

## КОМПЬЮТЕРНАЯ ПРОГРАММА ПОИСКА ПЛОСКОСТЕЙ ГРУППИРОВАНИЯ ГИПОЦЕНТРОВ ЗЕМЛЕТРЯСЕНИЙ

*Кролевец А. Н., Макеев А. М.*

*Камчатский государственный университет имени Витуса Беринга, г. Петропавловск-Камчатский,  
an@kamgu.ru*

### **Введение**

Особенностям пространственного распределения гипоцентров землетрясений посвящено множество работ. Принято считать, что гипоцентры группируются в областях сочленения литосферных плит и составляющих их блоков. Однако проблема идентификации границ блоков, даже самых крупных, размерами в сотни километров ещё далека от окончательного разрешения. Основная трудность заключена в существенных погрешностях определения координат гипоцентров. Весьма грубыми являются даже оценки величин самих погрешностей. В работе [1] была высказана гипотеза, что погрешность оказывается меньшей для гипоцентров уверенно регистрируемых землетрясений высших классов. Основанием этому послужило наблюдение того, что все гипоцентры афтершоковых землетрясений класса 13 вместе с гипоцентром основного события Кроноцкого землетрясения 5.12.1997 г. и выше (всего 15 событий) попадают лишь на три пространственные плоскости с отклонениями, не превышающими 3 км. Данное значение можно считать оценкой погрешности определения координат указанных событий. Поиск плоскостей группирования гипоцентров в работе [1] осуществлялся визуально с помощью графических возможностей пакета программных средств Maple. Такая процедура является весьма трудоёмкой. В случае анализа большего числа гипоцентров возникает вероятность пропуска значительной части плоскостей группирования. Проверить гипотезу для ряда других сильных событий мешало отсутствие подходящего программного обеспечения. Несмотря на публикацию алгоритма [2], в принципе решающего поставленную задачу, до последнего времени отсутствовала эффективная компьютерная программа, которая могла бы обрабатывать данные о сотнях и более числа гипоцентров за разумное время и не вызывать сбоев из-за исчерпания ресурсов компьютеров.

В данной работе представлена такая компьютерная программа и приведено её описание. Представлены также блок-схема алгоритма и программный код наиболее важных процедур ее исполнения. На компьютере, с процессором Intel Pentium Dual-Core E5200 данная программа обрабатывает список до 1000 гипоцентров, в среднем за 2-3 минуты. Программа реализована на языке Object Pascal.

### **Описание программы**

Исходными данными для работы программы является список гипоцентров, вместе с их географическими координатами и глубиной из каталога, декартовыми координатами, энергетическим классом, идентификационным номером, например, из каталога. Пересчёт представленных в каталогах географических координат гипоцентров и глубин в декартовы можно выполнять, например, по формулам, предложенным в работе [1].

Блок-схема алгоритма программы представлена на рис. 1. При запуске программы происходит загрузка динамического списка исходных данных. Далее пользователь вводит необходимые для поиска группирования критерии:  $m$  – минимально допустимое количество гипоцентров на плоскости,  $D$  – максимально допустимое удаление гипоцентра от плоскости. Далее программа запускает ряд вложенных циклов, осуществляющих перебор всех возможных ориентаций плоскостей. Каждой ориентации плоскости соответствует вектор нормали  $\mathbf{n}_j$ . Концы этих векторов попадают на верхнюю единичную полусферу. Угловое расстояние любой точки верхней полусферы от ближайшего вектора нормали не превышает 1 – 2 градуса. Полное число дискретных ориентаций (нормалей) – около 91 000.

Для каждой из выбранных ориентаций  $\mathbf{n}_j$  возможных плоскостей группирования осуществлялся поиск среди плоскостей, имеющих различные расстояния от начала координат. В цикле вычислялись проекции  $d_{ji} = \mathbf{n}_j \cdot \mathbf{r}_i$  радиус-векторов  $\mathbf{r}_i = (x_i, y_i, z_i)$ , проведенных к каждому  $i$ -му гипоцентру, на направление  $\mathbf{n}_j$ . Далее из значений проекций  $d_{ji}$  формировался динамически список, сортирующийся при заполнении по возрастанию. Затем, в списке отбирались события: с минимальным значением проекции – первое в списке и  $m$ -ное в списке. Вычислялась разность  $|d_{jM} - d_{j1}|$ . Здесь штрихи означают, что номер землетрясения соответствует не номеру по каталогу, а номеру в отсортированном списке. Если разность оказывалась меньше  $2D$ , то все  $M$  первых гипоцентра оказываются на одной плоскости. Если же это не так, то тестировалась разность  $|d_{jM+1} - d_{j2}|$  и т.д. до конца списка. Если, в общем случае,

