

```

1  //////////////////////////////////////
2  //おまじない（組込み済み関数の参照用）
3  //これはそのまま
4  #include <stdio.h>
5  #include <math.h>
6  #include <stdlib.h>
7  //////////////////////////////////////
8
9  //////////////////////////////////////
10 //プログラムで扱う変数のうち
11 //計算中に変更されないパラメータを定義
12 //以下のプログラム中では、以下の文字は
13 //それぞれの定義に従い、コンパイル時に
14 //書きかえられる
15 //例：NXは51で自動的に書きかえられる
16 //x方向の格子線の数
17 #define NX 51
18 //y方向の格子線の数
19 #define NY 51
20 //計算領域のx方向の幅
21 #define XMAX 100.
22 //計算領域のy方向の幅
23 #define YMAX 100.
24 //クーラン条件により時間刻み幅dtを決定する際の
25 //安全係数
26 #define CFL_SAFE_COFF 0.1
27 //拡散方程式を解く際の安定条件Cdiffに対する
28 //安全係数
29 #define VIS_SAFE_COFF 0.1
30 //////////////////////////////////////
31
32
33 //////////////////////////////////////
34 //プログラムで使用する変数の定義（大域変数）
35 //計算格子に関する変数
36 //////////////////////////////////////
37 //格子のインデックス
38 int i,j;
39 //格子線数(nx:x方向,ny:y方向)
40 int nx,ny;
41 //計算時に一時的に使用する格子点のインデックス
42 int iup,jup,idn,jdn;
43 //x方向の格子線の座標(xx[0]～xx[NX+1]まで使用可能)
44 double xx[NX+2];
45 //y方向の格子線の座標(yy[0]～yy[NY+1]まで使用可能)
46 double yy[NY+2];
47 //x方向の格子幅（等間隔格子）
48 double dx;
49 //y方向の格子幅（等間隔格子）
50 double dy;
51 //x方向の計算領域の幅
52 double xmax;
53 //y方向の計算領域の幅
54 double ymax;
55
56 //////////////////////////////////////
57 //Color Functionの値を格納する2次元配列

```

```

58 //値(color_func[0][0]~color_func[NX+1][NY+1]まで使用可能)
59 double color_func[NX+2][NY+2];
60 //x方向微分
61 double color_func_gx[NX+2][NY+2];
62 //y方向微分
63 double color_func_gy[NX+2][NY+2];
64
65 //////////////////////////////////////
66 // x方向の流速Uを格納する2次元配列
67 //値(uu[0][0]~uu[NX+1][NY+1]まで使用可能)
68 double uu[NX+2][NY+2];
69 //x方向微分
70 double uu_gx[NX+2][NY+2];
71 //y方向微分
72 double uu_gy[NX+2][NY+2];
73
74 //////////////////////////////////////
75 // y方向の流速Vを格納する2次元配列
76 //値(vv[0][0]~vv[NX+1][NY+1]まで使用可能)
77 double vv[NX+2][NY+2];
78 //x方向微分
79 double vv_gx[NX+2][NY+2];
80 //y方向微分
81 double vv_gy[NX+2][NY+2];
82
83 //////////////////////////////////////
84 //Color Functionから見積もられる
85 //実際の密度  $\rho$  の値を保存する2次元配列
86 //(density[0][0]~density[NX+1][NY+1]まで使用可能)
87 //(CIP法により移流は計算しないので微分は無い)
88 double density[NX+2][NY+2];
89
90 //////////////////////////////////////
91 //Poisson方程式から求める圧力p
92 //(pressure[0][0]~pressure[NX+1][NY+1]まで使用可能)
93 //(CIP法により移流は計算しないので微分は無い)
94 double pressure[NX+2][NY+2];
95
96 //////////////////////////////////////
97 //種類の物理定数を格納する変数
98 //動粘性係数 (気体でも水でも同一)
99 double viscosity_coff_mol;
100 //水の部分の密度 (一定値とする)
101 double density_water;
102 //気体の部分の密度 (一定値とする)
103 double density_air;
104 //重力加速度
105 double gravity;
106
107 //////////////////////////////////////
108 //時間に関する変数
109 //Time Step "n"
110 int time_step;
111 //計算を終了する最大のTime Step数
112 int end_time_step;
113 //時刻 "tn"
114 double time;

```

```

115 //時間刻み幅  $\Delta t$ 
116 double dt;
117 //time_step=0 -> 1(n=0 -> 1)における時間刻み幅
118 double dt_initial;
119 //計算を終了する最大時刻
120 double end_time;
121
122 //////////////////////////////////////
123 //出力に関する変数
124 //中間結果を出力するファイルに付ける通し番号
125 int output_file_number;
126 //中間結果を出力する時刻を保存する変数
127 double output_time;
128 //中間結果を出力する時間間隔
129 double output_time_dt;
130 //出力時にファイルを開くための変数
131 FILE *fp;
132 //出力ファイル名を一時的に保存する文字配列
133 char filename[256];
134
135 //////////////////////////////////////
136 //各種計算時に一時的に必要な変数
137 //[i+1][j]における値を一時的に保存
138 double iup_value;
139 //[i-1][j]における値を一時的に保存
140 double idn_value;
141 //[i][j+1]における値を一時的に保存
142 double jup_value;
143 //[i][j-1]における値を一時的に保存
144 double jdn_value;
145 //時間刻み幅をクーラン数から決定する際に必要となる
146 //流速Uの絶対値の最大値を保存するのに使用
147 double umax;
148 //時間刻み幅をクーラン数から決定する際に必要となる
149 //流速Vの絶対値の最大値を保存するのに使用
150 double vmax;
151 //時間刻み幅をクーラン数から決定する際に必要となる
152 //流速Uの最大値を格子幅dxで割った値(umax/dx)を保存するのに使用
153 double uu_by_dx;
154 //時間刻み幅をクーラン数から決定する際に必要となる
155 //流速Vの最大値を格子幅dyで割った値(vmax/dy)を保存するのに使用
156 double vv_by_dy;
157 //時間刻み幅をクーラン数から決定する際に必要となる
158 //流速UがCFL条件を満たすように決定した仮の時間刻み幅
159 double dt_uu;
160 //時間刻み幅をクーラン数から決定する際に必要となる
161 //流速VがCFL条件を満たすように決定した仮の時間刻み幅
162 double dt_vv;
163 //移流計算を行う場合に使用
164 //スタガード格子における変数夫々の定義位置に応じた
165 //流速値を一時的に計算するのに使用する
166 double uu_tmp;
167 double vv_tmp;
168
169 //各種計算の際に計算前の古い値を格納しておくために使用
170 //Color Functionの値の古い値
171 double old_color_func[NX+2][NY+2];

