complex Model):**

Mathematical Modeling of Three-Dimensional Cell Cultures in Perfusion Bioreactors  
(F. Coletti et al., 2006)

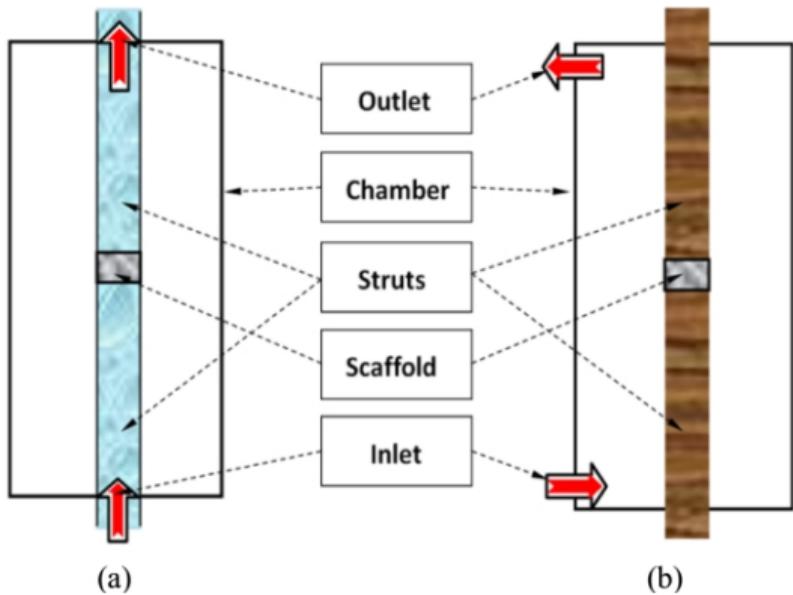
Materials similar to Example 2 are useful for students to be more familiar with the engineering aspects of Bioreactors.

## 4 Example 1: Flow Model

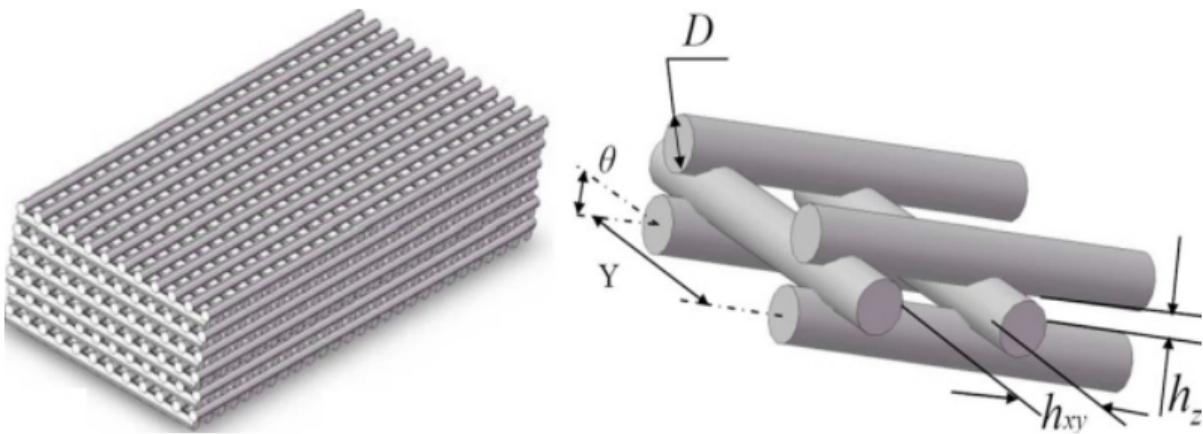
To control the cultivating process, the knowledge of the fluid flow inside and around a scaffold in the bioreactor is essential.

Figure: Schematic of bioreactors:  
(a) perfusion bioreactor  
(b) non-perfusion bioreactor

(X. Yan et al., 2011)



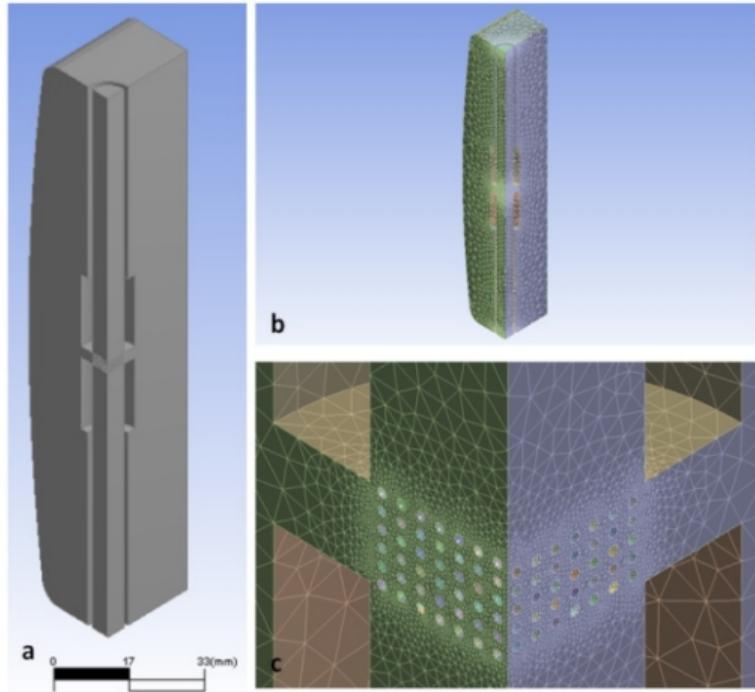
## 4 Scaffold Geometry



Geometric parameters for tissue scaffold

## 4 Model and Mesh Setup

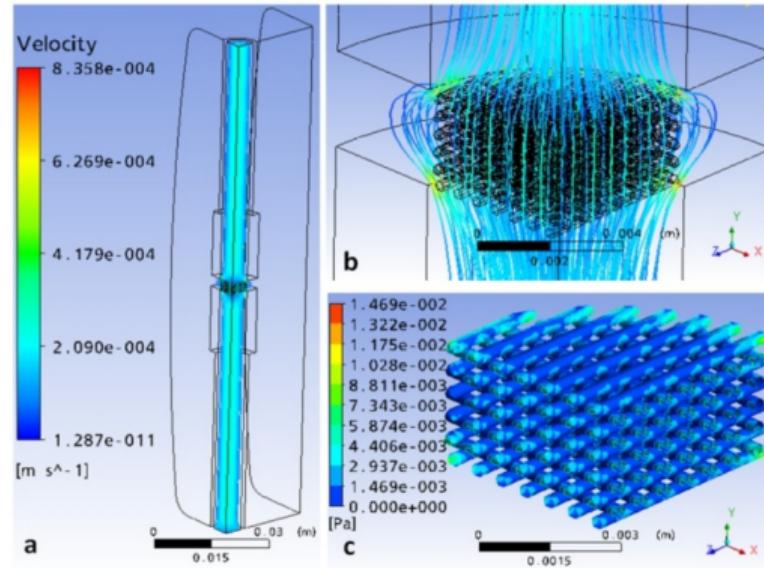
- a) Final geometric model
- b) Mesh
- c) Refined mesh around tissue scaffold



