



ODSS
(Ordered-Disordered-Solid-Solution)
Ver.1. -binar

Программа расчета неупорядоченных сверхячеек для
моделирования твердых растворов замещения

Р.З.Деянов, Н.Н.Еремин, В.С.Урусов
urusov@geol.msu.ru

© Кафедра кристаллографии и кристаллохимии Геологического факультета
Московского Государственного Университета
© Р.З.Деянов, Н.Н.Ерёмин, В.С.Урусов

Москва, 2006-2007.

Содержание

1. Постановка задачи	3
2. Ввод информации для счета	6
3. Вывод информации	8
4. Примеры расчетов	13
5. Текст программы	16

1. Постановка задачи

Как известно, последние десятилетия знаменуются качественным скачком быстродействия компьютеров, что позволяет бурно прогрессировать те области знания, в которых вычислительный эксперимент позволяет дополнить или в ряде случаев подменить физический. Поэтому в последние 10 лет в энергетической кристаллохимии появилась практическая возможность перейти от моделирования идеальных кристаллов к структурно несовершенным реальным кристаллам, причем как полуэмпирическими методами парных потенциалов, так и методами из первых принципов. В реальном разупорядоченном кристалле распределение различных сортов атомов по кристаллографически эквивалентным позициям в общем случае подчиняется статистическим законам. В связи с этим, вспыхнувший интерес к моделированию твердых растворов, столкнулся с проблемой правильного распределения атомов в сверхячейке для имитации неупорядоченности. Очевидно, что чем ближе теоретическая модель воспроизводит неупорядоченное распределение атомов по позициям, тем вероятнее добиться согласия предсказанных свойств твердых растворов с экспериментальными результатами. Эта проблема с той или иной степенью успеха решается различными приемами.

В бурно развивающихся методах Монте-Карло и *ab-initio* существуют свои приемы имитации неупорядоченности в твердом растворе, например *Cluster Variation Approach*. Суть приема состоит в том, что для моделирования используется небольшой фрагмент кристаллической структуры (кластер) с характерной конфигурацией, а интересующие свойства большой сверхячейки ячейки определяются из частоты встречаемости различных кластеров по нескольким координационным сферам. Такие подходы позволяют достаточно хорошо определить термодинамические свойства твердого раствора, но, не способны воспроизвести локальную структуру твердого раствора с достаточной достоверностью и оставляют проблему учета **правильного** распределения атомов по ячейке в силе.

В рамках полуэмпирического моделирования твердых растворов до настоящего времени использовались несколько подходов, основанных на анализе заполнения второй координационной сферы центрального атома (анализ чисел связей катион-катион в случае изоморфизма в катионной подрешетке). Эти подходы изложены в работах [Урусов В.С., Петрова Т.Г., Ерёмин Н.Н. / Компьютерное моделирование свойств твердых растворов MgO-CaO с учетом ближнего порядка./ Доклады АН, 2 (387) стр. 1-5, 2002] и [Урусов В.С., Ерёмин Н.Н., Петрова Т.Г., Талис Р.А. / Структуры и свойств бинарных оксидных

твердых растворов со структурой корунда. /4-ая Национальная кристаллохимическая конференция, Черногоровка. Тезисы докладов, стр. 238, 2006]. Однако, одни из них, требуют знания максимально упорядоченной сверхструктуры для каждого конкретного состава твердого раствора, что не всегда возможно и, кроме того, используют прогнозные (экстраполяционные) величины, что снижает их достоверность. Другие подходы, используют в качестве критерия порядка-беспорядка в сверхячейке величины, которые являются необходимыми, но недостаточными для объявления конфигурации разупорядоченной. Так, используемый критерий q (равный отношению числа разнородных пар А-Б к общему числу пар катионов во второй координационной сфере и усредненный для всех катионов сверхячейки структуры) для разупорядоченного твердого раствора равен удвоенному произведению концентраций чистых компонентов, но из его равенству этой величине не может быть сделано никаких выводов о степени упорядоченности полученной конфигурации.

В настоящей программе предлагается более совершенный алгоритм, лишенный недостатков вышеописанных подходов. В качестве критерия степени неупорядоченности конфигурации предлагается использовать величину суммы квадратов отклонений числа разнородных связей во второй координационной сфере в случайной конфигурации от статистической теоретической гистограммы (критерий согласия Пирсона). Множество случайных конфигураций анализируется на величину отклонения от идеальной разупорядоченной гистограммы частоты встречаемости разнородных вторых соседей для каждого состава (χ^2 диаграммы). Идеальную разупорядоченную конфигурацию находят методами комбинаторики с поправкой на относительную концентрацию каждого катионного компонента. При этом результирующая кривая получается как суперпозиция двух вкладов от атомов сорта А и Б.

К достоинствам предложенной программы относятся:

- 1) Возможность в рамках *конечной ячейки максимально приблизиться к* статистически неупорядоченному *распределению в бесконечном кристалле*;
- 2) Возможность в дальнейших расчетах термодинамических свойств обойтись одной «оптимальной» конфигурацией, что немаловажно для минимизации расчетного времени.
- 3) Удобная выдача для последующего использования в программе GULP [The General Utility Lattice Program, J.D. Gale and A.L. Rohl, *Mol. Simul.*, **29**, 291 (2003)];
- 4) Наличие *однозначной количественной* оценки качества конфигурации.

Очевидно, что в рамках конечной сверхячейки не существует абсолютно разупорядоченной конфигурации с $\chi^2 = 0$ (из-за дробных значений вероятностей). Также понятно, что простым перебором решить задачу нахождения наилучшей конфигурации невозможно, несмотря даже на то, что число *топологически* различных конфигураций существенно меньше *общего* числа вариантов, рассчитанных методами математической комбинаторики, что связано с собственной симметрией сверхячейки (см. рис. 1). Видно, что вариант 2-17-22-27 получается из варианта 5-7-10-26 вращением вокруг оси третьего порядка. В связи с этим оба различных (с математической точки зрения) варианта абсолютно равнозначны по взаимному расположению синих и зеленых точек (атомов) и обеспечивают одинаковую топологию связей разного типа.

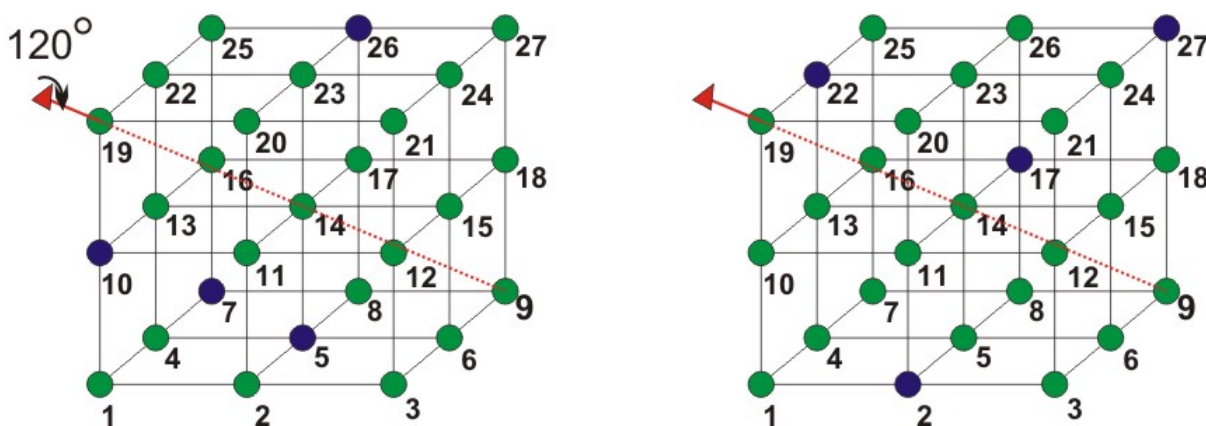


Рис. 1. Два топологически равнозначных варианта размещения 4 синих шаров в кубической сверхячейке 3*3*3 из 27 позиций.

Максимально симметричный случай кубической сверхячейки позволяет уменьшить число топологически неэквивалентных вариантов (по сравнению с общим числом вариантов) в 48 раз, что с одной стороны неплохо, но при оценке времени перебора всех вариантов этот понижающий коэффициент оказывается, к сожалению, явно недостаточным. Так, общее число вариантов размещения 32 синих шаров по ячейке из 64 атомов равно $\frac{64!}{32!(64-32!)}$ = 1,83262·10¹⁸, а число топологически разных вариантов в кубической ячейке такого размера будет равно 3,81797·10¹⁶. Однако, даже учитывая быстроедействие современного компьютера можно рассчитать, что время перебора всех уникальных вариантов составляет более тысячи лет даже для такой маленькой сверхячейки.

Пробные расчеты по программе показали, что за первые минуты счета удастся снизить расхождение до 2%, а если считать часы, то и до 1%. При этом различия в энергии конфигураций с 2% и 1% настолько незначительны на больших ячейках, что можно

рекомендовать величину $\chi^2 = 2\%$ как *качественную имитацию* разупорядоченной бесконечной конфигурации.

Программа написана на алгоритмическом языке FORTRAN и ориентирована на программный комплекс GULP.

2. Ввод информации для счета

Для запуска программы необходимо создать 2 файла:

- файл с названием input, содержащий входную информацию задачи;
- файл со структурной информацией (название файла указывается в первой строке файла input).

