

## Розширення функціональних можливостей пакету програм Surfer засобами мовного середовища GS Scripтер

*Москаль Н. М., Тарнавський В. Л., Заяць О. С.*  
Національний університет  
“Львівська політехніка”  
[moskal@polynet.lviv.ua](mailto:moskal@polynet.lviv.ua)

**The ways of extending functional capabilities of the Software package  
Surfer by means of the Software GS Scripтер.**

*Moscal Natalia, Tarnavskij Volodimir, Zajats Olexandr*  
National University  
“L`vivska Politechnica”

### **Abstract**

*Many users dealing in the fields of geology, geophysics, ecology have to solve problems connected with interpolation of basic data (which were got either through irregular network or specified in the shape of isolines) into regular network (GRID) with the following calculation aiming at space modeling and analysis. For these tasks the use of GIS-software appears very suitable but it demands considerable financing. Therefore it is still topical and profitable to use available specialized software packages for solving some applied problems. The article presents the text and description of ready for copying runtime-module designed in programming environment GS Scripтер. The module widens the standard set of functions of software package SURFER; it is orientated towards calculation of altitude of point on the base of digital regular model of relief (GRIP) and 2D, 3D visualization of basic data and calculation results.*

### **Вступ**

Багатьом користувачам, особливо в галузі геології, геофізики, екології, океанології, необхідно розв'язувати задачі, пов'язані з інтерполяцією вихідних даних (отриманих по нерегулярній моделі або заданих в вигляді ізоліній) на регулярну мережу. Часто при цьому необхідна наступна побудова ізоліній, тривимірна візуалізація рельєфу та проведення розрахунків з метою просторового моделювання та аналізу. Засоби для розв'язку перелічених задач включені в спектр можливостей повнофункціональних інструментальних ГІС: Arc/Info для робочих станцій і ПК (на базі триангуляційної – TIN та регулярної – GRID моделей), локалізованої (російськомовної) версії MapInfo Profesional 7 (TIN), модуль розширення Spatial Analist для Arc View (TIN). Проте придбання та освоєння цих систем вимагає немалих фінансових затрат і витрат часу, висуває істотні вимоги до конфігурації апаратних засобів (ЕОМ).

Застосування для цієї мети спеціалізованих пакетів програм, зокрема такого відомого та доступного як Surface Mapping System (Surfer), в багатьох випадках дає

більш економічний розв'язок проблеми. Проте Surfer не задовільняє всіх вимог користувачів. Наприклад, Surfer буде цифрову регулярну модель рельєфу (ЦМР-Grid) та надає можливість її візуалізації в ортогональній (карта ізоліній) та перспективній проекціях, але серед стандартних функцій цього пакету немає жодної розрахункової задачі на базі ЦМР. Користувачам часто ж необхідно обчислювати: висоти точок, для яких відомі планові координати; кути нахилу, кривину та інші морфометричні характеристики рельєфу; зони видимості і невидимості, тощо.

В даній публікації пропонується прокоментований текст працездатного runtime-модуля (**Altitude**), який реалізовано в середовищі програмування GS Scripтер 2.00, що поставляється разом із Surfer v.6.04. Представлений нами програмний продукт надає можливість обчислювати на базі цифрової моделі рельєфу (створеної раніше або в сеансі роботи з runtime-модулем) висоти точок, для яких відомі планові координати; здійснювати 2D та 3D візуалізацію вихідних даних та результатів розрахунків. Сподіваємось, що ознайомившись із структурою та поясненням до тексту даної програми кінцеві користувачі Surfer не тільки зможуть використати її в своїй роботі, але й отримають "ключ" для розробки власних прикладних програм.

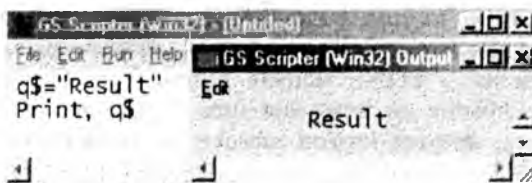
## 1. Основні відомості про мовне середовище GS Scripтер.

Щоб побудувати програмну систему, орієнтовану на розв'язок конкретних фахових задач, необхідно сумісно використовувати комерційний пакет програм та відповідне середовище розробки. Наприклад: для PC MapInfo з цією метою застосовується Map Basic, для ArcView – Avenue, тощо. Для Surfer'a роль інструмента для розробки прикладних програм користувача виконує мовне середовище GS Scripтер.

GS Scripтер – Basic-подібний інтерпретатор, який автоматично розташовується в каталог пакету під час інсталяції Surfer'a. Запуск мовного середовища здійснюється виконавчим модулем *GSMAC32.EXE*. Після старту GS Scripтер на екрані з'являється меню з чотирьох пунктів (мал. 1).

GS Scripтер дозволяє підготовляти тексти програм – скрипти (script), а також завантажувати і виконувати ці програмні модулі та стандартні макрокоманди, котрі використовуються в системі Surfer. Програми на мові GS Scripтер – це текстові файли, які створюються в вікні Edit GS Scripтер'a, або в вікні редактора Windows Notepad чи будь-якого іншого ASCII-редактора. Якщо script-файл відображений в вікні Edit, то його можна запустити на виконання командою **Start** з меню **Run** (або клавішею F5 – як і в більшості Бейсік-подібних мовних середовищ). Скрипт може містити команди, які забезпечують запуск модулів, що відповідають стандарту OLE 2.0.

Запуск пунктів меню Edit та Run спричиняє відкриття вікон – Edit та Output відповідно.