## 4 Results: Perfusion Bioreactor

D=0.3mm and Y=0.7mm

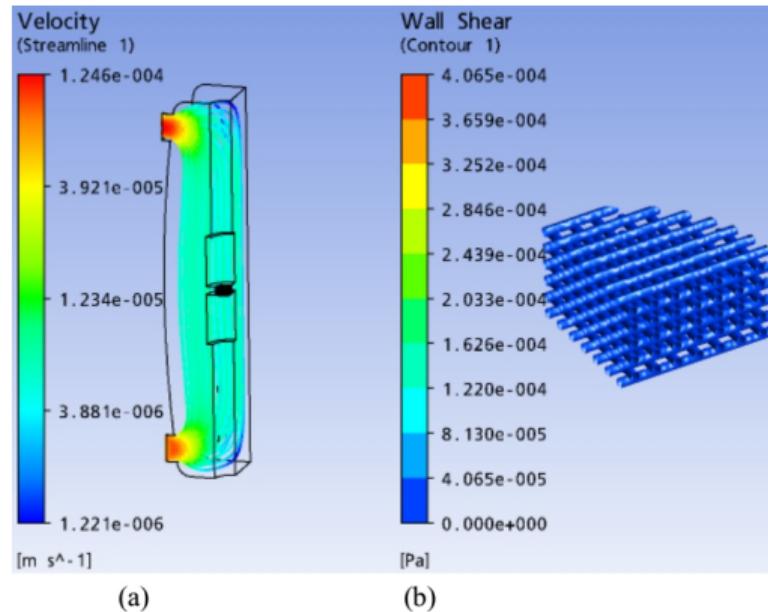
- a) Velocity streamlines in bioreactor
- b) Velocity streamlines around the tissue scaffold
- c) Surface shear stress distribution in the scaffold



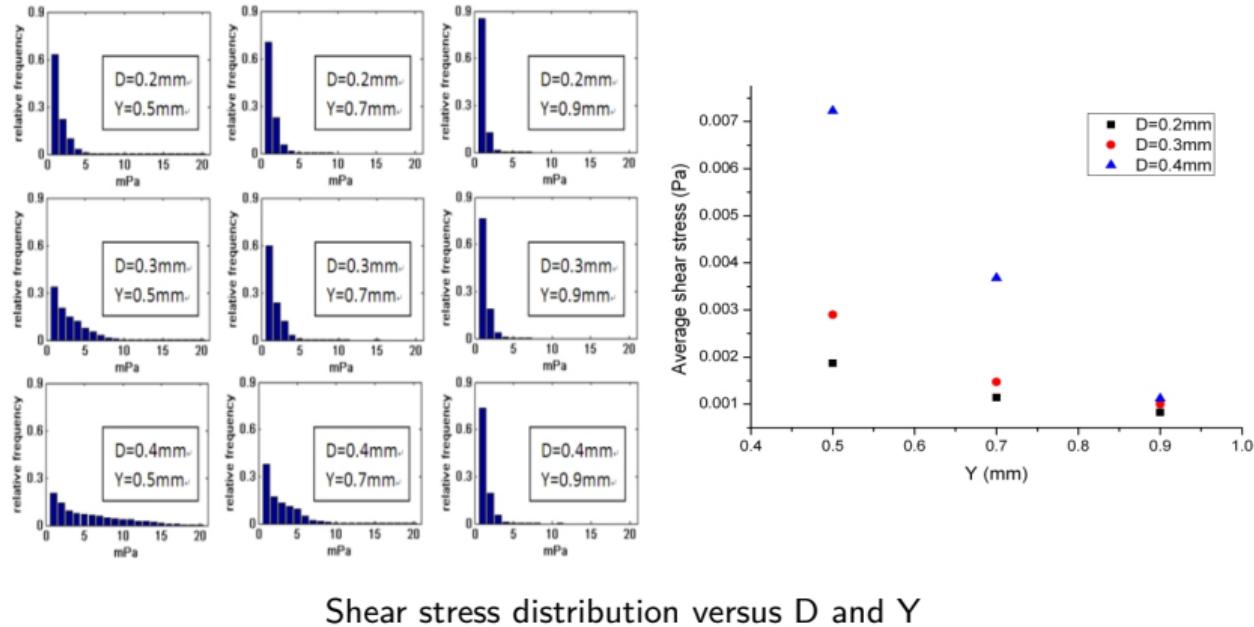
## 4 Results: Non-Perfusion Bioreactor

D=0.3mm and Y=0.7mm

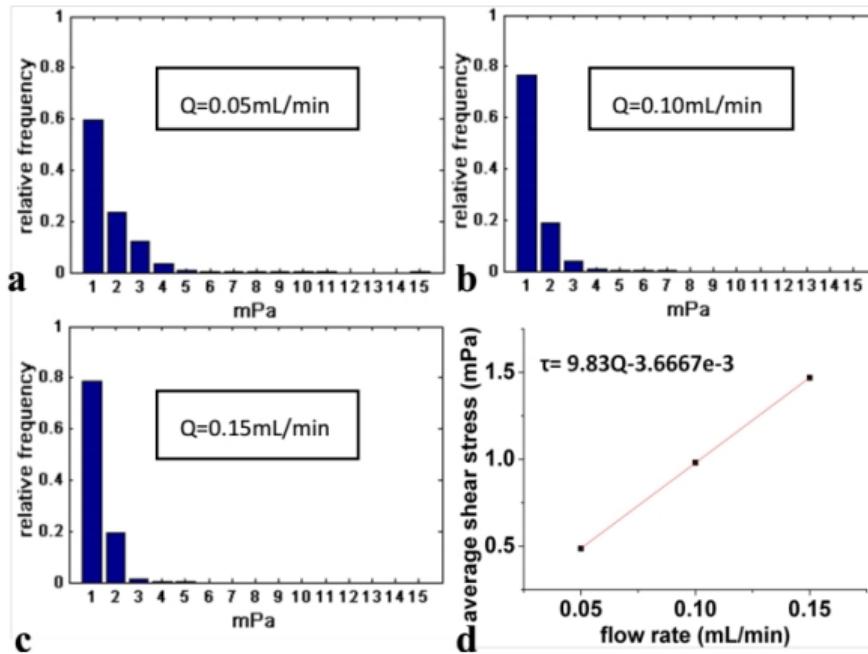
- a) Velocity streamlines in bioreactor
- b) Surface shear stress distribution in the scaffold



## 4 Results: Surface Shear Stress Distribution



## 4 Results: Surface Shear Stress Distribution



Shear stress distribution within scaffold for different flow rates

## 4 Example 2: Cell Cultures in Perfusion Bioreactor

Obtaining a proper oxygen supply, high cell density, and a uniform cell distribution in a 3D growth support are important challenges.

To study all together

→ Comprehensive mathematical model of perfusion bioreactor:

- ▶ Fluid Flow
- ▶ Convection (Advection)
- ▶ Diffusion
- ▶ Cell growth kinetics

(F. Coletti et al., 2006)

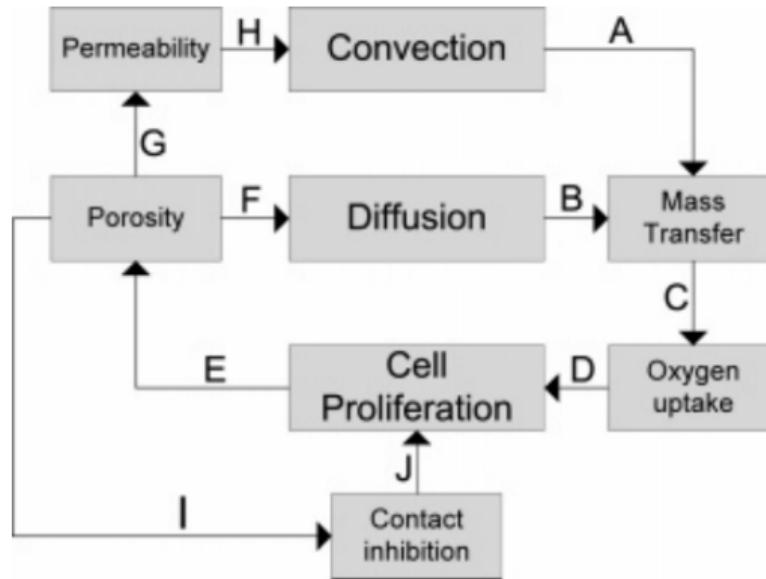
## 4 Governing Equations

To model spatial-temporal evolution of *oxygen concentration* and *cell density* within a 3D polymeric scaffold:

- ▶ Flow inside the bioreactor → Navier-Stokes equations
- ▶ Convection through the scaffold → Brinkman's extension of Darcy's law
- ▶ The oxygen uptake rate → Michaelis-Menten kinetics
- ▶ Cell growth → Function of oxygen concentration in the Contois equation

## 4 Conceptual Model

Schematic demonstration of the main interacting phenomena that occur in a perfusion bioreactor when convective and diffusive flux and cell proliferation take place within the scaffold



## 4 Line A

Momentum balance by the Navier-Stokes equations

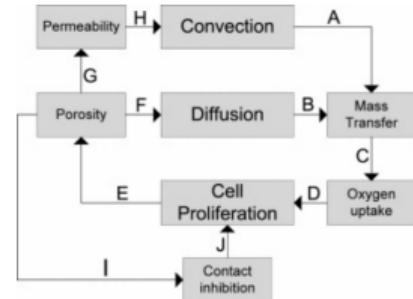
$$\rho \frac{\partial \vec{v}^0}{\partial t} + \rho (\vec{v}^0 \cdot \nabla) \vec{v}^0 = -\nabla P + \mu \nabla^2 \vec{v}^0 + \rho \vec{g}$$

Continuity equation for incompressible fluid

$$\nabla \vec{v}^0 = \frac{\partial v_z^0}{\partial z} + \frac{1}{r} \frac{\partial r v_r^0}{\partial r} + \frac{1}{r} \frac{\partial v_\vartheta^0}{\partial \vartheta} = 0$$

Brinkman's extension to Darcy's law for porous media

$$\rho \frac{\partial \vec{v}}{\partial t} = -\nabla P - \frac{\mu}{K} \vec{v} + \mu \nabla^2 \vec{v} + \rho \vec{g}$$



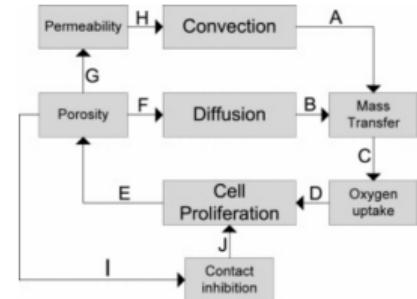
## 4 Line B

Species mass balance that includes convective and diffusion terms

$$\frac{\partial c_i}{\partial t} = - \left( \nabla \cdot c_i \vec{v}^0 \right) + \nabla \cdot (D_i \nabla c_i)$$

Material balance in the scaffold

$$\frac{\partial c_i}{\partial t} = - (\nabla \cdot c_i \vec{v}) + \nabla \cdot (D_{\text{eff}} \nabla c_i) + R_i$$



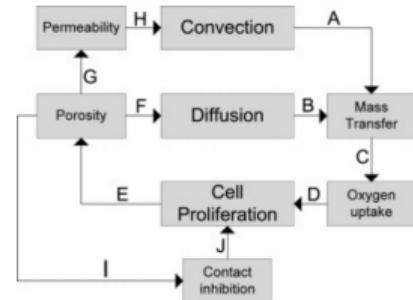
## 4 Line C and D

**Line C:** Reaction rate expressed in the form of Michaelis-Menten kinetics

$$R_{O_2} = \rho_{cell} \frac{Q_m c_{O_2}}{C_m + c_{O_2}}$$

**Line D:** The cell density variation with respect to time

$$\frac{\partial \rho_{cell}}{\partial t} = \left( \frac{\mu_{cell}^{\max} c_i}{K_c \rho_{cell} V_{cell} \rho_c + c_i} - k_d \right) \rho_{cell}$$



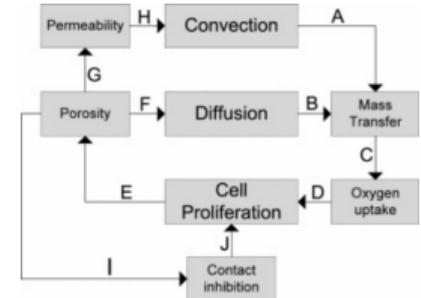
## 4 Line E and F

**Line E:** Reduction in scaffold porosity

$$\epsilon(z, r, t) = \epsilon(z, r, 0) - V_{\text{cell}} \rho_{\text{cell}}(z, r, t)$$

**Line F:** Effective diffusion coefficient as a function of porosity and tortuosity

$$D_{\text{eff}}(z, r, t) = \frac{\epsilon(z, r, t)}{\tau(z, r, t)} D$$



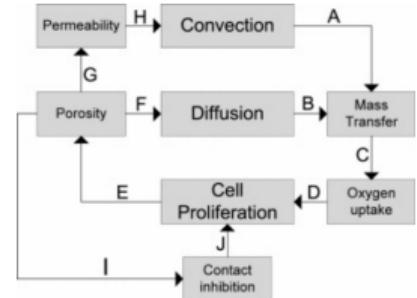
## 4 Line G and H

**Line G:** Tortuosity is modeled as a function of porosity

$$\tau = \left( \frac{2 - \epsilon}{\epsilon} \right)^2$$

**Line H:** The functional form of Koponen is used for permeability

$$K = \frac{\epsilon^3}{q\tau^2 s^2}$$



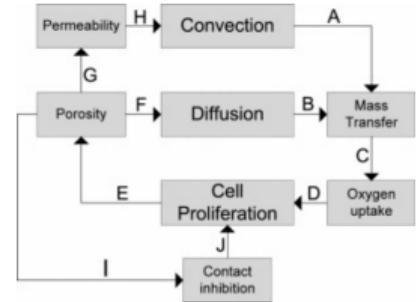
## 4 Line I and J

**Line I:** The Contois equation to describe cell growth

$$\mu_{\text{cell}} = \frac{\mu_{\text{cell}}^{\max} c_i}{K_c \rho_{\text{cell}} V_{\text{cell}} \rho_c + c_i}$$

**Line J:** Similar to Line D

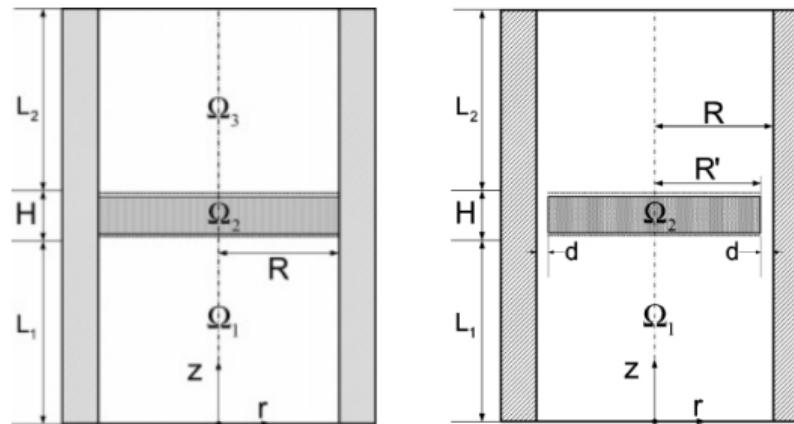
$$\frac{\partial \rho_{\text{cell}}}{\partial t} = \left( \frac{\mu_{\text{cell}}^{\max} c_i}{K_c \rho_{\text{cell}} V_{\text{cell}} \rho_c + c_i} - k_d \right) \rho_{\text{cell}}$$