Пример файла задания входной информации input.

Al2O3 - имя входного файла(параметры ячейки, координаты атомов)

Al - имя атома/группы

12 - кол-во атомов для этой группы

O - имя атома/группы

18 - кол-во атомов для этой группы

4 4 1 – трансляции (соотв. по осям X,Y,Z)

32 - кол-во заменяемых атомов

3.7 - радиус сферы захвата (в ангстремах)

St - имя атома для замены

1 - ограничение по времени в мин. для итераций сл.вбрасываний *

0 - критерий по хи-квадрат: 1-на возрастание, 0 - на убывание

0 - критерий по затравке с.ч.: 0 - затравка от процессора, 1- от встроенной процедуры **

5.0 - граница хи-квадрат, с кот. начинается выдача данных для GULP ***

1 - печать координат всех размноженных атомов (0 - нет, 1 - да)

1 - печать координат всех катионов (0 - нет, 1 - да)

1 - печать координат атомов попавших в сферу (0 - нет, 1 - да)

1 - печать минимальных значений хи-квадрат (0 - нет, 1 - да)

1 - печать распределения вероятности (0 - нет, 1 - да)

1 - печать координат конфигурации катионов (0 - нет, 1 - да)

1 - печать координат итоговой конфигурации (0 - нет, 1 - да)

1 - печать данных для GULP (0 - нет, 1 - да)

* - рекомендуется 30-60 мин.

** - рекомендуется затравка от процессора;

***- если критерий по хи-квадрат=1, то в GULP выдаются данные для хи-квадрат больше границы хи-квадрат; если критерий по хи-квадрат=0, то в GULP выдаются данные для хи-квадрат меньше границы хи-квадрат

Формат задания данных файла Al2O3.

```
comp opti prop
cell
4.7602 4.7602 12.9933 90.000000 90.000000 120.0
fractional
Al 0.0000000000 0. 0.3522 +1.89
Al 0. 0. 0.6478 +1.89
Al 0. 0. 0.1478 +1.89
Al 0. 0. 0.8522 +1.89
Al 0.3333333333 0.6666666666 0.018900 +1.89
Al 0.3333333333 0.6666666666 0.314500 +1.89
Al 0.3333333333 0.6666666666 0.814500 +1.89
Al 0.3333333333 0.6666666666 0.518900 +1.89
Al 0.6666666666 0.3333333333 0.685500 +1.89
Al 0.6666666666 0.3333333333 0.981100 +1.89
Al 0.6666666666 0.3333333333 0.481100 +1.89
Al 0.6666666666 0.3333333333 0.185500 +1.89
O 0.3061 0. 0.25 -1.26
O 0. 0.3061 0.25 -1.26
O 0.6939 0.6939 0.25 -1.26
O 0.6939 0.0000 0.75 -1.26
O 0.0000 0.6939 0.75 -1.26
O 0.3061 0.3061 0.75 -1.26
O 0.6394 0.6666666666 0.9166666 -1.26
O 0.333333 0.972800 0.9166666 -1.26
O 0.027200 0.360600 0.9166666 -1.26
O 0.027200 0.6666666666 0.4167 -1.26
O 0.3333333333 0.300600 0.4167 -1.26
O 0.639400 0.972800 0.4167 -1.26
O 0.972800 0.3333333333 0.5833 -1.26
O 0.6666666666 0.6394 0.5833 -1.26
O 0.360600 0.0272 0.5833 -1.26
O 0.360600 0.3333 0.0833 -1.26
O 0.6666666666 0.0272 0.0833 -1.26
O 0.972800 0.6394 0.0833 -1.26
space
1
buck
Al core O core 3494.019818 0.228 0.0 0.0 12.0
buck
O core O core 1437.028831 0.313123 0.048640 0.0 12
buck
Al core Al core 0.047457 0.064357 0.0 0.0 12.0 0 0 0
morse
Al core O core 6.138115 2.119270 0.684410 0.0 12.0
cutd 2.3
```

Формат задания координат: 10 знаков после запятой, нули можно не ставить.
Координаты даются относительно векторов ячейки **a,b,c**,

3. Вывод информации

В результате работы программы создаются 2 файла:

- файл с названием output, содержащий выходную информацию задачи;
- файл с названием data_for_gulp с координатами атомов выбранных конфигураций для счета по программе GULP.

Пример файла выходной информации output . (фрагмент, полный листинг 140 страниц)

Start time = 16 h. 32 min.
Total time limited = 1 min.
End time = 16 h. 33 min.

Координаты исходных атомов в долях маленькой ячейки (векторов a,b,c) :

1	Al	0.0000000000	0.0000000000	0.3522000000	1.89
2	Al	0.0000000000	0.0000000000	0.6478000000	1.89
3	Al	0.0000000000	0.0000000000	0.1478000000	1.89
4	Al	0.0000000000	0.0000000000	0.8522000000	1.89
5	Al	0.3333333333	0.6666666666	0.0189000000	1.89
6	Al	0.3333333333	0.6666666666	0.3145000000	1.89
7	Al	0.3333333333	0.6666666666	0.8145000000	1.89
8	Al	0.3333333333	0.6666666666	0.5189000000	1.89
9	Al	0.6666666666	0.3333333333	0.6855000000	1.89
10	Al	0.6666666666	0.3333333333	0.9811000000	1.89
11	Al	0.6666666666	0.3333333333	0.4811000000	1.89
12	Al	0.6666666666	0.3333333333	0.1855000000	1.89
1	O	0.3061000000	0.0000000000	0.2500000000	-1.26
2	O	0.0000000000	0.3061000000	0.2500000000	-1.26
3	O	0.6939000000	0.6939000000	0.2500000000	-1.26
4	O	0.6939000000	0.0000000000	0.7500000000	-1.26
5	O	0.0000000000	0.6939000000	0.7500000000	-1.26
6	O	0.3061000000	0.3061000000	0.7500000000	-1.26
7	O	0.6394000000	0.6666666666	0.9166666000	-1.26
8	O	0.3333330000	0.9728000000	0.9166666000	-1.26
9	O	0.0272000000	0.3606000000	0.9166666000	-1.26
10	O	0.0272000000	0.6666666666	0.4167000000	-1.26
11	O	0.3333333333	0.3006000000	0.4167000000	-1.26
12	O	0.6394000000	0.9728000000	0.4167000000	-1.26
13	O	0.9728000000	0.3333333333	0.5833000000	-1.26
14	O	0.6666666666	0.6394000000	0.5833000000	-1.26
15	O	0.3606000000	0.0272000000	0.5833000000	-1.26
16	O	0.3606000000	0.3333000000	0.0833000000	-1.26
17	O	0.6666666666	0.0272000000	0.0833000000	-1.26
18	O	0.9728000000	0.6394000000	0.0833000000	-1.26

Координаты размноженных атомов в трансляциях маленькой ячейки (векторов a,b,c) :

```
1 Al1 0.000 0.000 0.352
*****
192 Al192 3.667 3.333 0.185
```


1 O1 0.306 0.000 0.250

 288 O288 3.973 3.639 0.083

Координаты атомов катиона

по.	номер размножающего атома	X	Y	Z
1	1	0.0000	0.0000	0.3522
1296	156	4.6667	-0.6667	-0.8145

Текущий атом и атомы из сферы:

Номер атома = 1 Координаты: 0.000 0.000 0.352

по.	номер атома из сферы	координаты			расстояние
1	3	0.000	0.000	0.148	2.66
2	6	0.333	0.667	0.315	2.79
3	8	0.333	0.667	0.519	3.50
4	11	0.667	0.333	0.481	3.22
5	12	0.667	0.333	0.185	3.50
6	47	-0.333	0.333	0.481	3.22
7	48	-0.333	0.333	0.185	3.50
8	150	0.333	-0.333	0.315	2.79
9	152	0.333	-0.333	0.519	3.50
10	186	-0.667	-0.333	0.315	2.79
11	188	-0.667	-0.333	0.519	3.50
12	191	-0.333	-0.667	0.481	3.22
13	192	-0.333	-0.667	0.185	3.50

Текущий атом и атомы из сферы:

Номер атома = 192 Координаты: 3.667 3.333 0.185

по.	номер атома из сферы	координаты			расстояние
1	137	3.333	2.667	0.019	3.50
2	138	3.333	2.667	0.315	3.22
3	181	3.000	3.000	0.352	3.50
4	183	3.000	3.000	0.148	2.79
5	185	3.333	3.667	0.019	3.50
6	186	3.333	3.667	0.315	3.22
7	190	3.667	3.333	-0.019	2.66
8	145	4.000	3.000	0.352	3.50
9	147	4.000	3.000	0.148	2.79
10	149	4.333	3.667	0.019	3.50
11	150	4.333	3.667	0.315	3.22
12	1	4.000	4.000	0.352	3.50
13	3	4.000	4.000	0.148	2.79

Total time limited = 1 min.

no.iter	Rf	Q	Current spent time (min.)
1	15.978	0.269	0.0013
2	9.677	0.279	0.0013
8	8.283	0.274	0.0013
51	5.862	0.280	0.0016
94	5.287	0.279	0.0016
The estimate of max. iter. = 0.35 mln.			
1052	5.215	0.282	0.0044
2585	4.607	0.279	0.0096
8067	4.514	0.284	0.0255
24179	3.657	0.276	0.0740
28146	3.651	0.276	0.0857
28366	3.151	0.276	0.0865
108935	3.121	0.277	0.3245
269504	3.018	0.277	0.7987

Проведено 337900 случайных вбрасываний.
(выход из цикла сл. вбрасываний по ограничению времени равного = 1.0 мин.)

