MANCHESTER 1824
The University of Manchester

New rheology models, non-Newtonian formulations and multiphase flows

GEORGIOS FOURTAKAS

THE UNIVERSITY OF MANCHESTER

georgios.fourtakas@manchester.ac.uk

Outline

- Motivation
- Mathematical description of non-Newtonian flows
- Implementation
- Input configuration (xml file)
- Examples
- Work in progress
- Conclusions

Motivation

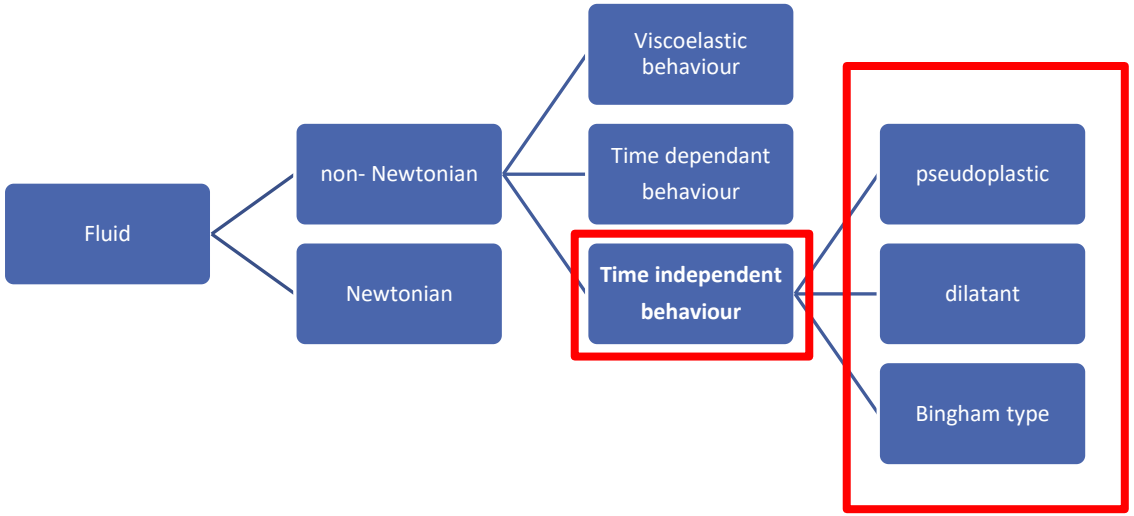
- Not all fluids are Newtonian
- Industrial applications use fluids that exhibit non-Newtonian characteristics
- These are usually multi-phase flows of complex fluids
- Two or more phases with:
  - different densities
  - different viscosities (at rest)
  - varying viscosity under stress
  - exhibit a yield strength

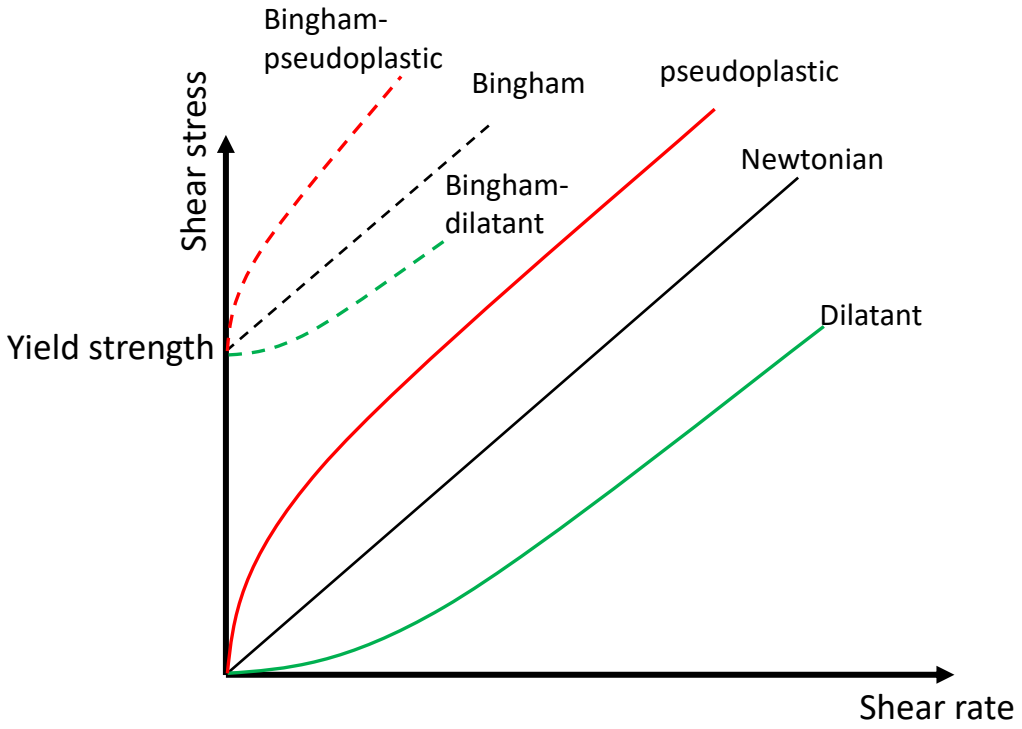
Corn starch at 30Hz

Credit: www.youtube.com by bendhoward

Description of non-Newtonian flows

- A non-Newtonian fluid viscosity is depended on shear stress/rate/history
- Usually, non-Newtonian fluids do not exhibit linear shear stress-rate behaviour
- In DualSPHysics v5.0 we have implemented time independent fluids
- A number of combinations and options are possible