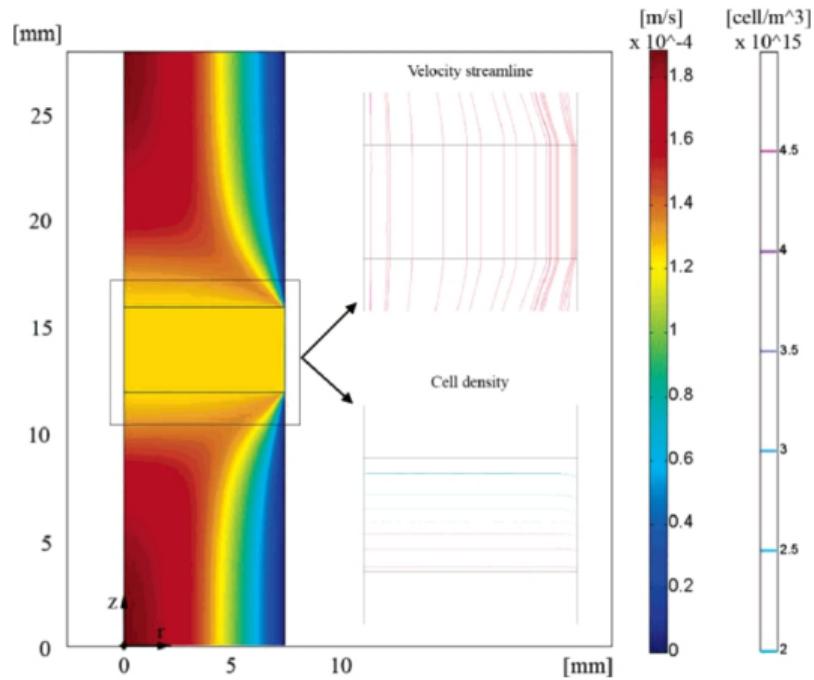
## 4 Simulating the Model

The model is used to simulate two conditions:

- ▶ **Case 1:** Total flow perfusion (left)
- ▶ **Case 2:** Partial flow perfusion with flow channelling at the walls (right)

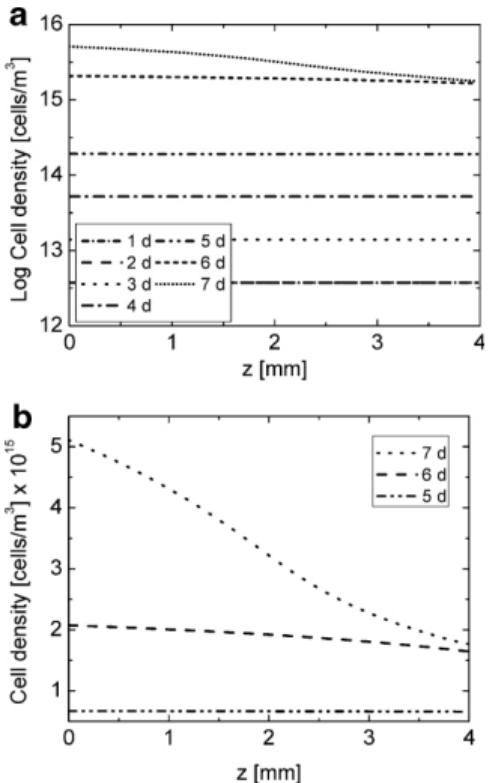


## 4 Results: Case 1, Total Flow Perfusion



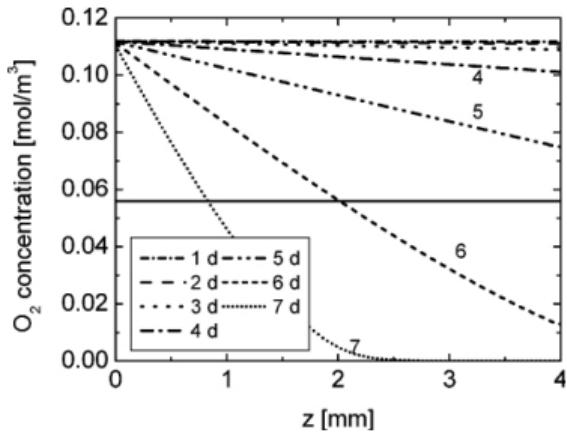
## 4 Results: Cell Density

Cell density in the scaffold on a logarithmic scale at 1-7 culture days (a) and for the last 3 culture days (b).



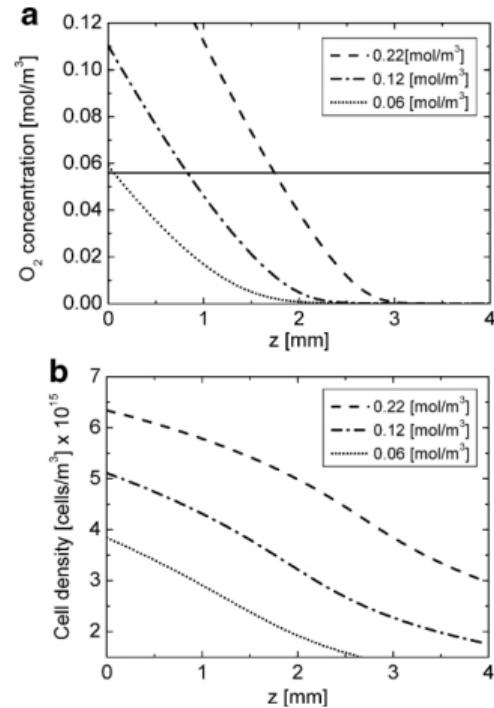
## 4 Results: Oxygen Concentration

Oxygen concentration in the medium fluid within the scaffold after 1-7 culture days. Between the 5th and the 6th day, the oxygen concentration decreases to values lower than the minimum value for cell viability indicated by the horizontal line



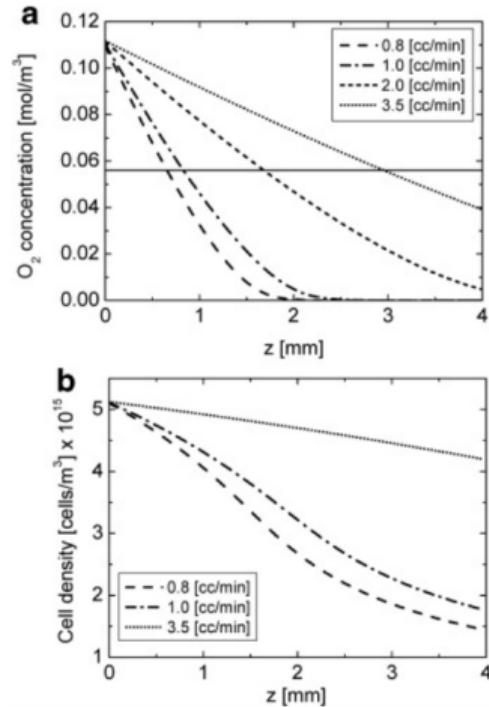
## 4 Results: Cell and Oxygen Density

Oxygen (a) and cell density (b) profiles within the scaffold domain after 7 culture days, for several oxygen concentrations in the medium inlet fluid

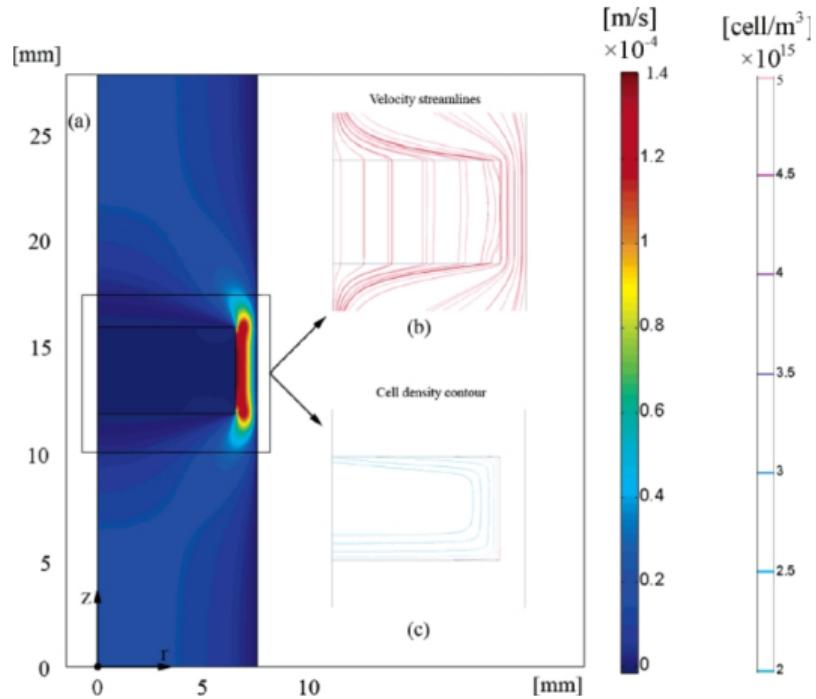


## 4 Results: Effect of Flow Rate

Effect of medium flow rate on oxygen profiles (a) and cell density (b) within the scaffold at day 7