```

```

172 //Color Functionのx微分の古い値
173 double old_color_func_gx[NX+2][NY+2];
174 //Color Functionのy微分の古い値
175 double old_color_func_gy[NX+2][NY+2];
176 //流速Uの値の古い値
177 double old_uu[NX+2][NY+2];
178 //流速Uのx微分の古い値
179 double old_uu_gx[NX+2][NY+2];
180 //流速Uのy微分の古い値
181 double old_uu_gy[NX+2][NY+2];
182 //流速Vの値の古い値
183 double old_vv[NX+2][NY+2];
184 //流速Vのx微分の古い値
185 double old_vv_gx[NX+2][NY+2];
186 //流速Vのy微分の古い値
187 double old_vv_gy[NX+2][NY+2];
188 //移流計算の際のxiからの上流点へのx方向の距離 UxDt
189 double udt;
190 //移流計算の際のyjからの上流点へのy方向の距離 VxDt
191 double vdt;
192 //移流計算の際のxiからの上流方向へのx方向格子幅 xiup-xi
193 double dxiup;
194 //移流計算の際のyjからの上流方向へのy方向格子幅 yjup-yj
195 double dyjup;
196
197
198 //CIP法の補間関数の係数
199 double C00,C10,C20,C30,C01,C02,C03,C31,C21,C11,C12,C13,AA;
200 //CIP法の微分の修正に用いる速度の勾配
201 double dudx,dudy,dvdx,dvdy;
202
203 //////////////////////////////////////
204 //使用する関数（サブルーチン）の宣言
205 int boundary();
206
207
208 //////////////////////////////////////
209 //////////////////////////////////////
210 //////////////////////////////////////
211 // 実際のプログラム本体
212 // 実行時にはここから順番に計算される
213
214 int main( )
215 {
216
217
218 //////////////////////////////////////
219 // 計算格子の設定
220 //////////////////////////////////////
221 //x方向の格子線数
222 nx=NX;
223 //y方向の格子線数
224 ny=NY;
225 //x方向の計算領域の幅
226 xmax=XMAX;
227 //y方向の計算領域の幅
228 ymax=YMAX;

```

```

229 //等間隔格子の場合のx方向の格子幅 Δx
230 dx=xmax/(nx-1);
231 //等間隔格子の場合のy方向の格子幅 Δy
232 dy=ymax/(ny-1);
233 //格子線のx座標 xi=xx[i]
234 //xx[0]~xx[nx+1] (xx[0],xx[nx+1]:仮想セル)
235 //実際の計算領域 xx[1]~xx[nx]
236 //xx[1]=0, xx[nx]=xmaxに一致
237 for(i=0;i<=nx+1;i++){
238     xx[i]=dx*(i-1);
239 }
240 //格子線のy座標 yj=yy[j]
241 //yy[0]~yy[ny+1] (yy[0],yy[ny+1]:仮想セル)
242 //実際の計算領域 yy[1]~yy[ny]
243 //yy[1]=0, yy[ny]=ymaxに一致
244 for(j=0;j<=ny+1;j++){
245     yy[j]=dy*(j-1);
246 }
247 ///////////////////////////////////////////////////
248 //計算格子の設定:ここまで
249 ///////////////////////////////////////////////////
250
251 ///////////////////////////////////////////////////
252 // 計算格子の座標点の出力(確認用)
253 ///////////////////////////////////////////////////
254 printf("Number of Grid Points: Nx=%d Ny=%d\n",nx,ny);
255 printf("Domain Size      : Xmax=%e Ymax=%e\n",xmax,ymax);
256 printf("Location of Grid Points is outputted into ¥\"Grid.dat¥\"¥n");
257 // File Prepare
258 fp=fopen("Grid.dat","w");
259 fprintf(fp,"i j xx[i] yy[j]¥n");
260 for(i=1;i<=nx;i++){
261     for(j=1;j<=ny;j++){
262         fprintf(fp,"%d %d %e %e¥n",i,j,xx[i],yy[j]);
263     }
264 }
265 fclose(fp);
266 ///////////////////////////////////////////////////
267 // 計算格子の座標点の出力:ここまで
268 ///////////////////////////////////////////////////
269
270
271
272 ///////////////////////////////////////////////////
273 //計算の時間方向に関する設定
274 ///////////////////////////////////////////////////
275 //計算開始時の時刻
276 time=0.;
277 //1 ステップ目の時間刻み Δt
278 dt_initial=1.e-4;
279 //計算を終了する時刻
280 end_time=100.;
281 //計算を終了するステップ数
282 end_time_step=1000;
283 ///////////////////////////////////////////////////
284 //計算の時間方向に関する設定:ここまで
285 ///////////////////////////////////////////////////