Mathematical description of non-Newtonian flows

- The governing equations read

$$\frac{\mathrm{d}\,\rho}{\mathrm{d}\,t} = -\rho\nabla\cdot\mathbf{u}$$

and

$$\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau} + \mathbf{F}$$

- In SPH formalism we obtain our usual classical form

$$\left\langle\frac{\mathrm{d}\,\rho}{\mathrm{d}\,t}\right\rangle = \rho_i\sum_j\frac{m_j}{\rho_j}\left(\mathbf{u}_i-\mathbf{u}_j\right)\cdot\nabla_i W_{ij} + \delta h c_0 D_i$$

and

$$\left\langle\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t}\right\rangle = -\sum_j m_j\left(\frac{p_j+p_i}{\rho_j\rho_i}\right)\nabla_i W_{ij} + \sum_j m_j\left(\frac{\boldsymbol{\tau}_i+\boldsymbol{\tau}_j}{\rho_i\rho_j}\right)\cdot\nabla_i W_{ij}$$

- We will be using a constitutive equation for the shear stresses (second term on the right-hand side of the momentum equation)

Mathematical description of non-Newtonian flows

- For a Newtonian fluid, the constitutive equation reads

$$\boldsymbol{\tau} = 2\eta\dot{\boldsymbol{\gamma}}$$

with

$$\dot{\boldsymbol{\gamma}} = \frac{1}{2}[\nabla\mathbf{u} + \nabla^T\mathbf{u}] - \frac{1}{3}(\nabla\cdot\mathbf{u})\boldsymbol{I}$$

- For a non-Newtonian fluid, the viscosity is related to the shear rate

$$\boldsymbol{\tau} = 2\eta_{app}\dot{\boldsymbol{\gamma}}, \quad \text{and} \quad \eta_{app} = f(\dot{\boldsymbol{\gamma}}, \tau_y, \dots)$$

- In DualSPHysics v5.0 we use the so-called **Herschel-Bulkley-Papanastasiou constitutive equation** which reads

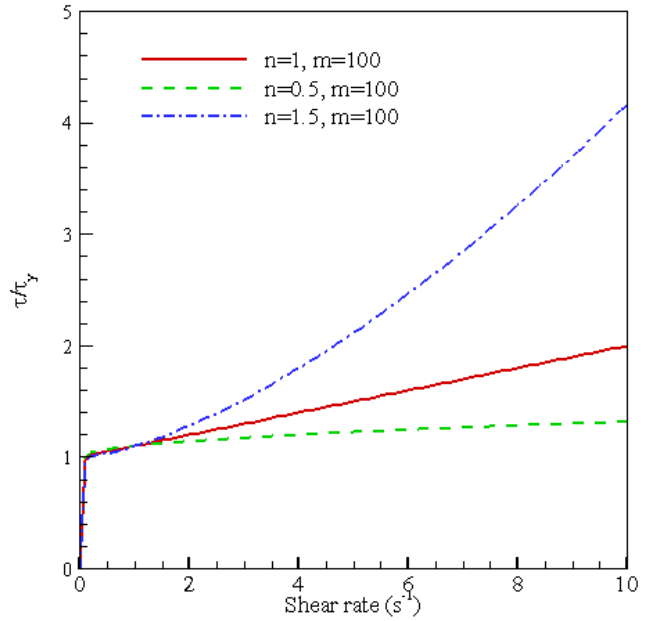
$$\eta_{app} = \eta|\dot{\gamma}|^{n_{HB}-1} + \frac{\tau_y}{|\dot{\gamma}|}\left[1 - e^{-m_P|\dot{\gamma}|}\right]$$

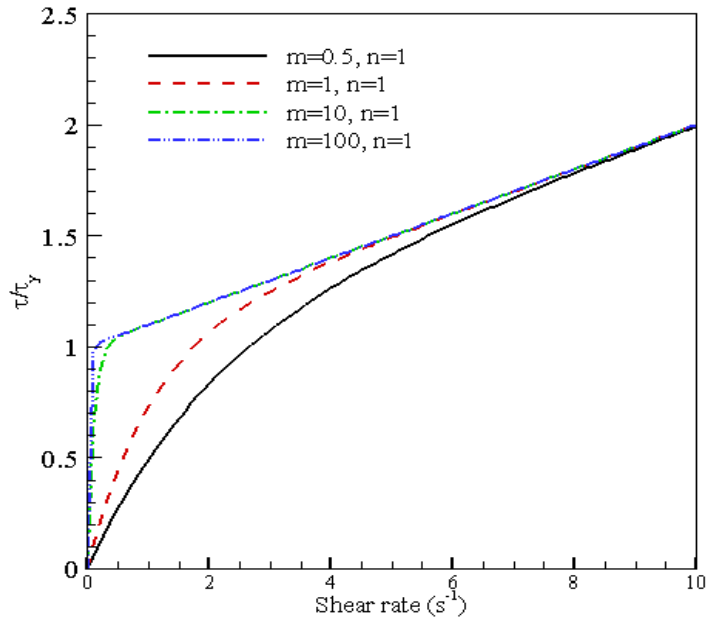
where, $|\dot{\gamma}|$ is the magnitude of the symmetric strain rate tensor (invariant), τ_y is the yield strength of the material, and parameters n_{HB} and m_P are the power law index and exponential shear rate growth parameter, respectively.