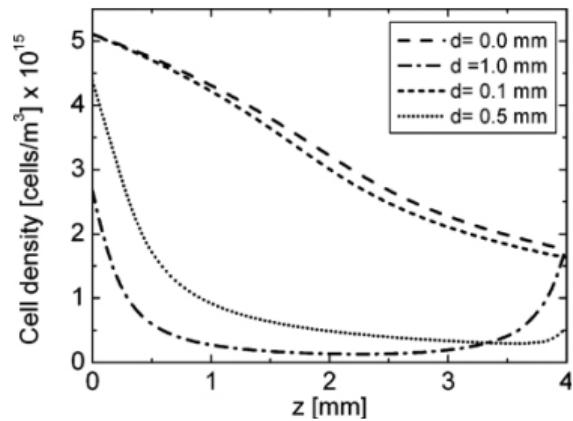


## 4 Results: Case 1, Partial Flow Perfusion



## 4 Results: Comparison of Cell Density

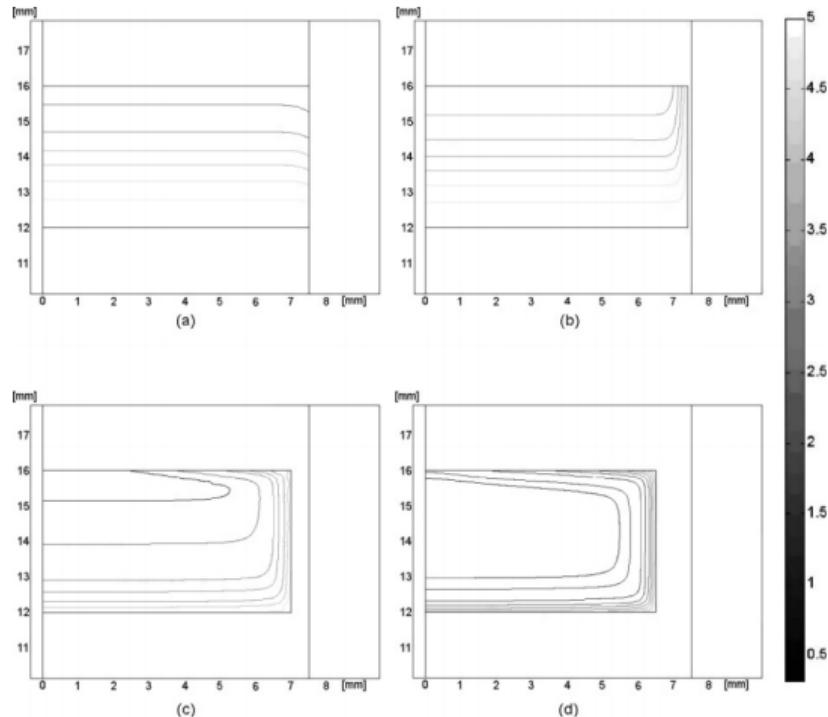
Cell density after 7 days within the scaffold for total ( $d = 0$ ) and partial perfusion ( $d > 0$ ) with different channel gaps  $d$ , at 4 mm from the scaffold center. When the channel gap ( $d$ ) is only 0.1 mm, the effect on cell distribution is quite small while the cell distribution differs significantly from the total perfusion case for ( $d > 1$  mm)



## 4 Results: Comparison of Cell Density

Cell density contours after 7 days of culture in a section at 4 mm from the scaffold center in the case of total perfusion (a) and partial perfusion with  $d = 0.1$  (b),  $d = 0.5$  (c), and  $d = 1$  mm (d).

The greater the channel gap, the lower the cell density obtained.



## 4 Example 2: Conclusion

The two cases illustrated, total and partial perfusion, represent situations where the dominant oxygen transfer mechanism changes from being essentially convective, when no channelling is present, to essentially diffusive, when large channelling gaps occur near the wall.

Although this example is relatively complex for students, it is suitable for teaching the concepts of diffusion and convection in Bioreactors and how they affect Tissue Engineering products.

We can overcome the complexity of this example using an effective educational tool!

## 5 Outline

- ① Introduction
- ② Bioreactors and Scaffolds
- ③ Design of Bioreactors
- ④ Bioreactor Modeling
- ⑤ Utilizing Jupyter Notebooks

## 5 Jupyter

Project Jupyter exists to develop open-source software, open-standards, and services for **interactive computing** across dozens of programming languages.



## 5 Why Jupyter Is Important?

### Jupyter

Considered as a **game-changer for research** because it enables scientific researchers to share detailed descriptions of raw code that then allow others to validate and build on their research.

### Jupyter Notebooks

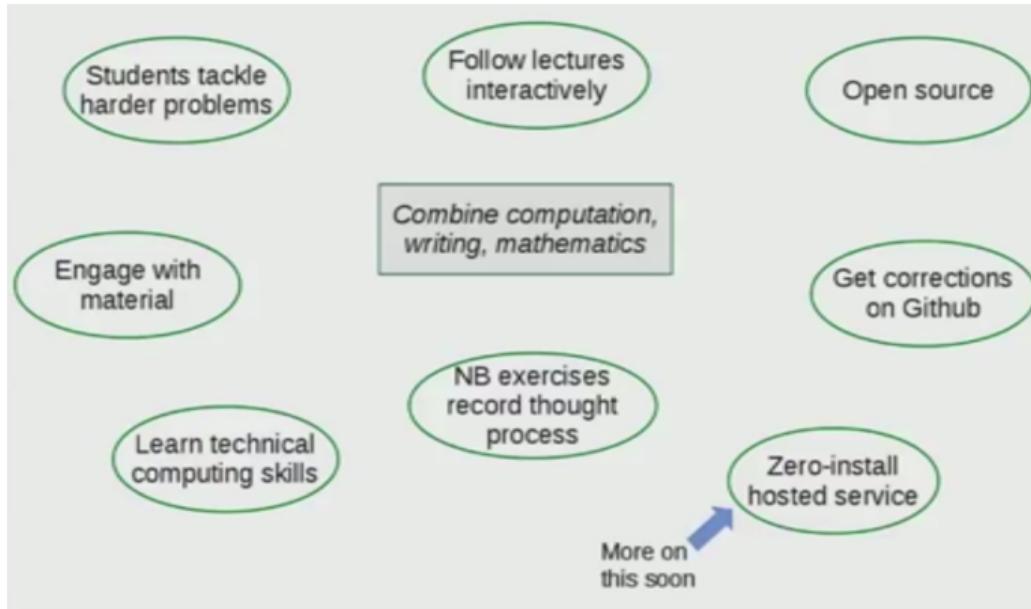
The computational notebooks are described as the computing equivalent of a scientist's lab notebook – an environment in which scientists worldwide can develop code and run it immediately in their notebook environment.

## 5 Why Using Jupyter Notebooks in Education?

### Jupyter Notebooks

- ▶ Make learning in the class active and interactive
- ▶ Eliminate the complexity of getting started
- ▶ Facilitate rapid iteration and hence faster learning.
- ▶ Allow embedding of support content (image, video, formula, text, links) right next to code
- ▶ Can be used for creating interactive assignments (even auto-graded!)

