

# Adam Blakey, PhD Student

Maths Careers Event

15 Feb 2021

# Talk Layout



---

Background



---

Experiences



---

Work



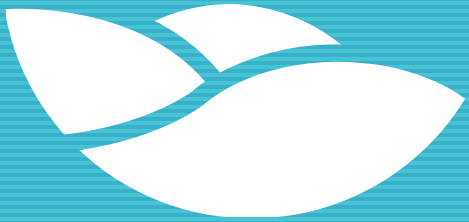
---

The PhD



---

Top Tips



# Adam's Background

# Background



- Geordie (Newcastle-ish)
- Clarinet
- Scouts
- Computers





# Adam's Experiences



# Experiences



- Study abroad
- Cub Leader
- Blowsoc





# Adam's Work

# Work



- ★ Research internship
- ★ Programming internship
- Website design



**EPSRC**  
Engineering and Physical Sciences  
Research Council

**Modelling Flow in 3D Printing**  
Adam M Blakey  
School of Mathematical Sciences, University of Nottingham

**University of Nottingham**  
UK | CHINA | MALAYSIA

### Introduction

- 3D printing is a popular means of prototyping and creating customised consumer goods [1], so it is important that we understand how the printing process affects the material properties.
- Polymers are complex molecules that deform easily under flow; deformed micro-structures may behave very differently to the material in equilibrium.
- The first step is to understand how flow through the nozzle deforms the polymer. It is often assumed, for simplicity, that the polymer molecules reach steady-state within the nozzle; however this condition will depend on temperature, print speed, and molecular weight.
- Polymers that have not reached steady-state will have a deformation memory of the contractions that occur in typical nozzle geometries.
- The strength of a 3D-printed part depends on what the micro-structure looks like, so we need to understand how the flow during printing affects the micro-structure.

### Simple Shear Flow (SSF)

- SSF occurs when a fluid is between a moving and stationary plate.
- For  $\mathbf{u} = (u, v, w)$ , we assume:
  - Flow only in  $x$ -direction ( $v = w = 0$ );
  - 2D geometry ( $\frac{\partial}{\partial z} = 0$ );
  - Conservation of mass ( $\frac{\partial u}{\partial x} = 0$ ).
- We also impose a no-slip boundary condition on  $u(y)$ , so that  $u(H) = V$  and  $u(0) = 0$ , so the velocity profile is given as  $u(y) = y\frac{V}{H}$ .
- Yields shear rate,  $\frac{\partial u}{\partial y} = \frac{V}{H} = \dot{\gamma}$ ,  $\dot{\gamma}$  is constant.

### The Rolie-Poly Equation

- Polymer melts consist of long chains of molecules that are well-entangled with each other. This idea is represented mathematically by a tube.
- Relaxation Times:
  - $\tau_d$ : Retraction time – time to retract through tubes.
  - $\tau_R$ : Rouse time – time to polymer stretch.
- The Weissenberg Number,  $Wi := \tau_d \dot{\gamma}$ , is defined by a timescale ( $\tau = \tau_d$  here).
- The deformation of the tube due to flow is described by the simple constitutive Rolie-Poly equation [2]:
 
$$\frac{d\mathbf{A}}{dt} = \mathbf{K} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{K}^T - \frac{1}{\tau_d} (\mathbf{A} - \mathbf{I}) - \frac{2}{\tau_d} \left( \frac{1}{\sqrt{\text{Tr}(\mathbf{A})}} \right) (\mathbf{A} + \mathbf{I}) \left( \frac{1}{\sqrt{\text{Tr}(\mathbf{A})}} \right) (\mathbf{A} - \mathbf{I})$$
- Specifically for SSF, we nondimensionalise the Rolie-Poly with respect to the retraction time, which gives:
 
$$\frac{d\mathbf{A}}{dt} = \mathbf{W} \left( \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{A} + \mathbf{A} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}^T - \frac{2\tau_d}{\tau_R} \left( \frac{1}{\sqrt{\text{Tr}(\mathbf{A})}} \right) (\mathbf{A} + \mathbf{I}) \left( \frac{1}{\sqrt{\text{Tr}(\mathbf{A})}} \right) (\mathbf{A} - \mathbf{I}) \right)$$
- $Wi \ll 1$  gives Newtonian-like behaviour.
- $Wi \gg 1$  gives an overshoot in the stress.