Вікно Edit це стандартний текстовий редактор ОС Windows, який дозволяє відкривати, створювати, редагувати і зберігати скрипти. Вивідне вікно Output показує результат виконання команд Print, які можуть використовуватися в прикладній програмі для друку значень змінних (див. на мал. 1 приклад виведення значення змінної q\$), та з метою візуального контролю в процесі відлагодження програми (для повного покрокового контролю в процесі

відлагодження програми краще використовувати трасування – Run > Trace).

## Мал. 1. Вигляд головного меню мовного середовища GS Scripter.

На GS Scripter'і можна реалізувати обчислювальні алгоритми будь-якої складності, оскільки синтаксис мови включає: спеціальні функції (математичні, для обробки символьних даних, роботи з датами, перетворення типів, тощо), оператори для організації розгалужених алгоритмів (`goto`, `if...then...else`, `while...wend`, `for...next`, `gosub`), функції та команди для організації інтерактивної надбудови (*MsgBox*, *InputBox*), стандартні макрокоманди Surfer'a. Назви макрокоманд максимально асоціюються із відповідними пунктами меню та опціям (назвами параметрів) діалогових вікон Surfer. Наприклад, для тривимірної візуалізації GRID вживається макрокоманда:

***object.MapSurface(InGrid, LineDir, xLineColor, xLineStyle, xLineWidth, yLineColor, yLineStyle, yLineWidth, zLineColor, zLineStyle, zLineWidth, zLevelFile, VisOptions, Border, ColorFile, ColorDir, ShowColorScale, BaseOptions, BaseHt, BaseLineColor, BaseLineStyle, BaseLineWidth, ID, ScaleID).***

Користувачу, який має мінімальний досвід роботи із Surfer, зрозуміло, що подані в дужках макрокоманди *object.MapSurface(...)* параметри дозволяють задати значення, аналогічні до значень параметрів діалогового вікна **Surface Plot** (запускається в Surfer з меню **Map > Surface...**).

GS Scripter має систему допомоги (пункт меню **Help**), яка містить інформацію достатню для освоєння можливостей мовного середовища та написання прикладної програми. Сумісне використання пакету програм Surfer та GS Scripter дозволяє не тільки реалізувати алгоритми розв'язку прикладних задач, але й побудувати власне інтерактивне середовище для організації діалогу з користувачем.

## 2. Опис функціональних можливостей та структури runtime-модуля (Altitude).

Основні моменти, з якими доводиться мати справу при створенні проблемно-орієнтованої програми загальновідомі: організація інтерактивного середовища для діалогу, ввід та зберігання даних, реалізація алгоритму для розв'язку задачі, візуалізація та аналіз результатів. Програма Altitude демонструє розв'язання цих проблем засобами мови GS Scripter.

### 2.1. Функціональні можливості програми

На мал.2 подано блок-схему, а на мал.3(а,б) – вихідний текст runtime-модуля Altitude, до функцій якого відносяться:

- **Побудова регулярної ЦМР.** Якщо для досліджуваної території така модель вже існує (побудована заздалегідь чи придбана), то програмі повідомляється тільки ім'я відповідного \*.grd-файлу.
- **Обчислення висот для масиву точок, заданих плановими координатами.** Проводиться шляхом інтерполяції відміток вузлів прямокутника (квадрата) із GRID.
- **Генерація табличного файлу результатів** обчислень висот Result.dat та вікон для візуалізації вмісту файлів вхідних даних та результатів.
- **Генерація файлів робочих просторів (Workspace)** для побудови карти ізоліній та тривимірної візуалізації рельєфу. В робочих просторах генерується шар, в якому виконується відображення положення опрацьовуваних точок на карті та на поверхні.
- **Організація виходу в середовище пакету Surfer, та відкриття (Plot) вікон** для візуалізації файлів даних та робочих просторів з метою надання користувачу можливості

