

Сейсмический калькулятор

Seismic Calculator

Инструкции пользователя

**ООО «Лаборатория Приезжева»**

**Copyright 2016, ООО «Лаборатория Приезжева». Все права защищены.**

Этот документ содержит конфиденциальную и служебную коммерческую тайну ООО «Лаборатория Приезжева» и не могут быть скопированы или сохранены в информационно-поисковой системы, переданной, используемой, распространять, переводить или передавать в любой форме или любыми средствами, электронными или механическими, полностью или частично без письменного разрешения владельца авторских прав.

**Старт:**

**Seismic attributes->**

**Seismic Calculator**

**Сейсмический калькулятор работает следующим образом:**

- 1. Определяемая процедура используется для вычисления каждого измерения (sample) в результирующем кубе.**
- 2. Имеется несколько предопределяемых переменных.**
  - **I, J, K – индекс для текущего точки измерения (sample).**
  - **X, Y, Z – координаты для текущего точки измерения (sample).**
  - **nI, nJ, nK – размер скользящего окна с имеющимися смежными примерами для вычисления вокруг текущих точек измерения.**
  - **Cube – массив со значениями в скользящем окне. Размер массива равен Cube[ncube][nI,nJ,nK], где ncube – это номер куба в списке кубов используемых для вычислений.**
  - **Seis0, Seis1, ..., Seis'ncube' – предопределяемые значения кубов, используемых как ввод для текущего примера (значения в центре скользящего окна).**
- 3. Результирующий куб процедуры может быть только один для одного прогона.**
- 4. Результирующий куб будет виртуальным.**
- 5. Имя результирующего куба будет определяться как переменная с '=' (TokenKind.ASSIGN). Например, если мы используем строку процедуры:**  
*cubeResult=100;*  
**это позволит построить новый виртуальный куб с именем "cubeResult" со значением константы =100.**
- 6. Если используется несколько строк для процедуры, в этом случае только последняя '=' ASSING будет определять результат процедуры. Например, если используется строка процедуры:**  
*d=(X-1000)\*(x-1000)+(y-3000)\*(y-3000)*

```
cubeResult=sqrt(d);
```

это позволяет создавать только один новый виртуальный куб с именем "cubeResult"

7. Процедура может иметь несколько предопределяемых функций `sin()`, `cos()`, `abs()`, `pow(,)`, `atan2(,)`, `sqrt()`, `exp()`, `log()`, `sign()`, `rand()`, `iif(,,)`.
8. Дополнительно могут использоваться все функции подобные `Math.Sqrt()`, ...
9. С# конструкции `for(;;) {}`, `if() else`, и все другие подобные могут быть использованы.
10. Ниже приводится пример вычисления простого атрибута когерентности:

```
double sum=0;  
int n=0;  
for(int k=0; k< nK; k++)  
{  
    for(int i=0; i< nI; i++)  
        for(int j=0; j< nJ; j++)  
            {  
                int ii =nI/2+1;  
                int jj =nJ/2+1;  
                var diff=abs(Cube[0][i,j,k]- Cube[0][ii,jj,k]);  
                var norm=abs(Cube[0][i,j,k]+ Cube[0][ii,jj,k]);  
                if(norm==0)norm=1000000; // protect to division to zero  
                var rez= (diff/norm);  
                sum+=rez;  
                n++;  
            }  
        }  
    coherence=sum/n;
```

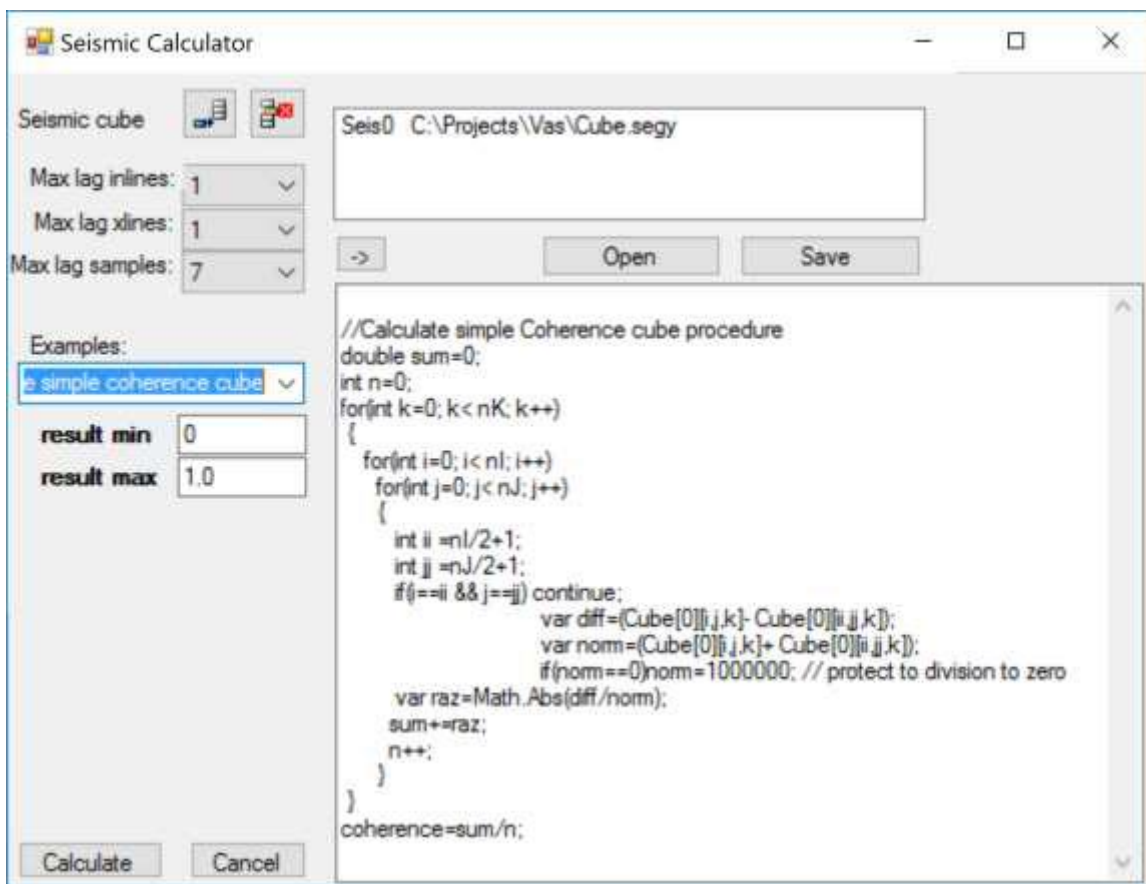


Рис. 1: Сейсмический калькулятор