Physics-Informed Neural Network Model for Cell Viability and Oxygen Consumption of Pancreatic Islets

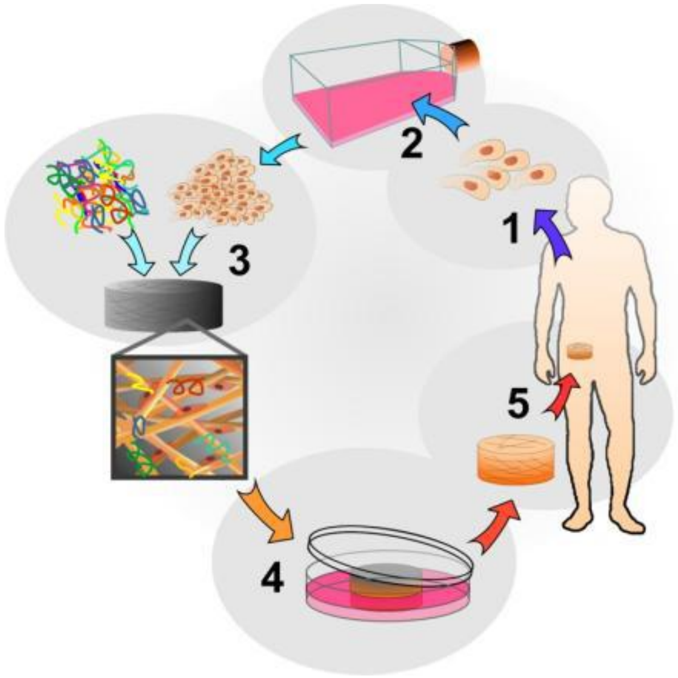
Mojtaba Barzegari, Liesbet Geris

Biomechanics Section, Department of Mechanical Engineering,

KU Leuven, Leuven, Belgium

Tissue Engineering

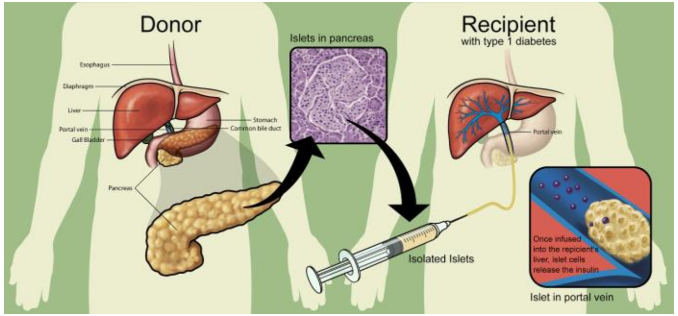
- An interdisciplinary field that applies the principles of engineering and life sciences toward the development of biological substitutes that restore, maintain, or improve tissue function or a whole organ (Langer & Vacanti, 1994)
- Cell, carrier, and growth factors

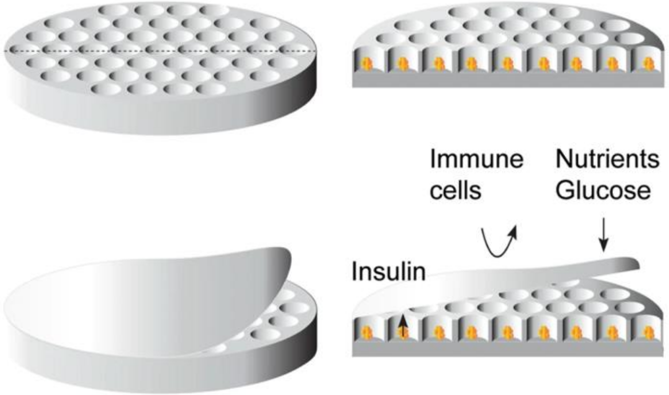


(source: J.H. George)

Pancreatic Islets Transplantation

- Islet transplantation to treat type 1 diabetes
- Proper preparation in a culturing process
- Viability of cultured islets is limited by hypoxia (death due to lack of vascularization and oxygen)
- Pancreatic islets are vulnerable due to large size
- Putting islets into micro-devices inside wells for gradient-driven oxygen diffusion





(sources: M.A. Naftanel et al., 2004, and K. Skrzypek et al., 2017)

KU LEUVEN

Problem Definition

- Challenge:
  - Tuning the culture device such that it prevents hypoxia
- Can be solved by:
  - Mathematical modeling of oxygen supply and consumption
  - Coupling oxygen transport models with vascularization models
  - Simulating the models by solving the transient mathematical equations

KU LEUVEN

Modeling Workflow

Underlying Phenomena → **Mathematical Models** → **Computational Models**

Underlying Phenomena	**Mathematical Models**	**Computational Models**
Oxygen consumption/supply	Partial differential equations	Finite element method
Hypoxia	Reaction-Diffusion-Convection	Physics-informed neural networks
Vascularization	Navier-Stokes equations	