Mathematical description of non-Newtonian flows

- The Herschel-Bulkley-Papanastasiou (HBP) constitutive equation is a general viscoplastic model which can represent a variety of time independent non-Newtonian fluids
  - Newtonian
  - Shear thinning
  - Shear thickening
  - Bingham thinning
  - Bingham thickening
- As $m \rightarrow \infty$ the HBP model approximates the original Bingham model model and when $n \neq 1$ the model reduces to a Herschel-Bulkley (shear thinning or thickening)

- For $n = 1$ and $m = 0$ the model reduces to a Newtonian fluid
- Most importantly, it does not exhibit a discontinuity at $\tau = \tau_y$





Mathematical description of non-Newtonian flows

$$-\frac{1}{3}(\nabla \cdot \mathbf{u})\boldsymbol{I} = 0$$

There are **two** options for the viscous term

$$-\frac{1}{3}(\nabla \cdot \mathbf{u})\boldsymbol{I} \neq 0$$

- Assume **incompressible flow**
- Isotropic material (isotropic stress tensor)
- Divergence-free $(\nabla \cdot \mathbf{u} = 0)$

$$\frac{1}{\rho}\nabla \cdot \boldsymbol{\tau} = \frac{\eta_{app}}{\rho}\nabla^2\mathbf{u}$$

- SPH Laplacian operator by Morris et. al., 1997

$$\frac{\eta_{app}}{\rho}\nabla^2\mathbf{u} = \frac{1}{\rho_i}\sum_j \frac{m_j}{\rho_j}\frac{(\eta_{app,j} + \eta_{app,i})}{{x_{ij}}^2}\mathbf{x}_{ij} \cdot \nabla W_{ij}\,\mathbf{u}_{ij}$$

- Assume (weakly) **compressible flow**
- Discretise the shear stress tensor without any additions/modifications

$$\frac{1}{\rho}\nabla \cdot \boldsymbol{\tau} = \sum_j m_j \left(\frac{\tau_i + \tau_j}{\rho_i \rho_j}\right) \cdot \nabla_i W_{ij}$$

with

$$\boldsymbol{\tau} = 2\eta_{app}\dot{\boldsymbol{\gamma}}$$

Mathematical description of non-Newtonian flows

Velocity gradients : there are again **two** options

- To obtain the apparent viscosity we use the HBP constitutive equation

$$\eta_{app} = \eta|\dot{\gamma}|^{n_{HB}-1} + \frac{\tau_y}{|\dot{\gamma}|}\left[1 - e^{-m_P|\dot{\gamma}|}\right]$$

- We need the velocity gradients

$$\dot{\boldsymbol{\gamma}} = \frac{1}{2}[\nabla\mathbf{u} + \nabla^T\mathbf{u}]$$

- Using a finite differences approach (FDA)

$$\nabla\mathbf{u}_{ij} = \frac{\mathbf{u}_{ij}\mathbf{x}_{ij}}{{x_{ij}}^2}$$

- Fast, but over dissipative (first order gradient)
- Calculate *on the fly*

- Use the SPH gradient approach (SPH)

$$\nabla\mathbf{u}_i = \sum_j V_j\left(\mathbf{u}_j - \mathbf{u}_i\right)\nabla_i W_{ij}$$

- Slow, but as accurate as your SPH scheme
- *Pre-particle loop* required

Mathematical description of non-Newtonian flows

- Summarizing:

Two shear stress operators:
- Morris operator for the Laplacian
- SPH form of the shear stress

Velocity gradients:
- Finite differences approach
- SPH gradient approach

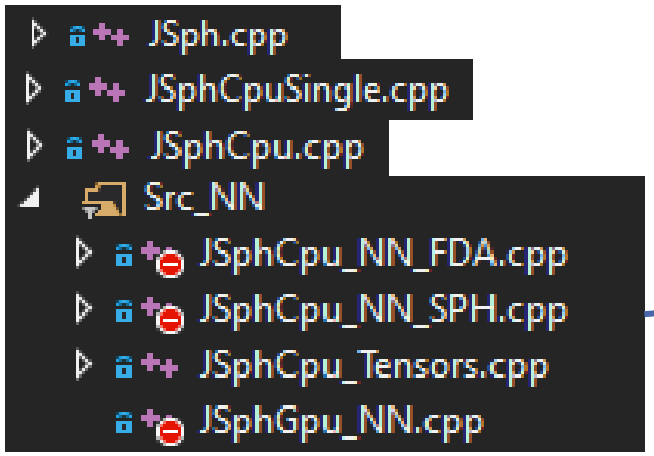
Any possible combination implemented in DualSPHysics v5.0

Multi-phase non-Newtonian solver capabilities:
- Newtonian
- Non-Newtonian (dilatant-pseudoplastic)
- Bingham (dilatant-pseudoplastic)
- Bi-viscosity fluids
- **And any combination of the above**

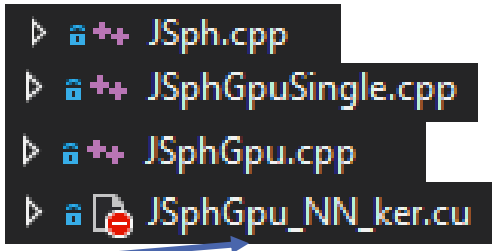
Implementation

- The computational implementation is split below in the CPU and GPU variants:

CPU files:



GPU files:



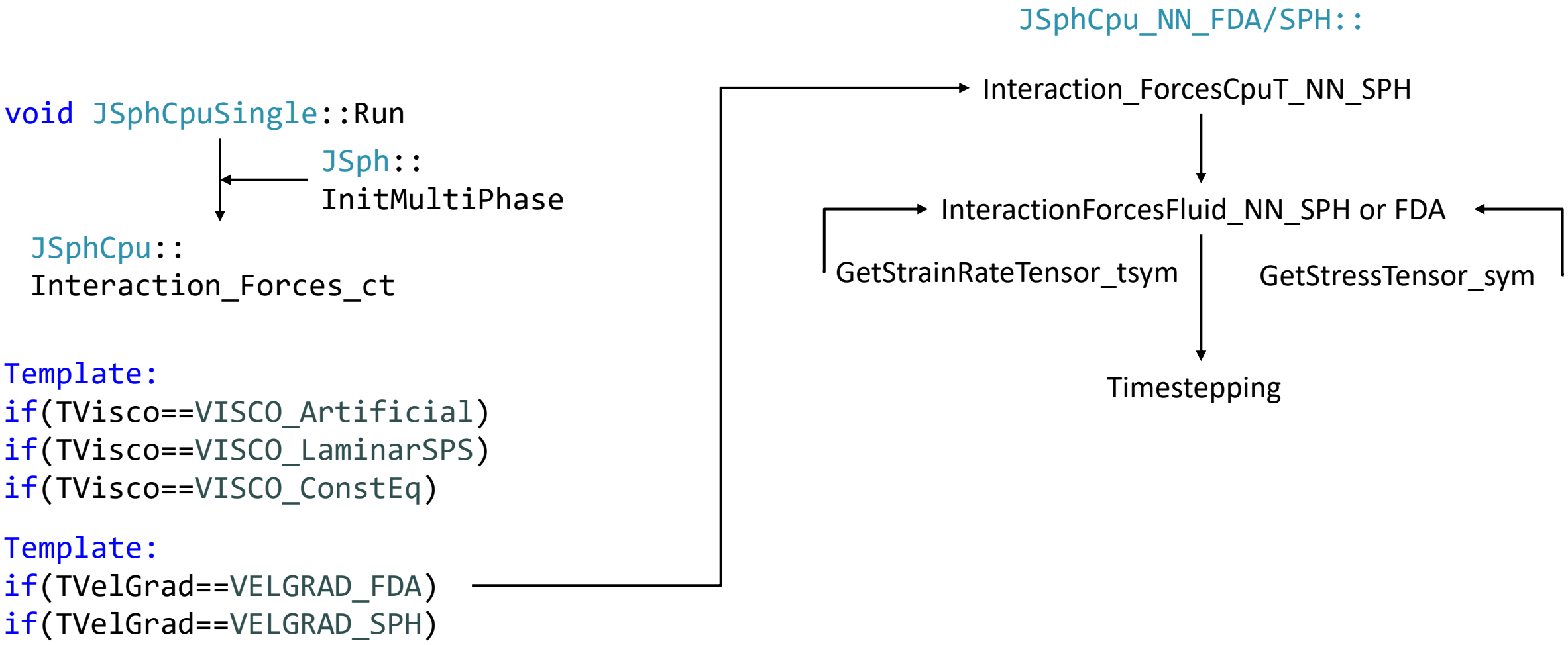
- All code changes in the main (excluding non-Newtonian specific) files are marked with the comment:

```
//<vs_non-Newtonian>
```