```

```

286
287
288 //////////////////////////////////////////////////
289 //計算結果の出力に関する設定
290 //////////////////////////////////////////////////
291 //途中結果の出力のファイルに付ける通し番号の開始値
292 output_file_number=0;
293 //途中結果を出力する時間間隔
294 output_time_dt=1.;
295 //途中結果を出力する次の時刻
296 output_time=output_time_dt;
297 //////////////////////////////////////////////////
298 //計算結果の出力に関する設定:ここまで
299 //////////////////////////////////////////////////
300
301
302 //////////////////////////////////////////////////
303 //初期条件の設定
304 //////////////////////////////////////////////////
305 //種々の物理定数
306 //動粘性係数
307 viscosity_coff_mol=1.e-6;
308 //水の密度(Projection法では一定とする)
309 density_water=1.e+3;
310 //気体の密度(Projection法では一定とする)
311 density_air=1.e0;
312 //重力加速度
313 gravity=-9.8;
314
315 //////////////////////////////////////////////////
316 // Color Functionの設定
317 //////////////////////////////////////////////////
318 // color_func[i][j]==1:水の領域
319 // color_func[i][j]==0:空気の領域
320 // 注意:Color Functionは計算セルの中央に定義するので
321 // 変数の範囲は, i=1 ~nx-1, j=1 ~ny-1
322 for(i=1;i<=nx-1;i++){
323     for(j=1;j<=ny-1;j++){
324         //水柱の領域の指定
325         if((xx[i]<0.3*xmax)&&(yy[j]<0.6*ymax)){
326             //水
327             color_func[i][j]=1.;
328         }else{
329             //空気
330             color_func[i][j]=0.;
331         }
332     }
333 //////////////////////////////////////////////////
334 // Color Functionの設定:ここまで
335 //////////////////////////////////////////////////
336
337 //////////////////////////////////////////////////
338 // 流速Uの設定
339 //////////////////////////////////////////////////
340 // 注意:Uはxi上に定義するので
341 // 変数の範囲は, i=1 ~nx, j=1 ~ny-1
342 for(i=1;i<=nx;i++){

```

```

343     for(j=1;j<=ny-1;j++){
344         uu[i][j]=1.;
345     }}
346     ///////////////////////////////////
347     // 流速Uの設定:ここまで
348     ///////////////////////////////////
349
350     ///////////////////////////////////
351     // 流速Vの設定
352     ///////////////////////////////////
353     // 注意:Vはyj上に定義するので
354     //     変数の範囲は, i=1~nx-1, j=1~ny
355     for(i=1;i<=nx-1;i++){
356         for(j=1;j<=ny;j++){
357             vv[i][j]=1.;
358         }}
359     ///////////////////////////////////
360     // 流速Vの設定:ここまで
361     ///////////////////////////////////
362
363     printf("Initial Condition:End¥n");
364
365     ///////////////////////////////////
366     //境界条件の設定
367     ///////////////////////////////////
368     //以上で設定されたColor Function, U, Vの値を基に
369     //仮想セルに境界条件に基づく値を設定(代入)する
370     boundary( );
371     ///////////////////////////////////
372     //境界条件の設定:ここまで
373     ///////////////////////////////////
374
375
376     ///////////////////////////////////
377     //CIP法で使用する微分(x勾配:gx, y勾配:gy)の
378     //初期条件を中心差分により設定
379     ///////////////////////////////////
380
381     ///////////////////////////////////
382     // Color Functionの微分の設定
383     ///////////////////////////////////
384     for(i=1;i<=nx-1;i++){
385         for(j=1;j<=ny-1;j++){
386             iup=i+1;
387             idn=i-1;
388             jup=j+1;
389             jdn=j-1;
390             iup_value=color_func[iup][j];
391             idn_value=color_func[idn][j];
392             jup_value=color_func[i][jup];
393             jdn_value=color_func[i][jdn];
394             //x方向中心差分
395             color_func_gx[i][j]=(iup_value-idn_value)/(2.*dx);
396             //y方向中心差分
397             color_func_gy[i][j]=(jup_value-jdn_value)/(2.*dy);
398         }}
399     ///////////////////////////////////

```

```

400 // Color Functionの微分の設定:ここまで
401 ///////////////////////////////////////////////////
402
403 ///////////////////////////////////////////////////
404 // 流速Uの微分の設定
405 ///////////////////////////////////////////////////
406 for(i=1;i<=nx;i++){
407     for(j=1;j<=ny-1;j++){
408         iup=i+1;
409         idn=i-1;
410         jup=j+1;
411         jdn=j-1;
412         iup_value=uu[iup][j];
413         idn_value=uu[idn][j];
414         jup_value=uu[i][jup];
415         jdn_value=uu[i][jdn];
416         //x方向中心差分
417         uu_gx[i][j]=(iup_value-idn_value)/(2.*dx);
418         //y方向中心差分
419         uu_gy[i][j]=(jup_value-jdn_value)/(2.*dy);
420     }
421 ///////////////////////////////////////////////////
422 // 流速Uの微分の設定:ここまで
423 ///////////////////////////////////////////////////
424
425 ///////////////////////////////////////////////////
426 // 流速Vの微分の設定
427 ///////////////////////////////////////////////////
428 for(i=1;i<=nx-1;i++){
429     for(j=1;j<=ny;j++){
430         iup=i+1;
431         idn=i-1;
432         jup=j+1;
433         jdn=j-1;
434         iup_value=vv[iup][j];
435         idn_value=vv[idn][j];
436         jup_value=vv[i][jup];
437         jdn_value=vv[i][jdn];
438         //x方向中心差分
439         vv_gx[i][j]=(iup_value-idn_value)/(2.*dx);
440         //y方向中心差分
441         vv_gy[i][j]=(jup_value-jdn_value)/(2.*dy);
442     }
443 ///////////////////////////////////////////////////
444 // 流速Vの微分の設定:ここまで
445 ///////////////////////////////////////////////////
446
447 ///////////////////////////////////////////////////
448 //境界条件の設定
449 ///////////////////////////////////////////////////
450 //以上で設定されたColor Function, U, Vの値を基に
451 //仮想セルに境界条件に基づく値を設定(代入)する
452 //計算領域内部の値が更新されるたびに境界条件を
453 //仮想セル上に書きしななければならない
454 boundary( );
455 ///////////////////////////////////////////////////
456 //境界条件の設定:ここまで