KU LEUVEN

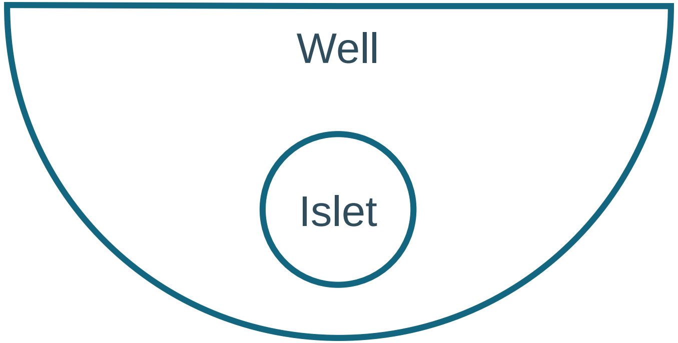
Constructing Mathematical Model

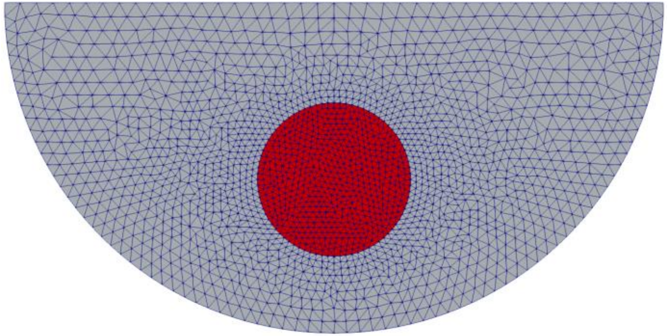
- Converting the biological phenomena into mathematical forms
- Reaction-diffusion-convection partial differential equations (PDE)
- Oxygen transport equation:

$$\frac{\partial C_{\mathrm{O_2}}}{\partial t} = \underbrace{\nabla.\left(D\nabla C_{\mathrm{O_2}}\right)}_{\text{Diffusion}} - \underbrace{\nabla.\left(\mathbf{u}C_{\mathrm{O_2}}\right)}_{\text{Convection}} + \underbrace{R_{\mathrm{max,O_2}} \frac{C_{\mathrm{O_2}}}{C_{\mathrm{O_2}} + C_{\mathrm{MM,O_2}}} . \varphi \frac{C_{\mathrm{gluc}}}{C_{\mathrm{gluc}} + C_{\mathrm{MM,gluc}}} . \delta\left(C_{\mathrm{O_2}} > C_{\mathrm{cr}}\right)}_{\text{Reaction}}$$

Implementing Computational Model

- Constructing the geometry
- Discretizing PDEs, finite element method
- Various techniques to linearize equations:
  - Picard-relaxation
  - Newton
- Requires a mesh to begin with
- Non-linear equations can become a problem





KU LEUVEN

Alternative Technique for Solving PDEs

- Commonly-used techniques: finite element and finite difference methods
- Approximating the derivative terms and variational formulation
- What if we employ some techniques from machine learning (ML)?
- Embedding the physics into the ML model
- Physics-informed neural networks (PINN)

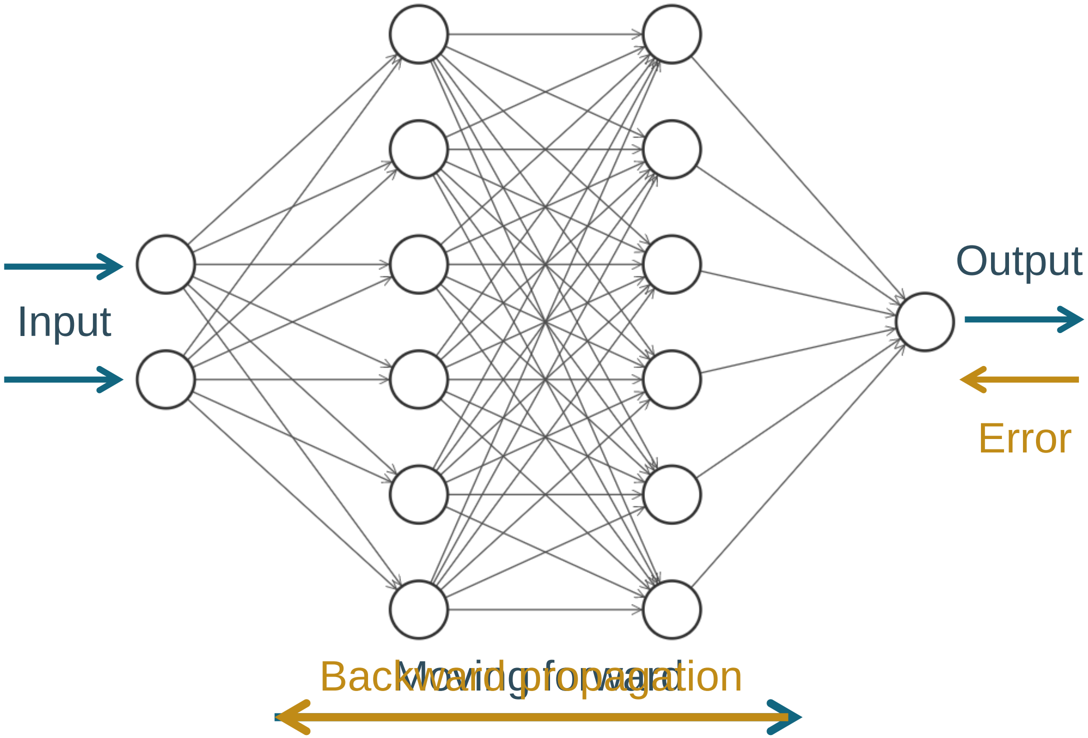
Why Physics-Informed Neural Networks?

