Towards improving predictions of non-Newtonian settling slurries with Delft3D: theoretical development and validation in 1DV

Appendixes – Jill L.J. Hanssen

May 12 2016

DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Civil Engineering and Geosciences
Department of Hydraulic Engineering

Table of Contents - Appendix

Appendix A

Rheological parameters of the three rheological models to compare with the data of A.D. Thomas (Thomas, 1999)

Model 1	
n_f	2.64
a	3.65
A_y	7.3E5
A_μ	9.3
β	0.27
$\phi_{sasi,max}$	0.6
μ_w	0.001

Model 2	
B_μ	2.64
B_y	4.75
K_y	6.7E4
K_μ	2.5
α	0.27
$\phi_{sasi,max}$	0.6
μ_w	0.001

Remark: Factors K_μ and K_y include A_{clay}, ρ_w and ρ_{sol}

Model 3	
C_y	4.75E5
p	5.61
D	17.7
$k_{yield}\,\phi_{sasi,max}$	0.9
$k_{visc}\,\phi_{sasi,max}$	0.75
$\phi_{sasi,max}$	0.6

Appendix B 1 – Rheological formulas Model 1

Mixture

$$\tau_{y,mix} = A_y \left(\frac{\phi_{cl}}{1-\phi_{sasi}} \right)^{2/(3-n_f)} \exp\{\beta\lambda\}$$

$$\mu_{mix} = \exp(\beta\lambda) \left[\mu_w + A_\mu \left(\frac{\phi_{cl}}{1-\phi_{sasi}} \right)_p^{\frac{2(a+1)}{3}} \left[\frac{1}{\dot{\gamma}} \right]^{\frac{(a+1)(3-n_f)}{3}} \right]$$

$$\lambda = \frac{1}{\left(\phi_{sasi,\max} / \phi_{sasi} \right)^{1/3} - 1}$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2}$$

$$\tau_{mixture} = \tau_{y,mix} \left(1 - \exp\{-m\dot{\gamma}\} \right) + \mu_w \dot{\gamma} + \mu_{sol-mix} \dot{\gamma}^n$$

$$0 < n < 1$$

Carrier fluid

$$\tau_{y,cf} = A_y \left(\frac{\phi_{cl}}{1-\phi_{sasi}} \right)^{2/(3-n_f)}$$

$$\mu_{cf} = \left[\mu_w + A_\mu \left(\frac{\phi_{cl}}{1-\phi_{sasi}} \right)_p^{\frac{2(a+1)}{3}} \left[\frac{1}{\dot{\gamma}} \right]^{\frac{(a+1)(3-n_f)}{3}} \right]$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2}$$

$$\tau_{carrier-fluid} = \tau_{y,cf} \left(1 - \exp\{-m\dot{\gamma}\} \right) + \mu_w \dot{\gamma} + \mu_{sol-cf} \dot{\gamma}^n$$

$$0 < n < 1$$

Appendix B 2 – Rheological formulas Model 2

Mixture

$$\tau_y = K_{\mathrm{y,mix}}\left(\frac{W}{PI}\right)^{B_y}\exp\left(\alpha\lambda\right)$$

$$\tau_{y,mix} = K_{\mathrm{y}}\left(\frac{\rho_w}{A_{clay}\rho_{solids}}\left(\frac{\phi_{cl}}{1-\phi_{solids}}\right)^{-1}\right)^{B_y}\times\exp\left(\beta\lambda\right)$$

$$\mu_{mix} = \left[\mu_w + K_\mu\left(\frac{\rho_w}{A_{clay}\rho_{solids}}\left(\frac{\phi_{cl}}{1-\phi_{solids}}\right)^{-1}\right)^{B_\mu}\right]\times\exp\left\{\beta\lambda\right\}$$

$$\lambda = \frac{1}{\left(\phi_{sasi,\max}/\phi_{sasi}\right)^{1/3}-1}$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2}$$

$$\tau_{mixture} = \tau_{y,mix}\left(1-\exp\left\{-m\dot{\gamma}\right\}\right)+\mu_{mix}\dot{\gamma}$$

Carrier fluid

$$\tau_{y,cf} = K_{\mathrm{y}}\left(\frac{\rho_w}{A_{clay}\rho_{solids}}\left(\frac{\phi_{cl}}{1-\phi_{solids}}\right)^{-1}\right)^{B_y}$$

$$\mu_{carrierfluid} = \left[\mu_w + K_\mu\left(\frac{\rho_w}{A_{clay}\rho_{solids}}\left(\frac{\phi_{cl}}{1-\phi_{clay}}\right)^{-1}\right)^{B_\mu}\right]$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2}$$

$$\tau_{carrier-fluid} = \tau_{y,cf}\left(1-\exp\left\{-m\dot{\gamma}\right\}\right)+\mu_{cf}\dot{\gamma}$$

Appendix B 3 – Rheological formulas Model 3

Mixture

$$x = \frac{\phi_{sand}}{\phi_{solids}} \qquad 0 \le \mathrm{x} \le 1$$

$$\tau_y = A_y \left[(1-x)\left(\frac{\phi_{solids}}{1 - x\phi_{solids}} \right) \right]^n \times \left[1 - \left(\frac{x\phi_{solids}}{k_{yield}\phi_{solids\max}} \right) \right]^{-2.5}$$

$$\mu_{slurry} = \left[1 - \frac{x\dfrac{\phi_{solids}}{1-\phi_{solids}}}{1+\dfrac{\phi_{solids}}{1-\phi_{solids}}} \frac{1}{k_{visc}\phi_{sand\max}} \right]^{-2.5} \times \exp\left\{ B(1-x)\frac{\phi_{solids}}{1-\phi_{solids}} \right\}$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2}$$

$$\tau_{mixture} = \tau_{y,mix}\left(1 - \exp\{-m\dot{\gamma}\}\right) + \mu_{mix}\dot{\gamma}$$

Carrier fluid

$$x = \frac{\phi_{sand}}{\phi_{solids}} \qquad 0 \le \mathrm{x} \le 1$$

$$\mu_{carrierfluid} = [1-0]^{-2.5} \times \exp\left\{ B(1-0)\frac{\phi_{clay}}{1-\phi_{clay}} \right\}$$

$$\dot{\gamma} = \sqrt{\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2}$$

$$\tau_{carrier-fluid} = \tau_{y,cf}\left(1 - \exp\{-m\dot{\gamma}\}\right) + \mu_{cf}\dot{\gamma}$$