Результаты 337899 случайных вбрасываний :
номер наилучшего вбрасывания = 269504
затраченное время : 47.84 сек.
Q = 0.277
R-factor = 3.018 %

Итоговое распределение вероятности (m - кол-во связей) :

P (экспер.)	P (теоретич.)	1000*delta
P(m= 0)= 0.072917	0.077887	-4.970
P(m= 1)= 0.203125	0.202505	0.620
P(m= 2)= 0.244792	0.243006	1.786
P(m= 3)= 0.182292	0.178205	4.087
P(m= 4)= 0.093750	0.089108	4.642
P(m= 5)= 0.031250	0.032128	-0.878
P(m= 6)= 0.005208	0.008896	-3.688
P(m= 7)= 0.000000	0.003422	-3.422
P(m= 8)= 0.010417	0.006672	3.745
P(m= 9)= 0.020833	0.017849	2.984
P(m=10)= 0.031250	0.035643	-4.393
P(m=11)= 0.046875	0.048601	-1.726
P(m=12)= 0.041667	0.040501	1.166
P(m=13)= 0.015625	0.015577	0.048

Координаты атомов наилучшего случайного вбрасывания:

1	Cr119	1.67	2.33	0.48
2	Cr131	2.67	2.33	0.48
3	Cr22	1.67	0.33	0.98
4	Cr189	3.67	3.33	0.69

5	Cr97	0.00	2.00	0.35
6	Cr158	1.00	3.00	0.65
7	Cr56	0.33	1.67	0.52
8	Cr166	1.67	3.33	0.98
9	Cr125	2.33	2.67	0.02
10	Cr101	0.33	2.67	0.02
11	Cr9	0.67	0.33	0.69
12	Cr40	3.00	0.00	0.85
13	Cr53	0.33	1.67	0.02
14	Cr44	3.33	0.67	0.52
15	Cr31	2.33	0.67	0.81
16	Cr58	0.67	1.33	0.98
17	Cr1	0.00	0.00	0.35
18	Cr164	1.33	3.67	0.52
19	Cr106	0.67	2.33	0.98
20	Cr163	1.33	3.67	0.81
21	Cr124	2.00	2.00	0.85
22	Cr168	1.67	3.33	0.19
23	Cr55	0.33	1.67	0.81
24	Cr150	0.33	3.67	0.31
25	Cr63	1.00	1.00	0.15
26	Cr171	2.00	3.00	0.15
27	Cr96	3.67	1.33	0.19
28	Cr35	2.67	0.33	0.48
29	Cr57	0.67	1.33	0.69
30	Cr110	1.00	2.00	0.65
31	Cr111	1.00	2.00	0.15
32	Cr33	2.67	0.33	0.69

Итоговая конфигурация после случайных вбрасываний.

no.	Атом	A/B	атомы из окружения								
1	Cr1	12	Al3	Al6	Al8	Al11	Al12	Al47	Al48	Cr150	Al152
	Al186	Al188	Al191	Al192							

192	Al192	2	Al137	Al138	Al181	Al183	Al185	Al186	Al190	Al145	
	Al147	Al149	Cr150	Cr1	Al3						

Затраты процессорного времени.

Задача	Время (сек.)
Расчет координат всех атомов по трансляциям	0.06
Расчет итераций	59.92
Суммарное время счета	59.98

Суммарное время счета(сек): 59.98 сек.

Общее время вычислений: 0 час. 0 мин. 59.98 сек.

Пример файла выходной информации data_for_gulp.
(фрагмент, полный листинг 66 страниц)

comp opti prop phon nofreq

cell

19.041 19.041 12.993 90.00 90.00 120.00

fractional

Rfact = 4.607 % Q = 0.279

Rfact = 3.018 % Q = 0.277

Cr1 0.0000000000 0.0000000000 0.3522000000 1.89

O288 0.9932000000 0.9098500000 0.0833000000 -1.26

space

1

buck

Al core O core 3494.019818 0.228 0.0 0.0 12.0

buck

O core O core 1437.028831 0.313123 0.048640 0.0 12

buck

Al core Al core 0.047457 0.064357 0.0 0.0 12.0 0 0 0

morse

Al core O core 6.138115 2.119270 0.684410 0.0 12.0

4. Примеры расчетов

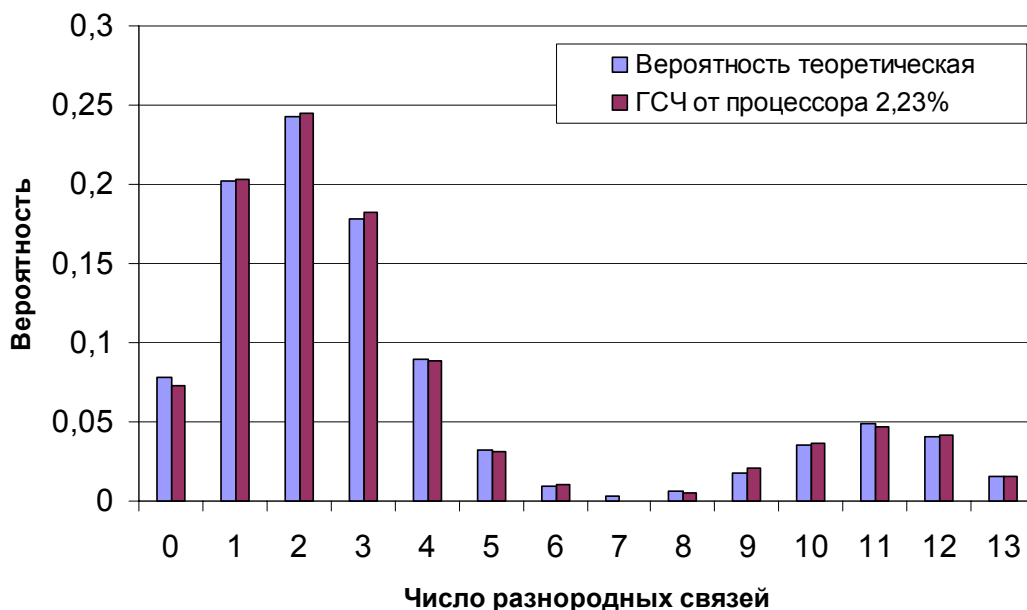
Пример 1

Система $Al_2O_3-Cr_2O_3$

13 соседей во второй координационной сфере
4-4-1 ячейка состав 1:5 (замена 32 из 192 атомов)

Генератор случайного числа - от процессора
(P4-3000, Windows XP), время счета – 60 минут.

№	χ^2	q	№ итерации	Время счета, сек
1	15,978	0,269	1	0,41
2	9,677	0,279	2	0,42
3	8,283	0,274	8	0,46
4	5,862	0,28	51	0,49
5	5,287	0,279	94	0,52
6	5,215	0,282	1052	0,70
7	4,607	0,279	2585	0,98
8	4,514	0,284	8067	2,35
9	3,657	0,276	24179	5,41
10	3,651	0,276	28146	6,17
11	3,151	0,276	28366	6,25
12	3,121	0,277	108935	20,09
13	3,018	0,277	269504	46,11
14	2,826	0,279	763920	125,67
15	2,722	0,276	1315735	214,36
16	2,504	0,277	1694145	275,90
17	2,425	0,28	6416244	1038,77
18	2,419	0,277	10068926	1627,47
19	2,232	0,279	15771529	2555,15



Пример 2

Система CaO-MgO

12 соседей во второй координационной сфере
4-4-4 ячейка состав 1:1 (замена 128 из 256 атомов)

Генератор случайного числа - от процессора
(P4-3000, Windows XP), время счета – 20 минут.

№	χ^2	q	№ итерации	Время счета, сек
1	18,525	0,49	1	0,44
2	11,429	0,507	3	0,47
3	10,21	0,512	6	0,50
4	7,223	0,495	8	0,52
5	6,304	0,508	37	0,55
6	6,286	0,507	111	0,59
7	6,075	0,491	167	0,64
8	5,957	0,496	365	0,72
9	5,887	0,5	582	0,80
10	5,054	0,51	810	0,88
11	4,802	0,491	827	0,91
12	4,74	0,499	888	0,95
13	3,719	0,504	1091	1,03
14	1,816	0,499	8640	3,05
15	1,645	0,497	562037	149,28
16	1,572	0,497	625343	165,94



