1 Exercice : Implémentation d'un mécanisme de protection reprise (dossier `life-io`)

Dans cette exercice, nous allons doter un code d'un mécanisme de protection reprise que permet de reprendre un calcul là où le code s'était arrété. Code le code est parallèle, nous souhaitons que les écritures et lectures soient faites en parallèles. D'où l'utilisation des MPI-IO.

Informations diverses :

- Il est conseiller de charger un module MPI récent : `> module avail mpi`, `> module load mpi/...`
- La commande `> make` vous permet de compiler le code.
- Le code se lance de la manière suivante :
  `> mpirun -n <nb_processes> mlife/mlife-mpiio -x <dim_x> -y <dim_y> -i <nb_iter> -r <restart_iter> -p <checkpoint_filename>`
- example d'essai fonctionnel dès le début :
  `> mpirun -n 4 ./mlife -x 50 -y 50 -i 20 -r -1 -p useless`
- example d'essai fonctionnel une fois l'écriture de fichier au point :
  `> mpirun -n 4 ./mlife-mpiio -x 50 -y 50 -i 20 -r -1 -p my_io_save`
- example d'essai final (reprends le calcul à l'itération 19):
  `> mpirun -n 4 ./mlife-mpiio -x 50 -y 50 -i 30 -r 19 -p my_io_save`

Question 1

Nous commencons ici avec l'écriture parallèle du fichier. Compléter les zones dans l'ordre, et vérifier le fonctionnement du mécanisme.

(a)
- Zone-1-A communicator duplication : comm into mlifeio_comm (MPI_Comm_dup).
- Zone-1-A-bis communicator duplication : free mlifeio_comm communicator (MPI_Comm_free)

(b)
- Zone-1-B define the correct opening mode for writing the protection file.
- Zone 1-B-bis open a MPI_file fh named "filename", with amode (MPI_File_open).
- Zone 1-B-ter close a MPI_file fh (MPI_File_close)

(c)
- Zone-1-C Create in vectype a MPI_Type_vector for the desired portion of the matrix
- Zone-1-C-bis Position vectype so that it covers the desired data in the matrix (MPI_Get_address, MPI_Type_create_hindexed)
- Zone-1-C-ter Free vectype (MPI_Type_free)

(d)
- Zone-1-D Create in newtype a MPI structure (MPI_Type_create_struct, MPI_Get_address, MPI_BOTTOM)
- Zone-1-D-bis free rowblk

(e)
- Zone-1-E compute the file offset for process 0
- Zone-1-E-bis compute the file offset for non-0 processes
- Zone-1-E-ter commit the new type so that it can be used (MPI_Type_commit)
- Zone-1-E-quater Explicitly write the data at the computed offset in file, use MPI_File_write_at_all
- Zone-1-E-quinquies free type

Question 2

Nous traitons maintenant la lecture du fichier. Compléter les zones dans l'ordre, et vérifier le fonctionnement du mécanisme.

(a)
- Zone-2-A define the correct opening mode for reading the protection file
- Zone 2-A-bis open a MPI_file fh named "filename", with amode
- Zone 2-A-ter close a MPI_file fh

(b)
- Zone 2-B make all processes read the header of the file : it contains 3 integers (rows, columns, iterations), use MPI_File_read_at_all
- Zone 2-B-bis use an allreduce max on err to check if there was an error in the read.

(c)
- Zone 2-C compute the offset in the file (myfileoffset) to access the desired data (use myoffset)
- Zone 2-C-bis commit the type and read the selected data (type and offset), use MPI_Type_commit and MPI_File_read_at_all
- Zone 2-C-ter free type