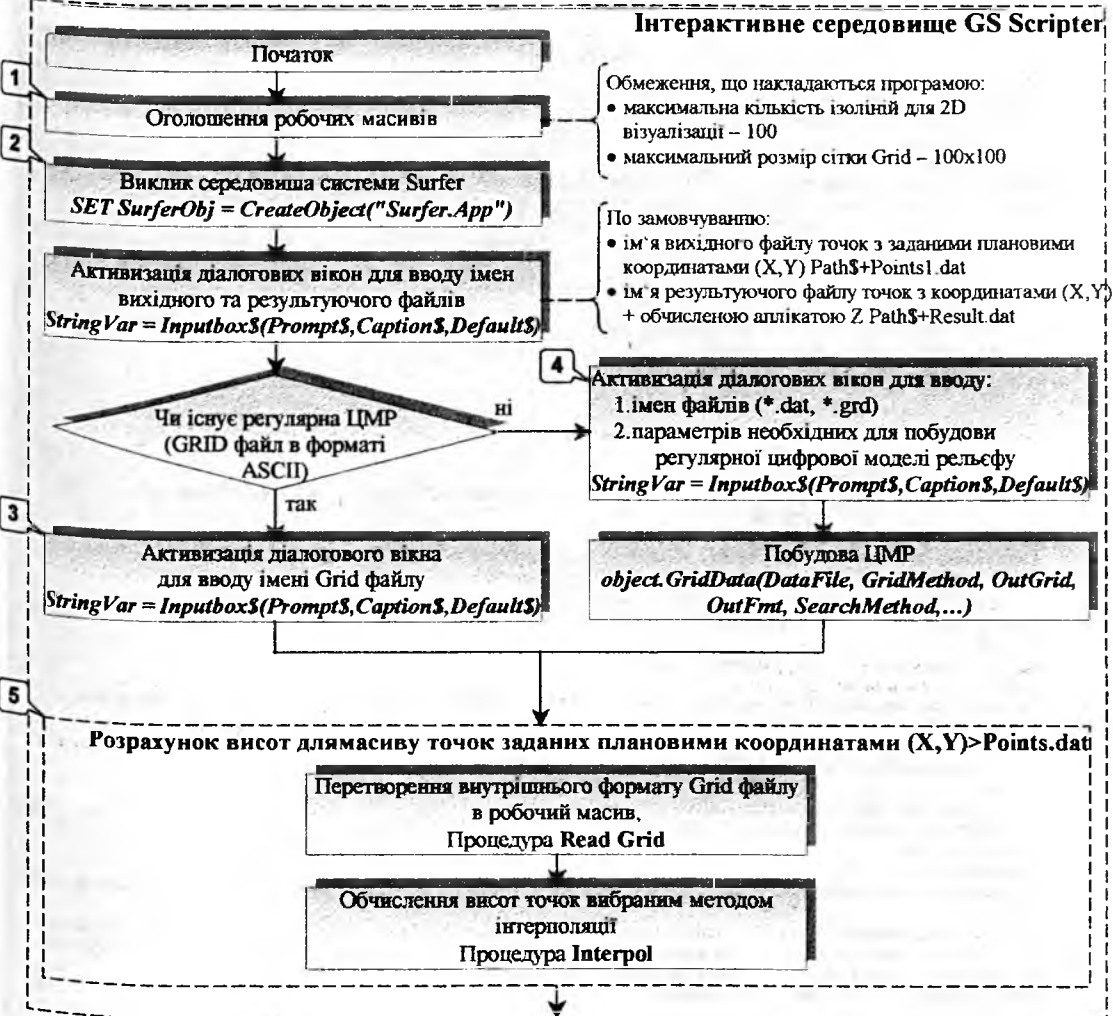
проводити візуальну оцінку результатів та вносити зміни в стильове оформлення вмісту графічних вікон.

В блок-схемі та тексті програми значком позначено відповідні етапи реалізації загального алгоритму розв'язку задачі. Для кожного блоку в схемі наведено ім'я макрокоманди Surfer або написаної нами в рамках runtime-модуля підпрограми-процедури, за допомогою якої виконується певна частина алгоритму (стандартні макрокоманди Surfer'a відмічені жирним курсивом, а підпрограми-процедури – жирним шрифтом).

Для формальних параметрів макрокоманд в блок-схемі збережено мнемоніку імен, що застосовується в системі Help GS Scripter'a. Тому, співставляючи синтаксис макрокоманди в блок-схемі і в тексті програми (тут вказуються значення фактичних параметрів) та користуючись Help SG Scripter'a можна на прикладі тексту Altitude навчитись використовувати макрокоманди для написання власних програм.

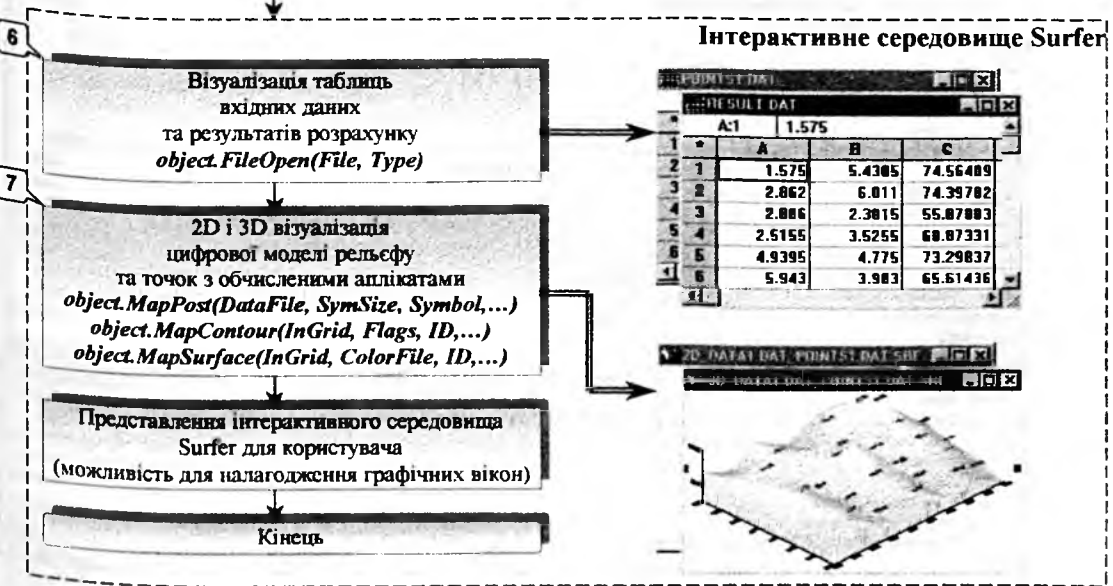
В тексті кожної підпрограми-процедури (Мал. 3б) прокоментовано її функціональне призначення. Виклик підпрограми з тексту головного модуля (Мал. 3а) виконується тоمانдою **GoSub <ім'я підпрограми (фактичні параметри)>**.