Appendix C 1 – Code rheological Model 1

Mixture

```
      subroutine mudvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c*****************************************************************************
c
c       D e l t a r e s   S o f t w a r e   C e n t e r
c
c           Module: Subroutine MUDVIC - HWCK
c         Function: Determine equivalent viscosity (VICMUD) derived from
c                   rheological model for water-clay-silt-sand flow.
c
c       Method used:
c
c       date       : 07-01-2016
c       Programmer : R.E. Uittenbogaard
c*****************************************************************************
      include    'pardef.inc'
      include    'dimens.inc'
      include    'physco.inc'
      include    'timefr.inc'
      include    'turcoe.inc'
      include    'hydarr.inc'
      include    'sedarr.inc'
      include    'turarr.inc'
      include    'wrkarr.inc'
c
      dimension  u1    (*),dudz  (0:kmaxd),rho   (*),
     +           phiss (1:kmaxd), solfrc(1:kmaxd), cl2   (1:kmaxd),
     +           conlin(1:kmaxd), ssinfy(1:kmaxd), ssinfv(1:kmaxd),
     +           actyie(1:kmaxd), actvic(1:kmaxd),
     +           xmuwat(0:kmaxd), xmusol(0:kmaxd)
c
c ********** input parameters *******************
c
      phissm = 0.6
      ayield = 729884.
      frcdim = 2.6426
      bety   = 0.2752
      watmu  = viscou*rhom
      Avic   = 9318.6/10.
      powa   = 3.65
      betv   = 0.2752
c++    actcl =
c
```

```
     shrco  = 5.E3
c
c
c
c*************** N E W - S L U R R Y ***********************************
c
c
c **************Vol. Frac SLURRY *****************
c
     do k = 1,kmax
       phiss(k)  = phisa(k) + phisi(k)
       if (phiss(k).eq.0.0) then
          phiss(k) = 10.e-99
       else
          phiss(k) = phisa(k) + phisi(k)
       endif
c
       solfrc(k) = phicl(k)/(1.-phiss(k))
c
c********* Linear concentration sand+silt
c
       cl1    = 1./3.
       cl2(k) = ((phissm/phiss(k))**cl1)-1.
       conlin(k) = 1./cl2(k)
       ssinfy(k)  = exp(bety*conlin(k))
     enddo
c
c ************* YIELD STRESS SLURRY ***************
c
c *********values at interface********
c
     powyie = 2./(3.-frcdim)
c
     do k=0,kmax
       kk = max(1,k)
       ku = min(k+1,kmax)
       ts = thick(kk)+thick(ku)
       solfri = (solfrc(kk)*thick(ku)+solfrc(ku)*thick(kk))/ts
       ssinyi = (ssinfy(kk)*thick(ku)+ssinfy(ku)*thick(kk))/ts
       tyield(k) = ayield*ssinyi*solfri**powyie
c
     enddo
c
c ******* VISCOSITY PARAMETERS SLURRY ********
c
     powvic = 2.*(powa+1.)/3.
     do k = 1,kmax
       ssinfv(k) = exp(betv*conlin(k))
```

```
c
      enddo
c
c ******* Equivalent viscosity derived from shear stress for mixture:
c
c *********values at interface********
c
cccc 0< Powshr < 1
      powshr = ((powa+1.)*(3.-frcdim))/3.
c
           rhoa    = 0.0
           do k    = 1,kmax
              rhoa = rhoa+rho(k)*thick(k)
           enddo
c
      do k=0,kmax
         kk = max(1,k)
         ku = min(k+1,kmax)
         ts = thick(kk)+thick(ku)
         solfri = (solfrc(kk)*thick(ku)+solfrc(ku)*thick(kk))/ts
         ssinvi = (ssinfv(kk)*thick(ku)+ssinfv(ku)*thick(kk))/ts
c
c
         shear      = abs(dudz(k))
         if (shear.eq.0.0) then
            vicmud(k) = 1.E6
         else
            shear = abs(dudz(k))
c
         xmuwat(k) = ssinvi*watmu*shear
         xmusol(k) = ssinvi*avic*(solfri**powvic)*(shear**(1-powshr))
c
         xmu   (k) = xmuwat(k) + xmusol(k)
c
         taubh (k) = tyield(k)*(1-exp(-shrco*shear))+xmu(k)
c
         vicmud(k) = taubh(k)/shear
c
         vicmud(k) = vicmud(k)/rhoa
         endif
      enddo
c
c
      return
      end
```

Carrier fluid

```
      subroutine cflvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c***********************************************************************
c
c       D e l t a r e s   S o f t w a r e   C e n t e r
c
c           Module: Subroutine CFLVIC
c         Function: Determine equivalent viscosity CARRIER FLUID derived from
c                   rheological model for water-clay-silt-sand flow.
c
c       Method used:
c
c       date       : 07-01-2016
c       Programmer : R.E. Uittenbogaard
c***********************************************************************
      include     'pardef.inc'
      include     'dimens.inc'
      include     'physco.inc'
      include     'timefr.inc'
      include     'turcoe.inc'
      include     'hydarr.inc'
      include     'sedarr.inc'
      include     'turarr.inc'
      include     'wrkarr.inc'
c
      dimension  u1    (*),dudz  (0:kmaxd),rho   (*),
     +           cffrc (1:kmaxd), cfclin(1:kmaxd), siinfy(1:kmaxd),
     +           siinfv(1:kmaxd), cl3   (1:kmaxd),
     +           cfmuwa(0:kmaxd), cfmuso(0:kmaxd)
c
c*************** N E W - C A R I E R  F L U I D ************************************
c
c ******** input parameters **********
c
      phisim = 0.6
      ayield = 729884.
      frcdim = 2.6426
      bety   = 0.2752
      watmu  = viscou*rhom
      Avic   = 9318.6/10.
      powa   = 3.65
      betv   = 0.2752
c
      shrco  = 5.E3
c
c************* Type carrier fluid clay *************
```

```
c
     if (carflu.eq.'cfclay') then
c
c ************* Volume Fraction *****************
        do k = 1,kmax
          cffrc(k) = phicl(k)
        enddo
c
c********* Linear concentration ******************************
c ******** only if carrier fluid is clay+silt, silt has influence
c
        do k = 1,kmax
          siinfy(k)=exp(bety*0.0)
        enddo
c
c ************* YIELD STRESS carrier fluid clay****************
c
     powyie = 2./(3.-frcdim)
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
          cffrci = (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinyi = (siinfy(kk)*thick(ku)+siinfy(ku)*thick(kk))/ts
          cfty(k)   = ayield*siinyi*cffrci**powyie
c
        enddo
C
c ******* VISCOSITY PARAMETERS CARRIER FLUID clay********
c
     powvic = 2.*(powa+1.)/3.
c
        do k=1,kmax
          siinfv(k)  = exp(betv*0.0)
c
        enddo
c
c ******* Equivalent viscosity derived from shear stress for carrier fluid clay:
c
     powshr = ((powa+1.)*(3.-frcdim))/3.
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
          cffrci = (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinvi = (siinfv(kk)*thick(ku)+siinfv(ku)*thick(kk))/ts
```

```
        rhocfi = (rhocf(kk)*thick(ku)+rhocf(ku)*thick(kk))/ts
c
        shear     = abs(dudz(k))
        if (shear.eq.0.0) then
          cfvic(k) = 1E4
        else
          shear = abs(dudz(k))
c
        cfmuwa(k) = siinvi*watmu*shear
        cfmuso(k) = siinvi*avic*(cffrci**powvic)*(shear**(1-powshr))
c
        cfmu  (k) = cfmuwa(k) + cfmuso(k)
c
        cftau(k) = cfty(k)*(1-exp(-shrco*shear))+cfmu(k)
c
        cfvic(k) = cftau(k)/shear
c
        cfvic(k) = cfvic(k)/rhocfi
c
        endif
       enddo
c
c
c ************************************************************
c****************** Type carrier fluid clay+silt **************
c
      elseif (carflu.eq.'cfclsi') then
c
c ************* Volume Fraction *****************
c
       do k = 1,kmax
        cffrc(k) = phicl(k)+phisi(k)
       enddo
c
c********* Linear concentration *******************************
c ******** only if carrier fluid is clay+silt, silt has influence
c
       cl1   = 1./3.
        if (phisi(k).eq.0.0) then
         phisi(k) = 10.e-99
        else
         phisi(k) = phisi(k)
        endif
c
       do k  =1,kmax
        cl3(k) = ((phisim/phisi(k))**cl1)-1.
        cfclin(k) = 1./cl3(k)
        siinfy(k)=exp(bety*cfclin(k))
```

```
        enddo
c
c ************* YIELD STRESS carrier fluid clay+silt ****************
c
      powyie = 2./(3.-frcdim)
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
          cffrci= (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinyi= (siinfy(kk)*thick(ku)+siinfy(ku)*thick(kk))/ts
          cfty(k)   = siinyi*ayield*cffrci**powyie
c
        enddo
C
c ******* VISCOSITY PARAMETERS CARRIER FLUID clay+silt********
c
      powvic = 2.*(powa+1.)/3.
c
        do k=1,kmax
          siinfv(k)  = exp(betv*cfclin(k))
c
        enddo
c
c ******* Equivalent viscosity derived from shear stress for carrier fluid clay+silt:
c
      powshr = ((powa+1.)*(3.-frcdim))/3.
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
          cffrci= (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinvi= (siinfv(kk)*thick(ku)+siinfv(ku)*thick(kk))/ts
          rhocfi = (rhocf(kk)*thick(ku)+rhocf(ku)*thick(kk))/ts
c
          shear     = abs(dudz(k))
          if (shear.eq.0.0) then
             cfvic(k)= 1E4
          else
             shear = abs(dudz(k))
c
          cfmuwa(k) = siinvi*watmu*shear
          cfmuso(k) = siinvi*avic*(cffrci**powvic)*(shear**(1-powshr))
c
          cfmu  (k) = cfmuwa(k) + cfmuso(k)
c
```

```
          cftau(k) = cfty(k)*(1-exp(-shrco*shear))+cfmu(k)
c
          cfvic(k) = cftau(k)/shear
c
          cfvic(k) = cfvic(k)/rhocfi
c
          endif
        enddo
      endif
c
      return
      end
```

