

Computer algebra describes flow of 3D turbulent floods via the Smagorinski large eddy closure

A. J. Roberts*

31 May 2012

Abstract

Consider the turbulent flow of a layer of fluid. The Smagorinski closure for turbulence, with its linear dependence of eddy viscosity upon the shear-rate, models turbulent dissipation. A slow manifold model of the dynamics of the fluid layer allows for large changes in layer thickness provided the changes occur over a large enough lateral length scale. The slow manifold is based on two macroscopic modes by modifying the spectrum: here artificially modify the boundary conditions on the free surface so that, as well as a mode representing conservation of fluid, a lateral shear flow with slip is a neutral critical mode. Then remove the modification to recover a model for turbulent floods.

*School of Mathematical Sciences, University of Adelaide, South Australia 5005. <mailto:anthony.roberts@adelaide.edu.au>

Contents

1	Introduction	3
1.1	Uniform acceleration	3
1.2	Overview	5
2	Preamble	6
2.1	Define order parameters	6
2.2	Stretch the coordinates with the free surface	7
2.3	Operators used elsewhere	7
3	Initialise with linear	9
4	Truncate the asymptotic expansion	11
5	Invoke the iterative loop	12
6	Update w with continuity and no flow through bed	12
7	Update the free surface evolution	13
8	Update pressure from vertical momentum and surface normal stress	13
9	Update stress, u and v from lateral momentum and surface tangential stress	14
9.1	Smagorinski large eddy stress-shear closure	14
9.2	Compute residuals of lateral momentum	16
9.3	Solve for updates to lateral velocity and rate-of-strain	17
10	Postprocessing	19
A	Trace prints from sample execution	20

1 Introduction

This approach to modelling turbulent floods develops from that of modelling non-Newtonian fluids which have a nonlinear dependence upon strain rate (Roberts 2007*a,b*). Bijvelds et al. (1999) required enhanced lateral mixing which is naturally predicted by this slow manifold approach. Roberts et al. (2008) reports preliminary results using the model derived with the computer algebra program documented herein.

Consider the three dimensional flow of a thin layer of turbulent fluid on a flat substrate. Let coordinates x and y measure distance along the substrate and coordinate z the distance normal to the substrate. Let the incompressible fluid have thickness $\eta(x, y, t)$, constant density ρ , and the nonlinear constitutive relation of the Smagorinski closure (Kim 2002, Marstop 2006, Özgökmen, Ilescu, Fischer, Srinivasan & Duan 2007, e.g.). The fluid flows with some varying velocity field $\vec{q}(x, y, z, t) = (u, v, w)$ and pressure field $p(x, y, z, t)$; these fields are the turbulent mean fields, that is, the fields averaged over realisations.

1.1 Uniform acceleration

Do not allow any lateral variations, $\partial_x = \partial_y = 0$. Then a low order slow manifold, errors $\mathcal{O}(\gamma^2 + g_x^2)$, is that in terms of the scaled vertical coordinate $\zeta = z/\eta$ the fluid fields are

$$w = 0, \quad (\text{shear flow}) \tag{1}$$

$$p = g_z(1 - \zeta)\eta, \quad (\text{hydrostatic}) \tag{2}$$

$$u = \bar{u} \frac{2(\zeta + c_u)}{1 + 2c_u} + \gamma \bar{u} \frac{(1 + c_u)[(1 + 4c_u)(c_u + \zeta) - 2(1 + 2c_u)(3c_u\zeta^2 + \zeta^3)]}{4(1 + 2c_u)^2(1 + 3c_u + 3c_u^2)}$$

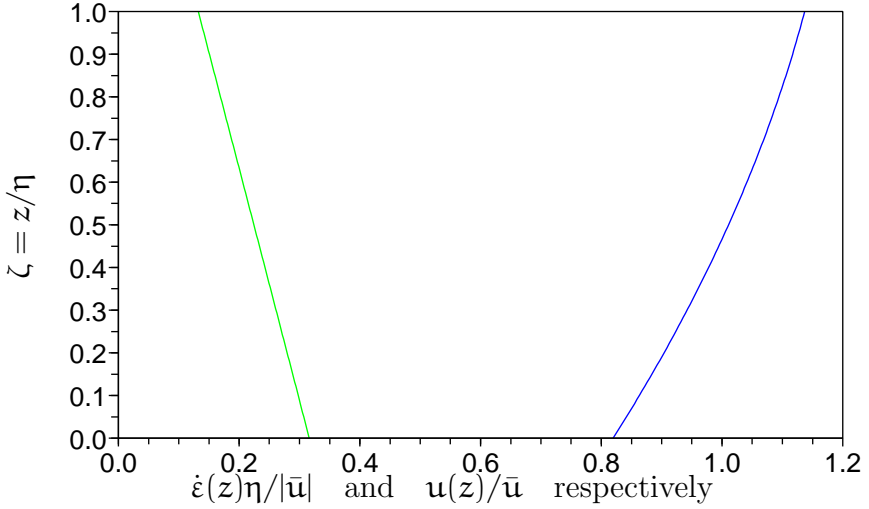


Figure 1: approximate vertical profiles at open channel flow equilibrium for which $g_x \eta / \bar{u}^2 = 0.0031$.