```



```

457 //////////////////////////////////////////////////
458
459 printf("Initial Derivative Estimation:End¥n");
460
461 //////////////////////////////////////////////////
462 //初期条件の設定:ここまで
463 //////////////////////////////////////////////////
464
465
466
467 //////////////////////////////////////////////////
468 //設定された初期条件を出力(確認用)
469 //////////////////////////////////////////////////
470
471 //////////////////////////////////////////////////
472 // Color Functionの出力
473 //////////////////////////////////////////////////
474 // Color_func_initial.datというファイル名で出力
475 fp=fopen("Color_func_initial.dat","w");
476 fprintf(fp,"xx[i] yy[j] color_func[i][j] color_func_gx[i][j] color_func_gy[i][j]¥n");
477 for(i=1;i<=nx-1;i++){
478     for(j=1;j<=ny-1;j++){
479         fprintf(fp,"%e %e %e %e %e¥n",xx[i],yy[j],
480             color_func[i][j],color_func_gx[i][j],color_func_gy[i][j]);
481     }
482     fprintf(fp,"¥n");
483 }
484 fclose(fp);
485 //////////////////////////////////////////////////
486 // Color Function:ここまで
487 //////////////////////////////////////////////////
488
489 //////////////////////////////////////////////////
490 // 流速Uの出力
491 //////////////////////////////////////////////////
492 // UU_initial.datというファイル名で出力
493 fp=fopen("UU_initial.dat","w");
494 fprintf(fp,"xx[i] yy[j] uu[i][j] uu_gx[i][j] uu_gy[i][j]¥n");
495 for(i=1;i<=nx;i++){
496     for(j=1;j<=ny-1;j++){
497         fprintf(fp,"%e %e %e %e %e¥n",xx[i],yy[j],
498             uu[i][j],uu_gx[i][j],uu_gy[i][j]);
499     }
500     fprintf(fp,"¥n");
501 }
502 fclose(fp);
503 //////////////////////////////////////////////////
504 // 流速U:ここまで
505 //////////////////////////////////////////////////
506
507 //////////////////////////////////////////////////
508 // 流速Vの出力
509 //////////////////////////////////////////////////
510 // VV_initial.datというファイル名で出力
511 fp=fopen("VV_initial.dat","w");
512 fprintf(fp,"xx[i] yy[j] vv[i][j] vv_gx[i][j] vv_gy[i][j]¥n");
513 for(i=1;i<=nx-1;i++){

```

```

514     for(j=1;j<=ny;j++){
515         fprintf(fp,"%e %e %e %e %e\n",xx[i],yy[j],
516             vv[i][j],vv_gx[i][j],vv_gy[i][j]);
517     }
518     fprintf(fp,"¥n");
519 }
520 fclose(fp);
521 //////////////////////////////////////
522 // 流速V:ここまで
523 //////////////////////////////////////
524
525 //////////////////////////////////////
526 //設定された初期条件を出力(確認用):ここまで
527 //////////////////////////////////////
528
529 //////////////////////////////////////
530 //////////////////////////////////////
531 //メインループ:時間発展を計算する本体
532 //////////////////////////////////////
533 // 次のtime_stepを1～1つつ増加させて計算を繰り返す
534 // ステップ数time_stepがend_time_stepより大きくなるか
535 // 時刻timeがend_timeより大きくなったら終了する
536 for(time_step=1;
537     (time_step<=end_time_step)&&(time<=end_time);
538     time_step++){
539
540     //////////////////////////////////////
541     //時間刻み幅Δtの決定
542     //////////////////////////////////////
543     //安定条件を満足するように決定する
544     //(B):CFL条件 (Udt/dx)<CFL_SAFE_COFF
545     //(F):拡散方程式に関する条件
546     //(vis_coff_mol*dt/dx/dx)<VIS_SAFE_COFF/2
547     //////////////////////////////////////
548
549     //////////////////////////////////////
550     //(B):CFL条件に関する条件
551     //////////////////////////////////////
552     //流速Uの絶対値の最大を求める
553     umax=-1.e+10;
554     for(i=1;i<=nx;i++){
555         for(j=1;j<=ny-1;j++){
556             //fabs(XX): XXの絶対値を計算する組み込み関数
557             if(fabs(uu[i][j])>umax){umax=fabs(uu[i][j]);}
558         }
559     }
560     //流速Vの絶対値の最大を求める
561     vmax=-1.e+10;
562     for(i=1;i<=nx-1;i++){
563         for(j=1;j<=ny;j++){
564             if(fabs(vv[i][j])>vmax){vmax=fabs(vv[i][j]);}
565         }
566     }
567     //Umax/dxとVmax/dyを一時的に計算する
568     uu_by_dx=umax/dx;
569     vv_by_dy=vmax/dy;
570     //UmaxからCFL条件を満足するようなΔtを決定(dt_uu)
571     if(uu_by_dx>0.){
572         dt_uu=CFL_SAFE_COFF/uu_by_dx;