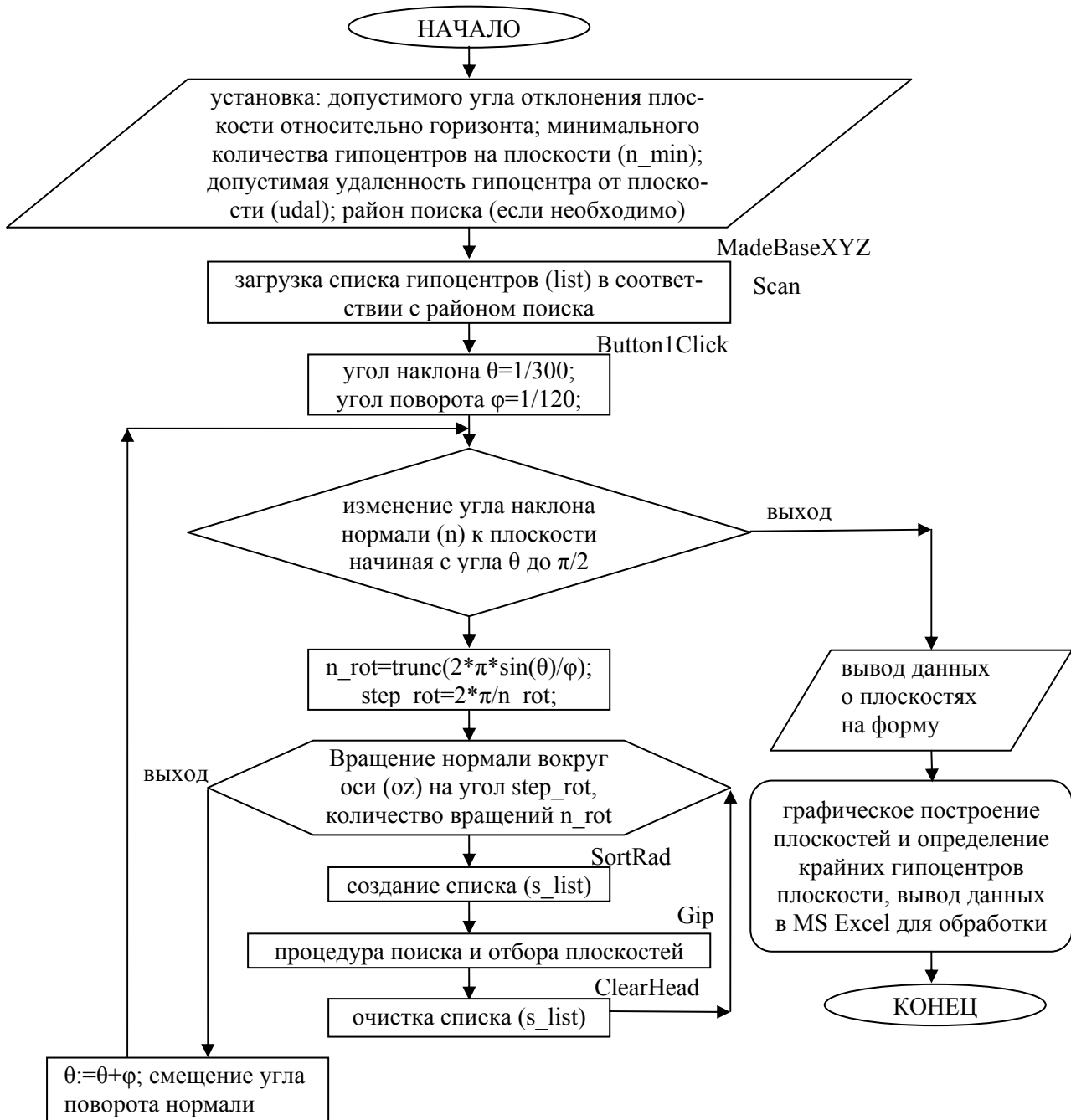


Рис. 1. Блок-схема алгоритма для поиска плоскостей группирования гипоцентров

оказывалось, что гипоцентры с номерами, начиная с  $t'$  и по  $(t+M-1)'$  принадлежали с точностью  $\pm D$  одной плоскости, список расширялся. Далее проверялось, не принадлежит ли плоскости ещё и гипоцентр, следующий в списке, то есть с номером  $(t+M)'$ . Для всей группы землетрясений, принадлежащих плоскости далее по методике, описанной в работе [1], определялись параметры плоскости: компоненты вектора нормали, расстояние от плоскости до начала координат, которые заносились в список плоскостей.