$$\begin{aligned}
 & + \frac{g_x \eta}{\bar{u}} \frac{[(5 + 6c_u)(c_u + \zeta) - 6(2 + 7c_u + 6c_u^2)\zeta^2 + 6(1 + 2c_u)^2\zeta^3]}{48\sqrt{2}c_t(1 + 3c_u + 3c_u^2)}, \\
 \dot{\epsilon} = & \frac{\bar{u}}{\eta} \frac{\sqrt{2}}{1 + 2c_u} + \frac{\gamma \bar{u}}{\eta} \frac{\sqrt{2}(1 + c_u)[(1 + 4c_u) - 6(1 + 2c_u)(2c_u\zeta + \zeta^2)]}{8(1 + 2c_u)^2(1 + 3c_u + 3c_u^2)} \\
 & + \frac{g_x}{\bar{u}} \frac{[(5 + 6c_u) - 12(2 + 7c_u)\zeta + 18(1 + 2c_u)^2\zeta^2]}{96c_t(1 + 3c_u + 3c_u^2)}. \quad (3)
 \end{aligned}$$

The parameter $c_u \eta$ is a ‘slip’ length on the bed, see boundary condition (15), and c_t parametrises the strength of Smagorinski’s turbulent mixing, see the eddy viscosity (12); they are determined from observations. Figure 1 displays a sample of the vertical profile of the rate of strain $\dot{\epsilon}$ and of the lateral velocity u . This equilibrium flow resolves the shear in the lateral velocity and the increase in rate of strain near the bed. Our analysis does not attempt to resolve the turbulent log layer: we assume the details of dynamical interest are those determined by the relatively large scale of the fluid depth.

The set of such profiles in the vertical, and their nonlinear interactions, form a slow manifold. The evolution on this slow manifold of the mean lateral velocities $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ is dictated by turbulent bed drag limiting gravitational forcing: ??

$$\begin{aligned} \frac{d\bar{\mathbf{u}}}{dt} = & -\frac{\sqrt{2}3c_t(1+c_u)}{(1+2c_u)(1+3c_u+3c_u^2)}\frac{\gamma\bar{\mathbf{u}}^2}{\eta} + \frac{\frac{3}{4}+3c_u+3c_u^2}{1+3c_u+3c_u^2}g_x \\ & + \mathcal{O}(\gamma^2 + g_x^2 + \partial_x) \end{aligned} \quad (4)$$

Upon putting the artificial parameter $\gamma = 1$ to recover the physical model, this evolution predicts an equilibrium channel flow at a mean velocity of

$$\bar{\mathbf{u}} = \frac{1}{2} \left[\frac{(1+2c_u)^3}{\sqrt{2}c_t(1+c_u)} \right]^{1/2} \sqrt{g_x\eta}. \quad (5)$$

For example, choosing $c_t = 0.020$ and $c_u = 1.848 \approx 13/7 \approx 11/6$ gives about the correct channel flow *and* gives about the correct eddy viscosity when compared with observations of open channel flow (Nezu 2005, e.g.).

1.2 Overview

Denote free surface thickness $\eta(\mathbf{x}, \mathbf{y}, t)$ by \mathbf{h} , mean lateral velocities $\bar{\mathbf{u}}(\mathbf{x}, \mathbf{y}, t)$ and $\bar{\mathbf{v}}(\mathbf{x}, \mathbf{y}, t)$ by \mathbf{uu} and \mathbf{vv} , and their evolution $\eta_t = \mathbf{gh}$, $\bar{\mathbf{u}}_t = \mathbf{gu}$ and $\bar{\mathbf{v}}_t = \mathbf{gv}$. The Reynolds number \mathbf{re} , and the coefficients of lateral and normal gravitational forcing are Gravity numbers \mathbf{grx} and \mathbf{grz} . Construct an asymptotic solution of the Smagorinski equations in terms of η , $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ to some order of nonlinearity in $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ and some order of lateral derivatives ∂_x and ∂_y .

Decide upon how the asymptotic expansions of the solution are to be truncated. Then iteratively update the velocity and pressure fields to solve the Smagorinski equations and boundary conditions. The iteration continues until the governing equations are satisfied; that is, their residuals are zero to the order of truncation.