```

```

571 }else{
572     //Umax=0の場合、即ちすべてのUが0の場合
573     //Uからの制限はない
574     dt_uu=0.;
575 }
576 //VmaxからCFL条件を満足するようなΔtを決定(dt_vv)
577 if(vv_by_dy>0.){
578     dt_vv=CFL_SAFE_COFF/vv_by_dy;
579 }else{
580     //Vmax=0の場合、即ちすべてのVが0の場合
581     //Vからの制限はない
582     dt_vv=0.;
583 }
584 //dt_uuとdt_vvの小さい方を次の時間刻みとする
585 //UとVともにある流速を持つ場合
586 if((dt_uu>0.)&&(dt_vv>0.)){
587     //dt_uuとdt_vvの小さい方を次の時間刻みとする
588     if(dt_uu<dt_vv){
589         dt=dt_uu;
590     }else{
591         dt=dt_vv;
592     }
593 }else{
594     //U=0でVがある流速を持つ場合
595     //dt_vvを採用
596     if((dt_uu<=0.)&&(dt_vv>0.)){
597         dt=dt_vv;
598     }
599     //V=0でUがある流速を持つ場合
600     //dt_uuを採用
601     if((dt_uu>0.)&&(dt_vv<=0.)){
602         dt=dt_uu;
603     }
604     //U=0かつV=0の場合:CFL条件に対する制限無し
605     //dt_initialを採用
606     if((dt_uu<=0.)&&(dt_vv<=0.)){
607         dt=dt_initial;
608     }
609 }
610 ///////////////////////////////////////////////////
611 //(B):CFL条件に関する条件:ここまで
612 ///////////////////////////////////////////////////
613 ///////////////////////////////////////////////////
614 ///////////////////////////////////////////////////
615 //(F):拡散方程式に関する条件
616 ///////////////////////////////////////////////////
617 //ここに(F)の条件に関するΔtの決定式を書く
618 ///////////////////////////////////////////////////
619 //(F):拡散方程式に関する条件:ここまで
620 ///////////////////////////////////////////////////
621 ///////////////////////////////////////////////////
622 //決定された時間刻みだけ時間を進め
623 //これから計算する次時刻tn+1の時刻をtimeで計算
624 ///////////////////////////////////////////////////
625 time=time+dt;
626
627

```

```

628 //////////////////////////////////////////////////
629 //決定された時間間隔等を画面に出力
630 //////////////////////////////////////////////////
631 printf("#####\n");
632 printf("Time Progress: n=%d -> n+1=%d\n",time_step-1,time_step);
633 printf("Time      : time(tn+1)=%e\n",time);
634 printf("Time Interval: dt=%e\n",dt);
635
636 //////////////////////////////////////////////////
637 //時間刻み幅Δtの決定:ここまで
638 //////////////////////////////////////////////////
639
640
641
642 //////////////////////////////////////////////////
643 //Step1: 移流相(移流方程式の計算)
644 //////////////////////////////////////////////////
645
646 //////////////////////////////////////////////////
647 //計算前の古い値fnをoldに格納
648 //////////////////////////////////////////////////
649 //以後の計算にはoldを用いて
650 //新しい値は上書きで保存する
651 //////////////////////////////////////////////////
652
653 //////////////////////////////////////////////////
654 // Color Functionの古い値
655 //////////////////////////////////////////////////
656 //注意:仮想セルの値もすべてoldに移す
657 //   i=0~nx, j=0~ny
658 for(i=0;i<=nx;i++){
659     for(j=0;j<=ny;j++){
660         old_color_func[i][j]=color_func[i][j];
661         old_color_func_gx[i][j]=color_func_gx[i][j];
662         old_color_func_gy[i][j]=color_func_gy[i][j];
663     }}
664 //////////////////////////////////////////////////
665 // Color Functionの古い値:ここまで
666 //////////////////////////////////////////////////
667
668 //////////////////////////////////////////////////
669 // 流速Uの古い値
670 //////////////////////////////////////////////////
671 //注意:仮想セルの値もすべてoldに移す
672 //   i=0~nx+1, j=0~ny
673 for(i=0;i<=nx+1;i++){
674     for(j=0;j<=ny;j++){
675         old_uu[i][j]=uu[i][j];
676         old_uu_gx[i][j]=uu_gx[i][j];
677         old_uu_gy[i][j]=uu_gy[i][j];
678     }}
679 //////////////////////////////////////////////////
680 // 流速Uの古い値:ここまで
681 //////////////////////////////////////////////////
682
683 //////////////////////////////////////////////////
684 // 流速Vの古い値

```

```

685 ///////////////////////////////////////////////////
686 //注意:仮想セルの値もすべてoldに移す
687 //    i=0~nx, j=0~ny+1
688 for(i=0;i<=nx;i++){
689     for(j=0;j<=ny+1;j++){
690         old_vv[i][j]=vv[i][j];
691         old_vv_gx[i][j]=vv_gx[i][j];
692         old_vv_gy[i][j]=vv_gy[i][j];
693     }
694 ///////////////////////////////////////////////////
695 // 流速Vの古い値:ここまで
696 ///////////////////////////////////////////////////
697
698 ///////////////////////////////////////////////////
699 //計算前の古い値fnをoldに格納:ここまで
700 ///////////////////////////////////////////////////
701
702
703
704 ///////////////////////////////////////////////////
705 //移流計算の本体
706 ///////////////////////////////////////////////////
707
708 ///////////////////////////////////////////////////
709 // Color Functionの移流
710 ///////////////////////////////////////////////////
711 //注意:仮想セルの値以外を計算し、
712 //    計算結果をold_colorに上書きする
713 //    i=1~nx-1, j=1~ny-1
714 for(i=1;i<=nx-1;i++){
715     for(j=1;j<=ny-1;j++){
716         //Color Functionは計算セルの中央
717         //中央での流速値を計算セルの境界面上に位置する
718         //流速uu[i][j],uu[i+1][j],vv[i][j],vv[i][j+1]
719         //から平均操作で推定
720         uu_tmp=0.5*(old_uu[i][j]+old_uu[i+1][j]);
721         vv_tmp=0.5*(old_vv[i][j]+old_vv[i+1][j]);
722         //上流側の格子点の決定
723         if(uu_tmp>0.){iup=i-1;}else{iup=i+1;}
724         if(vv_tmp>0.){jup=j-1;}else{jup=j+1;}
725         //移流する距離(上流点への距離)の計算
726         udt=-uu_tmp*dt;
727         vdt=-vv_tmp*dt;
728         //上流側の格子点への距離
729         // Δxi->iup=dxilup
730         dxilup=xx[iup]-xx[i];
731         // Δyj->jup=dyjup
732         dyjup=yy[jup]-yy[j];
733         //CIP法の計算
734         C00=....
735         C10=....
736         C01=....
737         C20=....
738         C02=....
739         C30=....
740         AA=....
741         C03=....