Appendix C 2 – Code rheological Model 2

Mixture

```
      subroutine mudvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c************************************************************************
c
c       D e l t a r e s   S o f t w a r e   C e n t e r
c
c          Module: Subroutine MUDVIC
c        Function: Determine equivalent viscosity (VICMUD) derived from
c                  rheological model for water-clay-silt-sand flow.
c
c       Method used:
c
c       date       : 07-01-2016
c       Programmer : R.E. Uittenbogaard
c************************************************************************
      include     'pardef.inc'
      include     'dimens.inc'
      include     'physco.inc'
      include     'timefr.inc'
      include     'turcoe.inc'
      include     'hydarr.inc'
      include     'sedarr.inc'
      include     'turarr.inc'
      include     'wrkarr.inc'
c
      dimension  u1    (*),dudz  (0:kmaxd),rho   (*),
     +           phiss (1:kmaxd), solfrc(1:kmaxd), cl2   (1:kmaxd),
     +           conlin(1:kmaxd), ssinfy(1:kmaxd), ssinfv(1:kmaxd),
     +           actyie(1:kmaxd), actvic(1:kmaxd),
     +           xmuwat(0:kmaxd), xmusol(0:kmaxd)
c
c ********** input parameters *******************
c
      phissm = 0.6
      ayield = 67191.
      powyie = 4.7544
      bety   = 0.2752
      watmu  = viscou*rhom
      Avic   = 2523.8/10.
      powvic = 2.6406
      betv   = 0.2752
c++     actcl =
```

```
c
      shrco  = 5000.
c
c
c
c*************** N E W - S L U R R Y ***********************************
c
c
c *************Vol. Frac SLURRY *****************
c
      do k = 1,kmax
        phiss(k)  = phisa(k) + phisi(k)
        if (phiss(k).eq.0.0) then
           phiss(k) = 10.e-99
        else
           phiss(k) = phisa(k) + phisi(k)
        endif
c
ccc phisol(k) from unesco
        solfrc(k) = phicl(k)/(1.-phisol(k))
c
c********* Linear concentration sand+silt
c
        cl1    = 1./3.
        cl2(k) = ((phissm/phiss(k))**cl1)-1.
        conlin(k) = 1./cl2(k)
        ssinfy(k)  = exp(bety*conlin(k))
      enddo
c
c ************* YIELD STRESS SLURRY ***************
c
c *********values at interface********
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        solfri = (solfrc(kk)*thick(ku)+solfrc(ku)*thick(kk))/ts
        ssinyi = (ssinfy(kk)*thick(ku)+ssinfy(ku)*thick(kk))/ts
        tyield(k) = ayield*ssinyi*solfri**powyie
c
c++        powyie    = -1.*powyie
c++        Actyie(k) = (rhow(k)/(actcl*rhosol(k)))**powyie
c++        Actyii    =(Actyie(kk)*thick(ku)+Actyie(ku)*thick(kk))/ts
c++        Tyield(k) = ssinyi(k)*ayield*Actyii(k)*solfri(k)**powyie
c
      enddo
c
c ******* VISCOSITY PARAMETERS SLURRY ********
```

```
c
      do k = 1,kmax
         ssinfv(k) = exp(betv*conlin(k))
c++            powvic = -1.*powvic
c++            actvic(k) = (rhow(k)/(actcl*rhosol(k)))**powvic
c
      enddo
c
c ******* Equivalent viscosity derived from shear stress for mixture:
c
c *********values at interface********
c
           rhoa    = 0.0
           do k    = 1,kmax
              rhoa = rhoa+rho(k)*thick(k)
           enddo
c
      do k=0,kmax
         kk = max(1,k)
         ku = min(k+1,kmax)
         ts = thick(kk)+thick(ku)
         solfri = (solfrc(kk)*thick(ku)+solfrc(ku)*thick(kk))/ts
         ssinvi = (ssinfv(kk)*thick(ku)+ssinfv(ku)*thick(kk))/ts
c++            actvii =(Actvic(kk)*thick(ku)+Actvic(ku)*thick(kk))/ts
c
c
         shear     = abs(dudz(k))
         if (shear.eq.0.0) then
             vicmud(k) = 1E4
         else
             shear = abs(dudz(k))
c
         xmuwat(k) = ssinvi*watmu
         xmusol(k) = ssinvi*avic*(solfri**powvic)
c++          xmusol(k) = ssinvi*avic*actvii*(solfri**powvic)
c
         xmu   (k) = xmuwat(k) + xmusol(k)
c
         taubh (k) = tyield(k)*(1-exp(-shrco*shear))+xmu(k)*shear
c
         vicmud(k) = taubh(k)/shear
c
         vicmud(k) = vicmud(k)/rhoa
         endif
      enddo
c
c
      return
```

```
end
```