**Інтерактивне середовище GS Scripter**



- Обмеження, що накладаються програмою:
- максимальна кількість ізоліній для 2D візуалізації – 100
  - максимальний розмір сітки Grid – 100x100
- По замовчуванню:
- ім'я вихідного файлу точок з заданими плановими координатами (X, Y) Path\$+Points1.dat
  - ім'я результуючого файлу точок з координатами (X, Y) + обчисленою аплікатою Z Path\$+Result.dat

**Інтерактивне середовище Surfer**



№	A	B	C
1	1.575	5.4305	74.56409
2	2.862	6.011	74.39782
3	2.886	2.3015	55.87883
4	2.5155	3.5255	68.87331
5	4.9395	4.775	73.29837
6	5.943	3.983	65.61436



Мал. 2. Блок-схема прикладної програми (runtime модуля Altitude)

## Основний модуль програми Altitude

```

1 Dim A(10)
2 Dim Grid(100,100)
  Set Surf = CreateObject("Surf.App")
  Surf.DocMaximize()
  Path$=CurDir$()
  Path$=InputBox$("Enter path to your files:", "Enter path", Path$)
  NameDataFile2$=InputBox$("Enter the name of the input data file (with coordinates X, Y) for which you would like to define altitude
  Z):", "Enter the name of data (*.dat) file", "Points1")
  NameDataFile3$=InputBox$("Enter the name of the output data file (containing X, Y, Z coordinates of the points):", "Enter the name of
  data (*.dat) file", "Result")
3 Result%=MsgBox("Do you have file of regular digital model (grid file)?", 36, "Do you have (*.grd) file?")
  If Result%<6 Then
    NameGrdFile$=InputBox$("Enter the name of the grid file of regular digital model:", "Enter the name of grid (*.grd) file", "Out1")
    OutGrdFile$=Path$+"\"+NameGrdFile$
    ID$=NameGrdFile$+"_grd "+NameDataFile2$+".dat"
4 EndIf
  If Result%<7 Then
    NameDataFile1$=InputBox$("Enter name of input data file with coordinates (X,Y,Z) for construction digital model:", "Enter the name of
    data (*.dat) file", "Data1")
    InDataFile1$=Path$+"\"+NameDataFile1$+".dat"
    NameGrdFile$=InputBox$("Enter the name of grid file (which will be created from "+NameDataFile1$+".dat):", "Enter the name of grid
    (*.grd) file", "Out1")
5 OutGrdFile$=Path$+"\"+NameGrdFile$
    GridMethod$=InputBox$("0=inverse distance 1=kriging 2=minimum curvature 3=polynomial regression 4=radial basis function
    5=stepard 6=triangulation 7=nearest neighbor:", "Enter grid method (value 0..7)", "1")
    SearchMethod$=InputBox$("0=None 1=Simple 2=Quadrant 3=Octant:", "Enter search method (value 0..3)", "2")
    If Surf.GridData(InDataFile1$, GridMethod=Val(GridMethod$), SearchMethod=SearchMethod$, OutFmt=2, OutGrid=OutGrdFile$)=0
    Then End
    ID$=NameDataFile1$+".dat "+NameDataFile2$+".dat"
    EndIf
    InterpolMethod$=InputBox$("Enter interpolation method for altitude Z 0=BiLinear 1=Medium Weight:", "Enter interpolation method
    (value 0,1, ...)", "1")
    If InterpolMethod$="1" Then
      StepIn$=InputBox$("Enter power:", "Enter inverse distance option", "1")
    EndIf
    InDataFile2$=Path$+"\"+NameDataFile2$+".dat"
    OutDataFile$=Path$+"\"+NameDataFile3$+".dat"
    GoSub ReadGrid
    GoSub Interpol
    SmoothAmount$=InputBox$("Enter smooth amount for CONTOUR MAP 0=Low 1=Medium 2=High:", "Enter smooth amount (value
    0..2)", "1")
    Minimum$=InputBox$("Enter level (Minimum value of Z is "+Str$(MinZ)+"): ", "Enter minimum contour level", Trim$(Str$(MinZ)))
    Maximum$=InputBox$("Enter level (Maximum value of Z is "+Str$(MaxZ)+"): ", "Enter maximum contour level", Trim$(Str$(MaxZ)))
    ContInterval$=InputBox$("Enter contour interval (default contour interval is 1):", "Enter interval:", "1")
6 FileSheetInData2=Surf.FileOpen(InDataFile2$,1)
    If FileSheetInData2=0 Then End
    FileSheetInData3=Surf.FileOpen(OutDataFile$,1)
    If FileSheetInData3=0 Then End
7 Surf.FileNew()
    Surf.MapPos(OutDataFile$, LabCol=2, LabPosType=2, LabNDig=2, SymSize=0.06, Symbol=12, LabType=0, FontColor="R255 G000
    B000", SymColor="R255 G000 B000")
    GoSub Str1
    GoSub Leveling
    If Surf.MapContour(OutGrdFile$, Minimum=Minimum$, Maximum=Maximum$, Interval=ContInterval$, Flags=3, LevelFile=LvlFile$,
    SmoothAmount=SmoothAmount$, ID=ID$)=0 Then End
    Kill(LvlFile$)
    Surf.Deselect()
    Surf.Select()
    Surf.ViewZoomSelected()
    Surf.MapOverlayMaps()
    Surf.FileSaveAs(Path$+"3d "+ID$+".srf",0)
    Surf.FileNew()
    GoSub Str2
    GoSub Leveling
    If Surf.MapSurface(OutGrdFile$, ColorFile=1, VFile$, ColorDir=3, ID=ID$)=0 Then End
    Kill(LvlFile$)
    Surf.MapPos(OutDataFile$, LabCol=2, LabPosType=2, LabNDig=2, SymSize=0.06, Symbol=12, LabType=0, FontColor="R255 G000
    B000", SymColor="R255 G000 B000")
    Surf.Deselect()
    Surf.Select()
    Surf.ViewZoomSelected()
    Surf.MapOverlayMaps()
    Surf.FileSaveAs(Path$+"3d "+ID$+".srf",0)
End