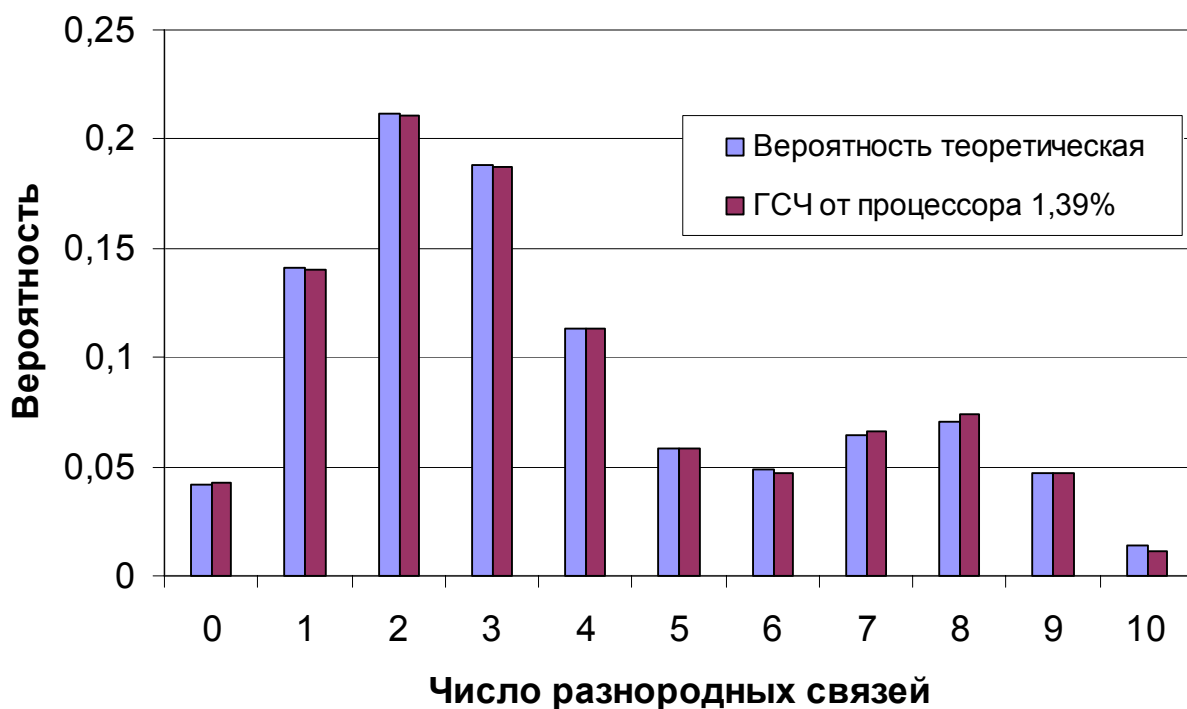
Пример 3

Система $\text{TiO}_2\text{-SnO}_2$

10 соседей во второй координационной сфере
4-4-8 ячейка состав 1:3 (замена 64 из 256 атомов)

Генератор случайного числа - от процессора
(P4-3000, Windows XP), время счета – 10 минут.

№	χ^2	q	№ итерации	Время счета, сек
1	20,816	0,377	1	0,41
2	18,414	0,373	2	0,44
3	11,306	0,378	3	0,49
4	9,169	0,377	9	0,52
5	5,778	0,378	47	0,56
6	5,24	0,373	352	0,65
7	5,207	0,37	1308	0,86
8	5,064	0,37	1401	0,92
9	4,671	0,378	2462	1,16
10	4,648	0,372	2993	1,30
11	4,25	0,38	4222	1,56
12	3,15	0,38	4759	1,69
13	2,987	0,373	55343	11,66
14	2,935	0,378	121878	24,55
15	2,878	0,378	149743	29,89
16	2,661	0,38	339685	65,31
17	2,342	0,375	809076	154,33
18	2,232	0,381	863270	164,41
19	1,386	0,375	1103966	208,94



5. Текст программы

```
program main
use commonmod

! *****
  call open_read
! N_rndm_all = m_X * m_Y * m_Z * Num_atomov_in_group(1) - кол-во размнож. атомов 1-
го сорта
! *****
! подготовка рабочих массивов:

  call XYZ0_in_XYZ

  Name_atom0 = Name_atom

! результат:
! N_all - кол-во всех атомов
! в XYZ(N_all,*) - координаты размноженных атомов всех сортов
! всего точек атомов 1-го сорта внутри и в области: k_work_rndm_all
! work_XYZ_rndm(k_work_rndm_all,3) - координаты атомов 1-го сорта внутри
! и в оболочке области
! index_XYZ_rndm(k_work_rndm_all) - номера этих атомов
!
! Number_atom(k_work_rndm_all,40) - в J-й строке находятся номера атомов,
! кот. захватываются сферой J-го атома
! *****
! Name_atom0 = Name_atom

! *****
! if(output_1==1)call output1
! *****

! вычисляем теоретическое распределение вероятностей
! N_rad - д.б. уже определен ( из XYZ0_in_XYZ)
  call calculation_P_theor
  call cpu_time(time_coord)
! *****
  write(2,'(//5x,"_____
"//)')
  write(2,'(//7x,"Total time limited = ",5x,i6," min.")')int(end_time_in_min)

  write(*,'(//4x,"no.iter",15x,"Rf",17x,"Q ",6x,"Current spent time (min.)"/)')
  write(2,'(//4x,"no.iter",15x,"Rf",17x,"Q ",6x,"Current spent time (min.)"/)')

! оценивается время для 1000 вбрасываний

  do k_iter=1,1000
```



```

!
write(2,'(5x, "_____")')
_____")')
! write(2,'(/5x, "Выбирается сл. образом N_rndm атомов из размноженных атомов 1-го
сорта:")')

call rndm_from_N_rndm_all
! в Name_atom - имена Ca,Mg; из них Mg - N_rndm
! в work_name_XYZ_rndm - имена Ca,Mg по всей области

call Rfactor
! write(*,*)k_iter

Name_atom = Name_atom0

! write(2,'(/5x, "Номер испытаний = ",i6,4x, "Q(среднее) = ",F6.3)')k,S_Q
! write(2,'(5x, "R-factor for P(theor),P(exp) = ",F7.3, " %")')R_f

end do ! k_iter=1000

call cpu_time(finish_time)

iter_estimate = int(1000*end_time_in_min*60/(finish_time-time_coord))
if(iter_estimate < 99999)then
write(*,'(/5x, "The estimate of max. iter. = ",i6/)')iter_estimate
write(2,'(/5x, "The estimate of max. iter. = ",i6/)')iter_estimate
else
write(*,'(/5x, "The estimate of max. iter. = ",f10.2, " mln."/)')iter_estimate/1000000.
write(2,'(/5x, "The estimate of max. iter. = ",f10.2, " mln."/)')iter_estimate/1000000.
end if
! if(iter_estimate < 1000000)then
! write(*,'(/5x, "The estimate of max. iter. = ",i6, " mln."/)')int(iter_estimate/1000000)
! write(2,'(/5x, "The estimate of max. iter. = ",i6, " mln."/)')int(iter_estimate/1000000)
! end if

k_iter= 1000

do while (.true.)

k_iter=k_iter+1

!
write(2,'(5x, "_____")')
_____")')
! write(2,'(/5x, "Выбирается сл. образом P% атомов из размноженных атомов 1-го
сорта:")')

call rndm_from_N_rndm_all

call Rfactor

```



```

! write(2,'(//5x,"Сумма квадратов разности между теор. и реальным распредел.
="f10.6/)'gistogr_min
write(2,'(//5x,"R-factor = ",F7.3," %")R_f_min_sqrt

! R_f = sqrt(R_f)*coeff_Rf

write(2,'(//5x,"Итоговое распределение вероятности (m - кол-во связей) :"/)')
write(2,'(13x,"P (экспер.) P (теоретич.) 1000*delta")')
do i=1,N_rad+1
write(2,'(5x,"P(m=",i2,")= ",2(f9.6,2x),2x,f12.3)'i-
1,P_real0(i),P_theor(i),1000*(P_real0(i)-P_theor(i))
end do

write(2,'(//5x,"Координаты атомов наилучшего случайного вбрасывания:"/)')
do k=1,N_rndm
write(2,'(//5x,i6,3x,A7,2x,3(f6.2,1x))'k,Name_atom_rndm_Q_min(k),(XYZ_rndm_Q_mi
n(k,j),j=1,3)
end do
call cpu_time(time_iter)

! 0. цикл по связям: 1,2,3,4,....,N_rad
! 1. находится номер наибольшего отклонения P_real0(i) от P_theor(i)
! 2. для всех точек имеющих данное число (i-1) связей(A-B,B-A) строим алгоритм
! замены центр.атома (A/B) с радиальным (B/A) - таких радиальных атомов (i-1)
! 2.1. для каждой такой замены находим Q и R-factor . Если новый R-factor меньше
! текущего, то данное распределение атомов запоминается. Их всех замен ( для всех
атомов)
! оставляется то рапределение, имеющее меньший R-factor
! 3. переход к следующему номеру
! write(*,*)Bad_Links_min(9)
! stop

write(2,'(///5x,"Итоговая конфигурация после случайных вбрасываний."/)')

write(2,'(///8x,"no. Атом A/B атомы из окружения"/)')

! do k=1,N_rndm_all
! if(Bad_Links_min(k)==0)then
! write(2,'(//5x,i6,3x,i4))'k,Bad_Links_min(k)
! else
! write(2,'(//5x,i6,6x,A7,2x,i2,4x,20(2x,A7))'k,Name_atom_min(k),Bad_Links_min(k),&
! (Name_atom_min(Number_atom(k,l)),l=1,N_rad)
! end if
! end do
! new:
do k=1,N_rndm_all
write(2,'(//5x,i6,6x,A7,2x,i2,4x,20(2x,A7))'k,Name_atom_min(k),Bad_Links_min(k),&
(Name_atom_min(Number_atom(k,l)),l=1,N_rad)
end do

! запись данных для программы GULP

