Introduction of GEM program files*

Liang Sun (sunl@ustc.edu.cn)

School of Earth and Space Sciences, University of Science and Technology of China, 230026, Hefei, China.

Genealogical Evolution Model (GEM) is an efficient logical model used to track dynamic evolution of mesoscale eddies in the ocean. It can distinguish different dynamic processes (e.g., merging and splitting) within a dynamic evolution pattern, which is difficult to accomplish using other tracking methods. To this end, GEM first uses a two-dimensional (2-D) vector rather than a scalar to measure the similarity between eddies, which effectively solves the "missing eddy" problem (temporally lost eddy in tracking). Second, GEM uses both parents and children in tracking, and the dynamic processes are described as birth and death of different generations. Additionally, a look-ahead approach with selection rules effectively simplifies computation and recording. All of the computational steps are linear and do not include iteration. Given the pixel number of the target region L, the maximum number of eddies M, the look-ahead time steps N, and the total number of time steps T, the total computation complexity is $O(LM(N+1)T)$. The tracking of each eddy is very smooth because we require that the snapshots of each eddy on adjacent days overlap one another.

0 Introduction

The project, supported by *the National Basic Research Program of China and the Natural Science Foundation of China* trying to solve eddy track problems in oceanography community, started in 2012. Both the model and the first version of the GEM programs (V00 in 2014) were accomplished in 2014. But the publication of them lagged too much. Not until in 2015, the watershed algorithm was published as a short note [Li and Sun, 2015]. As a conception model, the GEM itself does not grant a effective tracking program. For example, our previous tracking program (finished in 2012/2013) took almost one week to calculate 4.5 years of daily SLA data [Li et al., 2014], others showed us similar low efficiency (personal communications). It is a challenge to write such complex program (comparing with previous ones) with very high efficiency: the lowest storage cost, and the fastest calculation speed. We use many new algorithms and programming skills to accomplish this. So even a 20-year global eddy track computation only takes several hours in personal computer (by V00 in 2014). Now, we improve the 4th version of programs (V03 in 2016), which is even simpler and faster. It use only a fraction of storage that in V00, which is very useful for the data with large number and long-term eddies.

* This work was supported by the National Basic Research Program of China (Nos. 2012CB417402 and 2013CB430303), the National Natural Science Foundation of China (No. 41376017).

The programs are written in Fortran 90/95 standard on windows platform, including seven f90 program files. In a personal computer with CPU of i7-6700k and 4.00 GHz, program Mei.f90 uses 15 minutes to identify ***vortices*** (snapshot of eddy), program Vortex.f90 uses about 20 minutes to establish similarity, and program Eddy.f90 uses about 10 minutes to track eddies in the North Pacific Ocean (NPO).

Although we introduce the method only with SLA data and SLA-based eddy identification, the program Mei.f90 also includes ***other*** parameter-based (GV, OW, and hybrid) eddy identification algorithms. Readers can also use their own program to identify eddy, given the output files satisfying the input requirement of "Vortex.f90" and "Eddy.f90". It is well known that OW-based eddies may be different from SLA-based eddies. But we do not compare them at all (no one has done this yet). We point out here that the GEM programs can use to do so by evaluating the ratio of identified vortices to those be tracked as eddy, since "Eddy.f90" is ***independent*** of eddy identifications "Mei.f90". However, the users should be ***careful*** as some parameters are used in different program files (see section 4 below).

1 GEM Preparation

The present GEM programs use SLA data as input. We store all the SLA data files ('msla_19930101.dat') in a given directory as "SLApath", and all the data filename (19930101, 19930102, etc) in a txt file ('intxt =E:\sunl\WORK\Eddytrack\ini.txt'). Besides, we also use a topography file ("topofile=E:\sunl\WORK\Eddytrack\topo_721_1440.txt") since the SLA data have useless values on lands or coastal regions.

Then we need to prepare initial parameter file of 'inipara.txt' (see inipara in "Typeconst3.f90"), which includes these parameters.

```
open(22,file=inipara,form='formatted',action='read')
read(22,*) iconmin,iconmax, ias,ide !
read(22,*) intxt,topofile        ! datefile, topofile
read(22,*) SLApath                    ! SLA data path
read(22,*) vorpath,eddypath      ! vortices path, eddy path
close(22)
```

where iconmin=1 (first file), iconmax=7305 (last file of 20 year SLA data, 8035 for 22 year SLA data), ias=ide=1 is used for bi-way similarity calculations in "vortex.f90". The outputs of programs include identified vortices (vorpath) from SLA data and eddy tracks from vortices (eddypath).