2 Preamble

Improve printing by factoring with respect to these variables. It is a matter of taste and may be different depending upon what one wishes to investigate in the algebraic expressions.

```
1 on div; off allfac; on revpri;
2 factor vv,uu,qq,rqq,h,ct,gx,gz,gam,r2;
```

2.1 Define order parameters

Use the operator $h(m,n)$ to denote lateral derivatives of the fluid thickness, $\partial_x^m \partial_y^n \eta$, and similarly $uu(m,n)$ denotes lateral derivatives of the mean shear, $\partial_x^m \partial_y^n \bar{u}$. Also define readable abbreviations for η and its first spatial derivatives. Note: use d to count the number of lateral derivatives so we can easily truncate the asymptotic expansion.

```
3 operator h; operator uu; operator vv;
4 eta:=h(0,0); etax:=h(1,0)*d$ etay:=h(0,1)*d$
```

These operators must depend upon time and lateral space. Then lateral derivatives transform as $\partial_x h(m,n) = h(m+1,n)$, for example. Also, a time derivative transforms into lateral derivatives of the corresponding evolution: for example, $\partial_t h(m,n) = \partial_x^m \partial_y^n gh$.

```
5 %preamble
6 depend h,xx,yy,tt;
7 depend uu,xx,yy,tt;
8 depend vv,xx,yy,tt;
9 let { df(h(~m,~n),xx) => h(m+1,n)
10      , df(h(~m,~n),yy) => h(m,n+1)
11      , df(h(~m,~n),tt) => df(gh,xx,m,yy,n)
12      , df(uu(~m,~n),xx) => uu(m+1,n)
13      , df(uu(~m,~n),yy) => uu(m,n+1)
```

```

14      , df(uu(~m,~n),tt) => df(gu,xx,m,yy,n)
15      , df(vv(~m,~n),xx) => vv(m+1,n)
16      , df(vv(~m,~n),yy) => vv(m,n+1)
17      , df(vv(~m,~n),tt) => df(gv,xx,m,yy,n)
18      };

```

2.2 Stretch the coordinates with the free surface

Use stretched coordinates zz , xx , yy and tt to denote $Z = z/\eta(x, t)$, $X = x$, $Y = y$ and $T = t$. The free surface is then simply $Z = 1$.

```

19 depend xx,x,y,z,t;
20 depend yy,x,y,z,t;
21 depend zz,x,y,z,t;
22 depend tt,x,y,z,t;

```

Then space-time derivatives transform according to

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial X} - Z \frac{\eta_X}{\eta} \frac{\partial}{\partial Z}, \quad \frac{\partial}{\partial y} = \frac{\partial}{\partial Y} - Z \frac{\eta_Y}{\eta} \frac{\partial}{\partial Z}, \quad \frac{\partial}{\partial t} = \frac{\partial}{\partial T} - Z \frac{\eta_T}{\eta} \frac{\partial}{\partial Z}, \quad \frac{\partial}{\partial z} = \frac{1}{\eta} \frac{\partial}{\partial Z}$$

I neatly insert an automatic count of lateral x derivatives here, with d , in between ∂_x and ∂_X , and in between ∂_y and ∂_Y .

```

23 let { df(~a,x) => df(a,xx)*d-zz*etax/eta*df(a,zz)
24      , df(~a,y) => df(a,yy)*d-zz*etay/eta*df(a,zz)
25      , df(~a,t) => df(a,tt)-zz*gh/eta*df(a,zz)
26      , df(~a,z) => df(a,zz)/eta
27      };

```

2.3 Operators used elsewhere

For integrating in the vertical, use the linear operator `wsolv` as it is quicker than native integration. This is the definite integral that is zero at the bed

$\zeta = 0$.

```

28 operator wsolv; linear wsolv;
29 let {wsolv(zz^^~n,zz) => zz^(n+1)/(n+1)
30      ,wsolv(1,zz) => zz };

```

Similarly, it is quicker to use operators than to use the native integration to find the pressure. This is the vertical integral (negative) that is zero at the surface $\zeta = 1$.

```

31 operator psolv; linear psolv;
32 let {psolv(zz^^~n,zz) => (1-zz^(n+1))/(n+1)
33      ,psolv(1,zz) => (1-zz) };

```

Variable qq denotes the mean speed $\bar{q} = \sqrt{U^2 + V^2}$. Let rqq denotes its reciprocal. The following transformation rules should be correct. Either of the last two can be chosen.

```

34 depend qq,uu(0,0),vv(0,0);
35 let { qq^2=>uu(0,0)^2+vv(0,0)^2
36      , df(qq,~aa)=>(uu(0,0)*df(uu(0,0),aa)+vv(0,0)*df(vv(0,0),aa)
37      };
38 depend rqq,qq;
39 let { df(rqq,~aa)=>-rqq^2*df(qq,aa)
40      , rqq*qq=>1
41      , qq^2=>(uu(0,0)^2+vv(0,0)^2)
42      , vv(0,0)^2*rqq=>qq-uu(0,0)^2*rqq
43 %    , uu(0,0)^2*rqq=>qq-vv(0,0)^2*rqq
44      };