Carrier fluid

```
      subroutine cflvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c**********************************************************************
c
c       D e l t a r e s   S o f t w a r e   C e n t e r
c
c          Module: Subroutine CFLVIC
c         Function: Determine equivalent viscosity CARRIER FLUID derived from
c                rheological model for water-clay-silt-sand flow.
c
c       Method used:
c
c       date       : 07-01-2016
c       Programmer : R.E. Uittenbogaard
c**********************************************************************
      include    'pardef.inc'
      include    'dimens.inc'
      include    'physco.inc'
      include    'timefr.inc'
      include    'turcoe.inc'
      include    'hydarr.inc'
      include    'sedarr.inc'
      include    'turarr.inc'
      include    'wrkarr.inc'
c
      dimension   u1    (*),dudz  (0:kmaxd),rho   (*),
     +            cffrc (1:kmaxd), cfclin(1:kmaxd), siinfy(1:kmaxd),
     +            siinfv(1:kmaxd), cl3   (1:kmaxd), cfacty(1:kmaxd),
     +            cfactv(1:kmaxd), cfmuwa(0:kmaxd), cfmuso(0:kmaxd)
c
c**************** N E W - C A R I E R  F L U I D ************************************
c
c ******** input parameters **********
c
      phisim = 0.6
      ayield = 67191.
      powyie = 4.7544
      bety   = 0.2752
      watmu  = viscou*rhom
      Avic   = 2523.8/10.
      powvic = 2.6406
      betv   = 0.2752
c
      shrco  = 5000.
c
```

```
c************** Type carrier fluid clay **************
c
      if (carflu.eq.'cfclay') then
c
c ************* Volume Fraction *****************
c
        do k = 1,kmax
          cffrc(k) = phicl(k)/(1.-phicl(k))
        enddo
c
c********* Linear concentration ******************************
c ******** only if carrier fluid is clay+silt, silt has influence
c
        do k = 1,kmax
          siinfy(k)=exp(bety*0.0)
        enddo
c
c ************* YIELD STRESS carrier fluid clay****************
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
          cffrci = (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinyi = (siinfy(kk)*thick(ku)+siinfy(ku)*thick(kk))/ts
          cfty(k)   = ayield*siinyi*cffrci**powyie
c
c++           cfActy(k) = (rhow(k)/(actcl*rhosol(lclay)))**powyie
c++           cfAcyi =(cfacty(kk)*thick(ku)+cfacty(ku)*thick(kk))/ts
c++           powyie   = -1.*powyie
c++           cfTy(k)  = siinyi*ayield*cfAcyi*cffrci**powyie
c
        enddo
C
c ******* VISCOSITY PARAMETERS CARRIER FLUID clay*********
c
        do k=1,kmax
          siinfv(k)  = exp(betv*0.0)
c++           powvic    = -1.*powvic
c++           cfactv(k) = (rhow(k)/(actcl*rhosol(lclay)))**powvic
c
        enddo
c
c ******* Equivalent viscosity derived from shear stress for carrier fluid clay:
c
        do k=0,kmax
          kk = max(1,k)
          ku = min(k+1,kmax)
```

```
          ts = thick(kk)+thick(ku)
          cffrci = (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
          siinvi = (siinfv(kk)*thick(ku)+siinfv(ku)*thick(kk))/ts
          rhocfi = (rhocf(kk)*thick(ku)+rhocf(ku)*thick(kk))/ts
c++          cfacvi =(cfactv(kk)*thick(ku)+cfactv(ku)*thick(kk))/ts
c
          shear     = abs(dudz(k))
          if (shear.eq.0.0) then
             cfvic(k) = 1.E4
          else
             shear = abs(dudz(k))
c
          cfmuwa(k) = siinvi*watmu*shear
          cfmuso(k) = siinvi*avic*(cffrci**powvic)*shear
c++        cfmuso(k) = siinvi*avic*cfacvi*(cffrci**powvic)*shear
c
          cfmu  (k) = cfmuwa(k) + cfmuso(k)
c
          cftau(k) = cfty(k)*(1-exp(-shrco*shear))+cfmu(k)
c
          cfvic(k) = cftau(k)/shear
c
          cfvic(k) = cfvic(k)/rhocfi
c
          endif
        enddo
c
c ***********************************************************
c****************** Type carrier fluid clay+silt **************
c
      elseif (carflu.eq.'cfclsi') then
c
c ************* Volume Fraction *****************
c
        do k = 1,kmax
          cffrc(k) = phicl(k)/(1.-phicl(k)-phisi(k))
        enddo
c
c********* Linear concentration ******************************
c ******** only if carrier fluid is clay+silt, silt has influence
c
        cl1   = 1./3.
          if (phisi(k).eq.0.0) then
            phisi(k) = 10.e-99
          else
            phisi(k) = phisi(k)
          endif
c
```

```
      do k  =1,kmax
        cl3(k) = ((phisim/phisi(k))**cl1)-1.
        cfclin(k) = 1./cl3(k)
        siinfy(k)=exp(bety*cfclin(k))
      enddo
c
c ************* YIELD STRESS carrier fluid clay+silt ****************
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        cffrci= (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
        siinyi= (siinfy(kk)*thick(ku)+siinfy(ku)*thick(kk))/ts
        cfty(k)   = siinyi*ayield*cffrci**powyie
c
c++          cfActy(k)= (rhow(k)/(actcl*(rhosol(lclay)*phicl(k)+
c++    *              +rhosol(lsilt)*phisi(k))))**powyie
c++          cfAcyi= (cfActy(kk)*thick(ku)+cfActy(ku)*thick(kk))/ts
c++          powyie   = -1.*powyie
c++          cfTy(k)  = siinyi*ayield*cfAcyi*cffrci**powyie
c
      enddo
C
c ******* VISCOSITY PARAMETERS CARRIER FLUID clay+silt********
c
      do k=1,kmax
        siinfv(k)  = exp(betv*cfclin(k))
c++          powvic    = -1.*powvic
c++          cfactv(k) = (rhow(k)/(actcl*(rhosol(lclay)*phicl(k)+
c++    *             +rhosol(lsilt)*phisi(k))))**powvic
c
      enddo
c
c ******* Equivalent viscosity derived from shear stress for carrier fluid clay+silt:
c
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        cffrci= (cffrc(kk)*thick(ku)+cffrc(ku)*thick(kk))/ts
        siinvi= (siinfv(kk)*thick(ku)+siinfv(ku)*thick(kk))/ts
        rhocfi = (rhocf(kk)*thick(ku)+rhocf(ku)*thick(kk))/ts
c++         cfacvi=(cfactv(kk)*thick(ku)+cfactv(ku)*thick(kk))/ts
c
        shear     = abs(dudz(k))
        if (shear.eq.0.0) then
           cfvic(k)= 1.E4
        else
```

```
                shear = abs(dudz(k))
c
            cfmuwa(k) = siinvi*watmu*shear
            cfmuso(k) = siinvi*avic*(cffrci**powvic)*shear
c++            cfmuso(k) = siinvi*avic*cfacvi*(cffrci**powvic)*shear
c
            cfmu  (k) = cfmuwa(k) + cfmuso(k)
c
            cftau(k) = cfty(k)*(1-exp(-shrco*shear))+cfmu(k)
c
            cfvic(k) = cftau(k)/shear
c
            cfvic(k) = cfvic(k)/rhocfi
c
            endif
         enddo
      endif
c
      return
      end
```