In "vorpath", there are three series of files as "date-vor.dat", "date-voi.dat", "date-vet.dat", where "date" means "19930101, 19930102, etc."

In " eddypath", there are "Eddy-Eddt.dat" and "Eddy-Eddr.dat" (Eddy is identified as a integer number starting from 1000000), which contains long-term eddies (defined as life time>MP_Eddylif_CV_MIN). Besides there are two files as "TbEddy.txt" and "TpEddy.txt", which list parameters of all eddies (even life time< MP_Eddylif_CV_MIN).

2 GEM Programs

GEM programs include three independent parts：

1. Mononuclear Eddy Identification with SLA data by watershed
   Program files: Mei.f90+TypeConst.f90+Typemei.f90
   Input path: SLApath
   Input files: inipara.txt, topo_721_1440.txt,ini.txt, msla_date.dat
   Output path: vorpath
   Output files: "date-vor.dat", "date-voi.dat"
   Input data is sea surface height (SSH). We can get identified eddy information (the area of different eddy is marked by different integer number) in the output file "date-voi.dat" after running those programs. Two definition parameters for MEI model are used in this part:
   (1) MP_GRID_CV_MIN: minimum area of eddy.
   (2) MP_AMP_CV_MIN: minimum amplitude of eddy.

```
integer,parameter::MP_GRID_CV_MIN=9
real,parameter::MP_AMP_CV_MIN=3.E-2
real,parameter::MP_OW_CV_MIN=0.D-12
real,parameter::MP_OW_CV_MAX=1.D0
real,parameter::MP_GV_CV_MIN=-2.D0
real,parameter::MP_GV_CV_MAX=2.D0
real,parameter::MP_SSH_CV_ROB=1.D0
real,parameter::MP_SSH_CV_MIN=-3.E-2
real,parameter::MP_SSH_CV_MAX=3.E-2
integer,parameter::Max_Equator=381
integer,parameter::Min_Equator=341
integer,parameter::MAX_Road=250
integer,parameter::Flag_mark=1
integer,parameter::Fake_MEI=-1
```

2. Similarity vector
   Program files: Vortex.f90+TypeConst.f90+Typevors.f90
   Input path: Vorpath
   Input files: inipara.txt, ini.txt, "date-vor.dat", "date-voi.dat"
   Output path: Vorpath
   Output files: "date-vor.dat",
   Input data SSH and data Mei from the first step. We can get all the eddy information and the correlation (forward and backward) information of eddies. Also, two definitionparameters for GEM model are used in this part:
   (1) MP_Rc_CV_MIN: similarity vector (overlap ratio)
   (2) MP_MAX_DAY =N+2: N is the look-ahead time-steps (N=0 being the original linear closest eddy procedure without look-ahead)

real,parameter::MP_Rd_CV_MIN=0.75
real,parameter::MP_Rc_CV_MIN=2.0/3.0
integer,parameter::MP_MAX_DAY=4

3. Eddy tracks

Program files: Lines.f90+TypeConst.f90+Typeeddy.f90
Input path: Vorpath
Input files: inipara.txt, ini.txt, "date-vor.dat"
Output path: Vorpath   Output files: "date-vet.dat"
Output path: Eddypath
Output files: "Eddy-Eddt.dat","Eddy-Eddr.dat","TbEddy.txt" and "TpEddy.txt"
After the last step, we can get all the eddy information and the relationship between eddies and the eddy trajectories.

integer,parameter::MP_Eddylif_CV_MIN=15
integer,parameter::MAX_Eddy=1000000
integer,parameter::INI_Eddy=1000000
integer,parameter::MAX_Edds=1800
integer,parameter::MAX_Vors=1500
integer,parameter::MAX_XEdds=300
integer,parameter::MAX_LIFE=400
integer,parameter::MAX_XLIFE=2000
integer,parameter::MAX_CoList=100
integer,parameter::MAX_SpList=100
integer,parameter::Max_Gen=1
integer,parameter::MAX_EddBran=4
integer,parameter::MAX_EddProp=5
integer,parameter::MAX_EddDixi=3
integer,parameter::MAX_Pare=MAX_EddBran
integer,parameter::MAX_Chil=MAX_EddBran

Note that we track eddies only relay on "date-vor.dat". So any other method, which can generate seem data, can work with GEM programs.

