Leray-α turbulence model implementation

Lorena Gil Calo[1], Martial Boutet[1], Anne-Claire Bennis[1], Frédéric Dias[2]

[1] Morphodynamique Continentale et Côtière, Unicaen, CNRS

[2] ENS Paris Saclay & University College Dublin

Introduction

Lagrangian-Averaged Navier-Stokes alpha model
(Holm et al., 2006 ; Hecht et al., 2008)

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{v} + u_z\frac{\partial v_z}{\partial z} + \underbrace{\left[\nabla\mathbf{u}^T\cdot(\mathbf{v}-\mathbf{u}) - \alpha^2\nabla(|\nabla\mathbf{u}|^2)/2\right]}_{\text{LANS terms}} - \mathbf{f}\times\mathbf{u} = -\frac{1}{\rho_0}\nabla p + \mathcal{D}_{\mathbf{H}}(\mathbf{v}) + \frac{\partial}{\partial z}\left(\nu\frac{\partial \mathbf{v}}{\partial z}\right),$$

$$\frac{\partial p}{\partial z} = -\rho g,$$

$$\mathbf{u} - \alpha^2\Delta\mathbf{u} = \mathbf{v},$$

$$\nabla\cdot\mathbf{u} + \frac{\partial u_z}{\partial z} = 0,$$

$$\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = \mathcal{D}(C)$$

($\mathbf{v}$, v_z): rough velocity
($\mathbf{u}$, u_z): smooth velocity

Introduction

Lagrangian-Averaged Navier-Stokes alpha model
(Holm et al., 2006 ; Hecht et al., 2008)

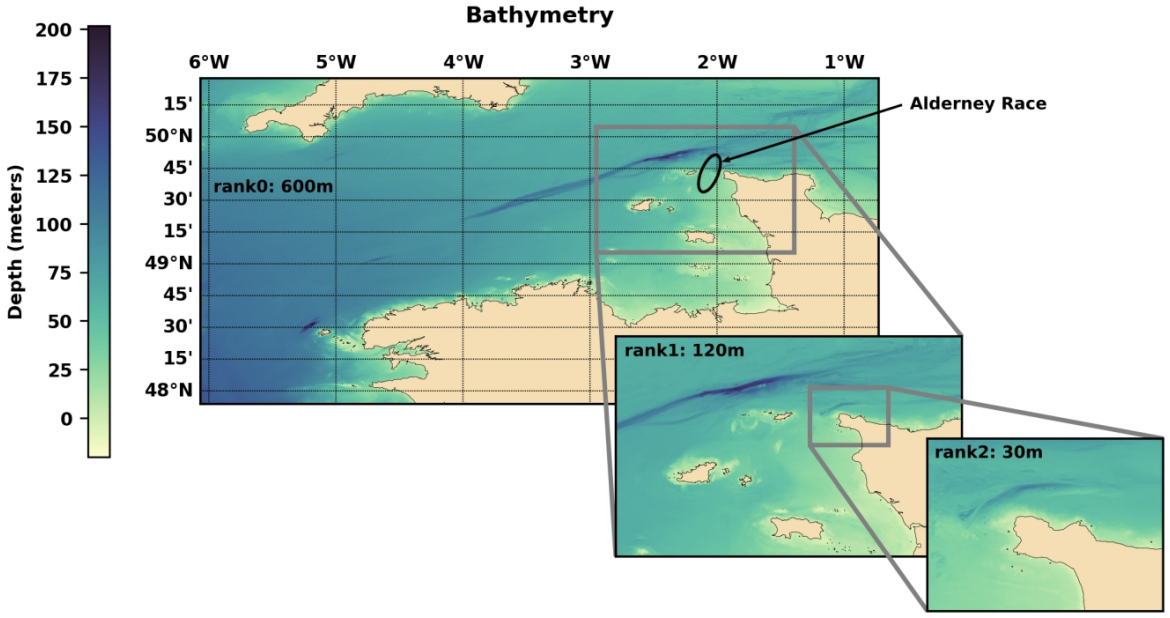
smoothing procedure (Petersen et al., 2008):

