

**Surface Calculator**

**User Manual**

**IPLAB**

**Copyright © 2016 IPLAB. All rights reserved.**

This work contains the confidential and proprietary trade secrets of IPLAB and may not be copied or stored in an information retrieval system, transferred, used, distributed, translated or retransmitted in any form or by any means, electronic or mechanical, in whole or in part, without the express written permission of the copyright owner.

**Start:**

**Surface attributes->**

**Surface Calculator**

**The surface calculator works according to the following rules:**

- 1. The defined procedure is used to calculate every node in result surface property.**
- 2. There are several predefined variables.**
  - **I, J – index for current surface node.**
  - **X, Y – coordinates for current surface node.**
  - **Z – depth or time for current node (property index =0)**
  - **nI, nJ – size of the moving window with available neighboring nodes for calculation around current surface node.**
  - **Prop – array all properties for current surface node.**
  - **Surf – array with values in moving window. Size of the array is Surf [nI,nJ][nK] where nK -is number properties in surface.**
  - **Variable with "PropName" allows to get values of this property for current node.**
- 3. Result surface property of the procedure can be only one for one run.**
- 4. Result property will be added to the surface.**
- 5. Name of the result property will be defined by variable with '=' (TokenKind.ASSIGN). For example, if we have a procedure string:**  
*surfResult=100;*  
**it will create a new property with the name "surfResult" with constant value=100.**
- 6. If we use several strings for procedure, only last ASSING will define procedure result. For example, if we have procedure string:**  
*d=(X-1000)\*(x-1000)+(y-3000)\*(y-3000)*  
*surfResult=sqrt(d);*  
**it creates only one new property with name "surfResult"**
- 7. Procedure can use several predefined function sin(), cos(), abs(), pow(), atan2(), sqrt(), exp(), log(), sign(), rand(), iif(,,).**
- 8. Additionally all function like Math.Sqrt(), ... can be used.**

9. C# constrictions like for(;;) {}, if() else, and any other can used.
10. The example below demonstrates the calculation of simple surface fault detection attribute:

```
//Fault detection attribute
double sum=0;
int n=0;
int k=0; //used property
for(int i=0; i< nI; i++)
    for(int j=0; j< nJ; j++)
    {
        int ii =nI/2+1;
        int jj =nJ/2+1;
        if(i==ii && j==jj) continue;
        var diff=abs(Surf[i,j][k]-Surf[ii,jj][k]);
        var norm=abs(Surf[i,j][k]+Surf[ii,jj][k]);
        if(norm==0)norm=1000000; // protect to division to
zero
        var raz=diff/norm;
        sum+=raz;
        n++;
    }
faultDetect=sum/n;
```

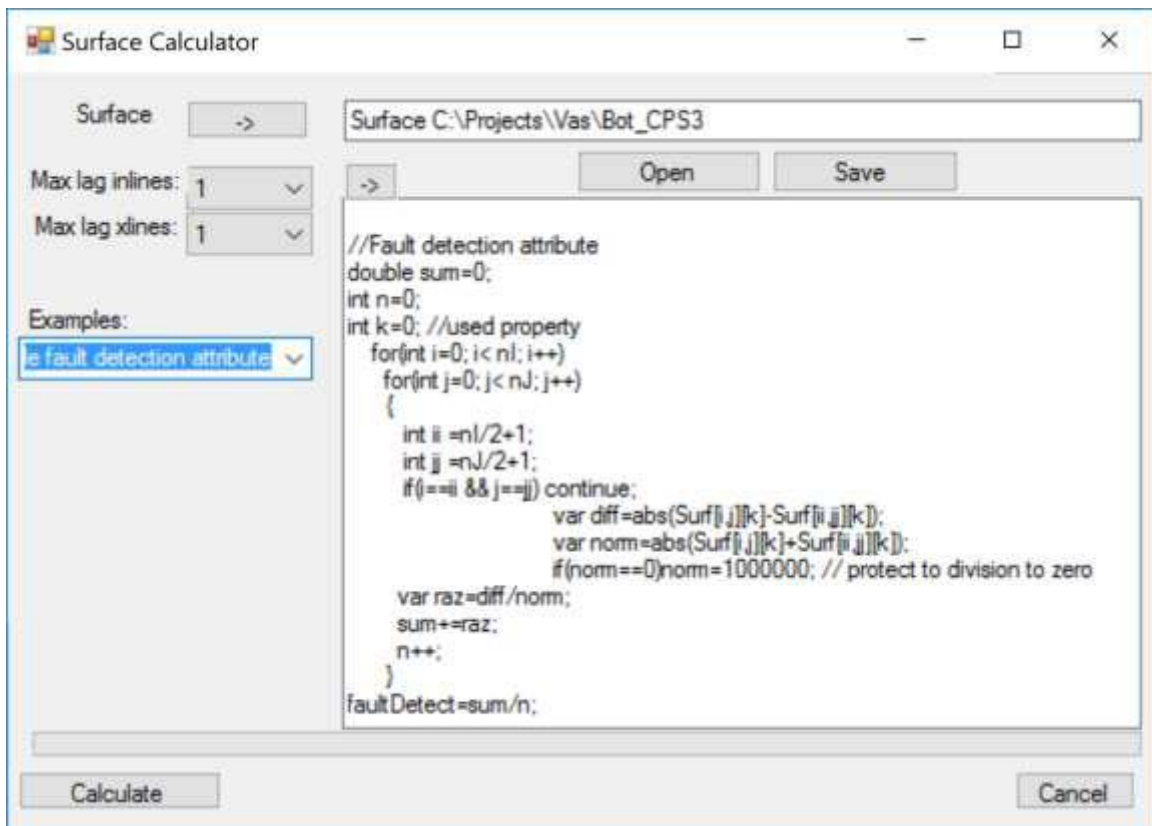


Figure 1: Surface calculator