```

```

742     C31=....
743     C13=....
744     C21=....
745     C12=....
746     C11=....
747
748     color_func[i][j]=...
749     color_func_gx[i][j]=...
750     color_func_gy[i][j]=...
751     }}
752     //////////////////////////////////////
753     // 微分の修正
754     //////////////////////////////////////
755     for(i=1;i<=nx-1;i++){
756         for(j=1;j<=ny-1;j++){
757             old_color_func_gx[i][j]=color_func_gx[i][j];
758             old_color_func_gy[i][j]=color_func_gy[i][j];
759         }}
760     for(i=1;i<=nx-1;i++){
761         for(j=1;j<=ny-1;j++){
762             //Color Functionは計算セルの中央
763             //速度の微分を中心差分で求める
764             dudx=(old_uu[i+1][j]-old_uu[i][j])/dx;
765             dvdy=(old_vv[i][j+1]-old_vv[i][j])/dy;
766             dudy=(0.5*(old_uu[i][j+1]+old_uu[i+1][j+1])
767                 -0.5*(old_uu[i][j-1]+old_uu[i+1][j-1]))/(2.*dy);
768             dvdx=(0.5*(old_vv[i+1][j]+old_vv[i+1][j+1])
769                 -0.5*(old_vv[i-1][j]+old_vv[i-1][j+1]))/(2.*dx);
770             color_func_gx[i][j]=old_color_func_gx[i][j]
771                 -dudx*old_color_func_gx[i][j]*dt
772                 -dvdx*old_color_func_gy[i][j]*dt;
773             color_func_gy[i][j]=old_color_func_gy[i][j]
774                 -dudy*old_color_func_gx[i][j]*dt
775                 -dvdy*old_color_func_gy[i][j]*dt;
776         }}
777
778
779     //////////////////////////////////////
780     // Color Functionの移流:ここまで
781     //////////////////////////////////////
782
783     //////////////////////////////////////
784     //境界条件の設定
785     //////////////////////////////////////
786     boundary( );
787     //////////////////////////////////////
788     //境界条件の設定:ここまで
789     //////////////////////////////////////
790
791
792     //////////////////////////////////////
793     // 流速Uの移流
794     //////////////////////////////////////
795     //注意:仮想セルの値以外を計算し、
796     // 計算結果をuuに上書きする
797     // i=1~nx, j=1~ny-1
798     for(i=1;i<=nx;i++){

```

```

799     for(j=1;j<=ny-1;j++){
800         //uu[i][j]はxi軸の上
801         //Vの流速値を付近の4点のVから平均して求める
802         //vv[i][j],vv[i][j+1],vv[i-1][j],vv[i-1][j+1]
803         uu_tmp=old_uu[i][j];
804         vv_tmp=0.25*(old_vv[i][j]+old_vv[i][j+1]
805             +old_vv[i-1][j]+old_vv[i-1][j+1]);
806         //上流側の格子点の決定
807         if(uu_tmp>0.){iup=i-1;}else{iup=i+1;}
808         if(vv_tmp>0.){jup=j-1;}else{jup=j+1;}
809         //移流する距離(上流点への距離)の計算
810         udt=-uu_tmp*dt;
811         vdt=-vv_tmp*dt;
812         //上流側の格子点への距離
813         //Δxi->iup=dx_iup
814         dx_iup=xx[iup]-xx[i];
815         //Δyj->jup=dy_jup
816         dy_jup=yy[jup]-yy[j];
817         //CIP法の計算
818         C00=....
819         C10=....
820         C01=....
821         C20=....
822         C02=....
823         C30=....
824         AA=....
825         C03=....
826         C31=....
827         C13=....
828         C21=....
829         C12=....
830         C11=....
831         uu[i][j]=....
832         uu_gx[i][j]=....
833         uu_gy[i][j]=....
834     }}
835     //////////////////////////////////////
836     // 微分の修正
837     //////////////////////////////////////
838     for(i=1;i<=nx;i++){
839         for(j=1;j<=ny-1;j++){
840             old_uu_gx[i][j]=uu_gx[i][j];
841             old_uu_gy[i][j]=uu_gy[i][j];
842         }}
843     for(i=1;i<=nx;i++){
844         for(j=1;j<=ny-1;j++){
845             //速度の微分を中心差分で求める
846             dudx=(old_uu[i+1][j]-old_uu[i-1][j])/(2.*dx);
847             dvdy=(0.5*(old_vv[i][j+1]+old_vv[i-1][j+1])
848                 -0.5*(old_vv[i][j]+old_vv[i-1][j]))/dy;
849             dudy=(old_uu[i][j+1]-old_uu[i][j-1])/(2.*dy);
850             dvdx=(0.5*(old_vv[i][j]+old_vv[i][j+1])
851                 -0.5*(old_vv[i-1][j]+old_vv[i-1][j+1]))/(dx);
852             uu_gx[i][j]=old_uu_gx[i][j]
853                 -dudx*old_uu_gx[i][j]*dt
854                 -dvdx*old_uu_gy[i][j]*dt;
855             uu_gy[i][j]=old_uu_gy[i][j]