## 5 Why Using Jupyter Notebooks in Education?



(EuroPython 2017, Jupyter notebooks for teaching and learning [[Link to video](#)])

## 5 Examples

Jupyter Notebooks  
powering UC Berkeley's  
classes

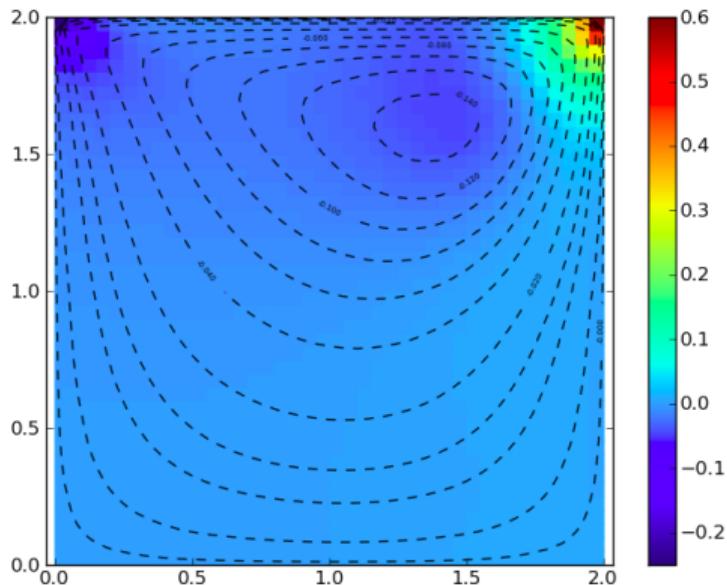


## 5 Examples

Lorena A. Barba group

A well-known powerful CFD course by taking advantage of Jupyter Notebooks [[Link](#)]

CFD Python: 12 steps to Navier-Stokes



## 5 Examples

*"Introduction to Computational Modelling for the Biosciences"* course in the University of Oslo

*One cannot learn programming from slide-based lectures. Much of the learning will have to be done 'by doing', in group work. I will use the Jupyter Notebook, the "killer app" in education according to professor Lorena Barba, in a flipped classroom approach where students study notebooks beforehand (each chapter of the course book can be turned into a notebook), formative assessment is used to gauge understanding, and students work with exercises during 'class'.*

*– Lex Nederbragt, the instructor*

# 5 Examples

Jupyter Notebooks are being used in the "Quantum Mechanics" course and lab of the Pacific University [[Link](#)]

The screenshot shows a Jupyter Notebook interface in Mozilla Firefox. The notebook displays several code cells and their outputs.

**In [59]:** `integrate(x**2,(x,0,1))`  
**Out[59]:**  $\frac{1}{3}$

The cell below will return an odd set of conditions on the result. This is because the solver doesn't want to assume anything about  $a$  and there is a special case where the answer would be different. If you look closely though, that special case isn't physically realistic so to ignore these special conditions, we add `conds='none'`. The next cell down does what you'd expect. From here on out, just add this to the `integrate` function and we'll get what we expect.

**In [64]:** `A = (c*cos((pi*x)/(2.0*a)))**2  
A.integrate((x,-a,a))`  
**Out[64]:** 
$$-c^2 \begin{cases} -a & \text{for } \frac{0.5a}{a} = 0 \\ -0.5a & \text{otherwise} \end{cases} + c^2 \begin{cases} a & \text{for } \frac{0.5a}{a} = 0.5a \\ a & \text{otherwise} \end{cases}$$

**In [65]:** `A = (c*cos((pi*x)/(2.0*a)))**2  
A.integrate((x,-a,a), conds='none')`  
**Out[65]:**  $1.0ac^2$

So this tells us the normalization constant should be  $c = \frac{1}{\sqrt{a}}$ . Check that it is normalized if we do that:

**In [68]:** `psi = 1/sqrt(a)*cos((pi*x)/(2.0*a)) # notice we can name the expression something useful  
B = psi**2  
B.integrate( (x,-a,a), conds='none')`  
**Out[68]:** 1.0

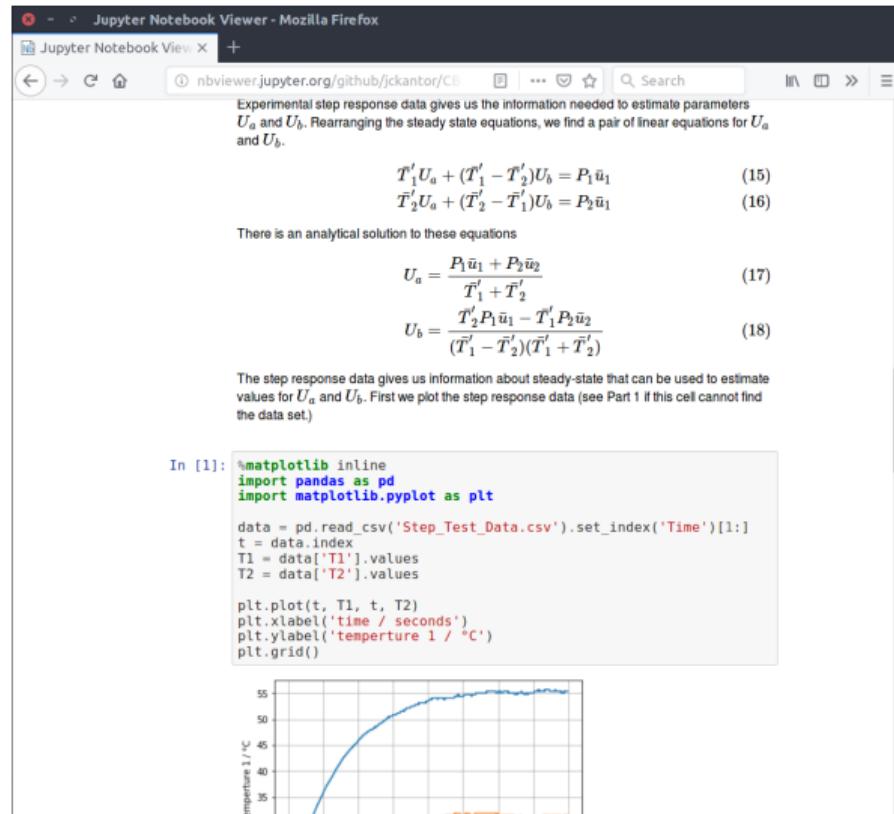
Because `psi` is a real function, we can calculate expectation values by integrating over  $x$  or  $x^2$  with `psi**2`:

**In [70]:** `C = x*psi**2  
C.integrate( (x,-a,a), conds='none')`  
**Out[70]:** 0

**In [71]:** `D = x**2 * psi**2  
E = D.integrate( (x,-a,a), conds='none')`

# 5 Examples

Jupyter Notebooks in support of "Chemical Process Control" course taught at the University of Notre Dame [Link]



## 5 Examples

Sample notebook to  
animate the  $\psi$  wave  
function [[Link](#)]

The screenshot shows a Jupyter Notebook Viewer window in Mozilla Firefox. The page title is "Jupyter Notebook Viewer - Mozilla Firefox". The URL in the address bar is "nbviewer.jupyter.org/github/atusis". The content of the notebook is as follows:

with  $L$  in nm and  $mc^2$  in eV. Then  $E_1$  has units of eV. Other energy eigenvalues are:

$$E_n = n^2 E_1.$$

Then, in time-dependent wavefunction, use

$$\frac{E}{\hbar} = \frac{2\pi(3e8 \text{ nm/ns})}{(1240 \text{ eV} \cdot \text{nm})} E_{eV}$$

where  $E_{eV}$  is in units of eV. Then  $\frac{E}{\hbar}$  has units of  $\text{ns}^{-1}$  and  $E/\hbar t$  can be written with  $t$  in ns.

```
In [6]: #constants
#use x in nm and t in ns
Erest=0.511e6 #rest energy for electron in eV
L=1 #nm
E1=(1240)**2/8/Erest/L**2
Eoverhbar=2*np.pi*3e8/1240 #(eV ns)^-1
```