```

The linear operator `usolv` solves $\partial_{\zeta}^2 u' = \text{RHS}$ such that the bed boundary condition (15) is always satisfied and that the mean of the solution u' is always zero to ensure \bar{u} remains the mean later velocity.

```

45 operator usolv; linear usolv;
46 let { usolv(zz^^~n,zz) => (zz^(n+2)

```



```

47      -(cu+zz)/(n+3)/(cu+1/2) )/(n+2)/(n+1)
48      , usolv(1,zz) => (zz^2 -(cu+zz)/3/(cu+1/2) )/2 };

```

Do not need it, but the linear operator `mean` quickly computes the average of some field over the fluid thickness.

```

49 operator mean; linear mean;
50 let { mean(zz^~~n,zz) => 1/(n+1)
51      , mean(1,zz) => 1 };

```

I like to see how the iteration is proceeding. For each equation, write out the number of terms in its residual throughout iteration. Could also write out time since last length written.

```

52 procedure mylength(res);
53 begin
54 %showtime;
55 %return res;
56 return if res=0 then 0 else length(res);
57 end;

```

3 Initialise with linear

Start the iteration from the linear solution that the lateral velocity $u = \bar{u}(c_u + \zeta)/(c_u + \frac{1}{2})$ and $v = ??$ and all other fields are zero, $w = p = 0$. The parameter c_u determines the turbulent slip on the bed and is to be determined to best fit experiment and/or observations. The evolution of the ‘order parameters’ is also zero: $\bar{u}_t = gu = 0$, $\bar{v}_t = gv = 0$ and $\eta_t = gh = 0$.

```

58 let r2^2=>2; % r2=sqrt2
59 u:=uu(0,0)*(cu+zz)/(cu+1/2);
60 v:=vv(0,0)*(cu+zz)/(cu+1/2);
61 p:=grz*(1-zz)*eta;
62 w:=gh:=gu:=gv:=0;

```

Also set initial strains from the linear solution.

```

63 exx:=df(u,x);
64 eyy:=df(v,y);
65 ezz:=df(w,z);
66 exz:=(df(u,z)+df(w,x))/2;
67 exy:=(df(u,y)+df(v,x))/2;
68 eyz:=(df(v,z)+df(w,y))/2;

```

Initially approximate the magnitude $\dot{\epsilon}$ of the strain-rate tensor: the above iteration step assumes the strain rate is $(\bar{u}, \bar{v})\sqrt{2}/\eta/(1 + 2c_u)$ to leading approximation.

```

69 ros:=qq*r2/eta/(1+2*cu);

```

In the Smagorinski model ([Özgökmen, Iliescu, Fischer, Srinivasan & Duan 2007](#), e.g.) $c_t = (c_s \Delta / \eta)^2$ where arguments indicate $c_s \approx 0.2$. To match observations ([Nezu 2005](#), e.g.) of open channel flow equilibria we set $c_t = 0.020$ from which the appropriate filter scale $\Delta \approx 0.7\eta$ consistent with significant mixing across the fluid layer as seen in Figures 14–15 by [Janosi et al. \(2004\)](#).

```

70 ct:=1/50;

```

Set initial values for the stress.

```

71 txx:=2*ct*eta^2*ros*exx;
72 tyy:=2*ct*eta^2*ros*eyy;
73 tzz:=2*ct*eta^2*ros*ezz;
74 txz:=2*ct*eta^2*ros*exz;
75 txy:=2*ct*eta^2*ros*exy;
76 tyz:=2*ct*eta^2*ros*eyz;

```

The bed slip-drag law requires coefficient.

```

77 cu:=11/6;

```

4 Truncate the asymptotic expansion

There are lots of ways to truncate the asymptotic model. The small parameters available are:

- **d** counting the number of **x** derivatives of the slowly varying lateral spatial structure in any term;
- the homotopy parameter **gam** varying between $\gamma = 0$ for the artificial base problem and $\gamma = 1$ for the physical fluid equations; and
- **grx**, **gry** and **grz** being the lateral and normal components of gravity.

Note: the velocity of the flow is not small; consider the parameter \bar{u} finite.

Usually we will make lateral gravity fairly small by scaling with the magnitude of ∂_x .

Initially omit all **x** derivatives by setting $\mathbf{d} = 0$, later we scale **d** with **eps** to get the relatively simple but interesting model with errors $\mathcal{O}(\gamma^{3/2} + g_x^{3/2} + g_z^3 + \partial_x^3)$.¹ We need not make normal gravity g_z small as here, but doing so removes some messy terms.

```

78 d:=eps;
79 grz:=eps*gz;
80 grx:=eps^2*gx;
81 gry:=0;
82 gamm:=eps*gam;
83 factor eps;
```

For now truncate to relatively low order, $\mathcal{O}(\gamma^{3/2} + g_x^{3/2} + g_z^3 + \partial_x^3)$, in spatial derivatives and boundary condition artifice (can do $\mathcal{O}(\epsilon^4)$ if no spatial variations):

```

84 let { eps^3=>0 };
```

¹We need to check the algorithm at $\mathcal{O}(g_x^3)$.