- Handshaking of HPC and AI for mechanistic modeling
- Dealing with nonlinearity
- Easier parameterization
- Enhanced inverse problems formulation
- PDE-constrained optimization
- Extensibility

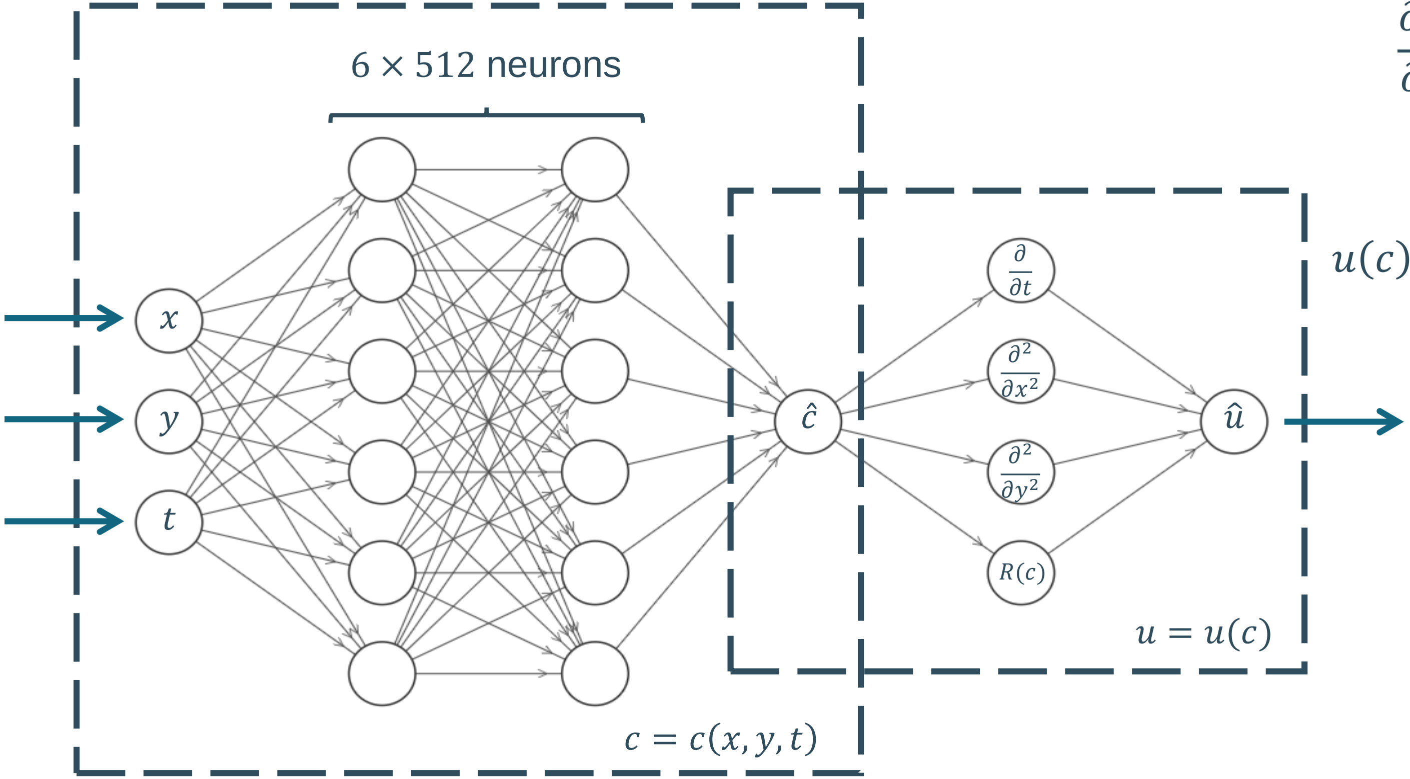
KU LEUVEN

Core Idea of PINN Models

- PINN models are deep neural network (NN) models in supervised learning
- Approximating functions using NN models
- By automatic differentiation, the derivatives of the function can be approximated