```

Мал. 3а Текст головного модуля програми Altitude

```

ReadGrid:
N=0
Open OutOrdFile$+"_grd" For Input As #1
Line Input #1,String$ Header
Line Input #1,String$ nx,ny
Gosub StrToArr
nx=A(1)
ny=A(2)
Line Input #1,String$ MinX,MaxX
Gosub StrToArr
MinX=A(1)
MaxX=A(2)
Line Input #1,String$ MinY,MaxY
Gosub StrToArr
MinY=A(1)
MaxY=A(2)
Line Input #1,String$ MinZ,MaxZ
Gosub StrToArr
MinZ=A(1)
MaxZ=A(2)
hx=(MaxX-MinX)/(nx-1)
hy=(MaxY-MinY)/(ny-1)
NPoints=0
While EOF(1)=0
Line Input #1,String$
Gosub StrToArr
If N<>0 Then
For i=1 To N
Npoints=Npoints+1
Ycur=Fix(Npoints/nx)
If (Npoints/nx)-Fix(Npoints/nx)>0 Then
Ycur=Ycur+1
EndIf
Xcur=Npoints-(Ycur-1)*nx
Grid(Ycur,Xcur)=A(i)
Next
EndIf
Wend
Close #1
Return
StrToArr:
Q=0
N=0
IFirst=1
For i=1 To Len(String$)
If Mid$(String$,i,1)="" Then
Q=Q+1
EndIf
Next
If Q<>Len(String$) Then
String$=Trim$(String$)+" "
For i=1 To Len(String$)
If Mid$(String$,i,1)=Chr$(32) Then
n=n+1
A(n)=Val(Mid$(String$,IFirst,i-IFirst))
IFirst=i+1
EndIf
Next
EndIf
Return
    
```

**Процедура перетворення символічної стрічки GRID в числовий формат**

```

Str1:
q$=Chr$(34)
sp$=Chr$(32)
s1$="1 "+q$+"Red Brown"+q$+sp$+q$+"Solid"+q$+"0 "+q$
s3$=q$+sp$+q$+"White"+q$+sp$+q$+"Solid"+q$+"2"
Return
Str2:
q$=Chr$(34)
sp$=Chr$(32)
s1$="1 "+q$
s3$=q$+sp$+q$+"Solid"+q$+"0"+q$+"Black"+q$+sp$+q$+"White"
+q$+sp$+q$+"None"+q$+"2"
Return
    
```

**Службова процедура для визначення діапазону кольорів для 2D.**  
**Службова процедура для визначення діапазону кольорів для 3D.**

```

Check:
If c=0 Then
Error=MsgBox("0 points from "+NameDataFile2$+" dat interpolated!",16,"Error")
End
    
```

**Процедура для перевірки просторової сумісності масиву точок (X,Y) та ЦМР.**

```

Interpol:
Open InDataFile2$ For Input As #1
Open OutDataFile$ For Output As #2
While EOF(1)=0
Line Input #1,String$ X,Y
Gosub StrToArr
X=A(1)
Y=A(2)
c=0
If MinX<=X And X<=MaxX And MinY<=Y And Y<=MaxY Then
c=c+1
X1=Fix((X-MinX)/hx)+1
Y1=Fix((Y-MinY)/hy)+1
X_=(X-MinX)-(X1-1)*hx
Y_=(Y-MinY)-(Y1-1)*hy
If InterpolMethod$="0" Then
GoSub InterpolBilinear
EndIf
If InterpolMethod$="1" Then
GoSub InterpolMedWeight
EndIf
Write #2,X,Y,Z
EndIf
Wend
Close #1
Close #2
GoSub Check
Return
InterpolBilinear:
a0=Grid(Y1,X1)
a1=(Grid(Y1,X1+1)-a0)/hx
a2=(Grid(Y1+1,X1)-a0)/hy
a3=(Grid(Y1+1,X1+1)+a0-Grid(Y1,X1+1)-Grid(Y1+1,X1))/(hx*hy)
Z=a0+a1*X_+a2*Y_+a3*X_*Y_
Return
InterpolMedWeight:
R1=Exp(Val(Stepin$)*log(Sqr(X_*X_*Y_*Y_)))
R2=Exp(Val(Stepin$)*log(Sqr(X_*X_+(Y_*Y_*(Y_*Y_))))
R3=Exp(Val(Stepin$)*log(Sqr((X_*hx)*(X_*hx)+(Y_*hy)*(Y_*hy))))
R4=Exp(Val(Stepin$)*log(Sqr((X_*hx)*(X_*hx)+Y_*Y_)))
If R1=0 then
Z=Grid(Y1,X1)
Else
If R2=0 then
Z=Grid(Y1+1,X1)
Else
If R3=0 then
Z=Grid(Y1+1,X1+1)
Else
If R4=0 then
Z=Grid(Y1,X1+1)
Else
Z=Grid(y1,x1)/R1+Grid(y1+1,x1)/R2+Grid(y1+1,x1+1)/R3+Grid(y1,x1+1)/R4
P=1/R1+1/R2+1/R3+1/R4
Z=Z/p
EndIf
EndIf
EndIf
EndIf
Return
    
```