```

```

856         -dudy*old_uu_gx[i][j]*dt
857         -dvdy*old_uu_gy[i][j]*dt;
858     }}
859
860     //////////////////////////////////////
861     // 流速Uの移流:ここまで
862     //////////////////////////////////////
863
864     //////////////////////////////////////
865     //境界条件の設定
866     //////////////////////////////////////
867     boundary( );
868     //////////////////////////////////////
869     //境界条件の設定:ここまで
870     //////////////////////////////////////
871
872
873     //////////////////////////////////////
874     // 流速Vの移流
875     //////////////////////////////////////
876     //注意:仮想セルの値以外を計算し、
877     // 計算結果をvvに上書きする
878     // i=1~nx-1, j=1~ny
879     for(i=1;i<=nx-1;i++){
880         for(j=1;j<=ny;j++){
881             //vv[i][j]はyj軸の上
882             //Uの流速値を付近の4点のUから平均して求める
883             //uu[i][j],uu[i+1][j],uu[i][j-1],uu[i+1][j-1]
884             uu_tmp=0.25*(old_uu[i][j]+old_uu[i+1][j]
885                 +old_uu[i][j-1]+old_uu[i+1][j-1]);
886             vv_tmp=old_vv[i][j];
887             //上流側の格子点の決定
888             if(uu_tmp>0.){iup=i-1;}else{iup=i+1;}
889             if(vv_tmp>0.){jup=j-1;}else{jup=j+1;}
890             //移流する距離(上流点への距離)の計算
891             udt=-uu_tmp*dt;
892             vdt=-vv_tmp*dt;
893             //上流側の格子点への距離
894             // Δxi->iup=dx_iup
895             dx_iup=xx[iup]-xx[i];
896             // Δyj->jup=dy_jup
897             dy_jup=yy[jup]-yy[j];
898             //CIP法の計算
899             C00=....
900             C10=....
901             C01=....
902             C20=....
903             C02=....
904             C30=....
905             C03=....
906             C31=....
907             C13=....
908             C21=....
909             C12=....
910             C11=....
911             vv[i][j]=....
912             vv_gx[i][j]=....

```



```

913     vv_gy[i][j]=....
914 }
915 //////////////////////////////////////////////////
916 // 微分の修正
917 //////////////////////////////////////////////////
918 for(i=1;i<=nx-1;i++){
919     for(j=1;j<=ny;j++){
920         old_vv_gx[i][j]=vv_gx[i][j];
921         old_vv_gy[i][j]=vv_gy[i][j];
922     }
923     for(i=1;i<=nx-1;i++){
924         for(j=1;j<=ny;j++){
925             //速度の微分を中心差分で求める
926             dudx=(0.5*(old_uu[i+1][j]+old_uu[i+1][j-1])
927                 -0.5*(old_uu[i][j]+old_uu[i][j-1]))/dx;
928             dvdy=(old_vv[i][j+1]-old_vv[i][j-1])/(2.*dx);
929             dudy=(0.5*(old_uu[i][j+1]+old_uu[i+1][j+1])
930                 -0.5*(old_uu[i][j]+old_uu[i+1][j]))/dy;
931             dvdx=(old_vv[i+1][j]-old_vv[i-1][j])/(2.*dx);
932             vv_gx[i][j]=old_vv_gx[i][j]
933                 -dudx*old_vv_gx[i][j]*dt
934                 -dvdx*old_vv_gy[i][j]*dt;
935             vv_gy[i][j]=old_vv_gy[i][j]
936                 -dudy*old_vv_gx[i][j]*dt
937                 -dvdy*old_vv_gy[i][j]*dt;
938         }
939     }
940     // 流速Vの移流:ここまで
941     //////////////////////////////////
942     //////////////////////////////////
943     //////////////////////////////////
944     //境界条件の設定
945     //////////////////////////////////
946     boundary( );
947     //////////////////////////////////
948     //境界条件の設定:ここまで
949     //////////////////////////////////
950
951
952     //////////////////////////////////
953     //途中結果の出力
954     //////////////////////////////////
955     //時刻が次に出力すべき時刻output_timeを超えたら出力
956     if(time>=output_time){
957         //出力ファイルの通し番号を+1する
958         output_file_number=output_file_number+1;
959         //次に出力する時刻に更新
960         output_time=output_time+output_time_dt;
961
962         //////////////////////////////////
963         // Color Function 出力
964         //////////////////////////////////
965         // Color_func_xxxx.datというファイル名で出力
966         // xxxxの部分は通し番号がつく
967         sprintf(filename,"Color_func_%4.4d.dat",output_file_number);
968         fp=fopen(filename,"w");
969         fprintf(fp,"xx[i] yy[j] color_func[i][j] color_func_gx[i][j] color_func_gy[i][j]¥n");

```

```

970     for(i=1;i<=nx-1;i++){
971         for(j=1;j<=ny-1;j++){
972             fprintf(fp,"%e %e %e %e %e\n",xx[i],yy[j],
973                 color_func[i][j],color_func_gx[i][j],color_func_gy[i][j]);
974         }
975         fprintf(fp,"\n");
976     }
977     fclose(fp);
978     ///////////////////////////////////
979     // Color Function 出力:ここまで
980     ///////////////////////////////////
981
982     ///////////////////////////////////
983     // 流速U 出力
984     ///////////////////////////////////
985     // UU_xxxx.datというファイル名で出力
986     // xxxxの部分は通し番号がつく
987     sprintf(filename,"UU_%4.4d.dat",output_file_number);
988     fp=fopen(filename,"w");
989     fprintf(fp,"xx[i] yy[j] uu[i][j] uu_gx[i][j] uu_gy[i][j]\n");
990     for(i=1;i<=nx;i++){
991         for(j=1;j<=ny-1;j++){
992             fprintf(fp,"%e %e %e %e %e\n",xx[i],yy[j],
993                 uu[i][j],uu_gx[i][j],uu_gy[i][j]);
994         }
995         fprintf(fp,"\n");
996     }
997     fclose(fp);
998
999     ///////////////////////////////////
1000    // 流速U 出力:ここまで
1001    ///////////////////////////////////
1002
1003    ///////////////////////////////////
1004    // 流速V 出力
1005    ///////////////////////////////////
1006    // VV_xxxx.datというファイル名で出力
1007    // xxxxの部分は通し番号がつく
1008    sprintf(filename,"VV_%4.4d.dat",output_file_number);
1009    fp=fopen(filename,"w");
1010    fprintf(fp,"xx[i] yy[j] vv[i][j] vv_gx[i][j] vv_gy[i][j]\n");
1011    for(i=1;i<=nx-1;i++){
1012        for(j=1;j<=ny;j++){
1013            fprintf(fp,"%e %e %e %e %e\n",xx[i],yy[j],
1014                vv[i][j],vv_gx[i][j],vv_gy[i][j]);
1015        }
1016        fprintf(fp,"\n");
1017    }
1018    fclose(fp);
1019    ///////////////////////////////////
1020    // 流速V 出力:ここまで
1021    ///////////////////////////////////
1022
1023 }
1024 ///////////////////////////////////
1025 //途中結果の出力:ここまで
1026 ///////////////////////////////////