4. System parameters

integer,parameter::MAX_LON=1440 ,
integer,parameter::MAX_LAT=720
real,parameter::Deft_SSH=-990
integer,parameter::MP_Bran_CV_MAX=3
integer,parameter::MAX_Extr=1500
integer,parameter::MP_MAX_LON_L=33
integer,parameter::MP_MAX_LAT_L=33
integer,parameter::MAX_Cov=15
integer,parameter::Min_Vorgrid=9,MAX_Vorgrid=1000

3 GEM Outputs

1. MEI Outputs

Mei includes two series output data files "date-vor.dat" and "date-voi.dat". The "date-vor.dat" is a ***txt*** file used in ***all*** program files, including Mei output (Mei.f90), Vortex output (vortex.f90) and Eddy output (Eddy.f90). This file lists the parameters of identified eddies (vortices) each day of "date". These parameters include the "vorid" (vortex number in each day), the "vorp" (vortex properties, such as latitude and longitude of vortex center, amplitude and area of vortex), and "vorr" (bi-way vortex relationship calculated by vortex.f90 with similarity vector method).

```
outfile=trim(vorpath)//trim(inlist(Id_file))//'-Vor.dat'
open(fout,file=outfile,form='formatted')
do i=1,Max_Vors
   write(fout,*)Vor(i)
end do
close(fout)
```

The "date-voi.dat" is a ***binary*** file of vortex map (two dimensional array of MAX_LON* MAX_LAT grids), which contains marks of vortices. According to the identification, vortex is set of grids (pixels). So each grid of map can be recorded as either an vortex (vortex number vorid) , non-vortex region (0), fake-vortex region with small amplitude or area (Fake_MEI), or land (Deft_SSH).

```
outfile=trim(vorpath)//trim(inlist(Id_file))//'-Voi.dat'
open(fout,file=trim(outfile),form='binary')
write(fout)((Voi(i,j),i=1,MAX_LON),j=1,MAX_LAT)
close(fout)
```

2. Vortex Outputs

The vortex.f90 have only one output file "date-vor.dat", which is produced previously by mei.f90. The purpose of vortex.f90 is calculation of "vorr" part in "date-vor.dat". The "vorr" part includes following parameters

```
integer::File_for,ntree_for
integer::File_bak,ntree_bak
integer::tree_for(MP_Bran_CV_MAX,2)
integer::tree_bak(MP_Bran_CV_MAX,2)
```

Note that the relationship is bi-way (forward and backward), which is not mentioned in paper. This is because we use both relationship to track eddies. However, we can't write such notation in the paper as readers may be confused by so much technical details and programming skills.

Another notation is that we track eddies only relay on "date-vor.dat". So any other method, which can generate seem data, can work with GEM programs.

3. Eddy Outputs

The Eddy.f90 have use only file "date-vor.dat" to generate eddy records but the records are output in two directories "Vorpath " and "Eddypath". In "Eddypath", the first "Eddy-Eddt.dat" is an list of time, center location, amplitude area of long-lived eddy (life time>MP_Eddylif_CV_MIN). The second "Eddy-Eddr.dat" records the roles and events of relative eddies. A typical file may contains generation type, termination type, parents, children, combining events and splitting events (date, eddy, locations, area, role). A example file may like this.

```
                         0         0       2       1
Parents            0
Children            0
Combining             2
19930104      4   1000016      1042      407      102       1
19930111     11   1000780      1052      403       43       1
Splitting           1
19930121     21   1001200      1043      404       32       1
```

We also record parameters of all eddies (including life time<MP_Eddylif_CV_MIN) in two files "TbEddy.txt" and "TpEddy.txt" (see Typeeddy.f90 for definition), which may be useful for the users.

In "Vorpath ", there is "date-vet.dat", which is a table maps from vortex to eddy in each day. One may want to know one vortex (vorid) identified on a special day (date) corresponding to which one eddy. This file gives the answer. For example, the first number 5 (vorid) is the snapshot of eddy 1000731 from the following record.

```
    5     529     405     51        -0.0483559        -0.1343027    1000731
```

4 GEM Parameters

The GEM programs include many parameters except for r_c and N mentioned in the paper. The following ones are special as they might be used in different files. Since the three individual steps must work together, some parameters in each program "mei.f90", "Vortex.f90" and "Eddy.f90" should be the same. Otherwise, it might be inconsistence or wrong.