**Процедура для розрахунку висот точок заданими плановими координатами. базові дані для інтерполяції – регулярна ЦМР (GRID)**

**Процедура реалізації методу білінійної інтерполяції**

**Процедура реалізації методу середнього вагового. вага представляється степенню для вагового коефіцієнта 1/d^n**

```

Leveling:
Lv1File$=Path$+"Lvl1"
Header1$="LVL2"
Header2$="Level Flags LColor LStyle LWidth FPGColor FBGColor
FPattern FMode"
Inter=Val(ConInterval$)
Min=Fix(Val(Minimum$))-Inter
Max=Chr(Val(Maximum$))+Inter
Nlvl=(Max-Min)/Inter-1
Open LvlFile$ For Output As #1
Print #1,Header1$
Print #1,Header2$
For i=1 To Nlvl
R=Fix(255/Nlvl*i)
G=255
B=0
R$=Right$(Str$(R),Len(Str$(R))-1)
G$=Right$(Str$(G),Len(Str$(G))-1)
B$=Right$(Str$(B),Len(Str$(B))-1)
Print #1,Min+i*Inter,";s1$+R"+R$+"G"+G$+"B"+B$+s3$
Next
    
```

**Генерація файлу \*.lvl – кольорової гамми для "виднявки рельєфу" та кольорової гамми для представлення осей X,Y в 3D візуалізації.**

## 2.2. Дані

Для функціонування програми необхідно мати такі дані:

- а) Для випадку, коли ЦМР існує, необхідно вказати ім'я \*.GRD файлу, що містить значення відміток у вузлах регулярної моделі (за замовчуванням в полі імені діалогового вікна пропонується – Out1.grd), та файл координат (X,Y) точок для яких необхідно обчислити висоту (за замовчуванням – Points1.dat). Обидва файли повинні бути представлені в ASCII- форматі.
- б) Для випадку коли GRID немає, для її побудови програма додатково опитує ім'я файлу координат (X,Y,Z) точок (за замовчуванням – Data1.dat), на базі яких слід будувати ЦМР та параметри побудови моделі (метод інтерполяції, спосіб вибору точок, крок сітки, тощо).

Для організації вводу програмою генерується система діалогів (див. приклади діалогів на мал. 4). Для цього в тексті програми застосовуються макрокоманди (*MsgBox, InputBox*).

## 2.3. Результати

В результаті функціонування програми отримуємо:

- ASCII-файл координат X,Y,Z точок, який зберігається на диску (за замовчуванням – Result.dat) та придатний для вводу в будь-яке програмне середовище;
- файли робочих просторів (2D\_Data1.dat\_Points1.dat.srf, 3D\_Data1.dat\_Points1.dat.srf);
- чотири відкриті Plot вікна Surfer, в яких відображається зміст цих файлів.

ASCII-файли відображаються в вікні Plot у вигляді таблиць, доступних для редагування. Робочий простір (SRF-файл) містить інформацію, яка необхідна для відновлення в вікні Plot графічного зображення точок, ліній полігонів, поверхонь, підписів, а також асоційованих об'єктів, які створюються спеціальними інструментами. В графічному вікні Plot користувач може проводити всі види редагування що передбачені для цього типу вікна: міняти стильове оформлення об'єктів, вносити зміни в значення стрічок табличних файлів, проводити операції редагування, які доступні для Plot вікон карти – 2D та поверхні – 3D. SRF-файл зберігається і може бути використаний в наступних сеансах роботи з Surfer. При його відкритті Surfer відображає карту або поверхню тільки при умові існування відповідного \*.grd файлу.

## 2.4. Список та опис призначення макрокоманд Surfer, які використані в програмі:

- object.DocMaximize(void)* – збільшення активного вікна документа;
- object.GridData(DataFile, GridMethod, OutGrid, OutFmt, SearchMethod,...)* – створення цифрової моделі рельєфу (\*.GRD – файл);
- object.FileOpen(File, Type)* – відкриття файлів та відповідних їм вікон в середовищі Surfer;
- object.FileNew(Type)* – створення нових графічних вікон – Plot, вікон робочих таблиць – Worksheet або вікон текстового редактора – Editor;
- object.MapPost(DataFile, SymSize, Symbol, FontColor, LabType,...)* – нанесення на карту точкових об'єктів за даними з файлу, який містить координати X, Y або X, Y, Z;
- object.MapContour(InGrid, Flags, Minimum, Maximum, Interval, LevelFile, SmoothAmount, ID,...)* – створення карти ізоліній, використовуючи дані GRID-файлу;
- object.Select(ID)* – вибір об'єктів у графічному вікні Plot;
- object.Deselect(ID)* – відміна виділення вибраних об'єктів у графічному вікні Plot;



**object.ViewZoomSelected(void)** – зміна масштабу відображення об'єктів у графічному вікні Plot;

**object.MapOverlayMaps(void)** – створення єдиного об'єкту для всіх вибраних елементів одного вікна. В єдиний об'єкт можна об'єднати будь-який ряд просторово сумісних об'єктів: карту ізоліній і шар точкових об'єктів, карту ізоліній і 3D поверхню, тощо;

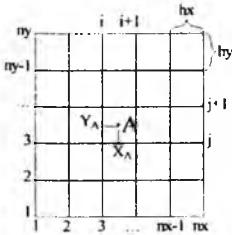
**object.FileSaveAs(File, Backup)** – збереження вмісту активного графічного вікна Plot;

**object.MapSurface(InGrid, ColorFile, ColorDir, ID,...)** – створення тривимірного зображення рельєфу,

використовуючи дані GRID-файлу.