5 Invoke the iterative loop

```
85 for iter:=1:9 do begin ok:=1;
86 write "ITERATION ",iter;
```

6 Update w with continuity and no flow through bed

The nondimensional PDEs for the incompressible fluid flow include the continuity equation

$$\vec{\nabla} \cdot \vec{q} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (6)$$

to be solved with no flow through the bed,

$$w = 0 \quad \text{on} \quad z = 0. \quad (7)$$

As with all field variables in this model, the quantities u , v , w and p are averaged over the ensemble of turbulent flows. Compute the residual of the continuity equation, then update the vertical velocity w by integrating from the bed, $\zeta = 0$. The variable `ok` stores whether all residuals are so far zero in this iteration.

```
87 resc:=df(u,x)+df(v,y)+df(w,z);
88 write length_resc:=mylength(resc);
89 ok:=if resc=0 then ok else 0;
90 w:=w+(dw:=-eta*wsolv(resc,zz));
```

The ζ component of the stress and rate-of-strain tensor should be, and might need to be, updated from this correction to the normal velocity; the magnitude of the rate-of-strain tensor is unaffected (to leading order). But surely we do not have to do this here as it is computed again a little later??

```

91 ezz:=ezz+df(dw,zz)/eta;
92 tzz:=tzz+2*r2*ct/(1+2*cu)*qq*df(dw,zz);

```

7 Update the free surface evolution

The kinematic condition at the free surface,

$$\frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} + v \frac{\partial \eta}{\partial y} = w \quad \text{on} \quad z = \eta, \quad (8)$$

gives the evolution of the fluid thickness h . It dominantly arises from the normal velocity.

```

93 gh:=sub(zz=1,w-u*etax-v*etay);

```

8 Update pressure from vertical momentum and surface normal stress

The nondimensional momentum equation is

$$\frac{\partial \vec{q}}{\partial t} + \vec{q} \cdot \vec{\nabla} \vec{q} = -\vec{\nabla} p + \vec{\nabla} \cdot \vec{\tau} + \vec{g}, \quad (9)$$

where $\vec{\tau}$ is the nondimensional deviatoric eddy stress tensor, and \vec{g} is the direction of gravity; when the substrate slopes, \vec{g} is not normal to the substrate. The vertical momentum equation is solved with the surface condition that the turbulent mean, normal stress to the free surface is zero, that is,

$$-p + \frac{\tau_{zz} - 2\eta_x \tau_{xz} - 2\eta_y \tau_{yz} + \eta_x^2 \tau_{xx} + 2\eta_x \eta_y \tau_{xy} + \eta_y^2 \tau_{yy}}{1 + \eta_x^2 + \eta_y^2} = 0 \quad \text{on} \quad z = \eta. \quad (10)$$

9 Update stress, u and v from lateral momentum and surface tangential stress¹⁴

Compute the residuals of the vertical momentum equation and the zero normal stress on the free surface. The recent change to the normal velocity affects the pressure update.

```
94 resw:=df(w,t)+u*df(w,x)+v*df(w,y)+w*df(w,z)
95      +df(p,z) +grz -df(txz,x)-df(txy,y)-df(tzz,z);
96 write length_resw:=mylength(resw);
97 restn:= sub(zz=1,-p*(1+etax^2+etay^2)
98      +tzz -2*etax*txz -2*etay*tyz
99      +etax^2*txx+2*etax*etay*txy+etay^2*tyy );
100 write length_restn:=mylength(restn);
101 ok:=if {resw,restn}={0,0} then ok else 0;
```

Update the pressure field p by integrating down from the free surface $\zeta = 1$; we use the linear operator psolv to solve $\partial_{\zeta} p' = -\text{RHS}$ such that $p' = 0$ at $\zeta = 1$.

```
102 p:=p+eta*psolv(resw,zz)+restn;
```

9 Update stress, u and v from lateral momentum and surface tangential stress

9.1 Smagorinski large eddy stress-shear closure

Now the strain-rate tensor

$$\dot{\epsilon}_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

```
103 exx:=df(u,x);
104 eyy:=df(v,y);
105 ezz:=df(w,z);
106 exz:=(df(u,z)+df(w,x))/2;
```

```

107 exy:=(df(u,y)+df(v,x))/2;
108 eyz:=(df(v,z)+df(w,y))/2;