Solving PDEs Using PINN Models



$$\frac{\partial c}{\partial t} = \nabla.(D\nabla c) + R(c)$$

$$c = c(x, y, t)$$

$$u(c) = \frac{\partial c}{\partial t} - \nabla.(D\nabla c) - R(c)$$

$$u(c) = 0$$

KU LEUVEN

Training PINN Models

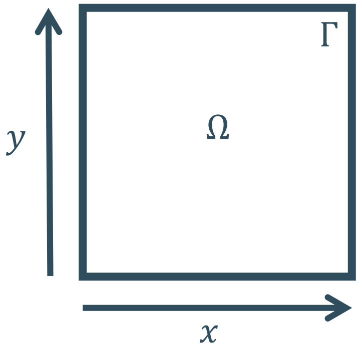
- Defining the problem, the domain of interest, and the initial and boundary conditions

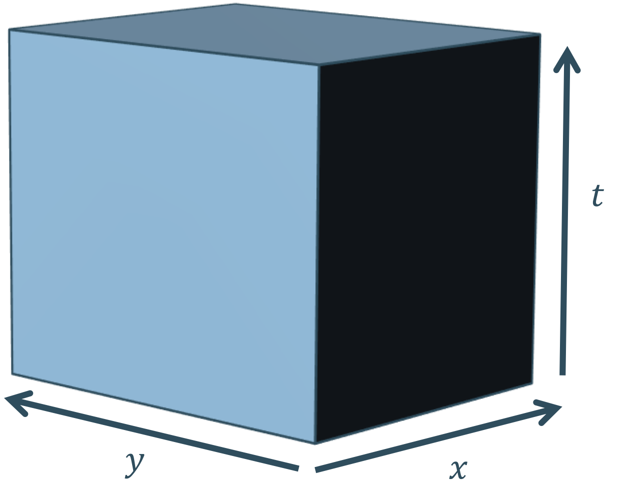
$$u(c) = \frac{\partial c}{\partial t} + \nabla.(-D\nabla c) - R(c) = 0$$

$$c = c(x, y, t), \qquad x, y \in \Omega, t \in [0, T]$$

$$c(x, y, 0) = c_{IC}, \qquad x, y \in \Omega$$

$$c(x, y, t) = c_{BC}\,, \qquad x, y \in \Gamma, t \in [0, T]$$





Training PINN Models

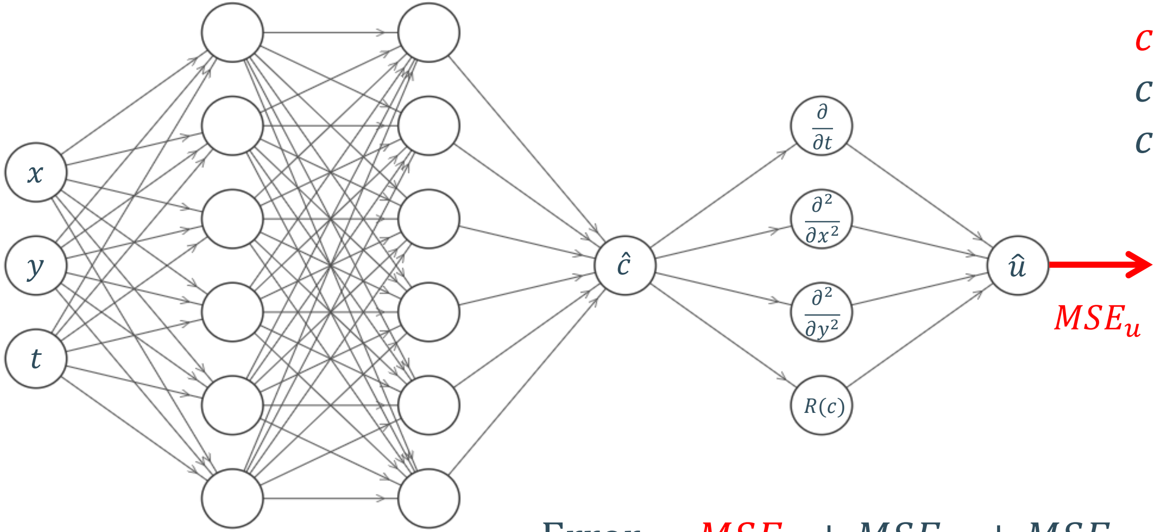
- Deriving the error for gradient descent

$$u(c) = \frac{\partial c}{\partial t} + \nabla.(-D\nabla c) - R(c) = 0$$

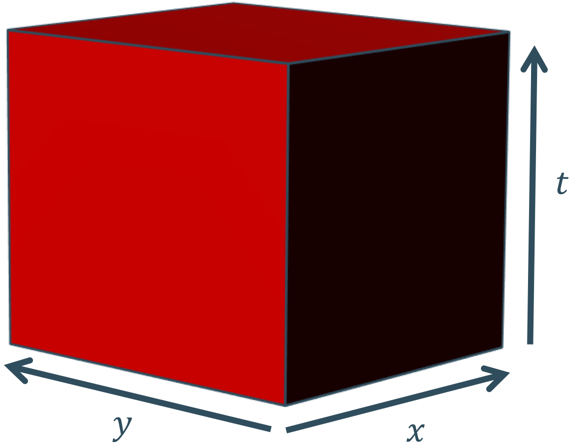
$$c = c(x, y, t), \qquad x, y \in \Omega, t \in [0, T]$$