```

```

do i=1,10
  read(3,'(A60)')text60
  write(4,'(A60)')text60
end do

! *****

!
-----
write(2,'(/5x,"*****")')
*****')')
write(2,'(/5x,"Затраты процессорного времени."/')')

write(2,'(/5x,"
-----
write(2,'(/5x,"          Задача          Время (сек.)          "')')

write(2,'(/5x,"
-----
write(2,'(/5x,"Расчет координат всех атомов по трансляциям          ",f12.2/')')time_coord -
start_time1
write(2,'(/5x,"Расчет итераций          ",f12.2/')')time_iter - time_coord
write(2,'(/5x,"Суммарное время счета          ",f12.2/')')time_iter - start_time1
write(2,'(5x,"
-----
write(2,'(/5x,"          ")')

time = time_iter - start_time1
hours = int(time/3600.)
minutes =int(time/60 -hours*60 + 0.00001)
seconds = time - hours*3600 - minutes*60

if(hours > 0)then
  write(2,'(/5x,"Суммарное время счета(час/мин/сек):",1x,i2," час.",2x,i2,"
мин.",2x,f6.2," сек."') &
  int(hours),int(minutes),seconds
else if(minutes > 0)then
  write(2,'(/5x,"Суммарное время счета(мин/сек):",1x,i2," мин.",2x,f6.2," сек."') &
  int(minutes),seconds
else
  write(2,'(/5x,"Суммарное время счета(сек):",1x,f6.2," сек."')seconds
end if

write(2,'(/5x,"*****")')
*****')')

call cpu_time(finish_time)

```

```

time = finish_time-start_time1
hours = int(time/3600.)
minutes =int((time-hours*3600)/60.)
seconds = time - hours*3600 - minutes*60
write(2,'(//5x,"Общее время вычислений:",1x,i2," час.",2x,i2," мин.",2x,f6.2," сек.")')
&
int(hours),int(minutes),seconds

call close_stop
end !

subroutine open_read
use commonmod

S_Q_min = 100.

! write(2,*)P_theor
open(1,file='input.txt')
open(2,file='output.txt')
open(4,file='data_for_gulp.txt')

! write(2,*)P_theor
! stop

read(1,*)name_file,text
read(1,*)N_group,text
! N_group - число сортов/групп

do i=1,N_group
read(1,*)Name_group(i),text
read(1,*)Num_atomov_in_group(i),text
end do
! Name_group() - содержит имена исходных сортов атомов

read(1,*)m_X,m_Y,m_Z,text
! m_X,m_Y - заданные трансляции по осям
read(1,*)N_rndm,text
! N_rndm - число замещаемых позиций

! write(*,*)m_X,m_Y,m_Z,N_rndm

if(N_rndm > Num_atomov_in_group(1)*m_X*m_Y*m_Z)then
write(*,'(//2x,"ERROR: N_rndm = ",i4," > Num_atomov_in_group(1)*m_X*m_Y*m_Z
= ",i4//)')&
N_rndm,Num_atomov_in_group(1)*m_X*m_Y*m_Z
write(2,'(//2x,"ERROR: N_rndm = ",i4," > Num_atomov_in_group(1)*m_X*m_Y*m_Z
= ",i4)')&
N_rndm,Num_atomov_in_group(1)*m_X*m_Y*m_Z
stop
end if

if(N_rndm < 1)then

```

```

write(*, '(//2x, "ERROR: N_rndm = ", i4, " < 1"//)') N_rndm
write(2, '(//2x, "ERROR: N_rndm = ", i4, " < 1"//)') N_rndm
stop
end if

```

! N_rndm_all = общее кол-во атомов (сортов первой группы) кот. будут случайно заменяться

```

N_rndm_all = m_X * m_Y * m_Z * Num_atomov_in_group(1)
! write(*,*) N_rndm_all

```

```

read(1,*) Radius, text
read(1,*) name_remont, text
! read(1,*) iter, text
read(1,*) output_1, text
read(1,*) output_2, text
read(1,*) end_time_in_min, text
read(1,*) criteriy_for_Rf, text
read(1,*) criteriy_for_rndm, text

```

! фиксация времени:

```

call cpu_time(start_time1)
call gettim(ihours, iminutes, iseconds, iseconds100)
write(*, '(//7x, "Start time      = ", i2, " h. ", i2, " min."//)') ihours, iminutes
write(*, '(//7x, "Total time limited = ", 5x, i6, " min."//)') int(end_time_in_min)
write(2, '(//7x, "Start time      = ", i2, " h. ", i2, " min."//)') ihours, iminutes
write(2, '(//7x, "Total time limited = ", 5x, i6, " min."//)') int(end_time_in_min)
k = int((iminutes + int(end_time_in_min)) / 60.)
ihours = mod(ihours + k, 24)
iminutes = iminutes + int(end_time_in_min) - 60 * k
write(*, '(//7x, "End time      = ", i2, " h. ", i2, " min."//)') ihours, iminutes
write(2, '(//7x, "End time      = ", i2, " h. ", i2, " min."//)') ihours, iminutes

```

! *****

```

if(criteriy_for_Rf==0) R_f_min = 1.e+10
if(criteriy_for_Rf==1) R_f_min = 0.

```

```

open(3, file=trim(name_file)//'.txt')
read(3, '(A60)') text60
write(4, '(A60)') text60
read(3, '(A60)') text60
write(4, '(A60)') text60
read(3, *) a, b, c, alpha, beta, gamma

```

```

write(4, '(2x, 3(f6.3, 2x), 3(f6.2, 2x)/)') &
a * m_X, b * m_Y, c * m_Z, alpha, beta, gamma
read(3, '(A60)') text60
write(4, '(A60)') text60

```

```

! *****
a2=a*a
b2=b*b
c2=c*c
ab=2*a*b*cos(gamma*PI180)
ac=2*a*c*cos(beta*PI180)
bc=2*b*c*cos(alpha*PI180)
Radius2 = Radius*Radius

! *****

! write(2,'(/2x,"Координаты исходных атомов в долях большой ячейки :"/)')

write(2,'(//2x,"Координаты исходных атомов в долях маленькой ячейки (векторов
a,b,c) :"/)')

k=0
N_all_atomov = 0
do i=1,N_group
no = 0
N_all_atomov = N_all_atomov + Num_atomov_in_group(i)
! Name_group_i = Name_group(i)
do j=1,Num_atomov_in_group(i)
no = no + 1
k=k+1
read(3,*)text2,(XYZ_0(k,L),L=1,3),dummy
if(i==1)dummy1=dummy
if(i==2)dummy2=dummy
write(2,'(2x,i2,2x,a2,6x,3(f12.10,2x),2x,f6.2)')no,text2,(XYZ_0(k,L),L=1,3),dummy
! XYZ_0(k,1)=m_X*XYZ_0(k,1)
! XYZ_0(k,2)=m_Y*XYZ_0(k,2)
! XYZ_0(k,3)=m_Z*XYZ_0(k,3)
end do ! j
write(2,'(/)')
end do ! i

! write(2,'(//2x,"Координаты исходных атомов в долях маленькой ячейки (векторов
a,b,c) :"/)')

! k=0
! do i=1,N_group
! no = 0
! Name_group_i = Name_group(i)
! do j=1,Num_atomov_in_group(i)
! no = no + 1
! k=k+1
! write(2,'(2x,i2,2x,a2,6x,3(f8.6,2x)')no,Name_group_i,(XYZ_0(k,L),L=1,3)
! end do ! j
! write(2,'(/)')
! end do ! i

```

```

write(2,'(/2x,3(3x,i2),A25/)')m_X,m_Y,m_Z," трансляции по X,Y,Z"

write(2,'(/2x,3(f6.3,2x),3(f6.2,2x),A40/)') &
a,b,c,alpha,beta,gamma," a,b,c,alpha,beta,gamma"
write(2,'(/2x,3(f6.3,2x),3(f6.2,2x),A40/)') &
a*m_X,b*m_Y,c*m_Z,alpha,beta,gamma," a*m_X,b*m_Y,c*m_Z,alpha,beta,gamma"

return
end

subroutine XYZ0_in_XYZ
use commonmod

! размножение атомов всех сортов ( в XYZ )

! m_group - для перехода к следующей группе атомов внутри XYZ_0
m_group = 0

N_all = 0
write(2,'(/2x,"Координаты размноженных атомов в трансляциях маленькой ячейки
(векторов a,b,c) :"/)')

do m=1,N_group
no = 0
! цикл по сортам атомов
k_name = 0
! для каждого сорта новая нумерация ( с единицы)
Name_group_m = Name_group(m)
! задаем имя сорта
! размножение группы(одного сорта) атомов

do k=1,m_Z
do j=1,m_Y
do i=1,m_X
do j_group=1,Num_atomov_in_group(m)
! цикл внутри сорта атома
N_all = N_all + 1
no = no + 1
XYZ(N_all,1) = XYZ_0(m_group + j_group,1) + dble(i - 1)
XYZ(N_all,2) = XYZ_0(m_group + j_group,2) + dble(j - 1)
XYZ(N_all,3) = XYZ_0(m_group + j_group,3) + dble(k - 1)
! write(2,'(5x,i6,3x,3(f5.3,2x))')L,(XYZ_0(L,j),j=1,3)

! запоминаем имена размноженных атомов только от 1-го сорта
! if(m==1)then
k_name = k_name + 1
write(ch_k_name,'(i5)')k_name
Name_atom(N_all)=trim(Name_group_m)//trim(adjustl(ch_k_name))
! проверено:
! write(2,'(5x,i6,3x,A7,2x,3(f5.3,2x))')N_all,Name_atom(N_all), &
! XYZ(N_all,1),XYZ(N_all,2),XYZ(N_all,3)