### Asymmetric Nozzle Flow

- We will now consider flow through a pipe in cylindrical polar coordinates.
- For  $\mathbf{u} = (u, v, w)$ , we assume:
  - Flow only in  $z$ -direction ( $u = v = 0$ );
  - Conservation of mass ( $\nabla \cdot \mathbf{u} = 0 \Rightarrow \frac{\partial w}{\partial z} = 0$ );
  - Axis-symmetric flow ( $\frac{\partial}{\partial \theta} = 0$ ).
- To conserve momentum in the nozzle, we introduce the following Cauchy conservation of momentum (CoM) equation:
 
$$\frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \frac{1}{\rho} \left( \frac{\partial \tau_{rz}}{\partial r} + \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rz}) \right)$$
- For the time-independent case, we can integrate the equation with respect to  $r$ , assuming  $\tau_{rz}(0) = 0$  and  $w(R_0) = 0$ . This yields a parabolic velocity profile and linear shear rate:
 
$$w = \frac{1}{4\mu} \frac{R_0^2}{L} \frac{\partial p}{\partial z} \left( 1 - \frac{r^2}{R_0^2} \right)$$
- We solve the time-dependent case for many times.
- We notice the steady-state curve is a parabola, as expected from the time-independent case.

### Polymer Melt Nozzle Flow

- We couple the Rolie-Poly equation with the following Cauchy CoM equation:
 
$$\frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \frac{1}{\rho} \left( \frac{\partial \tau_{rz}}{\partial r} + \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rz}) \right) + \left( \frac{\partial \tau_{rz}}{\partial t} + \frac{\partial}{\partial z} (w \tau_{rz}) \right)$$
- Due to the extra terms from the polymer contribution, the CoM equation can no longer be solved analytically.
- For a small Weissenberg number ( $Wi \approx 0.48$ ) and fixed parabolic velocity profile, we show the evolution of the polymer tensor in the next figure.

### Future Work

- Varying the polymer chain length,  $Z$ , against time;
- Coupling the heat equation;
- Experimenting with nozzle lengths for the polymer to reach steady-state.

### References

- [1] C. Milroy, P. D. Unwin, Deformation of an amorphous polymer during the fused-filament-fabrication method for additive manufacturing (Journal of Rheology 81, 2017).
- [2] A. E. Likhtman, R. S. Graham, Simple constitutive equation for linear polymer melts derived from molecular theory: Rolie-Poly equation (Journal of Non-Newtonian Fluid Mechanics, 2003).

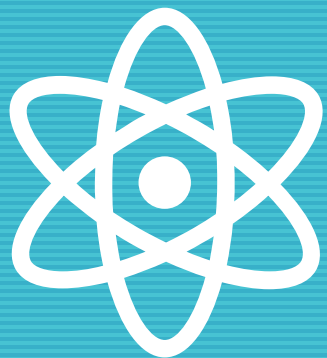
<https://r.blakey.family/3DPrinting>
Supervisors: Claire McIlroy and Richard Graham
adamblakey.family

**Keiron Anderson**

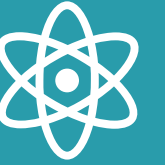
**Composer of modern classical music, conductor, teacher.**

**K**eiron Anderson is a consummate all-round musician experienced as a musical director, composer, performer, adjudicator and teacher. He is a firm believer in the accessibility of all the music played under his direction even though some of it may be unfamiliar to audiences. All his efforts with students, ensembles and audiences are designed to communicate music with emotion and energy "...no matter what else we do, we must get to the emotional content of the music and bring it to life for the audience ..."