$$c(x, y, 0) = c_{IC}, \qquad x, y \in \Omega$$

$$c(x, y, t) = c_{BC}\,, \qquad x, y \in \Gamma, t \in [0, T]$$



$$\text{Error} = MSE_u + MSE_{IC} + MSE_{BC}$$



Training PINN Models

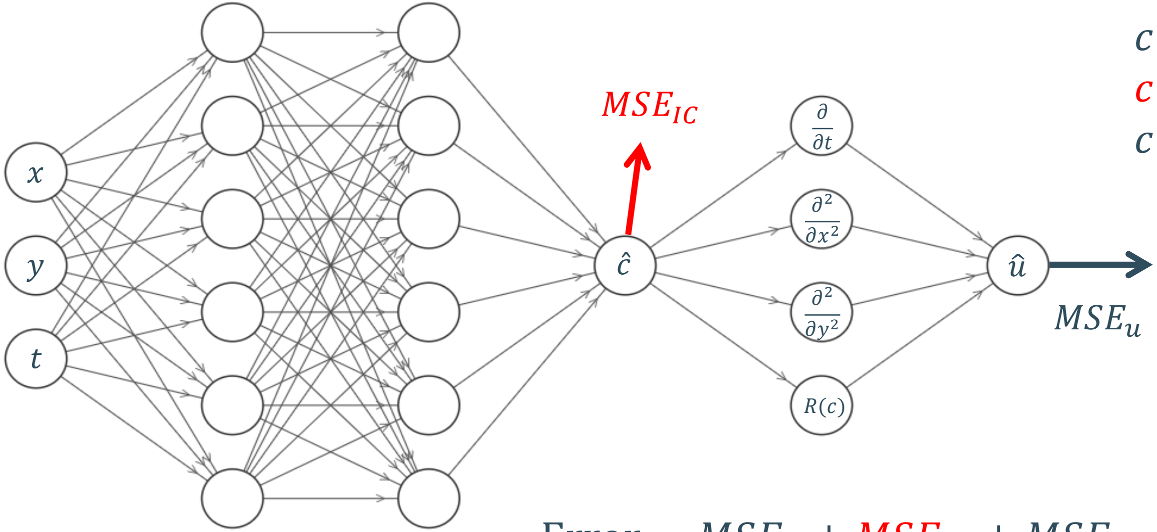
- Deriving the error for gradient descent

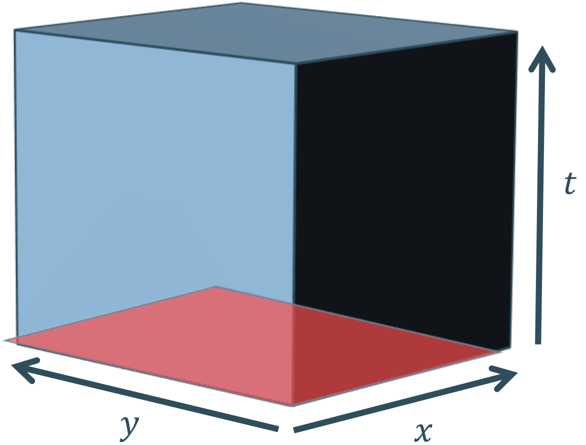
$$u(c) = \frac{\partial c}{\partial t} + \nabla.(-D\nabla c) - R(c) = 0$$

$$c = c(x, y, t), \qquad x, y \in \Omega, t \in [0, T]$$

$$c(x, y, 0) = c_{IC}, \qquad x, y \in \Omega$$

$$c(x, y, t) = c_{BC}\,, \qquad x, y \in \Gamma, t \in [0, T]$$





$$\text{Error} = MSE_u + MSE_{IC} + MSE_{BC}$$

KU LEUVEN

Training PINN Models

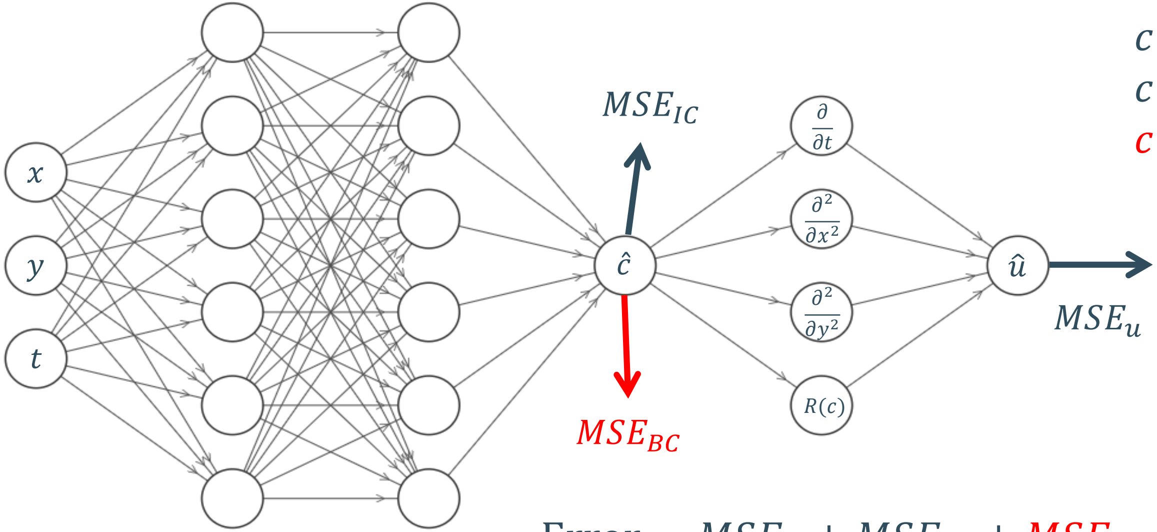
- Deriving the error for gradient descent