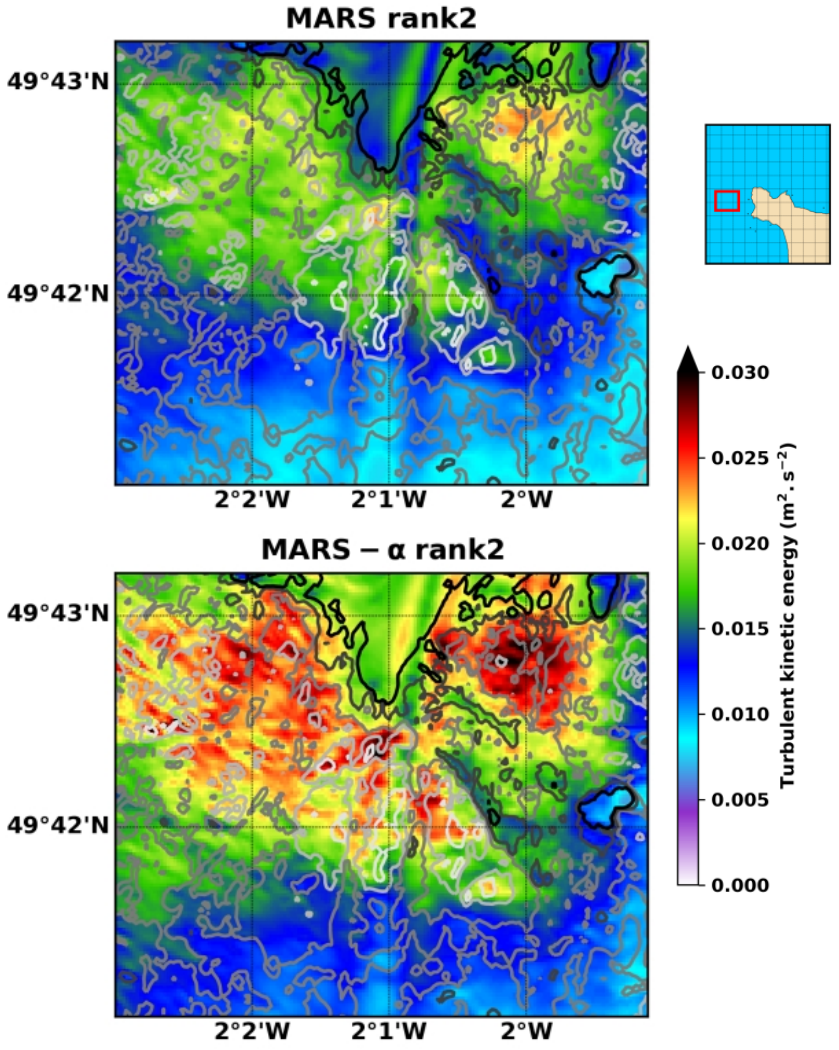
$$\mathbf{u}[i,j] = (\mathbf{v} * K_\alpha)[i,j] = \sum_{l=-\mathrm{fw}}^{\mathrm{fw}} \sum_{m=-\mathrm{fw}}^{\mathrm{fw}} w_\alpha[l,m]\mathbf{v}[i-l, j-m],$$

j+4	e^2	ed	ec	eb	e	eb	ec	ed	e^2
j+3	de	d^2	dc	db	d	db	dc	d^2	de
j+2	ce	cd	c^2	cb	c	cb	c^2	cd	ce
j+1	be	bd	bc	b^2	b	b^2	bc	bd	be
j	e	d	c	b	a	b	c	d	e
j–1	be	bd	bc	b^2	b	b^2	bc	bd	be
j–2	ce	cd	c^2	cb	c	cb	c^2	cd	ce
j–3	de	d^2	dc	db	d	db	dc	d^2	de
j–4	e^2	ed	ec	eb	e	eb	ec	ed	e^2
	i–4	i–3	i–2	i–1	i	i+1	i+2	i+3	i+4

Introduction

Implementation in MARS3D and application to Alderney Race (Bennis et al., 2021)





computational cost increase: ~ +170 %

Leray-α in CROCO

Primitive equations

$$\frac{\partial u}{\partial t} + \vec{\nabla}.\left(\vec{\mathbf{v}}u\right) - fv = -\frac{\partial \phi}{\partial x} + \mathcal{F}_u + \mathcal{D}_u$$

$$\frac{\partial v}{\partial t} + \vec{\nabla}.\left(\vec{\mathbf{v}}v\right) + fu = -\frac{\partial \phi}{\partial y} + \mathcal{F}_v + \mathcal{D}_v$$

$$\frac{\partial C}{\partial t} + \vec{\nabla}.\left(\vec{\mathbf{v}}C\right) = \mathcal{F}_C + \mathcal{D}_C$$

$$\frac{\partial \phi}{\partial z} = -\frac{\rho g}{\rho_0}$$

$$\vec{\nabla}.\vec{\mathbf{v}} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

Leray-α in CROCO

Primitive equations with Leray (1934) regularization

$$\frac{\partial u}{\partial t} + \vec{\nabla}.\left(\vec{\boldsymbol{u}}u\right) - fv = -\frac{\partial \phi}{\partial x} + \mathcal{F}_u + \mathcal{D}_u$$

$$\frac{\partial v}{\partial t} + \vec{\nabla}.\left(\vec{\boldsymbol{u}}v\right) + fu = -\frac{\partial \phi}{\partial y} + \mathcal{F}_v + \mathcal{D}_v$$

$$\frac{\partial C}{\partial t} + \vec{\nabla}.\left(\vec{\boldsymbol{u}}C\right) = \mathcal{F}_C + \mathcal{D}_C$$

$$\frac{\partial \phi}{\partial z} = -\frac{\rho g}{\rho_0}$$

$$\vec{\nabla}.\vec{\mathbf{v}} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

$$\mathbf{u}[i,j] = (\mathbf{v} * K_\alpha)[i,j] = \sum_{l=-\mathrm{fw}}^{\mathrm{fw}} \sum_{m=-\mathrm{fw}}^{\mathrm{fw}} w_\alpha[l,m]\mathbf{v}[i-l, j-m],$$