### Animate a wavefunction

$$\psi(x) = \frac{1}{\sqrt{2}} \psi_1 + \frac{1}{\sqrt{2}} \psi_2$$

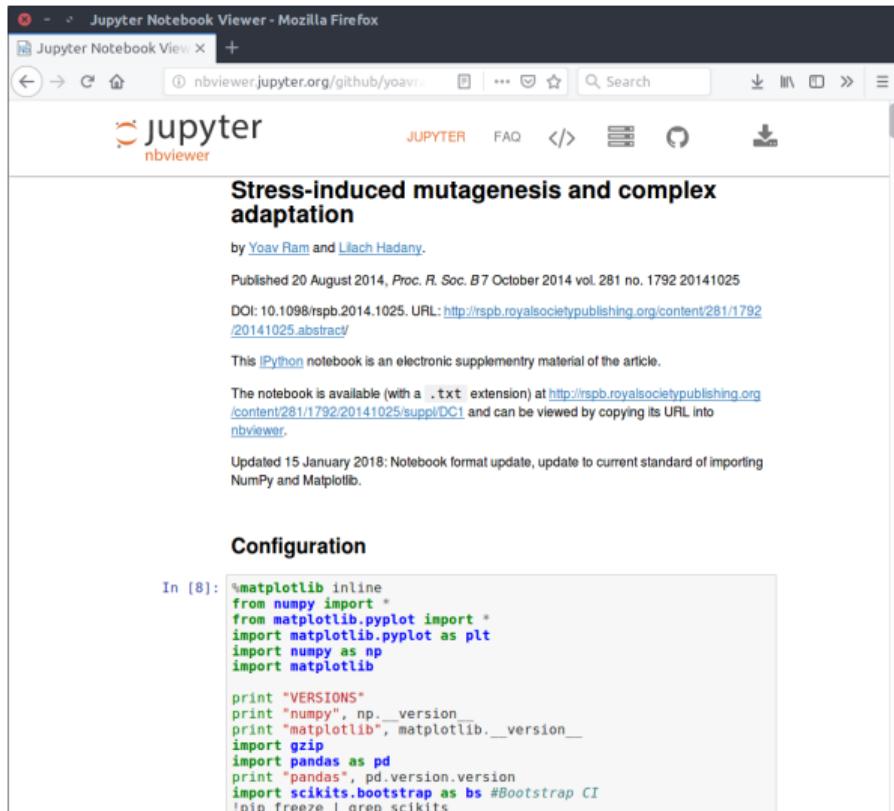
```
In [7]: #constants
A=np.sqrt(2/L)
k=np.pi/L
E2=4*E1

# initialization function: plot the background of each frame
def init():
    line.set_data([], [])
    return line,

# animation function. This is called sequentially
def animate(t):
    #t is time in ns
    #x is in nm
    x = np.linspace(0, 1, 1000)
```

## 5 Examples

It has become popular that scientists share the source code of their publications and its corresponding explanations in the notebook format [Link]



The screenshot shows a Jupyter Notebook Viewer window in Mozilla Firefox. The title bar reads "Jupyter Notebook Viewer - Mozilla Firefox". The main content area displays a publication titled "Stress-induced mutagenesis and complex adaptation" by Yoav Ram and Lilach Hadany. It includes a DOI link (10.1098/rspb.2014.1025) and a note about the Python notebook being an electronic supplement. A configuration cell at the bottom contains the following Python code:

```
In [8]: %matplotlib inline
from numpy import *
from matplotlib.pyplot import *
import matplotlib.pyplot as plt
import numpy as np
import matplotlib

print "VERSIONS"
print "numpy", np.__version__
print "matplotlib", matplotlib.__version__
import gzip
import pandas as pd
print "pandas", pd.__version__.version
import scikits.bootstrap as bs #Bootstrap CI
!pip freeze | grep scikits
```

## 5 Examples

An article to explain the details of a complex topic in Biology step by step using the Jupyter Notebook [Link]

The screenshot shows a browser window titled "Jupyter Notebook Viewer - Mozilla Firefox". The URL in the address bar is "nbviewer.jupyter.org/github/maayan". The page itself is titled "jupyter nbviewer" and features a main heading: "An open RNA-Seq data analysis pipeline tutorial with an example of reprocessing data from a recent Zika virus study". Below the heading, it lists authors: "Zichen Wang<sup>1</sup> and Avi Ma'ayan<sup>1</sup>". It also includes institutional information: "Department of Pharmacology and Systems Therapeutics; BD2K-LINCS Data Coordination and Integration Center; Mount Sinai Knowledge Management Center for Illuminating the Druggable Genome; Icahn School of Medicine at Mount Sinai, One Gustave L. Levy Place, Box 1603, New York, NY 10029 USA". A note about correspondence: "\*Correspondence: [avi.maayan@mssm.edu](mailto:avi.maayan@mssm.edu)". The page is divided into sections: "Abstract" and "RNA-seq analysis is becoming a standard method for global gene expression profiling. However, open and standard pipelines to perform RNA-seq analysis by non-experts remain challenging due to the large size of the raw data files and the hardware requirements for running the alignment step. Here we introduce a reproducible open source RNA-seq pipeline delivered as an IPython notebook and a Docker image. The pipeline uses state-of-the-art tools and can run on various platforms with minimal configuration overhead. The pipeline enables the extraction of knowledge from typical RNA-seq studies by generating interactive principal component analysis (PCA) and hierarchical clustering (HC) plots, performing enrichment analyses against over 90 gene set libraries, and obtaining lists of small molecules that are predicted to either mimic or reverse the observed changes in mRNA expression. We apply the pipeline to a recently published RNA-seq dataset collected from human neuronal progenitors infected with the Zika virus (ZIKV). In addition to confirming the presence of cell cycle genes among the genes that are downregulated by ZIKV, our analysis uncovers significant overlap with upregulated genes that when knocked out in mice induce defects in brain morphology. This result potentially points to the molecular processes associated with the microcephaly phenotype observed in newborns from pregnant mothers infected with the

# 5 Examples

A sample notebook to demonstrate how to solve the diffusion equation using Finite Different Method in Python [[Link](#)]

The screenshot shows a Jupyter Notebook Viewer in Mozilla Firefox. The page title is "Jupyter Notebook Viewer - Mozilla Firefox". The URL in the address bar is "nbviewer.jupyter.org/github/waltherm/numerical-methods-in-finite-difference-methods/blob/main/notebooks/1-Diffusion.ipynb". The content of the notebook includes the following text and equations:

$j = J - 1 : -\sigma U_{J-2}^{n+1} + (1 + 2\sigma)U_{J-1}^{n+1} - \sigma U_J^{n+1} = \sigma U_{J-2}^n + (1 - 2\sigma)U_{J-1}^n + \sigma U_J^n + \Delta t f(U_{J-1}^n).$

The problem here is that the values  $U_{-1}^n$  and  $U_J^n$  lie outside our grid.

However, we can work out what these values should equal by considering our Neumann boundary condition. Let us discretize our boundary condition at  $j = 0$  with the [backward difference](#) and at  $j = J - 1$  with the [forward difference](#):

$$\frac{U_1^n - U_0^n}{\Delta x} = 0,$$
$$\frac{U_J^n - U_{J-1}^n}{\Delta x} = 0.$$

These two equations make it clear that we need to amend our above numerical approximation for  $j = 0$  with the identities  $U_0^n = U_1^n$  and  $U_0^{n+1} = U_1^{n+1}$ , and for  $j = J - 1$  with the identities  $U_{J-1}^n = U_J^n$  and  $U_{J-1}^{n+1} = U_J^{n+1}$ .