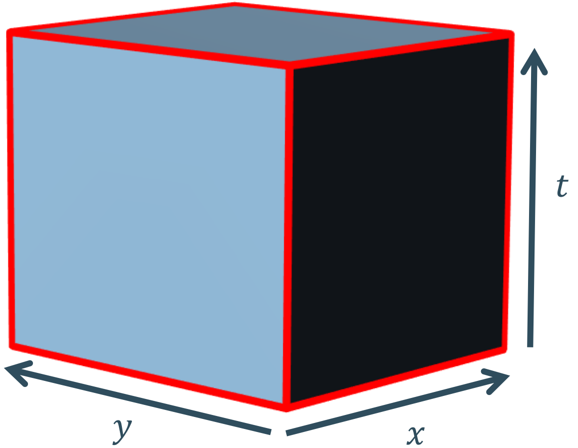
$$u(c) = \frac{\partial c}{\partial t} + \nabla.(-D\nabla c) - R(c) = 0$$

$$c = c(x, y, t), \qquad x, y \in \Omega, t \in [0, T]$$

$$c(x, y, 0) = c_{IC}, \qquad x, y \in \Omega$$

$$c(x, y, t) = c_{BC}\,, \qquad x, y \in \Gamma, t \in [0, T]$$





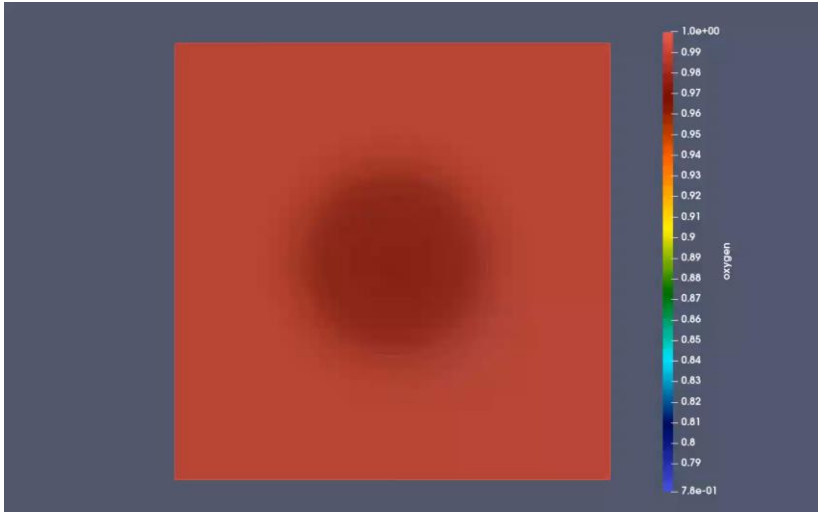
$$\text{Error} = MSE_u + MSE_{IC} + MSE_{BC}$$

Implementing PINN Model

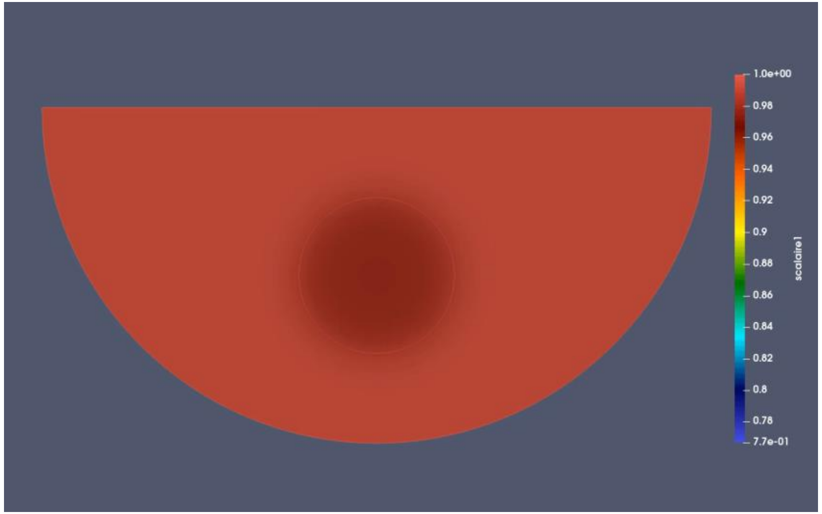
- Implemented using TensorFlow and NVIDIA SimNet
- Different diffusion rates for cells and culture environment
- Heaviside formulation to track variable diffusion and initial conditions
- Scaling and normalizing the problem is crucial
- Treating time becomes tricky for large time scales