Leray-α in CROCO

filter weights computation

```
1       subroutine filter_matrix (b,c,d,e,fw,w,w_sum3,w_sum5,w_sum7,w_sum9)
2
3         implicit none
4
5         real, intent (in) :: b, c, d, e
6         integer, intent (in) :: fw
7
8         real, intent (out), dimension (-4:4,-4:4) :: w
9         real, intent (out) :: w_sum9,w_sum7,w_sum5,w_sum3
10
11        w_sum9 = 4*(b**2+c**2+d**2+e**2+b+c+d+e)+8*(e*d+e*c+c*d+e*b+b*d+b*c)+1
12        w_sum7 = 4*(b**2+c**2+d**2+b+c+d)+8*(c*d+b*d+b*c)+1
13        w_sum5 = 4*(b**2+c**2+b+c)+8*b*c+1
14        w_sum3 = 4*(b**2+b)+1
15
16        ! 9x9 filter matrix :
17
18        w(0,0)=1
19
20        if (fw == 3) then
21
22                w(-1,1) = b**2
23                w(-1,-1) = b**2
24                w(1,1) = b**2
25                w(1,-1) = b**2
26
27                w(0,1) = b
28                w(0,-1) = b
29                w(1,0) = b
30                w(-1,0) = b
31
32        else if (fw == 5) then
33
34                w(-1,1) = b**2
35                w(-1,-1) = b**2
```

Leray-α in CROCO

filtering procedure

```
24          integer :: p, q, l, m, n
25          real :: weight_sum
26
27          v_smoothed = 0
28
29          v_smoothed(Istr,:) = vel(Istr,:)
30          v_smoothed(:,Jstr) = vel(:,Jstr)
31          v_smoothed(Iend,:) = vel(Iend,:)
32          v_smoothed(:,Jend) = vel(:,Jend)
33
34          do p = Istr+1,Iend-1
35            do q = Jstr+1,Jend-1
36              if (filter_width == 3 .or. p == Istr+1 .or. p == Iend-1 .or. q == Jstr+1 .or. q == Jend-1) then
37                weight_sum = weight_sum3
38                n = 1
39              else if (filter_width == 5 .or. p == Istr+2 .or. p == Iend-2 .or. q == Jstr+2 .or. q == Jend-2) then
40                weight_sum = weight_sum5
41                n = 2
42              else if (filter_width == 7 .or. p == Istr+3 .or. p == Iend-3 .or. q == Jstr+3 .or. q == Jend-3) then
43                weight_sum = weight_sum7
44                n = 3
45              else if (filter_width == 9) then
46                weight_sum = weight_sum9
47                n = 4
48              end if
49
50              do m = -n, n
51                do l = -n, n
52                  v_smoothed(p,q) = v_smoothed(p,q) + w(l,m)*vel(p-l,q-m)
53                end do
54              end do
55              v_smoothed(p,q)=(v_smoothed(p,q))/weight_sum
56            end do
57          end do
```

Leray-α in CROCO

advection with smoothed velocity
(rhs3d.F)

```
685       call filter(Huon(Istr-1:Iend+1,Jstr-1:Jend+1,k),
686      &                          Istr-1,Iend+1,Jstr-1,Jend+1,
687      &                 w,w_sum3,w_sum5,w_sum7,w_sum9,fw,Huon_sm)
688       call filter(Hvom(Istr-1:Iend+1,Jstr-1:Jend+1,k),
689      &                          Istr-1,Iend+1,Jstr-1,Jend+1,
690      &                 w,w_sum3,w_sum5,w_sum7,w_sum9,fw,Hvom_sm)
```

Leray-α in CROCO

advection with smoothed velocity (rhs3d.F)

UP3 advection scheme

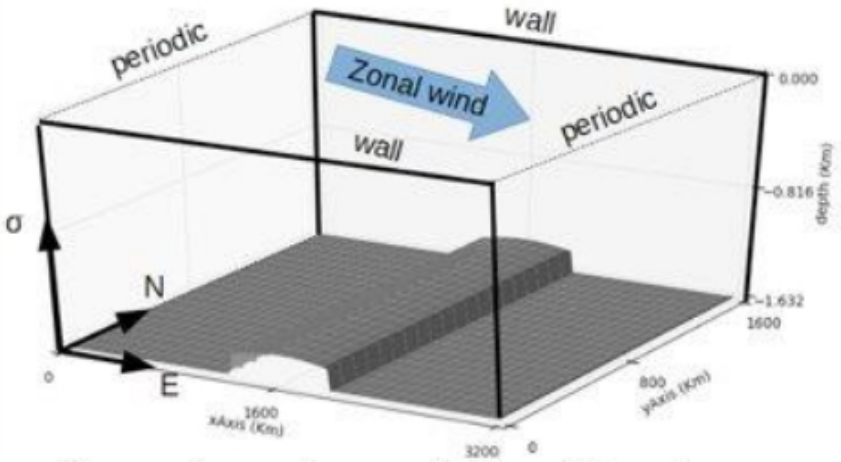
```
691           do j=Jstr,Jend
692             do i=IU_EXT_RANGE
693               uxx(i,j)=(u(i-1,j,k,nrhs)-2.*u(i,j,k,nrhs)
694       &                 +u(i+1,j,k,nrhs))  SWITCH umask(i,j)
695               Huxx(i,j)=(Huon(i-1,j,k)-2.*Huon(i,j,k)
696       &                  +Huon(i+1,j,k))   SWITCH umask(i,j)
697               Huxx_sm(i,j)=(Huon_sm(i-1,j)-2.*Huon_sm(i,j)
698       &                  +Huon_sm(i+1,j))   SWITCH umask(i,j)
699             enddo
700           enddo
```

```
745             cffX=u(i,j,k,nrhs)+u(i+1,j,k,nrhs)
746              if (cffX.gt.0.) then
747                curvX=uxx(i,j)
748              else
749                curvX=uxx(i+1,j)
750              endif
751              UFx(i,j)=0.25*(cffX+gamma*curvX)
752       &                          *(Huon_sm(i,j)+Huon_sm(i+1,j)
753       &                       -0.125*(Huxx_sm(i,j)+Huxx_sm(i+1,j)))
754
```

