

## 本選課題解説

スーパーコン2013審査委員会  
東京工業大学・大阪大学

### 宇宙の起源を探せ！(宇宙の起源探索問題)

ビッグバン直後の宇宙の極々初期の状態変化を解明するGK理論がスーパーコン研究所の菊池・権藤両博士によって提唱された。一方、ビッグバン直後の粒子<sup>1</sup>の状況については、遠藤博士が提唱した理論により、 $K$ 種類のパターンしかないことがわかっている。さらに遠藤博士の調査により、宇宙がある程度進化した時刻(仮に $T$ 時刻と呼ぶ)の粒子の分布状況については、 $L$ 種類( $L$ は $K$ に比べてかなり小さい)の候補に絞られることを突き止めた。この新たな調査結果を用い、GK理論で時刻をさかのぼり、宇宙の初期状態の候補を $K$ 種類から $L$ 種類に絞り込みたい。

この計算をGPUを使って高速に解くことを今回の目標にしよう。

### 課題

#### 宇宙の起源探索問題

2次元の $N \times N$ の格子平面上に3色の粒子が各々 $M$ 個存在し、各々が一定の速さ(向きは異なる)で、ある法則に従って移動する宇宙モデルを考える。入力としては、A群と呼ばれる $K$ 個の宇宙状態の集合とB群と呼ばれる $L$ 個の宇宙状態の集合が与えられる。ここで、宇宙状態とは、その宇宙における全粒子の状態(=粒子の位置と速度(向き))の全体のことである。

A群は初期の宇宙状態の候補である。一方、B群は約 $T$ 時刻後の可能な宇宙状態として遠藤博士が絞り込んだ宇宙状態の集合である。これらの入力に対して、B群の各状態に対応する初期の宇宙状態をA群の中から選び出せ。

注意！粒子には色の違いはあるが同色の粒子は同一視される。

### GK理論における粒子の動き

- (1) 時刻は整数で刻まれており、各粒子は毎時刻ごとに格子点の上からその隣の格子点に移動する(止まっている粒子はいない)。なお、格子点 $(x, y)$ の隣とは、 $(x, y \pm 1)$ ,  $(x \pm 1, y)$ ,  $(x \pm 1, y \pm 1)$ の8通りの格子点のこと。また、宇宙はドーナツ状(トーラスという)で、座標 $N$ は座標0に戻ってくるものとする。
- (2) A群ならびにB群の宇宙状態では、1つの格子点の上には高々1つの粒子しか存在しないが、一般には1つの格子点の上に複数の粒子が存在する場合がある。これは直観的には粒子が衝突した状況を表わしている。

<sup>1</sup>一般人向けに「星」と言う表現を使う場合もあります :-)

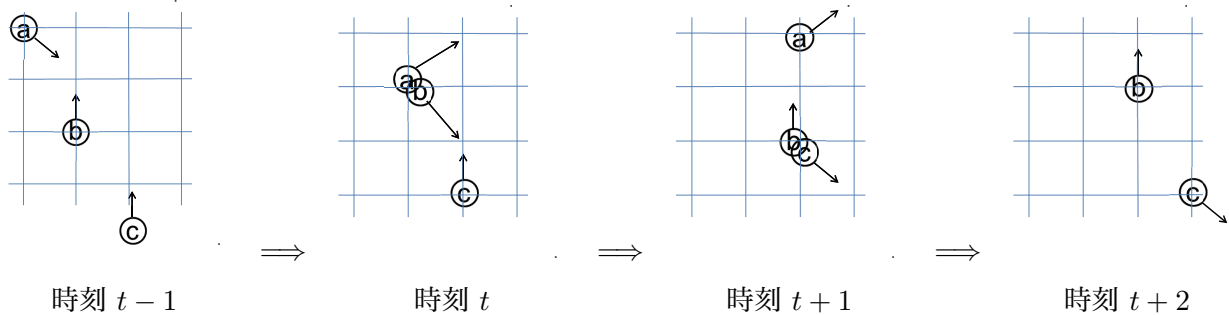


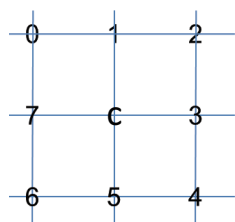
図 1: 粒子の動き

(3) ある時刻  $t$  において、1つの格子点  $(x, y)$  上に粒子が複数いたとする。その場合には、その各粒子は時刻  $t-1$  の時点で  $(x, y)$  の隣の格子点にいるはず。その各粒子の位置が時刻  $t+1$  の各粒子の位置を決める（詳細は付録の表参照）。たとえば、粒子は図 1 のように動く。図における矢印は各時刻での粒子の速度（動く向き）を表わしている。