- The name of the functions in CPU and GPU are identical

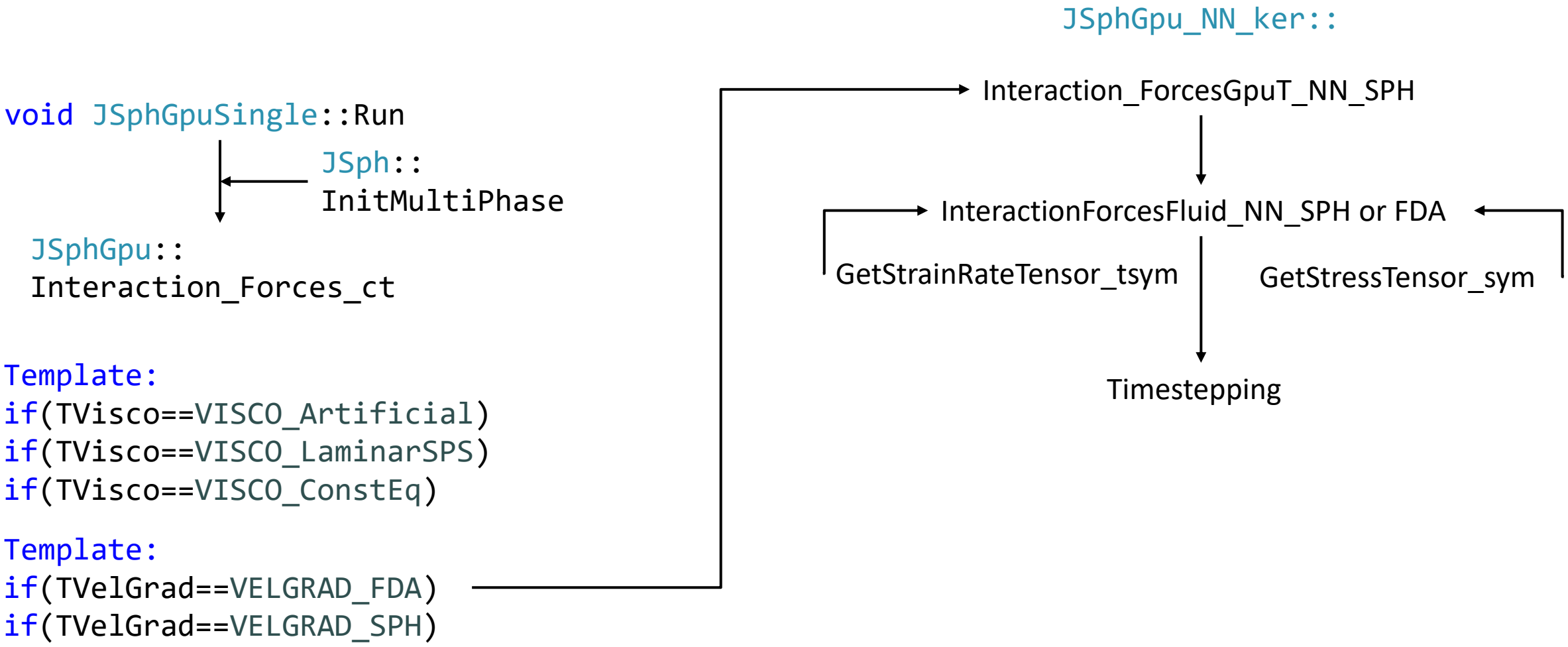
Implementation

CPU version flow chart



Implementation

GPU version flow chart



Implementation

- A compromise between computational efficiency and ease of development/modifications
- Hence, calculate many quantities which are not used but might be helpful to developers
- Main invariants and principal invariants are an example (for share rates and shear stress)
- Auxiliary arrays (aux) either for outputting directly from the code or store extra variables i.e., for outputting η_{app}
- CUDA kernels are split into two or more parts (for efficiency) to make the code readable/easy to change (see *Interaction_Forces_NN_XX* functions)

Implementation

- Herschel-Bulkley-Papanastasiou constitutive equation

```
//==============================================================================
__device__ void KerGetEta_Effective(const typecode ppx,float tau_yield,float D_tensor_magn,float visco
  ,float m_NN,float n_NN,float &visco_etap1)
{

  if(D_tensor_magn<=ALMOSTZERO)D_tensor_magn=ALMOSTZERO;
  float miou_yield=(PHASECTE[ppx].tau_max ? PHASECTE[ppx].tau_max/(2.0f*D_tensor_magn) : (tau_yield)/(2.0f*D_tensor_magn)); //HPB will adjust eta

  //if tau_max exists
  bool bi_region=PHASECTE[ppx].tau_max && D_tensor_magn<=PHASECTE[ppx].tau_max/(2.f*PHASECTE[ppx].Bi_multi*visco);
  if(bi_region) { //multiplier
    miou_yield=PHASECTE[ppx].Bi_multi*visco;
  }
  //Papanastasiou
  float miouPap=miou_yield *(1.f-exp(-m_NN*D_tensor_magn));
  float visco_etap1_term1=(PHASECTE[ppx].tau_max ? miou_yield : (miouPap>m_NN*tau_yield||D_tensor_magn==ALMOSTZERO ? m_NN*tau_yield : miouPap));

  //HB
  float miouHB=visco*pow(D_tensor_magn,(n_NN-1.0f));
  float visco_etap1_term2=(bi_region ? visco : (miouPap>m_NN*tau_yield||D_tensor_magn==ALMOSTZERO ? visco : miouHB));

  visco_etap1=visco_etap1_term1+visco_etap1_term2;

  /*
  //use according to you criteria
  - Herein we limit visco_etap1 at very low shear rates
  */
}
```

Input configuration (xml file)

- “Special-nnphases” tag

```
<special>
    <nnphases> %Defines non-newtonian phases parameters
        <phase mkfluid="0">
            <rhop value="1000" />
            <visco value="0.2" />
            <tau_yield value="0.0001" />
            <HBP_m value="100" />
            <HBP_n value="2" />
            <phasetype value="0" />
        </phase>
        <phase mkfluid="1">
            <rhop value="1000" />
            <visco value="0.1" />
            <tau_yield value="0.001" />
            <HBP_m value="10" />
            <HBP_n value="1" />
            <phasetype value="0" />
        </phase>
    </nnphases>
</special>
```

```
<!---Phase 1--->
<setmkfluid mk="0" />
<drawbox>
    <boxfill>solid</boxfill>
    <point x="0" y="0" z="0" />
    <size x="4" y="2" z="0.5" />
</drawbox>
<!---Phase 2--->
<setmkfluid mk="1" />
<drawbox>
    <boxfill>solid</boxfill>
    <point x="0" y="0" z="0.50" />
    <size x="1" y="2" z="0.25" />
</drawbox>
```

Input configuration (xml file)

- "Special-nnphases" phase tag

```
<phase mkfluid="0">
    <rhop value="1000" />
    <csound value="80" />
    <gamma value="7" />
    ...
    ...
    ...
</phase>
```

<rho /> : The tag defines the density of the phase of the specific **Mk** (not recommended for density ratios higher than two).

<csound /> : The tag defines the speed of sound for the phase to be used in the computation.
Note that, the maximum speed of sound of all phases with be used in the time step CFL restriction whereas the phase specific speed of sound will be used in the equation of state (obtaining the pressure). Thus, the model is bounded by the maximum phase speed of sound. This parameter is ***optional***, in this absence of this tag the speed of sound calculated by *gencase* will be used.

<gamma /> : The tag defines the polytropic index for the equation of state. This parameter is ***optional***, in the absence of this tag the default polytropic index defined in **<constantsdef/>** tag will be used.

Input configuration (xml file)

- “Special-nnphases” phase tag

```
<phase mkfluid="0">
    ...
    <visco value="0.2" />
    <tau_yield value="0.0001" />
    <HBP_m value="100" />
    <HBP_n value="2" />
    <phasetype value="0" />
</phase>
```

<visco /> : The tag defines the kinematic viscosity (or consistency index) of the phase.