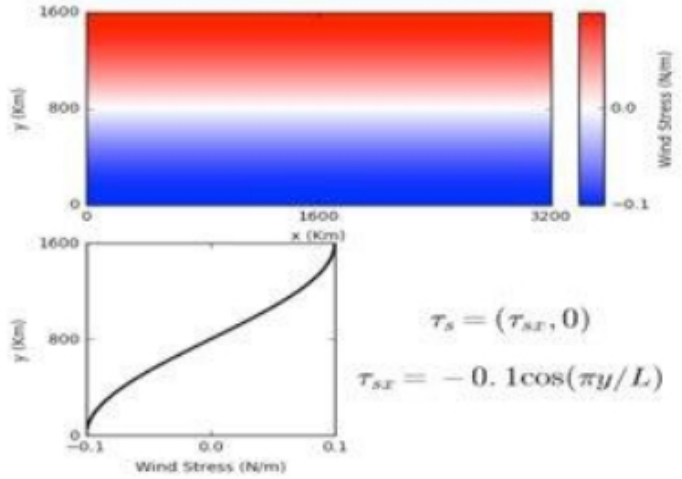
Leray-α in CROCO

Test case

Periodic channel domain inspired by (Hetch et al. 2008)



Geometry and computational domain



BC: Wind stress at the surface

Simulation parameters

Domain size

$Length: 3200\ Km$

$Width: 1600\ Km$

$Depth: 1.632\ Km$

Coarse mesh size

$\Delta x_1 = 40\ Km$

$\Delta y_1 = 40\ Km$

$\Delta z = 46\ m\ (34\ levels)$

Fine mesh size

$\Delta x_2 = 20\ Km$

$\Delta y_2 = 20\ Km$

$\Delta z = 46\ m\ (34\ levels)$

Numerical viscosity

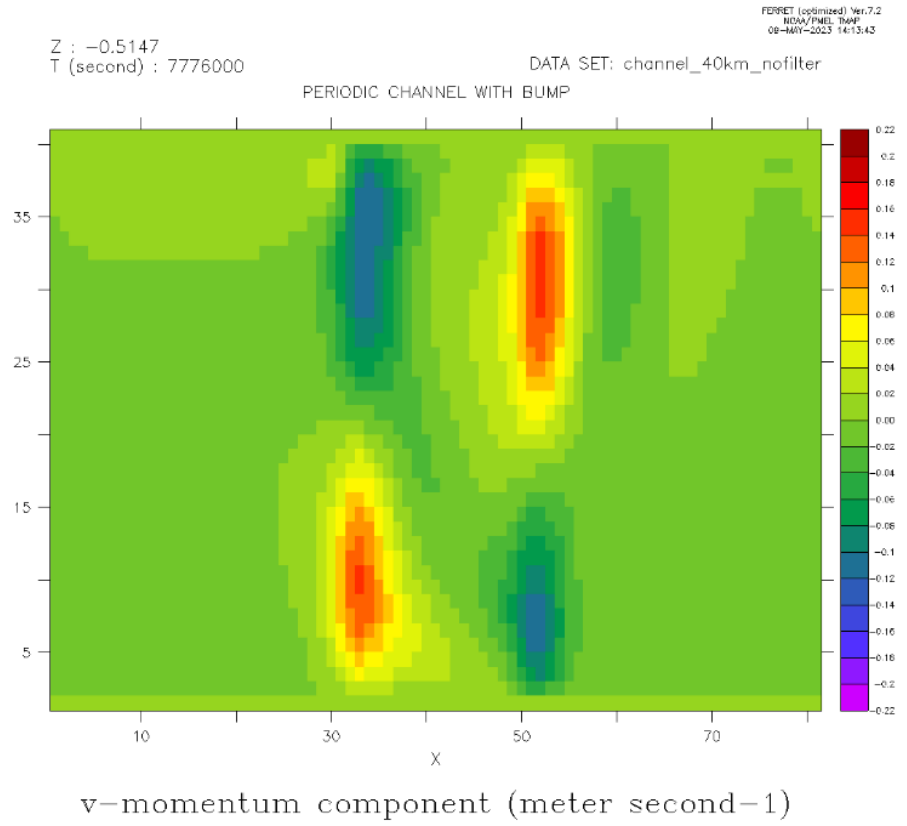
$\nu_H = 5 \times 0.01 \times (\Delta x_1)^{1.15}$