Let us reinterpret our numerical approximation of the line concentration of  $u$  in a fixed point in time as a vector  $\mathbf{U}^n$ :

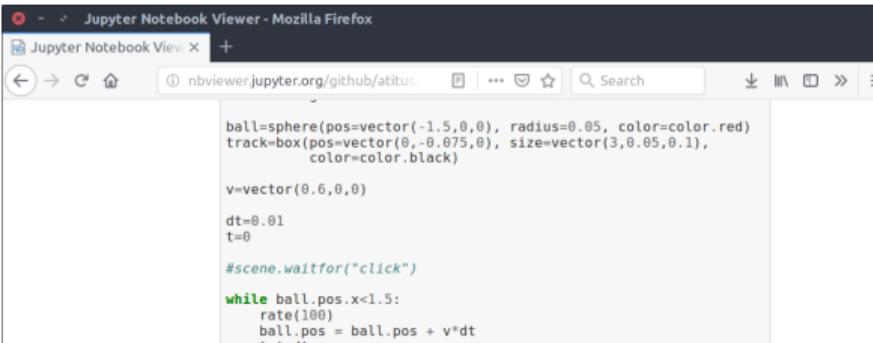
$$\mathbf{U}^n = \begin{bmatrix} U_0^n \\ \vdots \\ U_{J-1}^n \end{bmatrix}.$$

Using this notation we can now write our above approximation for a fixed point in time,  $t = n\Delta t$ , compactly as a linear system:

$$\begin{bmatrix} 1 + \sigma & -\sigma & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -\sigma & 1 + 2\sigma & -\sigma & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -\sigma & 1 + 2\sigma & -\sigma & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sigma & 1 + 2\sigma & -\sigma \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_J \end{bmatrix} = \mathbf{U}^n$$

## 5 Examples

Teaching Physics with Computation using Jupyter Notebook and VPython at the High Point University [[Link](#)]



```
ball=sphere(pos=vector(-1.5,0,0), radius=0.05, color=color.red)
track=box(pos=vector(0,-0.075,0), size=vector(3,0.05,0.1),
          color=color.black)

v=vector(0.6,0,0)

dt=0.01
t=0

#scene.waitfor("click")

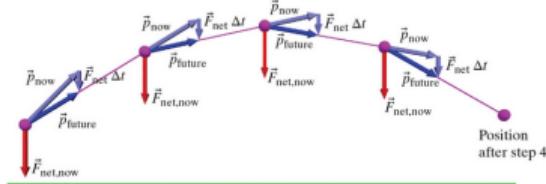
while ball.pos.x<1.5:
    rate(100)
    ball.pos = ball.pos + v*dt
    t=t+dt
```

### Momentum Update

For a small time interval  $\Delta t$ ,

$$\vec{p}_{future} = \vec{p}_{now} + \vec{F}_{net,now} \Delta t$$

This is the **update form** of the Momentum Principle.

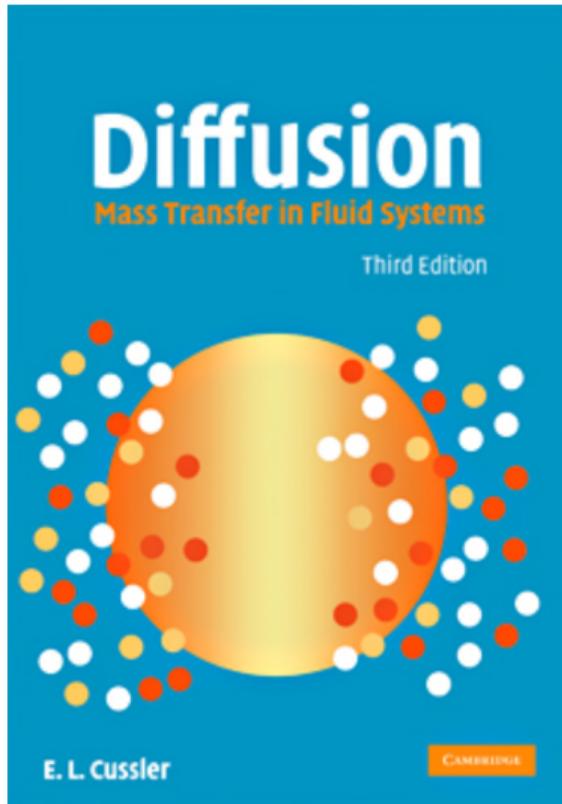


from Chabay and Sherwood, *Matter and Interactions*, 4th edition, Wiley (2015)

The momentum now contains the history of the impulses on the particle, but the future momentum depends on the net force now.

## 5 Examples

Textbook Companion open-source project [[Link](#)] aims to create a repository of solved examples of standard engineering textbooks in Jupyter Notebook format, coded in Python.



## 5 Language Support

Jupyter is not limited to Python and includes a variety of kernels for:

- ▶ MATLAB
- ▶ R
- ▶ C++
- ▶ Fortran
- ▶ Java
- ▶ C#
- ▶ Scala
- ▶ Mathematica
- ▶ ...

## 5 Other Tools and Considerations

- ▶ nbgrader
- ▶ OkPy.org
- ▶ nbconverter
- ▶ Hosted Notebooks
  - JupyterHub (even on a local sever)
  - Microsoft Azure Notebooks
  - Anaconda Cloud
  - ...

# 5 Time for a Demo!

Let's create a simple notebook.

## Simple Calculus

Let's calculate a simple integral. We use SymPy for this purpose.

```
In [14]: from sympy import init_session  
init_session(quiet=True)
```

IPython console for SymPy 0.7.6.1 (Python 3.5.2-64-bit) (ground types: python)

We should define the function, and then, we can use the *Integral* function to perform the integration.

```
In [15]: f = exp(x) * cos(x)  
a = Integral(f, x)
```

Now, to display the results, we use a combination of *Eq* and *doit* functions.

```
In [16]: Eq(a, a.doit())
```

```
Out[16]: 
$$\int e^x \cos(x) dx = \frac{e^x}{2} \sin(x) + \frac{e^x}{2} \cos(x)$$

```

## 5 Wrap It Up

- ▶ We can use Jupyter to teach the mathematical modeling of bioreactors step by step, which helps students understand the fundamental concepts of
  - Mass Transfer
  - Convection (Advection)
  - Fluid Flow
  - Practical Numerical Simulation
  - **Real World TE Modeling**in bioreactors and Tissue Engineering.
- ▶ Jupyter could also be considered as an effective platform for the assignments of this course and similar courses with a bunch of mathematical and computational operations.

## 5 References

- ▶ I. Martin et al., The role of bioreactors in tissue engineering, *TRENDS in Biotechnology*, 22(2), 2004
- ▶ J. Hansmann et al., Bioreactors in tissue engineering – principles, applications and commercial constraints, *Biotechnology Journal*, 8, 2013
- ▶ K.J. Blose et al., Bioreactors for Tissue Engineering Purposes, in book: *Regenerative Medicine Applications in Organ Transplantation*, 2014
- ▶ S. J. Hollister, Engineering Scaffold Mechanical and Mass Transport Properties, in book: *Comprehensive Biomaterials*, 2011
- ▶ J. Rosser et al., Bioreactor processes for maturation of 3D bioprinted tissue, in book: *3D Bioprinting for Reconstructive Surgery, Techniques and Applications*, 2018
- ▶ E. J. Levorson et al., Scaffolds: Flow Perfusion Bioreactor Design, in book: *Comprehensive Biomaterials*, 2011
- ▶ C. Julien et al., Bioreactor Monitoring, Modeling, and Simulation, *BioProcess International*, 2007
- ▶ X. Yan et al., Modeling of the Flow within Scaffolds in Perfusion Bioreactors, *American Journal of Biomedical Engineering*, 1(2), 2011
- ▶ F. Coletti et al., Mathematical Modeling of Three-Dimensional Cell Cultures in Perfusion Bioreactors, *Industrial & Engineering Chemistry Research*, 45, 2006

# Thank You