<tau_yield /> : The tag defines the specific yield stress of the material (τ/ρ) the fluid may experience before yielding.

<HBP_m /> : The tag defines the Papanastasiou exponential stress growth of the fluid. A value of zero reduces the yield stress to zero (Newtonian), a value of the order of ten to pseudo-Bingham and order of 100 to Bingham.

<HBP_m /> : The tag defines the Herschel-Buckley flow index. A value of one reduces the model to linear whereas a value smaller than one to shear thinning and greater that one to shear thickening fluid.

Input configuration (xml file)

- “Special-nnphases” phase tag

```
<phase mkfluid="1">
    ...
    ...
    <tau_yield value="0.001" />
    <tau_max value="0.001" />
    <Bi_multi value="10.0" />
    ...
    ...
</phase>
```

<tau_max /> : The tag defines the Bingham specific yield stress of the material (τ/ρ) the fluid may experience before yielding.

<Bi_multi /> : The tag defines the viscosity multiplier of a bi-viscosity model. This value multiplies the tau_max in the yielded region to avoid the singularity of the Bingham model.

Note, these two values are ***optional*** and should be used only if a Bingham model (or bi-visocity) is required, *both tags must be present*. Further, these two tags override the Papanastasiou **<HBP_m/>** value (**HBP_m** = 0).

Examples

- Parameters

```
%Choice of reology treatment, velocity gradient calculation and viscosity treatment
<parameter key="RheologyTreatment" value="2" />
<parameter key="VelocityGradientType" value="1" "1:FDA, 2:SPH (default=1)" />
<parameter key="ViscoTreatment" value="3" "1:Artificial, 2:Laminar+SPS, 3:Constitutive eq." />
```

<RheologyTreatment/> : The tag defines the rheology treatment is used. In the v5.0 version only option *1-2 Single and multi-phase* is allowed.

<VelocityGradientType/> : The tag defines which gradient formulation is used when calculating the velocity gradients.

<ViscoTreatment/> : The tag defines which formulation is used when calculating the viscous forces. Three choices are available, artificial viscosity (Newtonian only), laminar viscous operator by Morris (1997) and the SPH constitutive equation.

```
<parameter key="RelaxationDt" value="0.2" "(default=0.2)" />
```

<RelaxationDt/> : This tag defines a relaxation parameter for the viscous forces time step restriction in the form of $\lambda = \frac{1}{RelaxationDt}$.

Examples

- Lock gate test case

```
<nnphases> %Defines non-newtonian phases parameters
    <phase mkfluid="0">
        <rhop value="1200" comment="Density of the phase" />
        <visco value="0.1" comment="Kinematic viscosity (or consistency index) value for phase (m2/s)" />
        <tau_yield value="0.004" comment="Specific yield stress of phase (Pa m3/kg)" />
        <HBP_m value="10" comment="Use 0 to reduce Newtonian liquid, order of 10 for power law and order of 100 for Bingham (sec) " />
        <HBP_n value="1.25" comment="Use 1 to reduce to Newtonian, <1 for shear thinning >1 for shear thickenning " />
        <phasetype value="0" comment="Non-Newtonian=0 only option in v5.0" />
    </phase>
    <phase mkfluid="1">
        <rhop value="1000" comment="Density of the phase" />
        <visco value="0.01" comment="Kinematic viscosity (or consistency index) value for phase (m2/s)" />
        <tau_yield value="0.0005" comment="Specific yield stress of phase (Pa m3/kg)" />
        <HBP_m value="1" comment="Use 0 to reduce Newtonian liquid, order of 10 for power law and order of 100 for Bingham (sec) " />
        <HBP_n value="1" comment="Use 1 to reduce to Newtonian, <1 for shear thinning >1 for shear thickenning " />
        <phasetype value="0" comment="Non-Newtonian=0 only option in v5.0" />
    </phase>
</nnphases>
```

```
%Choice of reology treatment, velocity gradient calculation and viscosity treatment
<parameter key="RheologyTreatment" value="2" comment="Reology formulation 1:Single-phase classic, 2: Single and multi-phase" />
<parameter key="VelocityGradientType" value="2" comment="Velocity gradient formulation 1:FDA, 2:SPH" />
<parameter key="ViscoTreatment" value="3" comment="Viscosity formulation 1:Artificial, 2:Laminar+SPS, 3:Constitutive  eq." />
```