```

Then the stress tensor for the fluid is $\sigma_{ij} = -p\delta_{ij} + 2\rho\nu\dot{\epsilon}_{ij}$: when the viscosity ν is constant this models a Newtonian fluid; but here the Smagorinski closure for turbulent flow is that this eddy viscosity varies linearly with strain-rate magnitude (analogous to a shear thickening non-Newtonian fluid). Define the magnitude $\text{ros} = |\dot{\epsilon}|$, the second invariant of the strain-rate tensor, where

$$|\dot{\epsilon}|^2 = \sum_{i,j} \dot{\epsilon}_{ij}^2. \quad (11)$$

```

109 rese:=exx^2+ezz^2+eyy^2+2*exz^2+2*exy^2+2*eyz^2-ros^2;
110 write length_rese:=mylength(rese);
111 ok:=if rese=0 then ok else 0;
112 ros:=ros+rese*eta*(cu+1/2)/r2*rqq;

```

Approximate the eddy viscosity at any point in the fluid as proportional to the local strain-rate magnitude,

$$\nu = c_t \eta^2 \dot{\epsilon}, \quad (12)$$

where c_t is a dimensionless constant to be chosen to fit experiments and/or observations. For whatever c_t is chosen, the deviatoric stress tensor is $\tau_{ij} = 2\nu\dot{\epsilon}_{ij} = 2c_t\eta^2\dot{\epsilon}\dot{\epsilon}_{ij}$.

```

113 txx:=2*ct*eta^2*ros*exx;
114 tyy:=2*ct*eta^2*ros*eyy;
115 tzz:=2*ct*eta^2*ros*ezz;
116 txz:=2*ct*eta^2*ros*exz;
117 txy:=2*ct*eta^2*ros*exy;
118 tyz:=2*ct*eta^2*ros*eyz;

```

9.2 Compute residuals of lateral momentum

There must be no turbulent mean, tangential stress at the free surface,

$$(1 - \eta_x^2)\tau_{xz} + \eta_x(\tau_{zz} - \tau_{xx}) - \eta_y(\tau_{xy} + \eta_x\tau_{yz}) = 0 \quad \text{on } z = \eta. \quad (13)$$

$$(1 - \eta_y^2)\tau_{yz} + \eta_y(\tau_{zz} - \tau_{yy}) - \eta_x(\tau_{xy} + \eta_y\tau_{xz}) = 0 \quad \text{on } z = \eta. \quad (14)$$

Also, put a slip law on the mean bed to provide bed drag:

$$u = c_u \eta \frac{\partial u}{\partial z} \quad \text{on } z = 0, \quad (15)$$

$$v = c_u \eta \frac{\partial v}{\partial z} \quad \text{on } z = 0, \quad (16)$$

for some constant $c_u \approx 11/6$ to match open channel flow observations.

To get centre manifold theory support for the slow manifold model of shallow water flow, modify the surface condition (13) on the tangential stress to have an artificial forcing proportional to the square of the local, free surface, velocity:

$$\begin{aligned} & [(1 - \eta_x^2)\tau_{xz} + \eta_x(\tau_{zz} - \tau_{xx}) - \eta_y(\tau_{xy} + \eta_x\tau_{yz})] \\ &= \frac{(1 - \gamma)\sqrt{2}c_t}{(1 + c_u)(1 + 2c_u)} u \sqrt{u^2 + v^2} \quad \text{on } z = \eta. \end{aligned} \quad (17)$$

$$\begin{aligned} & [(1 - \eta_y^2)\tau_{yz} + \eta_y(\tau_{zz} - \tau_{yy}) - \eta_x(\tau_{xy} + \eta_y\tau_{xz})] \\ &= \frac{(1 - \gamma)\sqrt{2}c_t}{(1 + c_u)(1 + 2c_u)} v \sqrt{u^2 + v^2} \quad \text{on } z = \eta. \end{aligned} \quad (18)$$

When we evaluate at $\gamma = 1$ this artificial right-hand side becomes zero so the artificial surface condition (17) reduces to the physical surface condition (13). However, when both the parameter $\gamma = 0$ and the lateral derivatives are negligible, $\partial_x = \partial_y = 0$, then the lateral shear $u, v \propto c_u + \zeta$ becomes a neutral mode of the dynamics.

The Euler parameter of a toy problem suggests introducing a factor $(1 - \frac{1}{6}\gamma)$ into the left-hand side of the tangential stress boundary condition (17) in

order to improve convergence in the parameter γ when evaluated at the physically relevant $\gamma = 1$. This needs further exploration. For the moment omit such a factor.