注意！ この課題を通して、時刻  $t$  の粒子の 向き とは次に進むべき向きのこととする。つまり、時刻  $t$  で衝突していた場合には、衝突後の向きが時刻  $t$  の向きである。

(4) 各粒子に対して、その直前や直後の時刻での粒子の位置は通常、相対位置を表わす 0~7 の数字で表わされる。これを 相対位置番号 と呼ぶ。つまり、図 2 のように、今考えている粒子を中心  $c$  に、その隣の格子点の位置を 0~7 で表わす。

この番号は粒子の速度（向き）を表わすときにも使う。「向き」は次に進む方向なので、次の場所を表わす相対位置番号を向きと見なすのである。



$c$  の座標を  $(x, y)$  とすると

$$0 = (x-1, y+1), 1 = (x, y+1),$$

$$2 = (x+1, y+1), 3 = (x+1, y),$$

$$4 = (x+1, y-1), 5 = (x, y-1),$$

$$6 = (x-1, y-1), 7 = (x-1, y)$$

図 2: 相対位置

## 1. パラメータの定義ならびにデータの入出力方法

入出力は規定のテンプレート `sc13_template.c` を用いること（コンパイルにはヘッダーファイル `transition_rule.h` も必要）。それに従うと標準入出力を使ってデータのやり取りが行われる。以下では、パラメータと用語の解説、入力データと出力データの意味、テンプレートで用いられる変数の意味をそれぞれ述べる。

### 1.1. 用語, パラメータの説明 (パラメータ名はそのまま変数名としてテンプレートでも使用)

$N$  宇宙の格子点座標の上限. 座標  $(x, y)$  の範囲は,  $\{0, \dots, N - 1\}$  とする.

$M$  各色の粒子の個数. したがって, 粒子の総数は  $3M$  個.

$T$  B群の宇宙状態までの経過時刻の概数.

注意! 実際の経過時刻は各宇宙状態ごとに  $(0.5T, 1.5T)$  の範囲で異なる.

$K$  A群の宇宙状態数.

$L$  B群の宇宙状態数.

### 1.2. 入出力と関連の変数

入力はテンプレート中の変更不可の部分で行われ, 以下の変数に必要なデータが得られる.

入出力用構造体

```
typedef struct Star_{ // 粒子の状態情報のための構造体
    int x;           // x 座標
    int y;           // y 座標
    int dir;         // 速度=向き=次時刻の位置 (相対位置番号)
    int color;       // 色
} Star;

Star A[ MAXK ][ 3 * MAXM ]; // A群を表すデータ
Star B[ MAXL ][ 3 * MAXM ]; // B群を表すデータ
```

つまり, 各  $i, 0 \leq i \leq K - 1$ , に対して, 配列  $A[i]$  にはA群  $i$  番目の宇宙状態の情報が格納される (なお,  $i$  をA群の 宇宙状態インデックス と呼ぶことにする). より具体的には,  $A[i][0 \sim M - 1]$  には, その宇宙状態の0色粒子の状態 (位置と向き) が,  $A[i][M \sim 2M - 1]$  には1色粒子の状態が,  $A[i][2M \sim 3M - 1]$  には2色粒子の状態が, 各々与えられる.

なお, 同色の粒子間には差が無い. したがって, 1つの宇宙状態を表わす配列  $A[i]$  内 (同様に配列  $B[j]$  内) の同色の粒子の順番には意味がない.

出力もテンプレート中の変更不可の部分で行われ, 以下の変数に得られた答えが出力される.

```
int answer[ MAXL ]; // 出力用
// answer[ i ] = B[ i ] の過去であるA群の宇宙状態インデックス
```

答えは単純に, B群の各宇宙状態インデックスの順で, その状態の過去に当たるA群の宇宙状態インデックスを空白区切りで出力する.

(補足: 入力の仕様) 以上のように, 入出力はテンプレートで行うので, 具体的に入力としてデータを与える方法については, とくに詳細を知る必要はないかもしれないが, 参考のため以下に簡単に紹介しておく.

- ・ 1 行目に N, M, T, K, L が空白区切りで与えられる.
- ・ それ以降の K\*3\*M 行に A 群を表すデータが与えられる. B 群も同様.
- ・ 各 3\*M 行で一つの宇宙状態を与える. 各宇宙状態の各行では, 各粒子の状態を与える: 最初の M 行で色 0 の粒子, 次の M 行で色 1 の粒子, 最後の M 行で色 2 の粒子の位置と速度 (向き=相対位置番号) を与える.
- ・ 各粒子の状態は x y direction の形で与える.