```



```

!      end if

      write(2,'(5x,i6,3x,A7,2x,3(f6.3,2x))')no,Name_atom(N_all), &
XYZ(N_all,1),XYZ(N_all,2),XYZ(N_all,3)

      end do ! j_group
      end do ! i
      end do ! j
      end do ! k
      m_group = m_group + Num_atomov_in_group(m)
      end do ! m

!      stop

! результат:
! N_all - кол-во всех атомов
! в XYZ(N_all,*) - координаты размноженных атомов всех сортов
! в Name_atom(N_all) - имена размноженных атомов всех сортов
! N_rndm_all = m_X * m_Y * m_Z * Num_atomov_in_group(1) - кол-во размнож. атомов 1-
го сорта

!      write(2,*)N_all,N_rndm_all," - N_all,N_rndm_all"
!      stop

!
*****
***

! Строим массив work_XYZ_rndm(max_numb_work_points,3) - координаты атомов 1-го
сорта внутри
!
!                                     и на оболочке области
      num_sorta = 1
      Name_group_m = Name_group(num_sorta )
      m = Num_atomov_in_group(num_sorta)

! внутренние точки:
      do k=1,N_rndm_all
      work_XYZ_rndm(k,1) = XYZ(k,1)
      work_XYZ_rndm(k,2) = XYZ(k,2)
      work_XYZ_rndm(k,3) = XYZ(k,3)
      index_XYZ_rndm(k) = k
      end do ! k

      N = N_rndm_all

! k,j,i :
! один слой для фиксированного k:
! m*m_X*m_Y

! на K-м слое, J-м столбце, I-я коорд.:
! m*m_X*m_Y*(K-1) + m*m_X*(j-1) + m*(i - 1)

```

!

! дублируем верхний слой основного кубика: k=m_Z, j = 1,m_Y,i=1,m_X 5-6-7-8
! в верхний слой нижнего кубика под 1-2-3-4

```

m_m_X_m_Y = m*m_X*m_Y

k2 = m_Z
j2 = m_Y
i2 = m_X

do k = k2,k2
do j = 1,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

! дублируем верхнюю левую полосу : k=m_Z j = 1,m_Y i=1
! в нижнюю правую полосу
! 5 - 8 в 2-3

```

k2 = m_Z
j2 = m_Y
i2 = 1

do k = k2,k2
do j = 1,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j

```

```
end do ! k
```

```
!
```

```
! дублируем левую плоскость основного кубика: K=1,m_Z J= 1,m_Y, I=1  
! в левую часть правого кубика  
! 1-4-8-5 в 2-3-7-6
```

```
k2 = m_Z  
j2 = m_Y  
i2 = 1
```

```
do k = 1,k2  
do j = 1,j2  
do i = i2,i2  
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)  
do j_group=1,m  
N = N + 1  
index_XYZ_rndm(N) = L + j_group  
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)  
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)  
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)  
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &  
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)  
end do ! j_group  
end do ! i  
end do ! j  
end do ! k
```

```
!
```

```
! дублируем нижнюю левую полосу : k=1 j= 1,m_Y i=1  
! в верхнюю правую полосу  
! 1 - 4 в 6-7
```

```
k2 = 1  
j2 = m_Y  
i2 = 1
```

```
do k = k2,k2  
do j = 1,j2  
do i = i2,i2  
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)  
do j_group=1,m  
N = N + 1  
index_XYZ_rndm(N) = L + j_group  
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)  
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)  
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)  
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &  
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)  
end do ! j_group  
end do ! i
```

```
end do ! j
end do ! k
```

!

```
! дублируем нижнюю плоскость основного кубика: K=1 J = 1,m_Y, I=1,m_X
! в нижнюю часть верхнего кубика
! 1-2-3-4 в 5-6-7-8
```

```
k2 = 1
j2 = m_Y
i2 = m_X

do k = k2,k2
do j = 1,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k
```

!

```
! дублируем нижнюю правую полосу : k=1 j = 1,m_Y i=m_X
! в верхнюю левую полосу
! 2 - 3 в 5-8
```

```
k2 = 1
j2 = m_Y
i2 = m_X

do k = k2,k2
do j = 1,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
```

```

! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

```

! дублируем правую плоскость основного кубика: K=1,m_Z J = 1,m_Y, I=m_X
! в правую часть левого кубика
! 2-3-7-6 в 1-4-8-5

```

```

k2 = m_Z
j2 = m_Y
i2 = m_X

```

```

do k = 1,k2
do j = 1,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

!

```

! дублируем верхнюю правую полосу : k=m_Z j = 1,m_Y i=m_X
! в нижнюю левую полосу
! 6 - 7 в 1-4

```

```

k2 = m_Z
j2 = m_Y
i2 = m_X

```

```

do k = k2,k2
do j = 1,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m

```

```

N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(j - 1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

! дублируем заднюю плоскость основного кубика: K=1,m_Z J = m_Y I=1,m_X
! в заднюю часть переднего кубика
! 4-3-7-8 в 1-2-6-5

```

k2 = m_Z
j2 = m_Y
i2 = m_X

do k = 1,k2
do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

! дублируем переднюю плоскость основного кубика: K=1,m_Z J = 1 I=1,m_X
! в переднюю часть заднего кубика
! 1-2-6-5 в 4-3-7-8

```

k2 = m_Z
j2 = 1
i2 = m_X

do k = 1,k2

```

```

do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

```

! дублируем нижнюю переднюю полосу : k=1 j=1 i=1,m_X
! в верхнюю заднюю полосу
! 1 - 2 в 8-7

```

```

k2 = 1
j2 = 1
i2 = m_X

do k = k2,k2
do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

```

! дублируем нижнюю заднюю полосу : k=1 j=m_Y i=1,m_X
! в верхнюю переднюю полосу
! 4 - 3 в 5-6
k2 = 1
j2 = m_Y
i2 = m_X

do k = k2,k2

```

```

do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

```

! дублируем верхнюю заднюю полосу : k=m_Z j=m_Y i=1,m_X
! в нижнюю переднюю полосу
! 8 - 7 в 1-2

```

```

k2 = m_Z
j2 = m_Y
i2 = m_X

```

```

do k = k2,k2
do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

```

! дублируем верхнюю переднюю полосу : k=m_Z j=1 i=1,m_X
! в нижнюю заднюю полосу
! 5 - 6 в 4-3

```

```

k2 = m_Z
j2 = 1
i2 = m_X

```



```

do k = k2,k2
do j = j2,j2
do i = 1,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(i - 1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

! дублируем правую переднюю полосу : k=1,m_Z j=1 i=m_X
! в левую заднюю полосу
! 2 - 6 в 4-8

```

k2 = m_Z
j2 = 1
i2 = m_X

do k = 1,k2
do j = j2,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

!

! дублируем правую заднюю полосу : k=1,m_Z j=m_Y i=m_X
! в левую переднюю полосу
! 3 - 7 в 1-5

```

k2 = m_Z
j2 = m_Y
i2 = m_X

do k = 1,k2
do j = j2,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

!

```

! дублируем левую заднюю полосу : k=1,m_Z j=m_Y i=1
! в правую переднюю полосу
! 4 - 8 в 2-6

```

k2 = m_Z
j2 = m_Y
i2 = 1

do k = 1,k2
do j = j2,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

!

```

```
! дублируем левую переднюю полосу : k=1,m_Z j=1 i=1
! в правую заднюю полосу
! 1 - 5 в 3-7
```

```

k2 = m_Z
j2 = 1
i2 = 1

do k = 1,k2
do j = j2,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(k - 1)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j
end do ! k

```

```
! кубики-уголки ( всего 8-мь, по вершинам):
```

```
! 1-й кубик
```

```
! дублируем нижний передний левый кубик : k=1 j=1 i=1
```

```
! в верхний задний правый
```

```
k2 = 1
```

```
j2 = 1
```

```
i2 = 1
```

```

do k = k2,k2
do j = j2,j2
do i = i2,i2
L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
do j_group=1,m
N = N + 1
index_XYZ_rndm(N) = L + j_group
work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
end do ! j_group
end do ! i
end do ! j

```

```

end do ! k
!


---


! 2-й кубик
! дублируем нижний передний правый кубик : k=1 j=1 i=m_X
! в верхний задний левый
    k2 = 1
    j2 = 1
    i2 = m_X

    do k = k2,k2
    do j = j2,j2
    do i = i2,i2
    L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
    do j_group=1,m
    N = N + 1
    index_XYZ_rndm(N) = L + j_group
    work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
    work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
    work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
    end do ! j_group
    end do ! i
    end do ! j
    end do ! k
!


---


! 3-й кубик
! дублируем нижний задний правый кубик : k=1 j=m_Y i=m_X
! в верхний передний левый
    k2 = 1
    j2 = m_Y
    i2 = m_X

    do k = k2,k2
    do j = j2,j2
    do i = i2,i2
    L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
    do j_group=1,m
    N = N + 1
    index_XYZ_rndm(N) = L + j_group
    work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
    work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
    work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
! write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
    end do ! j_group
    end do ! i
    end do ! j
    end do ! k
!