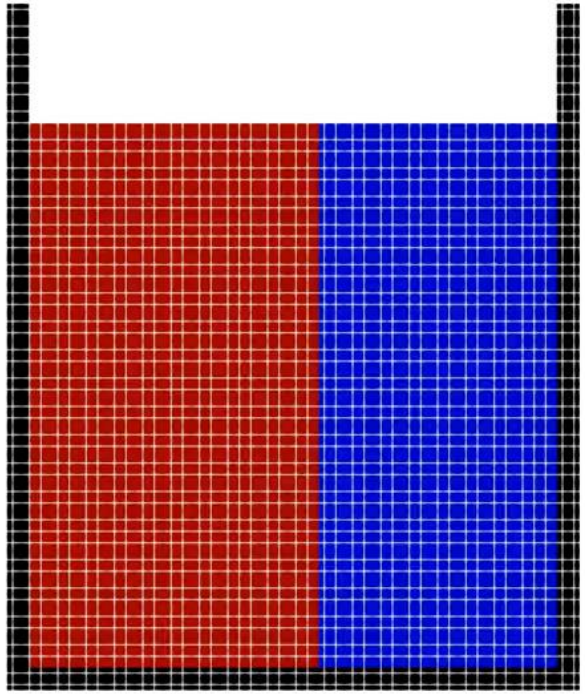
Examples

- Lock gate test case

CaseLockGateNN

Phase 1: Density=1200, Viscosity=0.10, Tau_yield=0.0040, HBP_n=1.5, HBP_m=10.0

Phase 2: Density=1000, Viscosity=0.01, Tau_yield=0.0005, HBP_n=1.0, HBP_m=1.0



Particles: 10,268
Physical time: 5 s
Runtime (GTX 2080): 600.9 s

Time: 0.00 s

Examples

- Impellers in a container

```
<special>
    <nnphases> %Defines non-newtonian phases parameters
        <phase mkfluid="0">
            <rhop value="1000" comment="Density of the phase" />
            <visco value="0.1" comment="Kinematic viscosity (or consistency index) value for phase (m2/s)" />
            <tau_yield value="0.21" comment="Specific yield stress of phase (Pa m3/kg)" />
            <tau_max value="0.21" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <Bi_multi value="20.0" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <HBP_m value="100" comment="Use 0 to reduce Newtonian liquid, order of 10 for power law and order of 100 for Bingham (sec) " />
            <HBP_n value="1" comment="Use 1 to reduce to Newtonian, <1 for shear thinning >1 for shear thickenning " />
            <phasetype value="0" comment="Non-Newtonian=0 only option in v5.0" />
        </phase>
        <phase mkfluid="1">
            <rhop value="1000" comment="Density of the phase" />
            <visco value="0.01" comment="Kinematic viscosity (or consistency index) value for phase (m2/s)" />
            <tau_yield value="0.021" comment="Specific yield stress of phase (Pa m3/kg)" />
            <tau_max value="0.021" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <Bi_multi value="20.0" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <HBP_m value="10" comment="Use 0 to reduce Newtonian liquid, order of 10 for power law and order of 100 for Bingham (sec) " />
            <HBP_n value="1" comment="Use 1 to reduce to Newtonian, <1 for shear thinning >1 for shear thickenning " />
            <phasetype value="0" comment="Non-Newtonian=0 only option in v5.0" />" />
        </phase>
        <phase mkfluid="2">
            <rhop value="1000" comment="Density of the phase" />
            <visco value="0.001" comment="Kinematic viscosity (or consistency index) value for phase (m2/s)" />
            <tau_yield value="0.0021" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <tau_max value="0.0021" comment="Maximum yield stress of phase" />
            <Bi_multi value="20.0" comment="User defined maximum specific yield stress of phase (Pa m3/kg)" />
            <HBP_m value="1" comment="Use 0 to reduce Newtonian liquid, order of 10 for power law and order of 100 for Bingham (sec) " />
            <HBP_n value="1" comment="Use 1 to reduce to Newtonian, <1 for shear thinning >1 for shear thickenning " />
            <phasetype value="0" comment="Non-Newtonian=0 only option in v5.0" />" />
        </phase>
    </nnphases>
</special>
```

Examples

- Impellers in a container

CaseImpellersNN

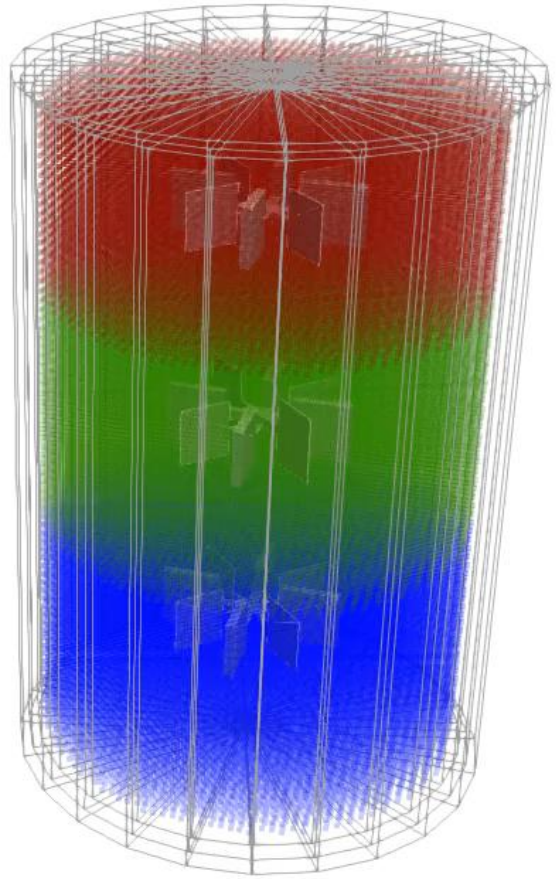
Phase 1: Density=1000, Viscosity=0.001,
Tau_yield=0.0021, HBP_n=1.0, HBP_m=1.0