Appendix C 3 – Code rheological Model 3

Mixture

```
      subroutine mudvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c******************************************************************************
c
c     Deltares Software Center
c
c       Module: Subroutine MUDVIC
c      Function: Determine equivalent viscosity (VICMUD) derived from
c                rheological model for water-clay-silt-sand flow.
c
c     Method used:
c
c     date       : 07-01-2016
c     Programmer : R.E. Uittenbogaard
c******************************************************************************
      include    'pardef.inc'
      include    'dimens.inc'
      include    'physco.inc'
      include    'timefr.inc'
      include    'turcoe.inc'
      include    'hydarr.inc'
      include    'sedarr.inc'
      include    'turarr.inc'
      include    'wrkarr.inc'
c
      dimension   u1    (*),dudz  (0:kmaxd),rho   (*),
     +           safrc(1:kmaxd)
c
c ********** input parameters *******************
c
      phissm = 0.6
      ayield = 7.45E5
      powyie = 5.61
      yieldk = 0.9/0.6
c       watmu  = viscou*rhom
      Bvic   = 17.7
      visck  = 0.75/0.6
c
      shrco  = 5000.
c
c
c
```

```
c**************** N E W - S L U R R Y ************************************
c
c
c **************Vol. Frac SLURRY ******************
c
ccc phisol(k) from unesco
        do k = 1,kmax
           safrc(k)  = phisa(k)/phisol(k)
        enddo
c
c
c ************* YIELD STRESS SLURRY *****************
c
c *********values at interface********
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        safri    = (safrc(kk)*thick(ku)+safrc(ku)*thick(kk))/ts
        phisoi   = (phisol(kk)*thick(ku)+phisol(ku)*thick(kk))/ts
        ty1      = ayield*((1-safri)*(phisoi/(1-safri*phisoi)))**powyie
        tyield(k)= ty1*((1.-((safri*phisoi)/(yieldk*phissm))))**(-2.5)
c
      enddo
c
c ******* Equivalent viscosity derived from shear stress for mixture:
c
c *********values at interface********
c
          rhoa    = 0.0
          do k    = 1,kmax
             rhoa = rhoa+rho(k)*thick(k)
          enddo
c
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        safri = (safrc(kk)*thick(ku)+safrc(ku)*thick(kk))/ts
        phisoi= (phisol(kk)*thick(ku)+phisol(ku)*thick(kk))/ts
c
c
        shear     = abs(dudz(k))
        if (shear.eq.0.0) then
           vicmud(k) = 1E3
        else
           shear = abs(dudz(k))
c
```

```
        xmu1      = ((safri*phisoi)/(1.-phisoi))/
     *            (1.+((phisoi)/(1.-phisoi)))
        xmu2      = 1.-xmu1*(1./(visck*phissm))
        xmu3      = (xmu2)**(-2.5)
        xmu(k)    = (1./10.)*(xmu3)*exp((bvic*(1.-safri)*
     *           (phisoi)/(1.-phisoi)))*shear

c
        taubh (k) = tyield(k)*(1-exp(-shrco*shear))+xmu(k)
c
        vicmud(k) = taubh(k)/shear
c
        vicmud(k) = vicmud(k)/rhoa
        endif
      enddo
c
c
      return
      end
```

Carrier fluid

```
      subroutine cflvic (zeta  ,dp    ,u1    ,dudz  ,rho )
c**********************************************************************
c
c     D e l t a r e s   S o f t w a r e   C e n t e r
c
c         Module: Subroutine CFLVIC
c       Function: Determine equivalent viscosity (CFLVIC) derived from
c                 rheological model for CARRIER FLUID.
c
c     Method used:
c
c     date       : 07-01-2016
c     Programmer : R.E. Uittenbogaard
c**********************************************************************
      include     'pardef.inc'
      include     'dimens.inc'
      include     'physco.inc'
      include     'timefr.inc'
      include     'turcoe.inc'
      include     'hydarr.inc'
      include     'sedarr.inc'
      include     'turarr.inc'
      include     'wrkarr.inc'
c
      dimension   u1    (*),dudz  (0:kmaxd),rho   (*),
     +            cfsafr(1:kmaxd)
c
c ********** input parameters ********************
c
      phissm = 0.6
      ayield = 7.45E5
      powyie = 5.61
      yieldk = 0.9/0.6
c      watmu  = viscou*rhom
      bvic   = 17.7
      visck  = 0.75/0.6
c
      shrco  = 5000.
c
c
c **************Vol. Frac SAND Carrier fluid *******************
c
ccc phisol(k) from unesco
        do k = 1,kmax
```

```
          cfsafr(k)  =0
        enddo
c
c
c ************* YIELD STRESS SLURRY ***************
c
c *********values at interface********
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        cfsafi = (cfsafr(kk)*thick(ku)+cfsafr(ku)*thick(kk))/ts
        phicli = (phicl(kk)*thick(ku)+phicl(ku)*thick(kk))/ts
        cfty1  = ayield*((1-cfsafi)*(phicli/(1-cfsafi*phicli)))**powyie
        cfty(k)= cfty1*((1.-((cfsafi*phicli)/(yieldk*phissm))))**(-2.5)
c
      enddo
c
c ******* Equivalent viscosity derived from shear stress for mixture:
c
c *********values at interface********
c
          rhoa    = 0.0
          do k    = 1,kmax
            rhoa = rhoa+rho(k)*thick(k)
          enddo
c
      do k=0,kmax
        kk = max(1,k)
        ku = min(k+1,kmax)
        ts = thick(kk)+thick(ku)
        cfsafi= (cfsafr(kk)*thick(ku)+cfsafr(ku)*thick(kk))/ts
        rhocfi = (rhocf(kk)*thick(ku)+rhocf(ku)*thick(kk))/ts
        phicli= (phicl(kk)*thick(ku)+phicl(ku)*thick(kk))/ts
c
c
        shear     = abs(dudz(k))
        if (shear.eq.0.0) then
           cfvic(k) = 1E4
        else
           shear = abs(dudz(k))
c
        xmu1      = ((cfsafi*phicli)/(1.-phicli))/
     *             (1.+((phicli)/(1.-phicli)))
        xmu2      = 1.-xmu1*(1./(visck*phissm))
        xmu3      = (xmu2)**(-2.5)
        cfmu(k)   = (1./10.)*(xmu3)*exp((bvic*(1.-cfsafi)*
     *            (phicli)/(1.-phicli)))*shear
```

```
c
c
       cftau (k) = cfty(k)*(1-exp(-shrco*shear))+cfmu(k)
c
       cfvic (k) = cftau(k)/shear
c
       cfvic (k) = cfvic(k)/rhocfi
       endif
     enddo
c
c
     return
     end
```