Завершающая часть программы вне всех циклов обрабатывает список найденных плоскостей. При этом предусмотрена возможность объединения близких по ориентации плоскостей. Настройки позволяют устанавливать максимальное количество точек плоскостей, отличных друг от друга, при этом пользователю предоставлена возможность выбора между объединением двух плоскостей в одну, либо без этого. Объединение здесь означает, что параметры объединённой плоскости должны вычисляться по объединению множества гипоцентров, найденных для каждой из плоскостей. По умолчанию выполняется исключение только повторяющихся записей. Такие могут быть, поскольку одни и те же группы гипоцентров, попадающих на одну плоскость, могут выявляться для двух, и более близ-

ких по ориентации векторов нормали  $n_j$ . Программа способна отбирать плоскости и по положению их в пространстве, то есть, предусмотрена возможность использования критерия отбирающего (или исключающего) плоскости по ориентации вектора нормали. Например, можно исключать плоскости, с ориентацией близкой к горизонтальной.

#### Код программы (язык Object Pascal)

```

procedure ClearHead();
//процедура очистки динамического списка,
//для всех типов используемых списков
//одинакова, с изменением только в типе
//переменной del и головы списка head3
var
del:TKord; //TKord структура для хранения
//списка гипоцентров
begin
//head3 переменная головы списка
while head3<>nil do
//цикл до конца списка гипоцентров
begin
del:=head3;
head3:=head3^.next;
Dispose(del);
//процедура очистки динамической переменной из
//оперативной памяти компьютера, для избегания
//утечки памяти
end;
end;

procedure MadeBaseXYZ(FileName:string);
//процедура загрузки базы гипоцентров
var
str1:string;
q:integer;
bl:boolean;
begin
ClearHead3(); //очистка списка координат, если
//загрузка производится повторно
head3:=nil;
AssignFile(f,FileName);
//подключение файла с гипоцентрами
{$I-} //отключение компилятора для
//обработки ошибок вручную
Reset(f); //открытие на чтение файла
//при ошибке открытия, выход из приложения
if IOResult<>0 then Application.Terminate;
{$I+} //включение компилятора
n_zps:=0; //счетчик гипоцентров
While not EoF(f) do //считывание файла до конца
begin
bl:=Scan(); //функция обработки строки
if (bl=true) and (Form1.CheckBox2.Checked=true)
then Inc(n_zps)
else if (Form1.CheckBox2.Checked=false) then
Inc(n_zps);
//пропускание тактов процессора
Application.ProcessMessages; end;
CloseFile(f); //закрытие файла
end;

procedure Button1Click(Sender: TObject);
//процедура запуска расчетов
var
tet,r,nf,r1,st,lam,dist1,grad:real;
i,ni:integer;
xyz:TKord;
begin
cm:=0;
head:=nil;
head4:=nil;
tet:=1/300; //угол наклона
r:=1/120; //угол поворота
while (tet<pi/2+0.0001) do
begin
nf:=2*pi*sin(tet)/r; //количество вращений
if (nf>=0) then ni:=round(nf)
else ni:=trunc(nf); //округление
r1:=2*pi/ni; //шаг вращения
for i:=0 to ni do //вращение нормали плоскости
begin
st:=sin(tet); //вычисление координат нормали с
//новыми углами наклона
lam:=i*r1;
nx:=st*cos(lam); ny:=st*sin(lam); nz:=cos(tet);
//указатель на начало списка с гипоцентрами с
//последующим его перебором
xyz:=head3;
while (xyz<>nil) do
begin
//расстояние от начала плоскости до гипоцентра
dist1:=nx*xyz.x+ny*xyz.y+nz*xyz.z;
SortRad(xyz.Number,dist1);
//заполнение списка с номерами гипоцентров
//в списке и удаленность от плоскости
xyz:=xyz^.next;
Application.ProcessMessages;
end;
grad:=(arccos(nz)*180)/pi;
//вычисление угла наклона плоскости
if grad>=25 then begin
Gip(); //процедура группирования плоскостей
Inc(cm); end;
ClearHead(); //очистка списка для сортировки
end;
tet:=tet+r; //смещение угла поворота нормали
Application.ProcessMessages;
end;
if head2<>nil then PlaneBruter;
//если список с плоскостями не
//пуст, то выводим найденные плоскости
//на форму для дальнейшего анализа
end;