Finite Element Results

- Oxygen concentration profiles showing consumption/supply



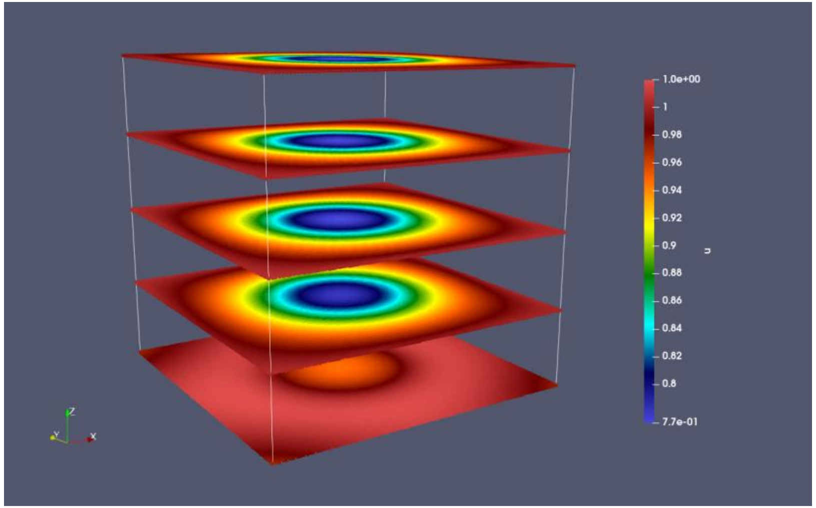
Rectangular well



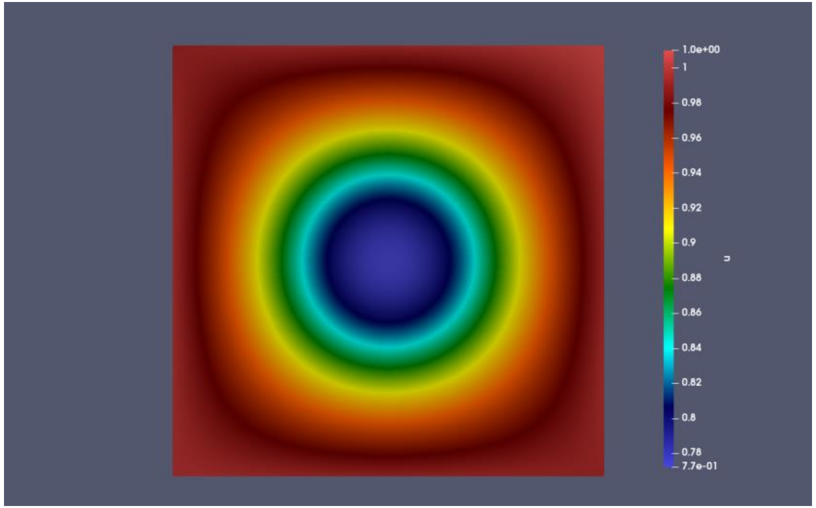
Circular well

PINN Results

- Oxygen concentration profiles showing consumption/supply

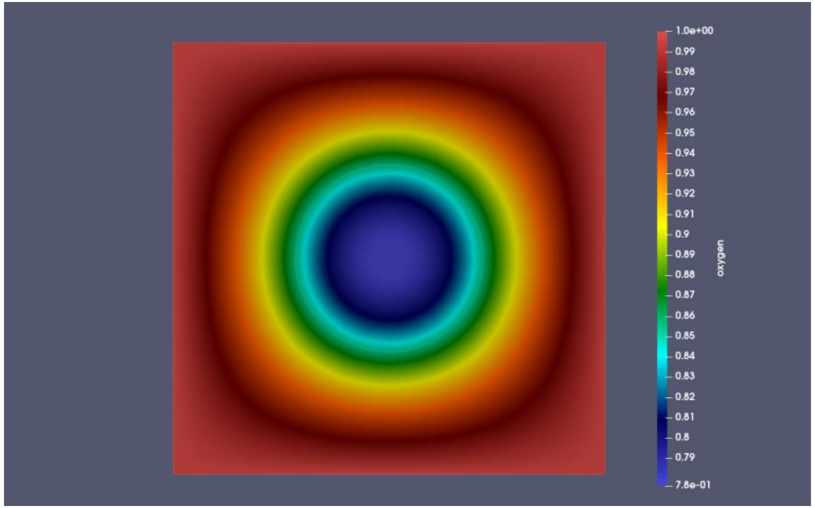


Time as the Z axis

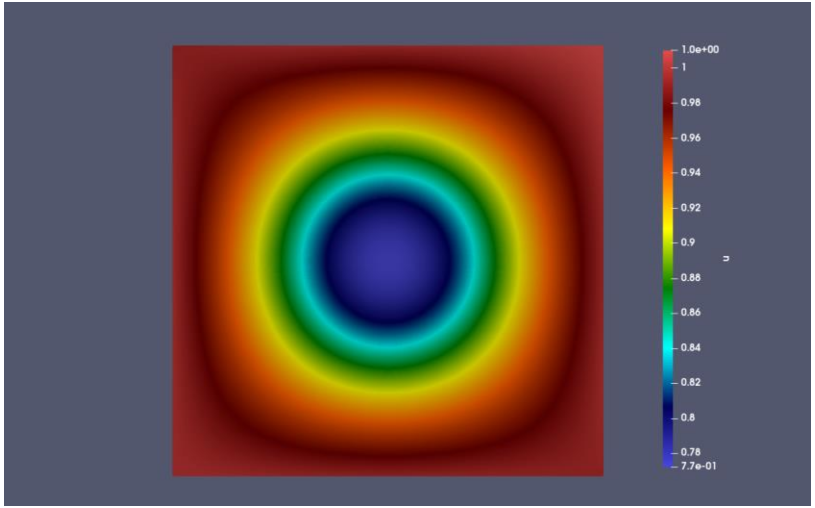


Final state

Numerical Simulation vs. PINN Solver

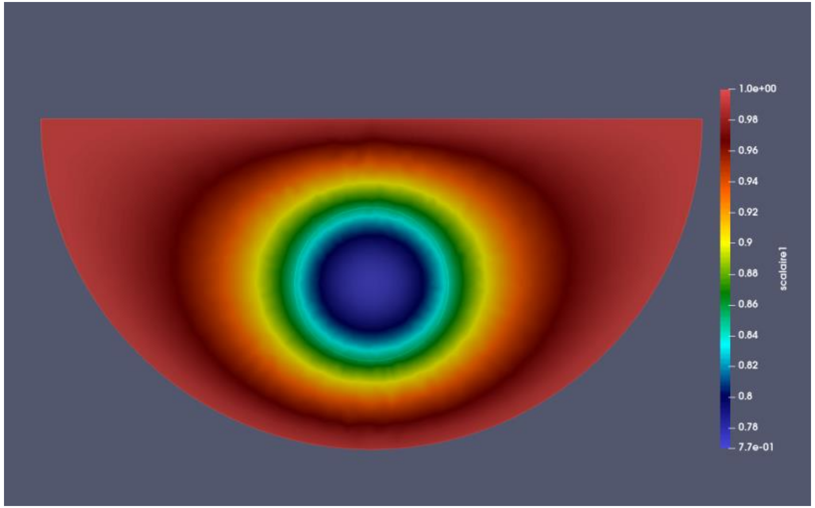


Final state using the
finite element solver

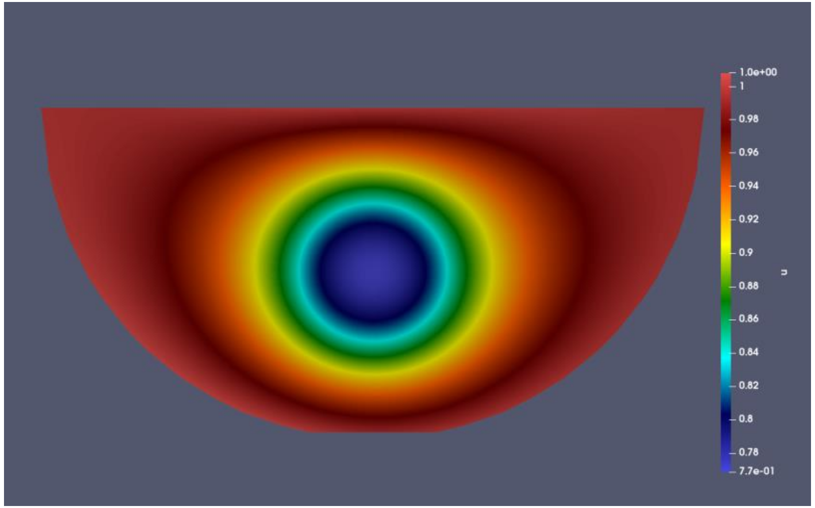


Final state using
the PINN solver

Numerical Simulation vs. PINN Solver



Final state using the
finite element solver



Final state using
the PINN solver

Conclusion

- Implanting pancreatic islets into microwells to increase viability
- Mathematical and computational modeling of the oxygen consumption and supply to assess cell viability and hypoxia
- Physics-informed neural networks (PINN) to solve governing equations
- Demonstrating the PINN model’s equivalence to the traditional numerical schemes in this case

KU LEUVEN

Thank you for your attention!

https://mbarzegary.github.io

Mojtaba.Barzegari@kuleuven.be

@MojBarz

@mbarzegary

KU LEUVEN