Appendix D – Segregation model

$$w_{s,eff} = (1-\phi_s)\left(1-\frac{\phi_{sa}}{\phi_{sa,\max}}\right)^2 w_{s,0s} = (1-\phi_s)\left(1-\frac{\phi_{sa}}{\phi_{sa,\max}}\right)^2 \frac{\alpha}{18}\frac{g(\rho_s-\rho_{cl})d^2}{\mu_{apparent-cf}}$$

Appendix E – Code segregation model

```
      subroutine fallve(dudz)
c**********************************************************************
c
c       D e l t a r e s  S o f t w a r e   C e n t e r
c
c           Module: Subroutine FALLVE
c         Function: Relation between sediment concentration
c                   and vertical fall velocity. Model for
c                   hindered settling.
c                   Fall velocity at layer interfaces.
c
c      Method used:
c             Date: 11-01-2016
c       Programmer: R.E. Uittenbogaard, Jill Hanssen
c
c**********************************************************************
      include     'pardef.inc'
      include     'dimens.inc'
      include     'physco.inc'
      include     'turcoe.inc'
c
      include     'conarr.inc'
      include     'hydarr.inc'
      include     'sedarr.inc'
      include     'turarr.inc'
c
      dimension    dudz   (0:kmaxd), dudzcr(0:kmaxd)       ,
     +             phisoi (1:kmaxd), wssa  (1:kmaxd,lsand),
     +             wssi   (1:kmaxd,lsilt)
c
      pi = 4.0*atan(1.0)
c
c     fall velocity dependent on carrier fluid type and sediment type
c
c********************************************************************
c
c     Carrier fluid water : stokes for sand/ silt, different for clay
c
      if (carflu.eq.'cfwat') then
c
      do l=1,lsed
c
        if (sedtyp(l).eq.'sand'.or.sedtyp(l).eq.'SILT') then
c
```

```
          do k  = 1,kmax
            ku = min(k+1,kmax)
            ts = thick(k)+thick(ku)
            rhoint=(thick(k)*rhow(ku)+thick(ku)*rhow(k))/ts
            rseint=(thick(k)*rsed0(ku,l)+thick(ku)*rsed0(k,l))/ts
            s = rhosol(l)/rhoint
            if (seddia(l).lt.1e-4) then
              ws(k,l) = (s - 1.0)*ag*seddia(l)**2/(18.*viscou)
            elseif (seddia(l).lt.1e-3) then
             ws(k,l) = 10.*viscou/seddia(l)*
     *                 (sqrt(1.+(s-1.)*ag*seddia(l)**3/
     *                     (100.*viscou**2))-1.)
            else
             ws(k,l) = 1.1*sqrt((s-1.)*ag*seddia(l))
            endif
c Settling velocity (- sign):
            ws(k,l) =-ws(k,l)
          enddo
c
        else if (sedtyp(l).eq.'clay') then
c
          do k     = 1,kmax
            ku     = min(k+1,kmax)
            ts     = thick(k)+thick(ku)
            salint = (thick(k)*r1(ku,lsal)+thick(ku)*r1(k,lsal))/ts
            rseint = (thick(k)*rsed0(ku,l)+thick(ku)*rsed0(k,l))/ts
            if (salint.lt.salmax(l)) then
              a   = 1.+wsm(l)/ws0(l)
              b   = a-2.
             ws(k,l) = 0.5*ws0(l)*(a-b*cos(pi*salint/salmax(l)))
            else
              ws(k,l) = wsm(l)
            endif
c
c Hindered settling Richardson & Zaki (1954):
c
            volsed     = rseint/rhosol(l)
            solid      = 1.0
            hinset     = max(0.0,(1.-volsed/solid))
            ws(k,l)    = ws(k,l)*hinset**4.65
          enddo
        endif
      enddo
c
      endif
c       (carflu.eq.'cfwat')
c**********************************************************************
c
```

```
c     Carrier fluid clay or clay +silt
c
c phisol(k) = total solid concentration calc in unesco
c
      retco  = 2.
      sasim  = 0.6
      shrco  = 1.
      cocr   = 1.
c
c
c********** fall velocity single grain in viscous fluid - clay**********
c
      if (carflu.eq.'cfclay') then
c
        do k  = 1 ,kmax
c
            kk = max(1,k)
            ku = min(k+1,kmax)
            ts = thick(kk)+thick(ku)
c
            rhocfi=(thick(kk)*rhocf(ku)+thick(ku)*
     *                rhocf(kk))/ts
c
            wssa(k,lsand) = shrco*(rhosol(lsand)-rhocfi)*ag
     *              *seddia(lsand)**2/(18*cfvic(k)*rhocfi)
c
            wssi(k,lsilt) = shrco*(rhosol(lsilt)-rhocfi)*ag
     *              *seddia(lsilt)**2/(18*cfvic(k)*rhocfi)
c
c********** Hindered settling **********
c
            retsa      = max(0.0,(1-phisa(ku)/sasim))
            retsi      = max(0.0,(1-phisi(ku)/sasim))
            buoyan     = max(0.0,(1-phisol(ku)))
c
            ws(k,lsand)= wssa(k,lsand)*buoyan*retsa**retco
            ws(k,lsilt)= wssi(k,lsilt)*buoyan*retsi**retco
c Settling velocity (- sign):
            ws(k,lsand) =-ws(k,lsand)
            ws(k,lsilt) =-ws(k,lsilt)
        enddo
c
c *****************************************************************
c********** fall velocity single grain in viscous fluid - clay+silt **********
c
      elseif (carflu.eq.'cfclsi') then
        do k  = 1 ,kmax
          kk = max(1,kmax)
```

```
          ku = min(k+1,kmax)
          ts = thick(kk)+thick(ku)
c
          rhocfi=(thick(kk)*rhocf(ku)+thick(ku)*
    *                rhocf(kk))/ts
          wssa(k,lsand) = shrset*(rhosol(lsand)-rhocfi)*ag
    *                *seddia(lsand)**2/(18*cfvic(k)*rhocfi)
c
c********** Hindered settling **********
c
          ku = min(k+1,kmax)
          retsa      = max(0.0,(1-phisa(ku)/sasim))
          buoyan     = max(0.0,(1-phisol(ku)))
          ws(k,lsand)= wssa(k,lsand)*buoyan*retsa**retco
c
c Settling velocity (- sign):
            ws(k,lsand) =-ws(k,lsand)
c
        enddo
      endif
c
      return
      end
```

Appendix F – Simulations non-Segregating flow

Parameters of all three Models for non-segregating flow simulations.