```

```

1027
1028
1029
1030
1031 }
1032 //////////////////////////////////////////////////
1033 //メインループ:時間発展を計算する本体:ここまで
1034 //////////////////////////////////////////////////
1035
1036
1037
1038 //////////////////////////////////////////////////
1039 //プログラムの終了
1040 //////////////////////////////////////////////////
1041
1042 return 0;
1043 }
1044
1045
1046
1047
1048
1049 //////////////////////////////////////////////////
1050 //////////////////////////////////////////////////
1051 //////////////////////////////////////////////////
1052 //////////////////////////////////////////////////
1053 //サブルーチンboundaryの定義
1054 //////////////////////////////////////////////////
1055
1056 int boundary( )
1057 {
1058
1059 //////////////////////////////////////////////////
1060 // 境界I(Y=Ymax)の仮想セルの設定
1061 // 自由境界
1062 //////////////////////////////////////////////////
1063 for(i=0;i<=nx;i++){
1064     color_func[i][ny]=color_func[i][ny-1];
1065     color_func_gx[i][ny]=color_func_gx[i][ny-1];
1066     color_func_gy[i][ny]=0.;
1067 }
1068 for(i=0;i<=nx+1;i++){
1069     uu[i][ny]=uu[i][ny-1];
1070     uu_gx[i][ny]=uu_gx[i][ny-1];
1071     uu_gy[i][ny]=0.;
1072 }
1073 for(i=0;i<=nx;i++){
1074     vv[i][ny+1]=vv[i][ny];
1075     vv_gx[i][ny+1]=vv_gx[i][ny];
1076     vv_gy[i][ny+1]=0.;
1077 }
1078 //////////////////////////////////////////////////
1079 // 境界I(Y=Ymax)の仮想セルの設定:ここまで
1080 //////////////////////////////////////////////////
1081
1082 //////////////////////////////////////////////////
1083 // 境界II(X=0)の仮想セルの設定

```

```

1084 // 壁面境界
1085 //////////////////////////////////////
1086 for(j=0;j<=ny;j++){
1087     color_func[0][j]=color_func[1][j];
1088     color_func_gx[0][j]=-color_func_gx[1][j];
1089     color_func_gy[0][j]=color_func_gy[1][j];
1090 }
1091 for(j=0;j<=ny;j++){
1092     //壁面上でUは0
1093     uu[1][j]=0.;
1094     uu[0][j]=-uu[2][j];
1095     uu_gx[0][j]=uu_gx[2][j];
1096     uu_gy[0][j]=-uu_gy[2][j];
1097 }
1098 //すべり無し境界
1099 for(j=0;j<=ny+1;j++){
1100     vv[0][j]=-vv[1][j];
1101     vv_gx[0][j]=vv_gx[1][j];
1102     vv_gy[0][j]=-vv_gy[1][j];
1103 }
1104 //////////////////////////////////////
1105 // 境界II(X=0)の仮想セルの設定:ここまで
1106 //////////////////////////////////////
1107
1108 //////////////////////////////////////
1109 // 境界III(Y=0)の仮想セルの設定
1110 // 壁面境界
1111 //////////////////////////////////////
1112 for(i=0;i<=nx;i++){
1113     color_func[i][0]=color_func[i][1];
1114     color_func_gx[i][0]=color_func_gx[i][1];
1115     color_func_gy[i][0]=0.;
1116 }
1117 //すべり無し境界
1118 for(i=0;i<=nx+1;i++){
1119     uu[i][0]=-uu[i][1];
1120     uu_gx[i][0]=-uu_gx[i][1];
1121     uu_gy[i][0]=uu_gy[i][1];
1122 }
1123 for(i=0;i<=nx;i++){
1124     //壁面上でV=0
1125     vv[i][1]=0.;
1126     vv[i][0]=-vv[i][2];
1127     vv_gx[i][0]=-vv_gx[i][2];
1128     vv_gy[i][0]=vv_gy[i][2];
1129 }
1130 //////////////////////////////////////
1131 // 境界III(Y=0)の仮想セルの設定:ここまで
1132 //////////////////////////////////////
1133
1134 //////////////////////////////////////
1135 // 境界IV(X=Xmax)の仮想セルの設定
1136 // 壁面境界
1137 //////////////////////////////////////
1138 for(j=0;j<=ny;j++){
1139     color_func[nx][j]=color_func[nx-1][j];
1140     color_func_gx[nx][j]=-color_func_gx[nx-1][j];

```

```

1141     color_func_gy[nx][j]=color_func_gy[nx-1][j];
1142 }
1143 for(j=0;j<=ny;j++){
1144     //壁面上でU=0
1145     uu[nx][j]=0.;
1146     uu[nx+1][j]=-uu[nx-1][j];
1147     uu_gx[nx+1][j]=uu_gx[nx-1][j];
1148     uu_gy[nx+1][j]=-uu_gy[nx-1][j];
1149 }
1150 //すべり無し境界
1151 for(j=0;j<=ny+1;j++){
1152     vv[nx][j]=-vv[nx-1][j];
1153     vv_gx[nx][j]=vv_gx[nx-1][j];
1154     vv_gy[nx][j]=-vv_gy[nx-1][j];
1155 }
1156 //////////////////////////////////////
1157 // 境界IV(X=Xmax)の仮想セルの設定:ここまで
1158 //////////////////////////////////////
1159
1160
1161     return 0;
1162
1163
1164 }
1165
1166
1167 //////////////////////////////////////
1168 //サブルーチンboundaryの定義:ここまで
1169 //////////////////////////////////////
1170

```