Phase 2: Density=1000, Viscosity=0.010,
Tau_yield=0.0210, HBP_n=1.0, HBP_m=10.0

Phase 3: Density=1000, Viscosity=0.100,
Tau_yield=0.2100, HBP_n=1.0, HBP_m=100.0

ONLY PARTICLES WITH
VELOCITY > 0.04 m/s

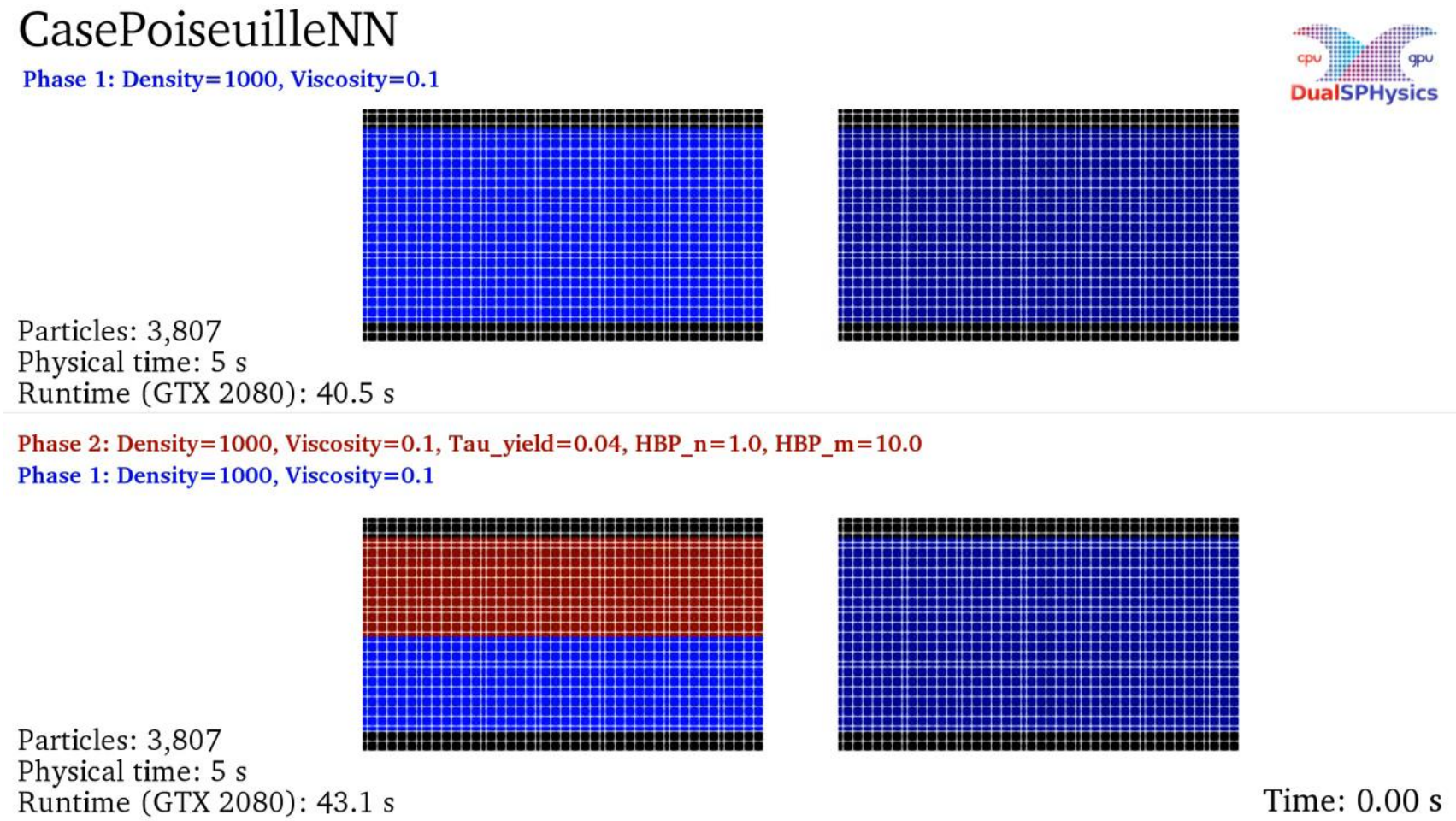
Particles: 153,810
Physical time: 6 s
Runtime (GTX 2080): 158.8 min



Time: 0.00 s

Examples

- Poiseuille flow



Work in progress

DualSPHysics v 5.xx

- Merging of the single and non-Newtonian solvers
- Sediment phase with a yield criterion (see Fourtakas and Rogers, 2016, AWR)

DualSPHysics v 6.++

- Time dependant non-Newtonian models (watch this space)
- Advanced multi-phase interface treatment (coming to your favourite SPH solver soon)
- Semi-implicit time integration (if interested in co-development, contact me)

- Many other improvements for viscous terms, stability and accuracy of multi-phase flows

Conclusions

- A general single and multi-phase solver
  - Variety of choices and formulations
- Applicable to a wide range of non-Newtonian fluids
  - From Newtonian to bi-viscosity and dilatant fluids
- CPU and GPU implementation
- Balance between efficiency
  - Usability for beginners/intermediate users
  - Tools/customisation for developers
- More developments on their way

Acknowledgements

The University of Manchester

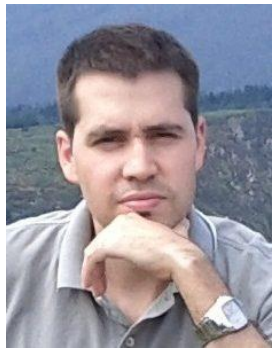
- Multi-phase branch developers

George Fourtakas
University of Manchester
georgios.fourtakas-at-manchester.ac.uk

Ben Rogers
University of Manchester

@SPH_Manchester
@GFourtakas
@benedictdrogers
@DualSPHysics

- Contributors



Jose Dominguez
University of Vigo

Renato Vacondio
University of Parma

- Industrial contributors

NATIONAL NUCLEAR LABORATORY

Brendan Perry
National Nuclear Laboratory