---



```

```

! 4-й кубик
! дублируем нижний задний левый кубик : k=1  j=m_Y  i=1
! в  верхний передний правый
      k2 = 1
      j2 = m_Y
      i2 = 1

      do k = k2,k2
      do j = j2,j2
      do i = i2,i2
      L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
      do j_group=1,m
      N = N + 1
      index_XYZ_rndm(N) = L + j_group
      work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
      work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
      work_XYZ_rndm(N,3) = XYZ_0(j_group,3) + dble(m_Z)
!       write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
      end do ! j_group
      end do ! i
      end do ! j
      end do ! k
! _____

```

```

! 5-й кубик
! дублируем верхний передний левый кубик : k=m_Z  j=1  i=1
! в  нижний задний правый
      k2 = m_Z
      j2 = 1
      i2 = 1

      do k = k2,k2
      do j = j2,j2
      do i = i2,i2
      L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
      do j_group=1,m
      N = N + 1
      index_XYZ_rndm(N) = L + j_group
      work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
      work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
      work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
!       write(2,(5x,i6,3x,A7,2x,2(f5.2,2x)))N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
      end do ! j_group
      end do ! i
      end do ! j
      end do ! k
! _____

```

```

! 6-й кубик

```

```

! дублируем верхний передний правый кубик : k=m_Z j=1 i=m_X
! в нижний задний левый
  k2 = m_Z
  j2 = 1
  i2 = m_X

  do k = k2,k2
  do j = j2,j2
  do i = i2,i2
  L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
  do j_group=1,m
  N = N + 1
  index_XYZ_rndm(N) = L + j_group
  work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
  work_XYZ_rndm(N,2) = XYZ_0(j_group,2) + dble(m_Y)
  work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
!   write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
  end do ! j_group
  end do ! i
  end do ! j
  end do ! k

```

```

! 7-й кубик
! дублируем верхний задний правый кубик : k=m_Z j=m_Y i=m_X
! в нижний передний левый
  k2 = m_Z
  j2 = m_Y
  i2 = m_X

  do k = k2,k2
  do j = j2,j2
  do i = i2,i2
  L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
  do j_group=1,m
  N = N + 1
  index_XYZ_rndm(N) = L + j_group
  work_XYZ_rndm(N,1) = XYZ_0(j_group,1) - dble(1)
  work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
  work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
!   write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
  end do ! j_group
  end do ! i
  end do ! j
  end do ! k
! _____

```

```

! 8-й кубик
! дублируем верхний задний левый кубик : k=m_Z j=m_Y i=1

```

```

! в нижний передний правый
  k2 = m_Z
  j2 = m_Y
  i2 = 1

  do k = k2,k2
  do j = j2,j2
  do i = i2,i2
  L = m_m_X_m_Y*(k-1) + m*m_X*(j-1) + m*(i - 1)
  do j_group=1,m
  N = N + 1
  index_XYZ_rndm(N) = L + j_group
  work_XYZ_rndm(N,1) = XYZ_0(j_group,1) + dble(m_X)
  work_XYZ_rndm(N,2) = XYZ_0(j_group,2) - dble(1)
  work_XYZ_rndm(N,3) = XYZ_0(j_group,3) - dble(1)
!   write(2,'(5x,i6,3x,A7,2x,2(f5.2,2x))')N_gran_points,Name_atom_gran(N_gran_points), &
! XYZ_gran(N_gran_points,1),XYZ_gran(N_gran_points,2)
  end do ! j_group
  end do ! i
  end do ! j
  end do ! k
!

```

```

! всего точек атомов 1-го сорта внутри и в области
  k_work_rndm_all = N
  write(2,'(///2x,"Координаты атомов 1-го сорта:"/)')
  write(2,'(/2x,"no. номер размножающего атома   X   Y   Z"/)')

```

```

! проверено :
  do j=1,k_work_rndm_all
  write(2,'(2x,i5,16x,i5,5x,3(f7.4,2x))')j,index_XYZ_rndm(j),&
  work_XYZ_rndm(j,1),work_XYZ_rndm(j,2),work_XYZ_rndm(j,3)
  end do

```

```

! результаты:
! work_XYZ_rndm(max_numb_work_points,3) - координаты атомов 1-го сорта внутри
!                                     и в оболочке области
! work_name_XYZ_rndm(max_numb_work_points) - имена этих атомов
! index_XYZ_rndm(max_numb_work_points) - номера этих атомов
! в index_XYZ_rndm - находятся индексы XYZ_rndm

```

```

! XYZ_rndm(max_XYZ_rndm,2) - координаты сл. выбранных из размноженных атомов 1-й
группы
! work_XYZ_rndm(max_numb_work_points,3) - координаты атомов 1-го сорта внутри
!                                     и на оболочке области
! work_name_XYZ_rndm(max_numb_work_points) - имена этих атомов
! index_XYZ_rndm(max_numb_work_points) - номера этих атомов
!

```

```

! Для каждого размнож. атома 1-го сорта ( их всего N_rndm_all) ищем номера атомов
попавших в сферу
! с центром в этом атоме Ca ( их всего N_rndm_all ) и заданным радиусом Radius

```

```

! в сферу должно попасть 12-ть атомов Ca - 3-х мерный случай
! в сферу должно попасть 8 атомов Ca - 2-х мерный случай

! 8-мь областей: каждая из них идентична исходной положительной осьмушки
! алгоритм:
! 2. цикл L по всем точкам 1-го сорта от 1 до N_rndm_all (точка L0 исключается)

!      write(2,'(///1x,"Номер атома      Кол-во атомов в сфере      Номера атомов из
сферы"//)')

! цикл L0 по всем точкам 1-го сорта от 1 до N_rndm_all ( внутри области)
do L0=1,N_rndm_all

! 1. фиксируем точку с коорд.      p0=XYZ(L0,1)      q0=XYZ(L0,2)      r0=XYZ(L0,3)
p0=XYZ(L0,1)
q0=XYZ(L0,2)
r0=XYZ(L0,3)
write(2,'(//2x,"
_____
_____")')
write(2,'(//2x,"Текущий атом и атомы из сферы:"//)')
write(2,'(2x,"Номер атома = ",i4,4x,"Координаты: ",3(f6.3,2x)/)')L0,p0,q0,r0

! в N_rad - кол-во захваченных
N_rad = 0
! в итоге N_rad д.б. равно кол-ву атомов захваченных сферой ( 12 ( или 8 в 2-у мерном
случае)

! цикл L по всем точкам 1-го сорта от 1 до k_work_rndm_all ( по всей области)

write(2,'(2x,"no.      номер атома из сферы      координаты      расстояние"//)')

do L=1,k_work_rndm_all
if(L==L0)cycle
! k=index_XYZ_rndm(L)
p=work_XYZ_rndm(L,1)
q=work_XYZ_rndm(L,2)
r=work_XYZ_rndm(L,3)

pp0=p-p0
qq0=q-q0
rr0=r-r0
! точки вне квадрата описанного около сферы:
if((abs(pp0) > Radius .or. abs(qq0) > Radius ) .or. abs(rr0) > Radius )cycle

pp02=pp0*pp0

qq02=qq0*qq0

rr02=rr0*rr0

```



```

R0i2=pp02*a2+qq02*b2+rr02*c2+pp0*qq0*ab+pp0*rr0*ac+qq0*rr0*bc

if(R0i2 > Radius2)cycle
N_rad = N_rad + 1
Number_atom(L0,N_rad) = index_XYZ_rndm(L)
! write(2,'(5x,i3," :",i4,4x,3(f6.3,2x),2x,f5.2)')N_rad,index_XYZ_rndm(L),p,q,r,sqrt(R0i2)
write(2,'(2x,i3,12x,i4,12x,3(f6.3,2x),2x,f5.2)')N_rad,index_XYZ_rndm(L),p,q,r,sqrt(R0i2)
)

end do ! L
! проверено :
! write(2,'(1x,i6,5x,i2," :",2x,20(i4,2x)//)')L0,N_rad,(Number_atom(L0,k),k=1,N_rad)

end do ! L0
! stop
return
end

! ***** Строим сферу с центром в каждом атоме Ca ( их всего N_rndm_all ) и
заданным радиусом Radius
! ***** в сферу должно попасть 12-ть атомов Ca - 3-х мерный случай
! 8-мь областей: каждая из них идентична исходной положительной осьмушки
! алгоритм:
! цикл L0 по всем точкам 1-го сорта от 1 до N_rndm_all
! 1. фикс. точка p0=XYZ(L0,1) q0=XYZ(L0,2)
! 2. цикл L по всем точкам 1-го сорта от 1 до N_rndm_all (точка L0 исключается)
! 2.1.
! результаты:
!
subroutine Rfactor
use commonmod
S_Q = 0.
P_real = 0.
do L0=1,N_rndm_all
Name_atom_L0 = Name_atom(L0)(1:2)
! Name_atom(L0)(1:2) - Ca/Mg
! if(output_2==1)then

!
write(2,'(//5x,"*****")')
*****

! write(2,'(//5x," No. Name X Y Z - координаты(в долях) текущего атома "/)')

! write(2,'(/5x,i6,3x,A7,2x,3(f5.3,2x)')L0,Name_atom(L0),(XYZ(L0,j),j=1,3)
! write(2,'(10x,"_____")')
! end if