### 1.3. テンプレートにおけるその他の主要変数

最も重要なのは衝突後の速度 (向き) の変化「遷移ルール」を表わす表だろう. これは transition\_rule.h というヘッダーファイルで, 次のように与えられる.

```
struct TransitionRule{ // 遷移ルールを表現する構造体
    int id;           // 遷移ルール id. 0 から始まる. 多分使うことはない
    int length;      // 衝突に関わる粒子数. length 番目以降の pattern, next は未定義
    int pattern[8]; // 衝突の直前時刻の各粒子の相対位置番号
    int next[8];    // 衝突の直後時刻の各粒子の相対位置番号=衝突時刻の粒子の向き
};

int TRANSITION_RULES_NUM = 255; // 遷移ルールの総数
TransitionRule transition_rules[ 255 ]; // 遷移ルールの表
```

その他に以下のような変数も定数として定義されている.

```
#define MAXN 500 // 空間の最大サイズ
#define MAXM 4000 // 1色あたりの粒子の最大数 ( 粒子の最大数は 3 * MAXM )
#define MAXT 5000 // 最大予測経過時間
#define MAXK 250 // A群の最大宇宙状態数
#define MAXL 100 // B群の最大宇宙状態数
```

## 2. パラメータの範囲, 審査方法

審査では以下のようなパラメータを用いる予定. プログラミングの際に用いるテスト用データも, このサイズのものを用意しておくが, 最初のうちは小さいサイズで実験することをお勧めする.

```
N = 500    M = 4000    T = 5000
```

なお, 初期宇宙状態の形はテスト用データとは異なるものを用いる予定.

### 2.1. 審査方法について (あくまで予定です)

審査は, 提出されたプログラムに対して, 第一次選抜を行いチーム数を少し絞った後で, 最終審査を行う. 第一次選抜では K = 25, L = 10 程度の宇宙状態数のデータを用いる. 一方, 最終審査では K = 250, L = 100 程度の宇宙状態数のデータを用いる.

### 3. プログラミングの注意

- (1) プログラミング言語は C 言語. プログラムは C 言語規格 (ANSI C や C99 など) に準拠する C 言語で書くこと. GCC や Visual C++ などに特有の仕様は, スーパーコンピュータ TSUBAME2.0 では使えない可能性が高い. PC でコンパイルできたものが TSUBAME 上でコンパイルできるとは限らないことに注意.
- (2) CPU 並列化はしない. 実行は TSUBAME 1 ノード上で行われる. その上には複数コア, 複数 GPU が割り当てられているが, プログラム上で使ってよいのは 1CPU コア, 1GPU のみとする. (CPU の) pthread や MPI による並列化は禁止とする. (厳密には, CUDA が内部的に pthread を用いる可能性があるが, それは違反とはしない.)
- (3) 今回は GPU 上の並列化をうまく利用するのが鍵となる. GPU に関する配布資料や講習などを参考にして欲しい.
- (4) データの読み込みと解答の出力には審査委員会で用意したテンプレート `sc13_template.c` を使う. テンプレートの限られた部分のみを書き換えてプログラミングすること.
- (5) 使用可能メモリサイズは CPU 側で約 48GB. その範囲内であれば, ひとつの配列として取れるサイズに上限はない. いっぽう, GPU 側での使用可能メモリサイズは約 2.5GB. これを超えるデータを扱う場合には, GPU 上の同じメモリ領域を使いまわす等の工夫が必要となる.
- (6) その他, 補足の説明, コンテスト中の質問への回答, 細かな注意などは, コンテスト Wiki に掲載するので参照すること.

付録：衝突に関する遷移ルール

id	leng	pattern	next
0	1	0	4
1	1	1	5
2	1	2	6
3	1	3	7
4	1	4	0
5	1	5	1
6	1	6	2
7	1	7	3
8	2	01	54
9	2	02	64
10	2	03	74
11	2	04	04
12	2	05	14
13	2	06	24
14	2	07	34
15	2	12	65
16	2	13	75
17	2	14	05
18	2	15	15
19	2	16	25
20	2	17	35
21	2	23	76
22	2	24	06
23	2	25	16
24	2	26	26
25	2	27	36
26	2	34	07
27	2	35	17
28	2	36	27
29	2	37	37
30	2	45	10
31	2	46	20
32	2	47	30
33	2	56	21
34	2	57	31
35	2	67	32
36	3	012	654
37	3	013	754
38	3	014	054
39	3	015	415
40	3	016	425
41	3	017	435
42	3	023	764
43	3	024	064
44	3	025	025
45	3	026	426
46	3	027	436
47	3	034	074
48	3	035	417
49	3	036	036
50	3	037	437
51	3	045	041
52	3	046	042
53	3	047	043
54	3	056	142
55	3	057	143
56	3	067	243
57	3	123	765
58	3	124	065
59	3	125	165
60	3	126	526
61	3	127	536
62	3	134	075
63	3	135	175