## 2.5. Програмна реалізація алгоритму обчислення висот точок за регулярною цифровою моделлю – GRID.

Загальний вигляд регулярної моделі можна представити схематично таким чином:



Сітка представляється двовірним масивом відміток в вузлах:

$G [ 1 .. ny , 1 .. px ]$ , де

$px$  – кількість вузлів вздовж осі X,  $ny$  – вздовж осі Y,

$hx, hy$  – крок сітки вздовж осі X та Y відповідно.

Відмітку точки A що потрапляє в клітинку сітки знаходимо використовуючи відмітки найближчих чотирьох вузлів. В runtime-модулі Altitude реалізовано два методи: середнього вагового (процедура **InterpolMedWeight**) та метод білінійної інтерполяції (процедура **InterpolBiLinear**). Обидві процедури викликаються із підпрограми **Interpol**, яка організовує діалоги для вводу параметрів (наприклад, при виборі методу середнього вагового параметром є значення показника степеня  $p$ ) та формування масиву результатів (Result.dat).

Для визначення відмітки в точці A за методом середнього вагового використовуються відомі співвідношення:

$$Z_A = \frac{\sum_{l=i-1}^{i+1} \sum_{k=j-1}^{j+1} G[k,l] \cdot P_{kl}}{\sum_{l=i-1}^{i+1} \sum_{k=j-1}^{j+1} P_{kl}}, \text{ де } P_{kl} = \frac{1}{R^p(A, G[k,l])}, \text{ } R(A, G[k,l]) - \text{відстань від точки A до}$$

вузла сітки  $G[k,l]$ ;  $p$  – показник степеня.

Для методу білінійної інтерполяції відмітка точки A –  $Z_A = a_0 + a_1 \cdot X_A + a_2 \cdot Y_A + a_3 \cdot X_A \cdot Y_A$ ,

де  $a_0, a_1, a_2, a_3$  – невідомі коефіцієнти, котрі знаходяться за наступним співвідношеннями:

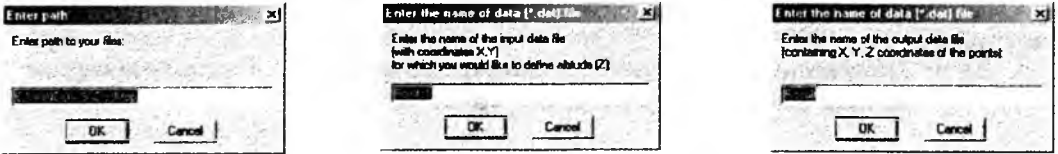
$$a_0 = G[j,i]; \quad a_1 = (G[j,i+1] - G[j,i]) / hx; \quad a_2 = (G[j+1,i] - G[j,i]) / hy; \quad a_3 = (G[j+1,i+1] + G[j,i] - G[j,i+1] - G[j+1,i]) / (hx \cdot hy).$$

Runtime-модуль залишається відкритим для реалізації будь-яких інших методів, що можуть цікавити дослідників. Для цього слід написати підпрограму-процедуру для реалізації методу та організувати її виклик із процедури **Interpol** (наприклад, за зразком виклику **InterpolMedWeight**) командою **GoSub <ім'я>**.

## 3. Алгоритм роботи з програмою Altitude

Послідовність дій користувача для запуску та роботи в середовищі runtime-модуля повинна бути наступною:

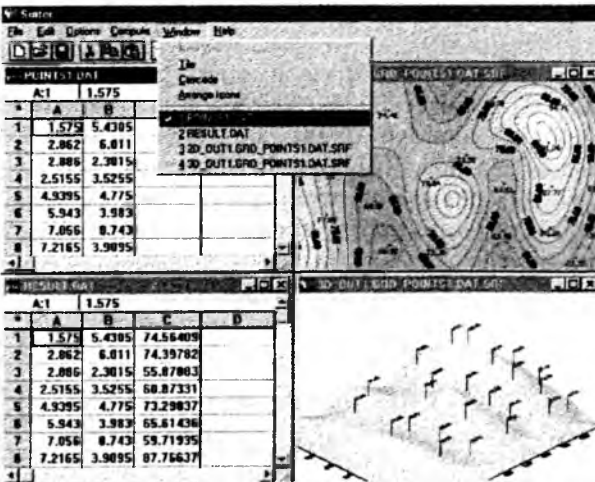
1. **Запустити виконавчий модуль GS Scriptor:** знайти та відмітити в програмному каталогі Surfer виконавчий модуль *GSMAC32.EXE*, натиснути ліву клавішу миші (або функціональну клавішу ENTER).
2. **Помістити текст runtime-модуля в вікно Edit Scriptor'a:** (File>Open<ім'я модуля.bas>) та запустити його на виконання (Run>Start або натиснути F5).
3. **Надавати відповіді на запитання, які появляються в діалогових вікнах інтерактивного середовища, що генеруються в сеансі роботи з модулем.** Можна просто підтверджувати відповіді, які підставляються в полях діалогових вікон за замовчуванням



(мал.4).

Мал.4. Послідовність діалогів при виконанні алгоритму від пункту 1 до 2.