Model 1	Sim B1.1	Sim B1.2	Sim B1.3	Sim B1.4
n_f	2.64	2.64	2.64	2.64
a	3.65	3.65	3.65	3.65
A_y	7.3E5	7.3E5	7.3E5	7.3E5
A_μ	9.3	930	930	930
β	0.27	0.27	0.27	0.27
$\varphi_{sasi,max}$	0.6	0.6	0.6	0.6
μ_w	0.001	0.001	0.001	0.001
φ_{clay}	0.16	0.16	0.16	0.12
φ_{sand}	0.04	0.04	0.04	0.16
φ_{sol}	0.20	0.20	0.20	0.28
SFR	1 : 4	1 : 4	1 : 4	4 : 3
ρ_{clay}	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860
ρ_{mixture}	1336	1336	1336	1348
m	5	5	5000	5000

Model 2	Sim B1.1	Sim B1.2	Sim B1.3	Sim B1.4
B_μ	2.64	2.64	2.64	2.64
B_y	4.75	4.75	4.75	4.75
K_y	6.7E4	6.7E4	6.7E4	6.7E4
K_μ	2.5	250	250	250
α	0.27	0.27	0.27	0.27
$\phi_{sasi,max}$	0.6	0.6	0.6	0.6
μ_w	0.001	0.001	0.001	0.001
φ_{clay}	0.16	0.16	0.16	0.12
φ_{sand}	0.04	0.04	0.04	0.16
φ_{sol}	0.20	0.20	0.20	0.28
SFR	1 : 4	1 : 4	1 : 4	4 : 3
ρ_{clay}	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860
ρ_{mixture}	1336	1336	1336	1348
m	5	5	5000	5000

Model 3	Sim B1.1	Sim B1.2	Sim B1.3	Sim B1.4
C_y	4.75E5	4.75E5	4.75E5	4.75E5
p	5.61	5.61	5.61	5.61
D	17.7	17.7	17.7	17.7
$k_{yield}\ \phi_{sasi,max}$	0.9	0.9	0.9	0.9
$k_{visc}\ \phi_{sasi,max}$	0.75	0.75	0.75	0.75
$\phi_{sasi,max}$	0.6	0.6	0.6	0.6
Multiplication μ	-	100	100	100
φ_{clay}	0.16	0.16	0.16	0.12
φ_{sand}	0.04	0.04	0.04	0.16
φ_{sol}	0.20	0.20	0.20	0.28
SFR	1 : 4	1 : 4	1 : 4	4 : 3
ρ_{clay}	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860
ρ_{mixture}	1336	1336	1336	1348
m	5	5	5000	5000

Solid content in gram per liter, volume concentration and mass concentration.

	Sand [g/l]	*Clay* [g/l]	φ_{sand} [-]	φ_{clay} [-]	φ_{solids} [-]	ξ_{sand} [-]	ξ_{clay} [-]	ξ_{solids} [-]	*SFR* [-]
B1.1	106	427	0.04	0.16	0.20	0.08	0.32	0.40	1 : 4
B1.2	106	427	0.04	0.16	0.20	0.08	0.32	0.40	1 : 4
B1.3-REF	106	427	0.04	0.16	0.20	0.08	0.32	0.40	1 : 4
B1.4	466	312	0.16	0.12	0.28	0.35	0.23	0.58	4 : 3

Modified parameters for non-segregating flow simulations. (*) this simulation is only done for Model 1.

	φ_{sand} [-]	φ_{clay} [-]	τ_{yield} [Pa]	μ_{tot} [Pa s]	m [-]	Grid [#]	Time [s]	*Sin(θ)* [-]	q [m²/s]	W_s [m/s]
B1.1	0.04	0.16	38	0.04	5	200	0.1	0.01	0.1	0
B1.2	0.04	0.16	38	4.3	5	200	0.1	0.01	0.1	0
B1.3-REF	0.04	0.16	38	4.3	5000	200	0.1	0.01	0.1	0
B1.3.1*	0.04	0.16	38	4.3	5E6	200	0.1	0.01	0.1	0
B1.4	0.16	0.12	32.6	3.5	5000	200	0.1	0.01	0.1	0

Appendix G – Simulations segregating flow

Horizontal bed

Parameters of Model 2 for segregating flow on a horizontal bed. Simulation A2.1 – A2.3

Model 2		A21 – A23			
B_μ	[-]	2.64	φ_{clay}	[-]	0.16
B_y	[-]	4.75	φ_{sand}	[-]	0.04
K_y	[-]	6.7E4	φ_{sol}	[-]	0.20
K_μ	[-]	2.5; 25; 250	ρ_{clay}	[kg/m^3]	2670
α	[-]	0.27	ρ_{sand}	[kg/m^3]	2860
$\phi_{sasi,max}$	[-]	0.6	$\rho_{mixture}$	[kg/m^3]	1336
μ_w	[Pa s]	0.001	**SFR**	[m]	1 : 4
M	[-]	5000	**u**	[m/s]	0.4
			h	[m]	0.5

Adapted parameter of Model 2 for segregating flow on a horizontal bed. Simulation A2.1 – A2.3.

	φ_{sand} [-]	φ_{clay} [-]	τ_{yield} [Pa]	μ_{tot} [Pa s]	m [-]	Grid [#]	Time [s]
A2.1	0.04	0.16	38	0.04	5000	200	0.1
A2.2	0.04	0.16	38	0.43	5000	200	0.1
A2.3	0.04	0.16	38	4.3	5000	200	0.1

Slope

Parameters of all three Models for segregating flow simulations along a slope.

Model 1	Sim B2.1	Sim B2.1.1	Sim B2.2	Sim B2.3	Sim B2.4	Sim B2.5
n_f	2.64	2.64	2.64	2.64	2.64	2.64
a	3.65	3.65	3.65	3.65	3.65	3.65
A_y	7.3E5	7.3E5	7.3E5	7.3E5	7.3E5	7.3E5
A_μ	930	930	930	930	930	930
β	0.27	0.27	0.27	0.27	0.27	0.27
$\varphi_{sasi,max}$	0.6	0.6	0.6	0.6	0.6	0.6
μ_w	0.001	0.001	0.001	0.001	0.001	0.001
φ_{clay}	0.16	0.16	0.16	0.16	0.16	0.16
φ_{sand}	0.04	0.04	0.04	0.04	0.04	0.04
φ_{sol}	0.20	0.20	0.20	0.20	0.20	0.20
SFR	1 : 4	1 : 4	1 : 4	1 : 4	1 : 4	1 : 4
ρ_{clay}	2670	2670	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860	2860	2860
$\rho_{mixture}$	1336	1336	1336	1336	1336	1336
m	5E3	5E6	5E3	5E3	5E3	5E3

Model 2	Sim B2.1	Sim B2.2	Sim B2.3	Sim B2.4	Sim B2.5
B_μ	2.64	2.64	2.64	2.64	2.64
B_y	4.75	4.75	4.75	4.75	4.75
K_y	6.7E4	6.7E4	6.7E4	6.7E4	6.7E4
K_μ	250	250	250	250	250
α	0.27	0.27	0.27	0.27	0.27
$\phi_{sasi,max}$	0.6	0.6	0.6	0.6	0.6
μ_w	0.001	0.001	0.001	0.001	0.001
φ_{clay}	0.16	0.16	0.16	0.16	0.16
φ_{sand}	0.04	0.04	0.04	0.04	0.04
φ_{sol}	0.20	0.20	0.20	0.20	0.20
SFR	1 : 4	1 : 4	1 : 4	1 : 4	1 : 4
ρ_{clay}	2670	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860	2860
$\rho_{mixture}$	1336	1336	1336	1336	1336
m	5000	5000	5000	5000	5000