```

**procedure SortRad(num: integer; ch: real);**

```
//процедура сортировки гипоцентров
var
add,now,prev:TGroup;
begin
New(add); //создание динамической переменной
add^.numb:=num; //заполнение полей
add^.dist:=ch; //структуры TGroup
now:=head;
prev:=nil;
//поиск в списке указателя, с большим значением,
чем у добавляемого, для вставки по возрастанию
while (now<>nil) and (add.dist>now^.dist) do
begin //сохранение предыдущего указателя
prev:=now; //для определения места вставки
now:=now^.next; //элемента add
end; //добавление в начало
if prev=nil then begin
add^.next:=head;
head:=add;
end //добавление в любое другое место списка
else begin //с перенаправлением указателей
add^.next:=prev^.next;
prev^.next:=add;
end;
end;
```

**procedure Scan():boolean;** //обработка строки

```
var
add:TKord;
//структура для списка гипоцентров
adding:boolean;
begin
adding:=false;
new(add);
ReadLn(f, add^.Number, add^.Long, add^.Lat,
add^.Energy, add^.x, add^.y, add^.z);
//если есть ограничения по местоположению, то
//проверяем, и при удачном сопоставлении
//разрешаем добавление в список
if (Form1.CheckBox2.Checked=true) and
(add.Long>=StrToFloat(Form1.Edit10.Text)) and
(add.Long<=StrToFloat(Form1.Edit12.Text)) and
(add.Lat>=StrToFloat(Form1.Edit11.Text)) and
(add.Lat<=StrToFloat(Form1.Edit13.Text)) then
adding:=true
//если нет ограничений, то просто разрешаем
//добавление в список гипоцентров
else if (Form1.CheckBox2.Checked=false) then
adding:=true;
if (adding=true) then begin
//добавление элемента с перенаправлением
add^.next:=head3; //указателей
head3:=add;
end;
Result:=adding;
end;
```

**procedure Gip;**

```
//процедура формирования плоскостей
Label GipLabel;
var
iw,jw,k,r,n_min:integer;
dism,d:real;
ans1,ans2:TGroup;
tt:string;
sta:TGr;
begin
d:=StrToFloat(Form1.Edit1.Text); //максимальное
//отклонение гипоцентра от плоскости
n_min:=StrToInt(Form1.Edit2.Text); //минимальное
//количество гипоцентров на плоскости
iw:=1; jw:=n_min;
GipLabel: //метка для возврата
while (jw<=(n_zps)) do
//перебор отсортированного списка гипоцентров
begin
//считывание из динамического списка
//записей с шагом n_min
fRec(jw,ans1); fRec(iw,ans2);
//если разность расстояний не подходит под наше
//отклонение, то сдвиг номеров и переход в начало
if (abs(ans1.dist-ans2.dist)>=d) then begin Inc(iw);
Inc(jw); goto GipLabel; end;
//проверка следующего значения в списке пока
//разница между расстояниями не станет больше
//заданной d - отклонения
while (abs(ans1.dist-ans2.dist)<d) do
begin
Inc(jw); //увеличение количества гипоцентров
//считывание следующего элемента в списке, при
//условии, что список не закончился
if (jw<=n_zps) then fRec(jw,ans1)
else Break;
end;
k:=jw-iw; dism:=0; tt:=',';
for r:=1 to k do
begin
fRec(iw,ans2);
//запись гипоцентров в список по возрастанию
Stack(ans2.numb);
dism:=dism+ans2.dist;
iw:=iw+1;
end;
//начало списка, сформированного
sta:=head4; //процедурой Stack
while (sta<>nil) do
begin //запись в строку гипоцентров из списка
tt:=tt+IntToStr(sta.numb)+',';
sta:=sta^.next;
end;
Application.ProcessMessages;
Spisok(k,dism/k,tt); //записи плоскости в список
iw:=jw; jw:=jw+n_min-1;
end; end;
```

### Тестирование программы

Тестирование выполнялось для поиска плоскостей группирования гипоцентров событий камчатской сейсмоактивной зоны. На рис. 2 представлены плоскости найденные по анализу каталога включающего около 700 событий, энергетического класса 11 и выше. Программа выполнялась около 90 секунд. Высокая производительность и скорость выполнения обусловлены использованием динамических типов данных и многопоточности приложения. Даже при анализе до 1000 гипоцентров программа выполняется за приемлемое время, которое напрямую зависит от используемого оборудования и введенных значений для группирования.

Алгоритм и программа не гарантируют, что найденные плоскости группирования соответствуют реально существующим в природе структурам и не являются проявлением случайного совпадения. Для выявления среди найденных плоскостей признаков реально существующих пространственных структур должны быть выполнены дополнительные исследования. Результаты этих исследований – являются предметом отдельной публикации.