! write(2,'(//5x," No. Name X Y Z RO - координаты(в долях) атомов
попавших в сферу; расстояние до точки "/)')

n_bad = 0

```

! в итоге n_bad д.б. равно кол-ву связей A/B, B/A по всей области

! N_rad0 = 0

! в итоге N_rad д.б. равно кол-ву атомов захваченных сферой (12 (или 8 в 2-у мерном случае)

```
do k=1,N_rad
  L = Number_atom(L0,k)
!   write(*,*)'k,L='k,L
  if(Name_atom_L0/=Name_atom(L)(1:2))n_bad = n_bad + 1

! name_sphera(L0,N_rad) - имя атома попавшего в сферу
!   name_sphera(L0,k) = Name_atom(L)
!
  if(output_2==1)write(2,'(//5x,i3,2x,A7,2(f6.2,2x),2x,f8.3)')N_rad0,Name_atom(L),p,q,sqr
t(R0i2)
!   write(2,'(//5x,i3,2x,A7,3(f6.2,2x),2x,f8.3)')N_rad,Name_atom(L),p,q,s,sqrt(R0i2)
  end do ! k

!   write(*,*)N_rndm_all,N_rad,L0,n_bad
  Bad_Links(L0) = n_bad
  Q = real(n_bad)/N_rad
  S_Q = S_Q + Q

  i=n_bad+1

!   write(*,*)i
  P_real(i) = P_real(i) + 1

!   if(output_2==1)&

! проверено:
!   write(2,'(//5x,i5,2x,A7,("i1,"-",i1,")",2x,"Q=",F6.3,3x,20(A7,2x),2x,f8.3)')&
!   L0,Name_atom(L0),N_rad-n_bad,n_bad,Q,(name_sphera(L0,i),i=1,N_rad)

  end do ! L0
! *****

  S_Q = S_Q/N_rndm_all
!   насчитываем сумму квадратов разностей между теорет. распределением и реальным

  P_real = P_real/N_rndm_all
!   p=0.
  R_f = 0.
  do i=1,N_rad + 1
!   write(2,'(5x,"P(m=",i1,")= ",2(f7.4,2x))')i-1,P_real(i),P_theor(i)
!   p=p+P_real(i)
  R_f = R_f + (P_real(i) - P_theor(i))**2
  end do ! i

! проверено:
```



```

subroutine rndm_from_N_rndm_all
use commonmod

!
write(2,'(//5x,"*****")')
*****")

! write(2,'(//5x,"No. Name(old) Name(new) X Y - координаты( в долях)"/)')

! ISEED = 2147483646/2
! call GSU1R(ISEED,N_rndm,R_rndm)
! write(2,*)(R_rndm(i),i=1,N_rndm)

if(criteriy_for_rndm == 1)call random_seed()
i=1
do while (i <= N_rndm)
call random_number(rndm)
k = int(rndm*N_rndm_all + 0.5)
if(k==0)k=1
if(k > N_rndm_all)k=N_rndm_all
repeat = .false.
if(i > 1)then
do j=1,i-1
if(k==k_repeat(j))then
repeat = .true.
exit
end if
end do ! j
end if
if(repeat)cycle

! заменяем в исходном наборе имя атома на новое(Mg):
! k - выбранный сл. образом индекс

Name_atom(k) = name_remont//Name_atom(k)(3:len_trim(Name_atom(k)))

!
write(2,'(5x,i6,2x,A7,2x,A7,3x,2(f5.3,2x))')i,Name_atom0(k),Name_atom_rndm(i),(XYZ
_rndm(i,j),j=1,2)

k_repeat(i)=k
i=i+1
end do ! i

! проверено :
! write(2,'(2x,i3,3x,A7)')(k,Name_atom(k),k=1,N_rndm_all)

! заносим эти имена в work_name_XYZ_rndm

do k=1,k_work_rndm_all
work_name_XYZ_rndm(k) = Name_atom(index_XYZ_rndm(k))

```

```

! проверено :
!
      write(2,'(2x,i3,3x,A7,3(2x,f6.1))'k,work_name_XYZ_rndm(k),work_XYZ_rndm(k,1),w
ork_XYZ_rndm(k,2),work_XYZ_rndm(k,3)
      end do
!      stop

      return
      end
      subroutine output_for_gulp
      use commonmod
      write(4,('_____'))

      write(4,'(1/x,"Rfact = ",2x,f7.3," %",5x,"Q = ",2x,f7.3/)'R_f_min_sqrt,S_Q_min

! атомы первого сорта:
      do k=1,N_rndm_all
      x_coord = dble(work_XYZ_rndm(k,1)/m_X)
      y_coord = dble(work_XYZ_rndm(k,2)/m_Y)
      z_coord = dble(work_XYZ_rndm(k,3)/m_Z)
      write(4,'(1x,A7,3(2x,f13.10),4x,f6.2)'Name_atom_min(k),x_coord,y_coord,z_coord,dum
my1
      end do

! атомы второго сорта:
      m = 2
      Name_group_m = Name_group(m)
      do k=N_rndm_all+1,N_all
      write(ch_k_name,'(i5)'k-N_rndm_all
      x_coord = dble(XYZ(k,1)/m_X)
      y_coord = dble(XYZ(k,2)/m_Y)
      z_coord = dble(XYZ(k,3)/m_Z)
      write(4,'(1x,A7,3(2x,f13.10),4x,f6.2)'trim(Name_group_m)//adjustl(ch_k_name),&
x_coord,y_coord,z_coord,dummy2
      end do

      return
      end

      subroutine close_stop
      close(1)
      close(2)
      close(3)
      close(4)
      stop
      return
      end

```

```

SUBROUTINE GSU1R(ISEED,N,R)
DIMENSION R(1)
INTEGER ISEED,N,I,D2P32M
REAL R
DOUBLE PRECISION Z,D2P31M,D2PN31,DMOD,DFLOAT
DATA D2P31M/2147483647.D0/,D2PN31/4.656612873077393D-10/,D2P32M/16807/
Z=DFLOAT(ISEED)
DO I=1,N
Z=DMOD(D2P32M*Z,D2P31M)
R(I)=Z*D2PN31
      end do
ISEED=Z
RETURN
END
! *****
!      subroutine calculation_C_m_l

      subroutine calculation_P_theor
      integer C_m_l(100),C(100)
      integer N_all,N_rndm_all,N_rad
      integer N_group,m_X,m_Y,m_Z,N_rndm,iter

      real gistogr_min,P_real,P_real0,P_theor

      common / R_factor /
R_f,R_f_min,R_f_min_sqrt,coeff_Rf,P_real(100),P_real0(100),P_theor(100)
      common / N_all / N_all,N_rndm_all,N_rad
      common / input /
N_group,m_X,m_Y,m_Z,N_rndm,Radius,iter,end_time_in_min,finish_time,start_time

! N_rad - кол-во атомов попавших в сферу
!!!! N_rad должно быть не менее 3-х
! P_theor(1) = P(k=0)

!      real :: gistogr_theor(9) = (/1.,8.,28.,56.,70.,56.,28.,8.,1./)

! концентрации смеси: p - нового сорта, q - заменяемого сорта
      p = real(N_rndm)/real(N_rndm_all)
      q = 1. - p
!      write(*,*)p,q
!      stop

!
!
! _____
!
! вычисляем коэфф. числа сочетаний l элементов из m :

      m = N_rad
      C(1) = 1
      C(2) = 2
      C(3) = 1
      C_m_l = C

```

```

do j=3,N_rad
do i=2,j
C_m_l(i) = C(i-1) + C(i)
end do ! i
C_m_l(j+1) = 1
C = C_m_l
! write(*,*)j,(C_m_l(i),i=1,j+1)
end do ! j
!
!

```

```

ppA = 1.
ppB = 1.

do j=1,N_rad+2
ppA = p*ppA
ppB = q*ppB
end do ! j

ppA = ppA/q
ppB = ppB/p

qp = q/p
pq = p/q

! sP = 0.
do j=1,N_rad+1
ppA = ppA*qp
ppB = ppB*pq
P_theor(j) = C_m_l(j)*(ppA + ppB)
! sP = sP + P_theor(j)
! write(2,*)j,C_m_l(j),P_theor(j)
end do ! j
! write(2,*)sP

! stop

coeff_Rf = 0.
do j=1,N_rad+1
coeff_Rf = coeff_Rf + P_theor(j)*P_theor(j)
end do
coeff_Rf = 100.*sqrt(1./coeff_Rf)

return
end

subroutine output1
use commonmod
write(2,'(//)')

```

```

write(2,'(/5x,"Кол-во всего размноженных атомов = ", i8)')N_all
write(2,'(/5x,"из них: кол-во помеченных атомов = ", i8)')N_rndm_all
write(2,'(/5x,"Кол-во трансляций по X,Y,Z = ", 3(2x,i3))')m_X,m_Y,m_Z
write(2,'(/5x,"*****")')

write(2,'(//5x," No. Name X Y Z- размноженные координаты(в долях) всех
сортов атомов "//)')

do i=1,N_all
write(2,'(5x,i6,3x,A7,2x,2(f5.3,2x))')i,Name_atom(i),XYZ(i,1),XYZ(i,2),XYZ(i,3)
end do ! i
write(2,'(/5x,"*****")')

return
end

```