Model 3	Sim B2.1	Sim B2.2	Sim 2.3	Sim B2.4	Sim B2.5
C_y	4.75E5	4.75E5	4.75E5	4.75E5	4.75E5
p	5.61	5.61	5.61	5.61	5.61
D	17.7	17.7	17.7	17.7	17.7
$k_{yield}\,\phi_{sasi,max}$	0.9	0.9	0.9	0.9	0.9
$k_{visc}\,\phi_{sasi,max}$	0.75	0.75	0.75	0.75	0.75
$\phi_{sasi,max}$	0.6	0.6	0.6	0.6	0.6
Multiplication μ	100	100	100	100	100
φ_{clay}	0.16	0.16	0.16	0.16	0.16
φ_{sand}	0.04	0.04	0.04	0.04	0.04
φ_{sol}	0.20	0.20	0.20	0.20	0.20
SFR	1 : 4	1 : 4	1 : 4	1 : 4	1 : 4
ρ_{clay}	2670	2670	2670	2670	2670
ρ_{sand}	2860	2860	2860	2860	2860
$\rho_{mixture}$	1336	1336	1336	1336	1336
m	5000	5000	5000	5000	5000

Input parameters of Sim. B2.1 - B2.5. Segregating flow along a slope. (*) this simulation is only done for Model 1.

	φ_{sand} [-]	φ_{clay} [-]	τ_{yield} [Pa]	μ_{tot} [Pa s]	m [-]	Grid [#]	Time [s]	$Sin(\theta)$ [-]	q [m^2/s]	W_s [m/s]
B2.1 REF	0.04	0.16	38	4.3	5000	200	0.1	0.1	0.1	#
B2.1.1*	0.04	0.16	38	4.3	5E6	200	0.1	0.1	0.1	#
B2.2	0.04	0.16	38	4.3	5000	600	0.1	0.1	0.1	#
B2.3	0.04	0.16	38	4.3	5000	200	0.01	0.1	0.1	#
B2.31	0.04	0.16	38	4.3	5000	200	1	0.1	0.1	#
B2.4	0.04	0.16	38	4.3	5000	200	0.1	0.025	0.1	#
B2.5	0.04	0.16	38	4.3	5000	200	0.1	0.1	0.5	#

Comparison of concentration increase of simulation B2.1, B2.4, B2.5 of Model 1 and 2

Model	Simulation	$\bar{u}$ [m/s]	Δt [s]	Concentration increase [kg/m]
1	B2.1	0.26	1538	1.50
1	B2.4 decreased slope	0.075	5333	1.10
1	B2.5 increased discharge	1.16	345	1.14
2	B2.1	0.24	1667	1.06
2	B2.4 decreased slope	0.077	5160	0.98
2	B2.5 increased discharge	0.91	440	0.58

Appendix H - Verification with experiment

Hydraulic parameters of (Pirouz et al., 2013)

Parameter		Value	unit
Diameter pipe	D	326	[mm]
Discharge	Q	13.79	[l/s]
Depth	H	61	[mm]
Width	W	254	[mm]
Flow Area	A_{wet}	0.0108	[m^2]
Surface velocity	$u_{surface}$	1.7	[m/s]
Average velocity	$u_{average}$	1.28	[m/s]
Slope	$\sin(\theta)$	0.05	[-]

Soil content and rheological parameters of (Pirouz et al., 2013).

Parameter		Value	unit
Solids concentration	φ_{sol}	0.40	[-]
Sand concentration	φ_{sa}	0.24	[-]
Clay concentration	φ_{cl}	0.16	[-]
Yield stress	τ_y	10.1	[Pa]
Surface velocity	μ	0.596	[Pa s]
Flow rate	n	0.573	[-]

Horizontal bed

Parameters of Model 1 to simulate the experiment to simulate the conditions after 6 s as if they were static.

Model 1		unit
n_f	2.5	[-]
a	1.56	[-]
A_y	2403	[-]
A_μ	3.975	[-]
β	2.7	[-]
$\phi_{sasi,max}$	0.6	[-]
μ_w	0.001	[Pa s]
m	5000	[-]
h	0.56	[m]
$\bar{u}$	1.2 – 1.3 – 1.35	[m/s]

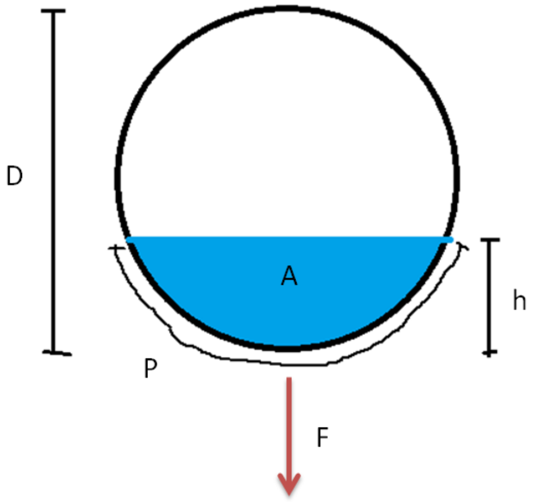
Slope

Parameters Model 1 to simulate segregating flow along a slope of (Pirouz et al., 2013)

Model 1		unit
n_f	2.5	[-]
a	1.56	[-]
A_y	2403	[-]
A_μ	3.975	[-]
β	2.7	[-]
$\phi_{sasi,max}$	0.6	[-]
d_{50}	250	[μm]
μ_w	0.001	[Pa s]
m	5000	[-]
h	0.61	[m]
$\bar{u}$	1.28	[m/s]
D-pipe	326	[mm]
Q	0.0781	[m^3/s]
Wet area	0.061	[m2]
Slope: $\sin(\theta)$	0.019	[-]

Appendix I – Calculation of parameters for experiment B. Pirouz

Slope model



$$F_{total} = \rho g \sin(\theta) * A_{wet\,pirouz}$$

$$\tau_{wall} = \frac{F_{total}}{P_{pirouz}}$$

$$R_{h-pirouz} = \frac{A_{wet\,pirouz}}{P_{pirouz}}$$

$$R_{h-1DV} = h$$

$$\tau_{wall-1DV} = \tau_{wall} * \frac{R_{h-pirouz}}{h}$$

$$\sin(\theta_{1DV}) = \frac{\tau_{wall-1DV}}{\rho g h}$$

$$\overline{u} = \frac{Q_{pirouz}}{\mathrm{A}_{wet-pirouz}}$$

$$Q_{1DV} = \overline{u} * \mathrm{A}_{wet-1DV}$$

Variable	Experiment	1DV model
D pipe	326 mm	
h	0,061 m	0,061 m
Area	0.0108 m2	0.061m2
Q	13.79E-3 m3	0.0781 m3
U- average	1.28m/s	1.28m/s
Slope	2.86 degrees	1.06 degrees
ρ	1655,64 kg/m3	1655,64 kg/m3
T_y	10.1 Pa	10.1 Pa
μ	0,596 Pa s	0,596 Pa s
T_{wall}	30 Pa	18.26
d50 sand	250μm	250μm