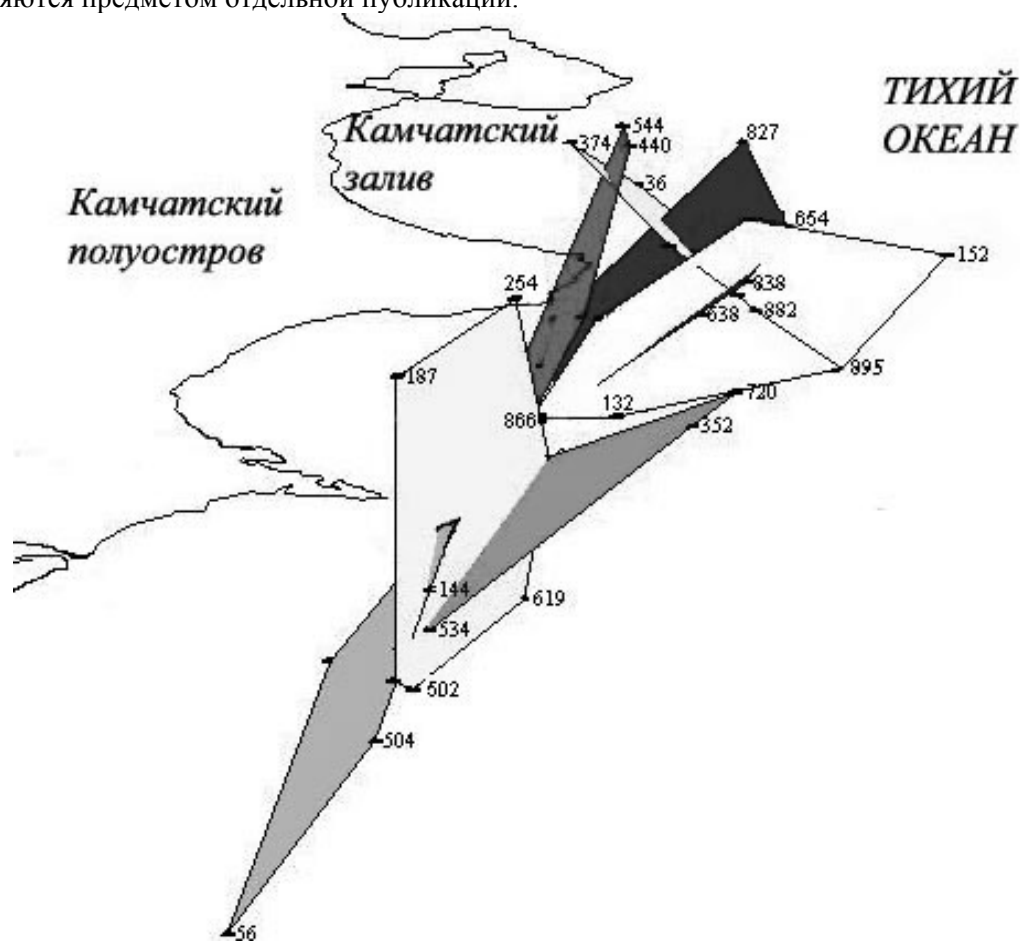


Рис. 2. Пример плоскостей, найденных по гипоцентрам событий, класса 12 и выше (региональный каталог землетрясений КФ ГС РАН)

### Заключение

Составлена компьютерная программа на языке Object Pascal, в среде программирования Delphi 7, реализующая алгоритм поиска групп гипоцентров, принадлежащих пространственным плоскостям. Представлены также блок-схема алгоритма и программный код наиболее важных процедур ее исполнения. На компьютере с процессором Intel Pentium Dual-Core E5200 данная программа обрабатывает список до 1000 гипоцентров в среднем за 2-3 минуты. Программа реализована на языке Object Pascal.

### Список литературы

1. Кролевец А.Н. Плоскости разломов Кроноцкого землетрясения 5 декабря 1997 г. Геофизический мониторинг Камчатки. Материалы научно-технической конференции 17 – 18 января 2006 г. Петропавловск - Камчатский: Изд-во Оттиск, 2006. С. 32-39.
2. Кролевец А.Н. Алгоритм поиска плоскостей группирования гипоцентров землетрясений. Геофизический мониторинг и проблемы сейсмической безопасности Дальнего Востока России: в 2 томах. Труды региональной научно-технической конференции. Т. 2. Петропавловск - Камчатский: Изд-во Оттиск, 2008. С. 29-33.