$\nu_V = 10^{-6}\ Pa \cdot s$

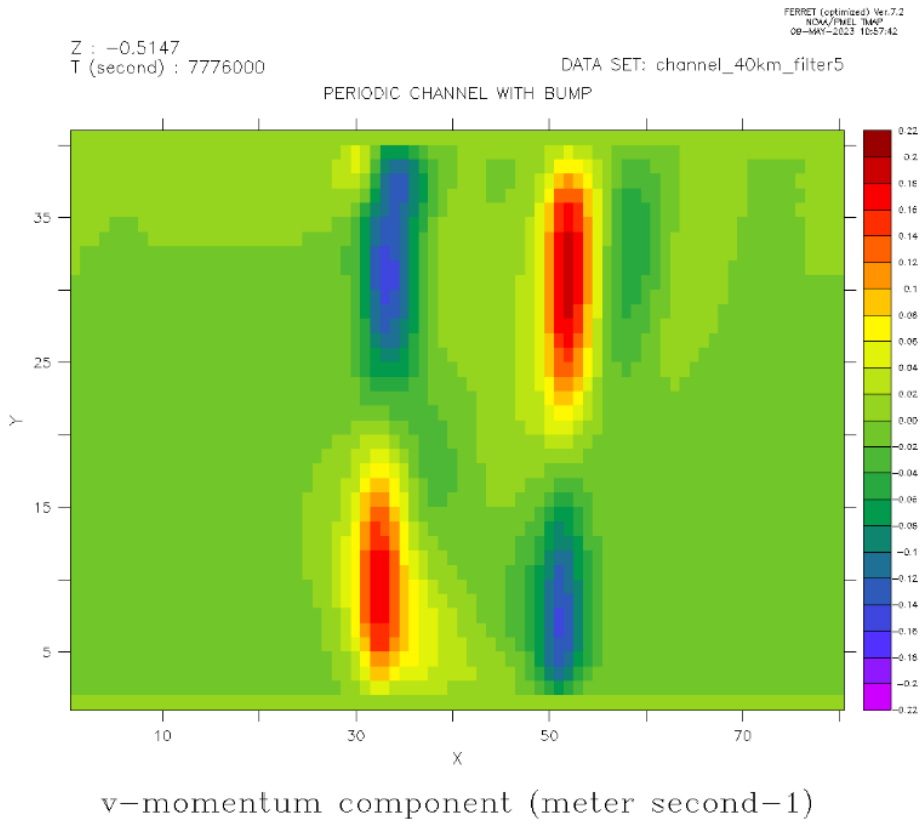
Leray-α in CROCO

Test case

meridional velocity at mid-depth



CROCO 40km

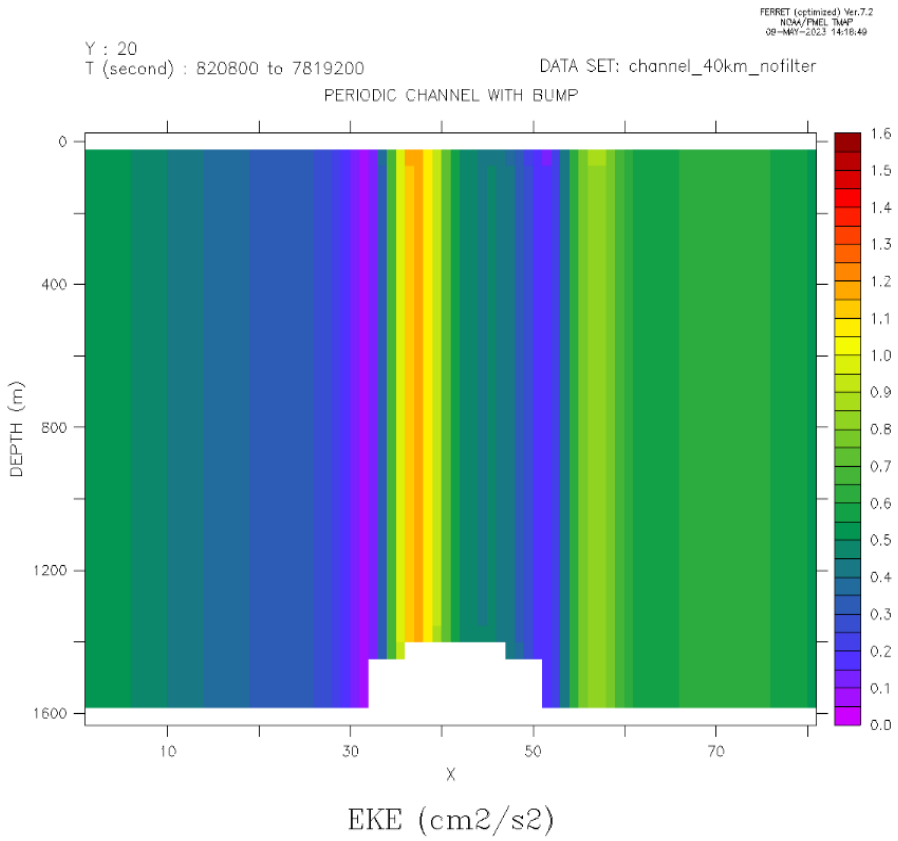


CROCO Leray-α 40km
filter size: 5

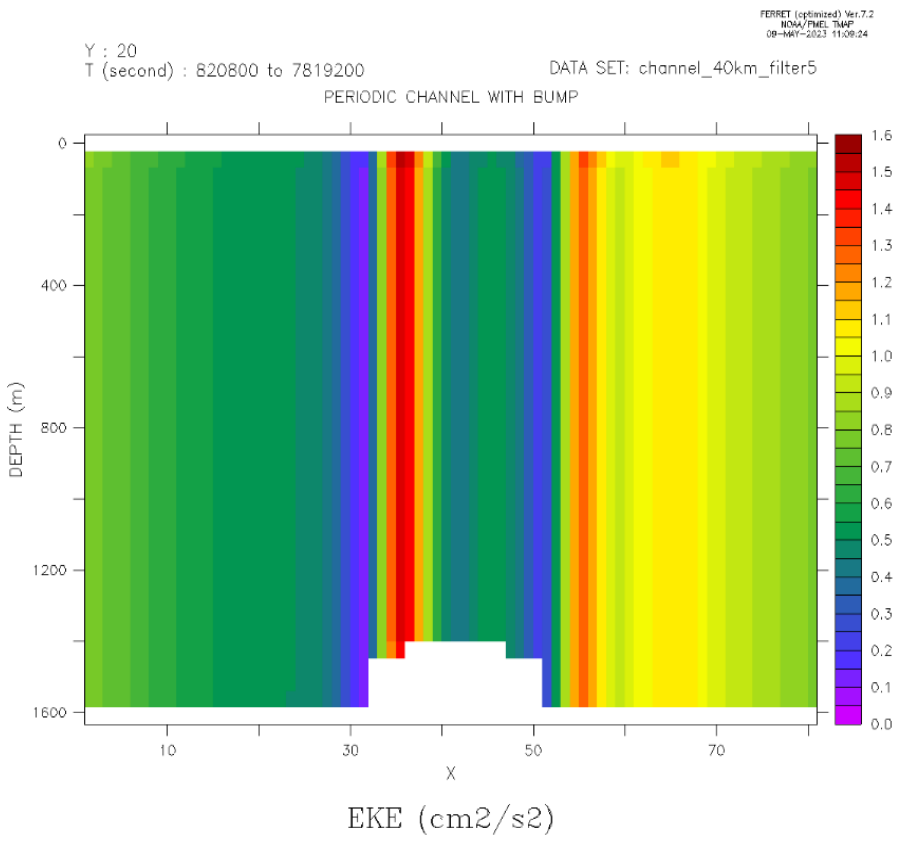
Leray-α in CROCO

Test case

eddy kinetic energy zonal vertical section