Compute the residuals of the lateral momentum equation, an artificial tangential stress on the free surface, and the bed boundary. See that when $\gamma = 0$ the free surface condition is effectively $\eta \partial_z u = u / (1 + c_u)$, leading to our neutral mode $u \propto c_u + \zeta$, namely $u = \bar{u}(x, t)(c_u + \zeta) / (c_u + \frac{1}{2})$; whereas when $\gamma = 1$ the free surface condition reduces to zero tangential stress.

```

119 resu:= df(u,t)+u*df(u,x)+v*df(u,y)+w*df(u,z)
120      +df(p,x) -grx -df(txx,x)-df(txy,y)-df(txz,z);
121 write length_resu:=mylength(resu);
122 resv:= df(v,t)+u*df(v,x)+v*df(v,y)+w*df(v,z)
123      +df(p,y) -gry -df(tyy,y)-df(txy,x)-df(tyz,z);
124 write length_resv:=mylength(resv);
125 resbu:=sub(zz=0,-u+cu*eta*df(u,z));
126 write length_resbu:=mylength(resbu);
127 resbv:=sub(zz=0,-v+cu*eta*df(v,z));
128 write length_resbv:=mylength(resbv);
129 ok:=if {resv,resu,resbu,resbv}={0,0,0,0} then ok else 0;

```

9.3 Solve for updates to lateral velocity and rate-of-strain

Update the lateral fields using an as yet unknown change in the evolution for the lateral mean velocities. The lateral fields are coupled by the nonlinear stress-strain relation of the Smagorinski turbulent flow.

```

130 u:=u+(du:=eta*(1+2*cu)*r2/(4*ct)*rq^3*usolv(
131      +(qq^2+vv(0,0)^2)*resu
132      -uu(0,0)*vv(0,0)*resv ,zz));
133 v:=v+(dv:=eta*(1+2*cu)*r2/(4*ct)*rq^3*usolv(

```

9 Update stress, u and v from lateral momentum and surface tangential stress¹⁸

```
134      +(qq^2+uu(0,0)^2)*resv
135      -uu(0,0)*vv(0,0)*resu ,zz));
136  ros:=ros+(uu(0,0)*df(du,zz)+vv(0,0)*df(dv,zz))/(r2*eta)*rqq;
```

Now use the tangential stress on the free surface to determine the evolution corrections gud and gvd . But first need to update part of the stress from changes in lateral velocity and rate-of-strain: these updates to stress and strain are indeed needed to ensure that the residual of the lateral momentum equations are satisfied. I would think that one extra iteration would do just as well, but upon testing find that it does not.

```
137  exz:=(df(u,z)+df(w,x))/2;
138  eyz:=(df(v,z)+df(w,y))/2;
139  txz:=2*ct*eta^2*ros*exz;
140  tyz:=2*ct*eta^2*ros*eyz;
```

Then compute residuals of tangential stress equations.

```
141  resttu:=(-sub(zz=1,
142      (1-0*gamm)*((1-etax^2)*txz+etax*(tzz-txx)-etay*(txy+etax*t
143      -(1-gamm)*r2*ct/(cu+1)/(2*cu+1)*u*qq) );
144  write length_resttu:=mylength(resttu);
145  resttv:=(-sub(zz=1,
146      (1-0*gamm)*((1-etay^2)*tyz+etay*(tzz-tyy)-etax*(txy+etay*t
147      -(1-gamm)*r2*ct/(cu+1)/(2*cu+1)*v*qq) );
148  write length_resttv:=mylength(resttv);
149  ok:=if {resttu,resttv}={0,0} then ok else 0;
```

Update the lateral evolution based upon these residuals.

```
150  gu:=gu-3*(1+2*cu)*(1+cu)
151      /2/eta/(3+11*cu+12*cu^2)/(1+3*cu+3*cu^2)/(3+4*cu)
152      *(((1+5*cu+8*cu^2)*uu(0,0)^2*rqq^2
153      -(9+45*cu+80*cu^2+48*cu^3))*resttu
154      +(1+5*cu+8*cu^2)*uu(0,0)*vv(0,0)*rqq^2*resttv);
155  gv:=gv-3*(1+2*cu)*(1+cu)
```

```

156      /2/eta/(3+11*cu+12*cu^2)/(1+3*cu+3*cu^2)/(3+4*cu)
157      *(((1+5*cu+8*cu^2)*vv(0,0)^2*rqq^2
158      -(9+45*cu+80*cu^2+48*cu^3))*resttv
159      +(1+5*cu+8*cu^2)*uu(0,0)*vv(0,0)*rqq^2*resttu);

```

10 Postprocessing

End the iterative loop.

```

160 showtime;
161 if ok then write iter:=100000+iter;
162 end;

```

I may use these transformations to check on the dimensionality of various expressions. But not at the moment.

```

163 dims:={ h(~m)=>nh*ll
164      , uu(~m)=>mu*ll/tt
165      , gx=>ngx*ll/tt^2 }$

```

Write out the final evolution on the slow manifold.

```

166 r2:=sqrt(2)$ %eps:=1$
167 on rounded; print_precision 4;
168 write dhdt:=length(gh);
169 write dudt:=length(gu);
170 write dvdt:=length(gv);

```

Fin.

```

171 end;