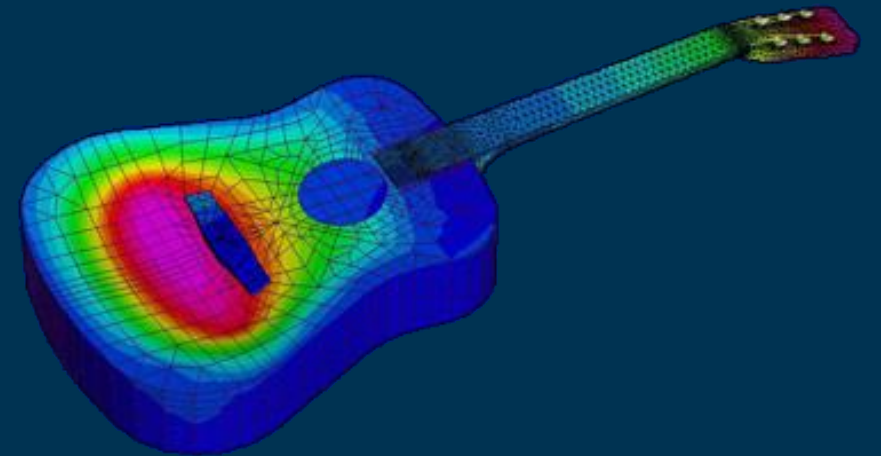


# Adam's PhD



## ‘Finite Element Methods for Fluid-Structure Interaction Problems’

- Scientific Computation
- Mathematical Biology
- Fluid Dynamics
- Multiscale Modelling
- Data-Driven Modelling





## A Typical Day:

- Reading papers
- Meetings
  - Supervisor
  - PGR Rep
- Seminars
- Programming
- Teaching
- Socialising

**Programming**  
08:30 – 10:00

**PGR Rep Catchup**  
10:00 – 11:00

**Reading papers**  
11:00 – 12:00

**Lunch!**  
12:00 – 13:00

**Core Python Programming**  
13:00 – 14:00

**Doing some calculations**  
14:00 – 15:00

**Fluid Mechanics Seminar**  
15:00 – 16:00

**Supervisor Meeting**  
16:00 – 17:00

**Read emails, 17:00**  
**Backup work, 17:15**

```

module myMesh
  implicit none
  type(element_interval)
  integer :: elementNo
  integer :: nodes
  integer :: dimension(2)
  integer :: dimension(1), allocatable
  integer :: polynomialDegree

  elementNo = 1
  nodes = 2
  nodeIndices(1) = 1
  nodeIndices(2) = 2
  polynomialDegree = 1

  allocate(nodeCoordinates(2))
  nodeCoordinates(1) = 0
  nodeCoordinates(2) = 1
  call my_elementConstructor(elementNo, nodes, nodeIndices,
    print = *, my_elementType, localise)
  call my_elementDestructor()
  deallocate(nodeCoordinates)

  type(myMesh) :: myMesh
  type(mySolution) :: mySolution

  real(dp) :: epsilon

  call myMeshConstructor_ex(0)
  call mySolutionConstructor(myMesh, f, epsilon, c)

  print *, "Adam"
  print *, myMeshNodes(2)element_typeMap_localToGlobal()
end module myMesh

procedure :: testadam
  use common
  real(dp) :: x
  real(dp) :: double_double
  interface
    real(dp) :: double_double
  end interface
  contains
    subroutine testadam(this)
      class(mySolution) :: this
      real(dp) :: x
      real(dp) :: double_double
      integer :: n

      this%SolutionMesh = a_mesh ! is this making a copy?
      this%epsilon = a_epsilon
      this%epsilon = a_epsilon
      this%epsilon = a_epsilon

      call this%testadam()
      call this%allocate_hofs()

      this%Dofs = 4
      n = this%Dofs
      allocate(this%startDofs(n))
      allocate(this%u(n))
    end subroutine testadam
  end interface
end module mySolution
  
```





A PhD was for me because:

- I enjoyed my research internship
- I enjoyed my dissertation and project
- I enjoyed programming
  - Not necessary for a PhD, but it helped me!



# Adam's Top Tips



# Top Tips



- Say yes!
- Try new experiences
- PhDs are not like UG degrees



Studying for a PhD is less about being “smart”, and more about:

- Self-motivation
- Eagerness to learn
- Willingness to persevere



# Thank You!

[adam.blakey@nott.ac.uk](mailto:adam.blakey@nott.ac.uk)

<https://adam.blakey.family>