CROCO 40km

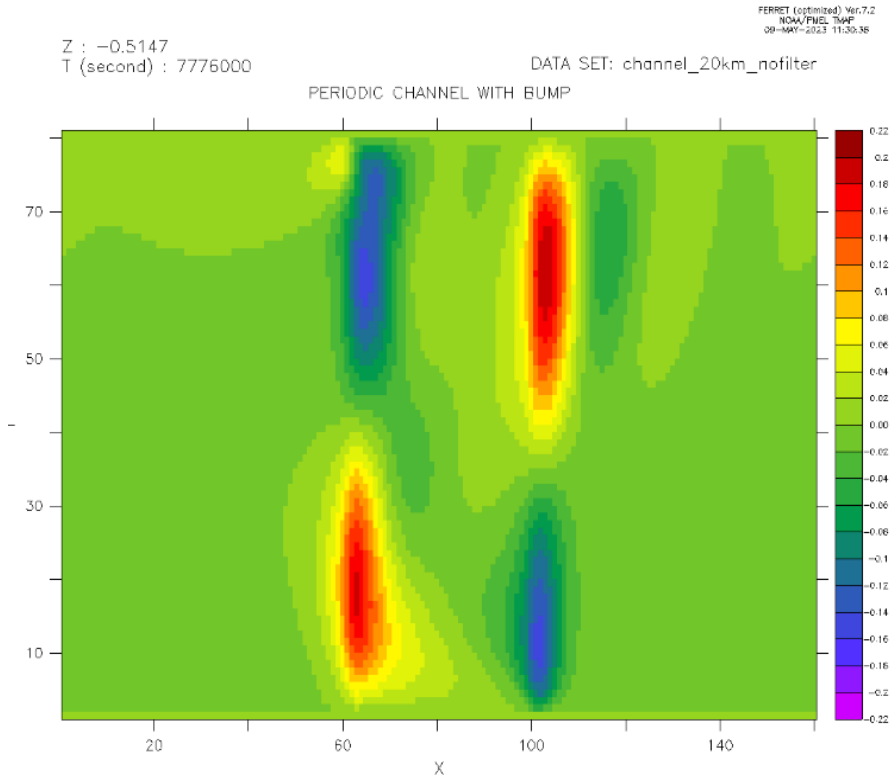


CROCO Leray-α 40km
filter size: 5

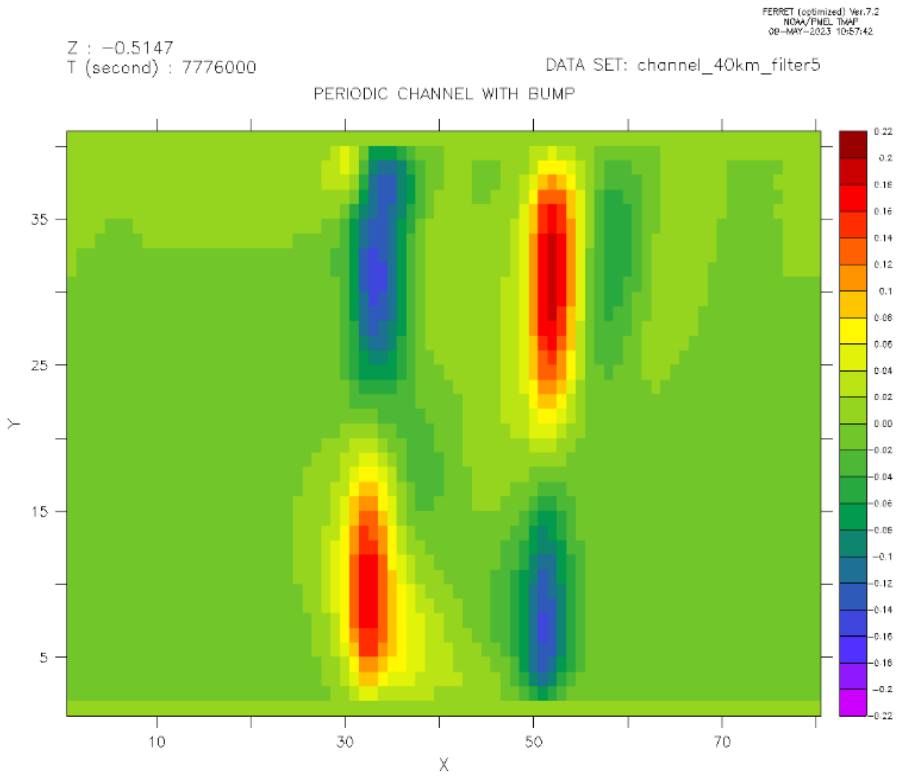
Leray-α in CROCO

Test case

meridional velocity at mid-depth



CROCO 20km

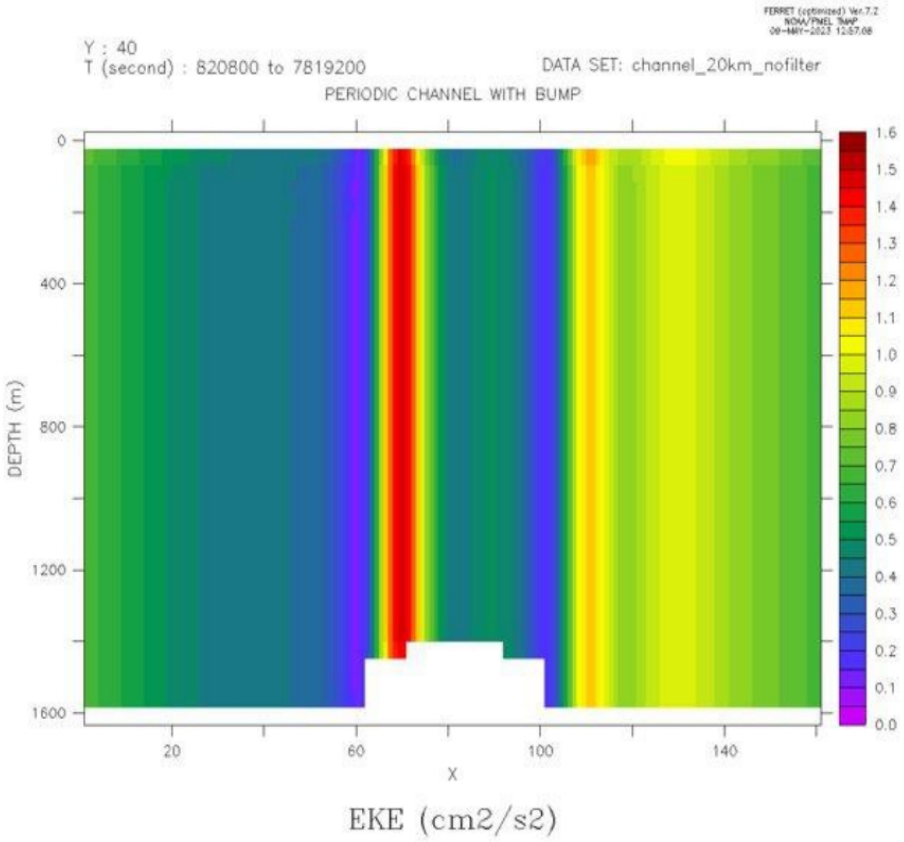


CROCO Leray-α 40km
filter size: 5

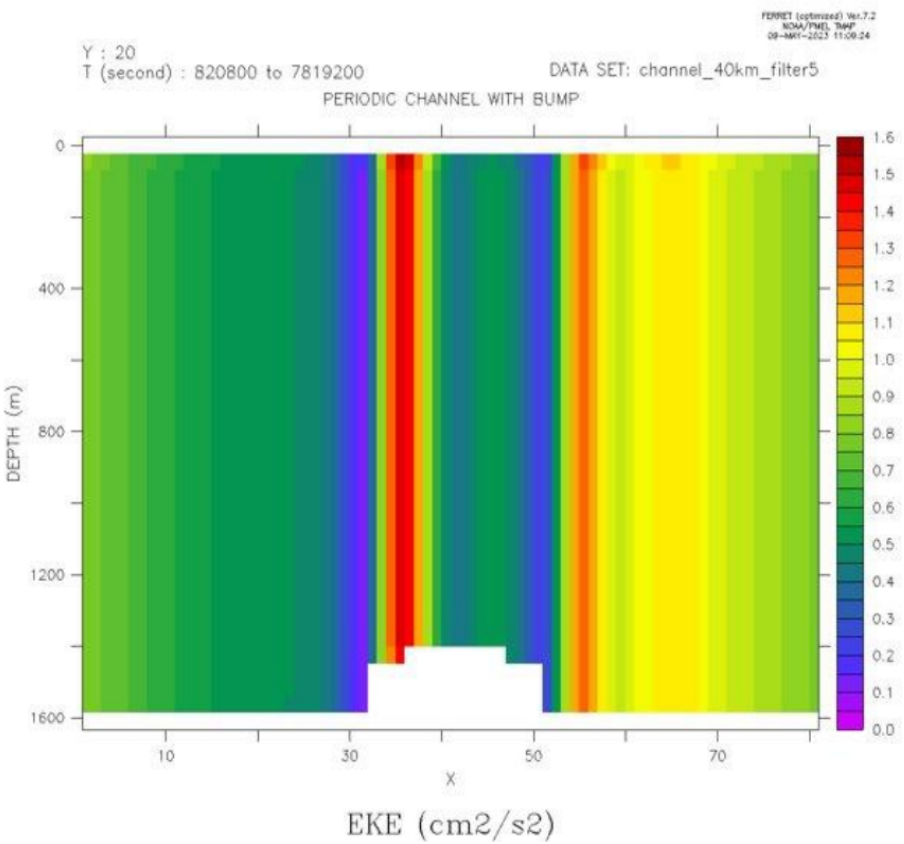
Leray-α in CROCO