4. **Переглянути табличні файли та робочі простори** стандартними засобами Surfer. Вікна для перегляду (мал.5) модуль генерує без втручання користувача. При необхідності, можна внести зміни в таблиці даних та налагодити на свій смак графічні зображення карти і просторової візуалізації ЦМР. Якщо проводиться редагування, то слід зберегти таблиці і робочі простори, в вікнах яких вносилися зміни (File>Save).
5. **Завершити роботу з Surfer. Завершити роботу з Script.**



Мал. 5. Вигляд інтерактивного середовища Surfer після завершення обчислень висот точок.

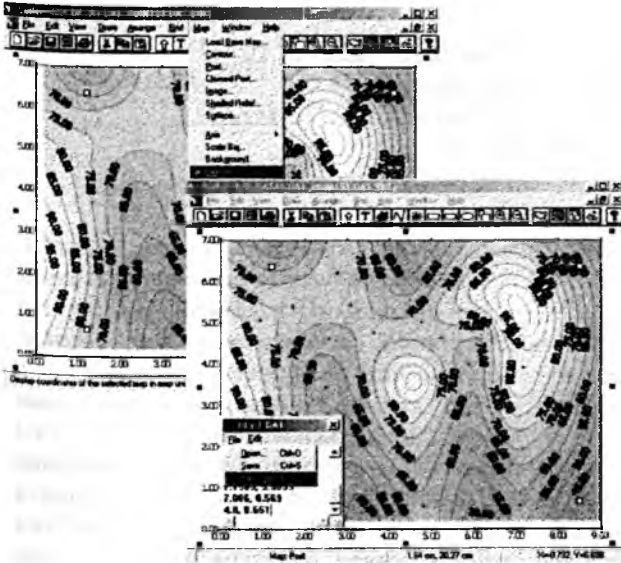
(Вікна відображають: Points1.dat – координати (X,Y) точок; Result.dat – файл координат X,Y,Z точок, для яких знаходили висоту; 2d\_Data1.dat\_Points1.dat.srf – карту ізоліній, побудовану на базі ЦМР (\*.GRD файлу), та точки з файлу Result.dat з підписаними відмітками; 3d\_Data1.dat\_Points1.dat.srf – 3D

відображення ЦМР і точок з Result.dat).

#### 4. Технологія 3D дигіталізації в середовищі Surfer з використанням runtime-модуля Altitude.

Серед стандартних можливостей пакету Surfer є функція цифрування по карті ізоліній, яка доступна в вікні графічного типу карти (Plot). Для активізації функції необхідно вибрати опцію **Digitize** з меню **Map**, після чого в вікні появляється особливий курсор у формі перехрестя. Наведення курсора на певне місце на карті і натискання лівої клавіші миші призводить до визначення координат (X,Y) положення курсора та запису значень в робочий файл (мал.6). Використовуючи далі runtime-модуль Altitude для точок з цього файлу, можна обчислити їх висоту.

Отже метою 3D-дигіталізації в середовищі Surfer з використанням runtime-модуля Altitude слід діяти так:



1. Запустити пакет програм Surfer;
2. Побудувати для існуючого файлу регулярної ЦМР (\*.grd) карту ізоліній: **Map>Contours...**;
3. Вибрати режим цифрування по карті: **Map>Digitize**;
4. Провести цифрування по карті та зберегти файл результатів: **File>Save As<Digit.dat>**. При збереженні на диск можна змінити назву цього файлу;
5. Вийти з середовища

пакету Surfer: **File>Exit**;

6. Запустити середовище GS Scripтер **GSMAC32.EXE**;
7. Завантажити runtime-модуль для виконання (див. алгоритм роботи з модулем).
8. Переглянути та при необхідності налагодити стиль відображення оцифрованих точок на карті чи 3D-зображенні.
9. Завершити роботу з Surfer.
- 10.

## Висновки

Представлена в статті програма Altitude є працездатним runtime-модулем, який можна використати підготувавши текст в вікні Edit SG Scripтер'a, або отримавши від авторів ([moskal@polynet.lviv.ua](mailto:moskal@polynet.lviv.ua)). Користувачі також можуть використати фрагменти та окремі процедури тексту модуля для програмування власних задач в мовному середовищі SG Scripтер.

Рецензію на статтю склав проф. Дорожинський О.Л.

## **Розширення функціональних можливостей пакету програм Surfer засобами мовного середовища GS Scripтер**

*Москаль Н. М., Тарнавський В. Л., Заяць О. С.*

Національний університет

“Львівська політехніка”

[moskal@polynet.lviv.ua](mailto:moskal@polynet.lviv.ua)

### **Анотація**

*Багатьом користувачам, особливо в галузях геології, геофізики, екології, необхідно розв'язувати задачі, пов'язані з інтерполяцією вихідних даних (отриманих по нерегулярній мережі або заданих в вигляді ізоліній) на регулярну мережу та проведення розрахунків з метою просторового моделювання та аналізу. Застосування з цією метою повнофункціональних інструментальних ГІС є привабливим, однак вимагає значних фінансових затрат на придбання та витрат часу на освоєння. Тому, для окремих прикладних задач залишається актуальним і виявляється економічно вигідним використання відомих і доступних спеціалізованих пакетів програм. В статті пропонується готовий runtime-модуль, написаний в мовному середовищі для програмування – GS Scripтер, який дозволяє розширювати стандартний перелік функцій спеціалізованого пакету програм Surfer : надає можливість обчислювати на базі цифрової моделі рельєфу(створеної раніше або в сеансі роботи runtime-модуля) аплікати точок, що задані плановими координатами; здійснювати 2D та 3D візуалізацію вихідних даних та результатів розрахунків.*