1. The parameters should be kept the same in all 3 programs.
   These parameters mainly are for definition of “type vortex” in “TypeConst3.f90” and output file “-Vor.dat”.
   ```
   integer,parameter::MAX_Vors
   integer,parameter::MP_Bran_CV_MAX
   ```

2. The parameters should be kept the same in "mei.f90", "Vortex.f90".
   These parameters are used mainly as definition of regional grids.
   ```
   real,parameter::Start_LON=0
   real,parameter::Start_LAT=-Pi/2.0
   integer,parameter::MP_GRID_CV_MIN
   ```

```
integer,parameter::MAX_LON=1440
integer,parameter::MAX_LAT=720
integer,parameter::MAX_LON_R=1440
integer,parameter::MAX_LAT_R=720
real,parameter::Deft_SSH=-990
real,parameter::Pi=3.1415926535
```

3. The parameter should be kept the same in "Vortex.f90" and "Eddy.f90". The parameter is used in definition of type(Vortex1)::Vortices(MAX_Vors,MP_MAX_DAY) and integer::Num_Vors(MP_MAX_DAY)

```
integer,parameter::MP_MAX_DAY
```

5 GEM Extension

The GEM programs can have many extensions.

1. Identification with OW or GV

If the SLA data is not available, or for some special reasons one may use other parameters (GV, OW, or hybrid) or other method to identify eddy. Since the calculation may use second derivative of SLA, we add a subroutine for SLA smoothing (Figure 14 in paper). The present "mei.f90" and "Typemei.f90" include OW-based identification methods, one "subroutine MonoExtrId_USTC_NEW" with watershed, another "subroutine MonoExtrId_OW_Identify" without. Besides, user can also use their own method to deal with the eddy identification. Comparing with SLA-based method, OW-based method identifies smaller eddies. And the OW-based eddy region will easily be affected by background flow. So the parameter "MP_Rc_CV_MIN2=2/3" used in "Vortex.f90" should reduce to "1/3" or smaller, otherwise many small eddy can't be tracked. Even this, the tracking eddies are much short-lived than SLA-based method.

2. Regional zone

The identify program "Typemei.f90" is written in default value for global region with $0.25^{o} \times 0.25^{o}$ resolution. If user want to apply this program to a regional zone with a different resolution (e.g. $0.1^{o} \times 0.1^{o}$ resolution of OFES data), some changes may required before running the programs. The simplest but most expensive one (with least change in program files but largest computation and storage) is simply changing the resolution then read the data right at the corresponding region (other regions are marked as land or default value -999). The complex one is changing the program files, since there is no cycle constraint for longitude. Lots of subroutines in "Typemei.f90" and " Typeeddy.f90" should be changed. The middle one is changing the parameters and change the input and output files, see "subroutine MonoExtrId_ReadSSH_OFES" and "subroutine MonoExtrId_OutVor_OFES".

3. Tracking in 3D

As the eddies in the ocean is three-dimension (3D), user may want to track them with 3D simulation data. There might be two different ways to do so. The first one is directly using 2D algorithm by some vertical average. Although the eddy has a 3D structure in ocean, we can reduce it into a series of 2D maps using vertical layers. In each layer, there is a cross section of 3D structure. According to a universal structure theory [Zhang et al., 2013], the eddy in each layer (2D map) represents the whole eddy (3D structure). Then one can use any 2D algorithm to track eddies.

However, some recent observations [Zhang et al., 2016] reported that the top and bottom of eddy may shift more than 100 km. This large distance may mistake one surface eddy from another bottom eddy, or mistake one eddy as two different eddies. So we need a method tracking 3D eddies without these mistakes.

The GEM can accomplish this 3D tracking since the GEM model is an logically conceptive model which suits for any dimension problem. To achieve this goal, the user only extend the input of "Vortex.f90" into 3D, and modify the relative subroutines for 3D array, mainly in "subroutine Vortmap_compare".

References:

Li, Q. Y., Sun, L., Liu, S. S., Xian, T., & Yan, Y. F. (2014). A new mononuclear eddy identification method with simple splitting strategies. Remote Sensing Letters, 5(1), 65-72. doi:10.1080/2150704X.2013.872814.

Li, Q. Y., & Sun, L. (2015). Technical Note: Watershed strategy for oceanic mesoscale eddy splitting. Ocean Science, 11(2), 269-273. doi: 10.5194/os-11-269-2015.

Li, Q.-Y., Sun, L., and Lin, S.-F.: GEM: A Dynamic Tracking Model for Mesoscale Eddies in the Ocean, Ocean Sci. Discuss., doi:10.5194/os-2016-49, in review, 2016

Zhang, Z., Zhang, Y., Wang, W., & Huang, R. X. (2013). Universal structure of mesoscale eddies in the ocean. Geophysical Research Letters, 40(14), 3677-3681. doi: 10.1002/grl.50736.

Zhang, Z. et al (2016). Observed 3D Structure, Generation, and Dissipation of Oceanic Mesoscale Eddies in the South China Sea. Sci. Rep. 6, 24349; doi: 10.1038/srep24349.