Test case

eddy kinetic energy zonal vertical section



CROCO 20km



CROCO Leray-α 40km
filter size: 5

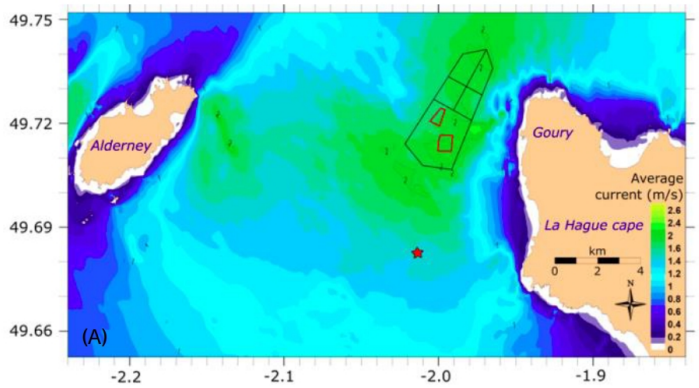
Conclusions & perspectives

Conclusions

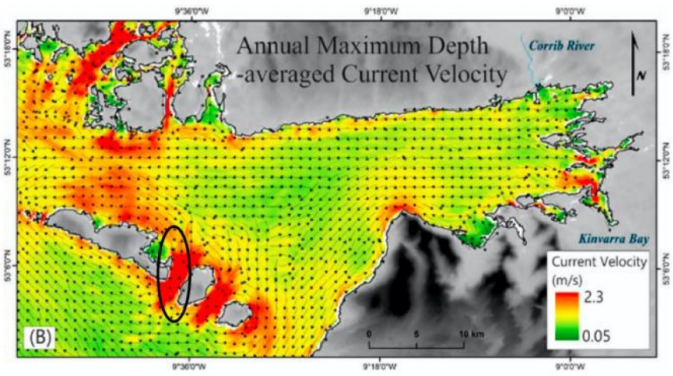
- Leray-α regularization re-energizes the flow: stronger eddy kinetic energy
- Applied to coarser configurations Leray-α can be used to simulate finer resolutions, with a lower computational cost

Perspectives

- MPI parallelization implementation (halo size increase and MPI exchanges)
- Application to coastal realistic configurations (Wave-current interactions, non-hydrostatic, ...)



Alderney race



Gregory sound
(Aran Islands, Ireland)