```

A Trace prints from sample execution

```
172 d := eps
173
174 grz := gz*eps
175
176           2
177 grx := gx*eps
178
179 gry := 0
180
181 gamm := gam*eps
182
183 ITERATION 1
184
185 length_resc := 7
186
187 length_resw := 76
188
189 length_restn := 3
190
191 length_rese := 51
192
193 length_resu := 92
194
195 length_resv := 87
196
197 length_resbu := 0
198
199 length_resbv := 0
200
201 length_rettu := 100
```

```
202
203 length_resttv := 101
204
205 Time: 320 ms
206
207 ITERATION 2
208
209 length_resc := 274
210
211 length_resw := 235
212
213 length_restn := 57
214
215 length_rese := 127
216
217 length_resu := 477
218
219 length_resv := 517
220
221 length_resbu := 0
222
223 length_resbv := 0
224
225 length_resttu := 74
226
227 length_resttv := 84
228
229 Time: 4370 ms
230
231 ITERATION 3
232
233 length_resc := 240
234
```

```
235 length_resw := 206
236
237 length_restn := 62
238
239 length_rese := 109
240
241 length_resu := 361
242
243 length_resv := 410
244
245 length_resbu := 0
246
247 length_resbv := 0
248
249 length_restvu := 74
250
251 length_restvu := 84
252
253 Time: 5610 ms   plus GC time: 10 ms
254
255 ITERATION 4
256
257 length_resc := 0
258
259 length_resw := 89
260
261 length_restn := 31
262
263 length_rese := 0
264
265 length_resu := 147
266
267 length_resv := 167
```

```
268
269 length_resbu := 0
270
271 length_resbv := 0
272
273 length_rettu := 0
274
275 length_rettv := 0
276
277 Time: 7290 ms   plus GC time: 20 ms
278
279 ITERATION 5
280
281 length_resc := 0
282
283 length_resw := 0
284
285 length_retn := 0
286
287 length_rese := 0
288
289 length_resu := 0
290
291 length_resv := 0
292
293 length_resbu := 0
294
295 length_resbv := 0
296
297 length_rettu := 0
298
299 length_rettv := 0
300
```

```

301 Time: 6990 ms   plus GC time: 10 ms
302
303 iter := 100005
304
305 dhdt := 4
306
307 dudt := 103
308
309 dvdt := 104

```

References

- Bijvelds, M. D. J. P., Kranenburg, C. & Stelling, G. S. (1999), ‘3d numerical simulation of turbulent shallow-water in square harbor’, *Journal of Hydraulic Engineering* **125**(1), 26–31. [http://dx.doi.org/10.1061/\(ASCE\)0733-9429\(1999\)125:1\(26\)](http://dx.doi.org/10.1061/(ASCE)0733-9429(1999)125:1(26)).
- Janosi, I. M., Jan, D., Szabo, K. G. & Tel, T. (2004), ‘Turbulent drag reduction in dam-break flows’, *Experiments in Fluids* **37**, 219–229. <http://dx.doi.org/10.1007/s00348-004-0804-4>.
- Kim, T. K. (2002), A Modified Smagorinsky Subgrid Scale Model for the Large Eddy Simulation of Turbulent Flow, PhD thesis, Univeristy of California, Davis. [<http://arXiv.org/abs/math.sc/0209377>].
- Marstop, L. (2006), Subgrid-scale modelling for large-eddy simulation including scalar mixing in rotating turbulent shear flows, PhD thesis, Royal Institute of Technology KTH Mechanics SE-100 44 Stockholm, Sweden.
- Nezu, I. (2005), ‘Open-channel flow turbulence and its research prospect in the 21st century’, *J. Hydraulic Engineering* p. 229. [http://dx.doi.org/10.1061/\(ASCE\)0733-9429\(2005\)131:4\(229\)](http://dx.doi.org/10.1061/(ASCE)0733-9429(2005)131:4(229)).
- Özgökmen et al.

- Özgökmen, T. M., Iliescu, T., Fischer, P. F., Srinivasan, A. & Duan, J. (2007), ‘Large eddy simulation of stratified mixing and two-dimensional dam-break problem in a rectangular enclosed domain’, *Ocean Modelling* **16**, 106–140. <http://dx.doi.org/10.1016/j.ocemod.2006.08.006>.
- Roberts, A. J. (2007a), Computer algebra models the inertial dynamics of a thin film flow of power law fluids and other generalised newtonian fluids, Technical report, University of Southern Queensland. <http://eprints.usq.edu.au/archive/00002010/>.
URL: <http://eprints.usq.edu.au/archive/00002010/>
- Roberts, A. J. (2007b), ‘The inertial dynamics of thin film flow of non-newtonian fluids’, *Physics Letters A* **372**, 1607–1611. <http://dx.doi.org/10.1016/j.physleta.2007.10.014>.
- Roberts, A. J., Georgiev, D. J. & Strunin, D. V. (2008), Model turbulent floods with the Smagorinski large eddy closure, Technical report, <http://arxiv.org/abs/0805.3192>.