id	leng	pattern	next
64	3	136	752
65	3	137	537
66	3	145	105
67	3	146	146
68	3	147	305
69	3	156	152
70	3	157	153
71	3	167	253
72	3	234	076
73	3	235	176
74	3	236	276
75	3	237	637
76	3	245	106
77	3	246	206
78	3	247	247
79	3	256	216
80	3	257	631
81	3	267	263
82	3	345	107
83	3	346	207
84	3	347	307
85	3	356	217
86	3	357	317
87	3	367	327
88	3	456	210
89	3	457	310
90	3	467	320
91	3	567	321
92	4	0123	7564
93	4	0124	0564
94	4	0125	4165
95	4	0126	4526
96	4	0127	4536
97	4	0134	0574
98	4	0135	7145
99	4	0136	4275
100	4	0137	4537
101	4	0145	0145
102	4	0146	0542
103	4	0147	0543
104	4	0156	4152
105	4	0157	4153
106	4	0167	4253
107	4	0234	0674
108	4	0235	7641
109	4	0236	4276
110	4	0237	4637
111	4	0245	0641
112	4	0246	0246
113	4	0247	0643
114	4	0256	4216
115	4	0257	4316
116	4	0267	4263
117	4	0345	0741
118	4	0346	0742
119	4	0347	0347
120	4	0356	1742
121	4	0357	1347
122	4	0367	4327
123	4	0456	0412
124	4	0457	0413
125	4	0467	0423
126	4	0567	1423
127	4	1234	0675

id	leng	pattern	next
128	4	1235	1675
129	4	1236	5276
130	4	1237	5637
131	4	1245	1605
132	4	1246	5206
133	4	1247	0653
134	4	1256	1256
135	4	1257	1356
136	4	1267	5263
137	4	1345	1705
138	4	1346	5207
139	4	1347	0357
140	4	1356	1257
141	4	1357	1357
142	4	1367	2357
143	4	1456	1052
144	4	1457	1350
145	4	1467	5320
146	4	1567	1523
147	4	2345	1706
148	4	2346	2706
149	4	2347	6307
150	4	2356	2716
151	4	2357	1367
152	4	2367	2367
153	4	2456	2016
154	4	2457	1063
155	4	2467	2063
156	4	2567	2163
157	4	3456	2017
158	4	3457	3017
159	4	3467	3027
160	4	3567	3127
161	4	4567	3120
162	5	01234	05674
163	5	01235	41675
164	5	01236	45276
165	5	01237	45637
166	5	01245	01645
167	5	01246	05246
168	5	01247	05643
169	5	01256	41256
170	5	01257	41653
171	5	01267	45263
172	5	01345	01745
173	5	01346	05742
174	5	01347	05347
175	5	01356	41752
176	5	01357	41357
177	5	01367	45327
178	5	01456	01452
179	5	01457	01453
180	5	01467	05423
181	5	01567	41523
182	5	02345	06741
183	5	02346	02746
184	5	02347	06347
185	5	02356	42716
186	5	02357	46317
187	5	02367	42367
188	5	02456	02416
189	5	02457	06413
190	5	02467	02463
191	5	02567	42163

id	leng	pattern	next
192	5	03456	07412
193	5	03457	03417
194	5	03467	03427
195	5	03567	43127
196	5	04567	04123
197	5	12345	16705
198	5	12346	52706
199	5	12347	56307
200	5	12356	12756
201	5	12357	16357
202	5	12367	52367
203	5	12456	12056
204	5	12457	16053
205	5	12467	52063
206	5	12567	12563
207	5	13456	17052
208	5	13457	13057
209	5	13467	53027
210	5	13567	13527
211	5	14567	10523
212	5	23456	27016
213	5	23457	63017
214	5	23467	23067
215	5	23567	23167
216	5	24567	20163
217	5	34567	30127
218	6	012345	016745
219	6	012346	052746
220	6	012347	056347
221	6	012356	412756
222	6	012357	416357
223	6	012367	452367
224	6	012456	012456
225	6	012457	016453
226	6	012467	052463
227	6	012567	412563
228	6	013456	017452
229	6	013457	013457
230	6	013467	053427
231	6	013567	413527
232	6	014567	014523
233	6	023456	027416
234	6	023457	063417
235	6	023467	023467
236	6	023567	423167
237	6	024567	024163
238	6	034567	034127
239	6	123456	127056
240	6	123457	163057
241	6	123467	523067
242	6	123567	123567
243	6	124567	120563
244	6	134567	130527
245	6	234567	230167
246	7	0123456	0127456
247	7	0123457	0163457
248	7	0123467	0523467
249	7	0123567	4123567
250	7	0124567	0124563
251	7	0134567	0134527
252	7	0234567	0234167
253	7	1234567	1230567
254	8	01234567	01234567