ЗМІСТ

Вступ

. Метод сіток

.1 Постановка задачі

.2 Розв’язання задачі Діріхле для рівняння Лапласа методом сіток у прямокутної області

. Текст програми

# ВСТУП

Значна кількість задач фізики і техніки приводять до диференціальних рівняння в приватних похідних (рівняння математичної фізики). Усталені процеси різної фізичної природи описуються рівняннями еліптичного типу. Точні розв'язки крайових задач для еліптичних рівнянь вдається отримати лише в окремих випадках. Тому ці завдання вирішують в основному наближено. Одним з найбільш універсальних та ефективних методів, отримали в даний час широке поширення для наближеного рішення рівнянь математичної фізики, є метод кінцевих різниць або метод сіток.

Метод сіток або, інакше, метод кінцевих різниць найбільш поширений для наближеного рішення диференціальних рівнянь в приватних похідних. Основна ідея методу сіток полягає в тому, що диференціальне рівняння, початкові і граничні умови замінюються системою звичайно-різницевих рівнянь алгебри, наближено представляють дану крайову задачу. Розглянемо застосування методу сіток до вирішення задач теплопровідності на прикладі двомірної задачі.

Для лінійних рівнянь в приватних похідних одним з очевидних прийомів є застосування методу сіток. Тоді рівняння в приватних похідних замінюється системою лінійних алгебраїчних рівнянь з числом невідомих, залежних від числа взятих точок. Основною перешкодою тут є недостатній обсяг пам'яті існуючих машин. Для нелінійних рівнянь в приватних похідних застосування методу сіток призводить, як правило, до непереборним труднощам.

Застосування методу сіток дозволяє звести диференціальну крайову задачу до системи нелінійних в загальному випадку алгебраїчних рівнянь щодо невідомих вузлових значень функцій.

Особливості в застосуванні методу сіток до рівнянь в приватних похідних викликані насамперед тим,що для рівнянь різних типів можуть бути коректно поставлені завдання різного виду з початковими чи граничними умовами. Метод сіток тісно пов'язаний з типом диференціального рівняння і для рівнянь різних типів потрібно обирати, взагалі кажучи, сітки різних видів.

Однак при застосуванні методу сіток до розглянутої електростатичної задачі виникають деякі принципові труднощі. З найбільшою повнотою було досліджено застосування методу сіток до рівнянь еліптичного типу.

Це не служить перешкодою до застосування методу сіток для областей з неідеальними кордонами. Однак для обґрунтування потрібні інші, більш тонкі теореми. Зокрема рівномірна метрика, якою ми користуємося у цій главі, стає мало природною.

# 1. МЕТОД СІТОК

Методи скінченних різниць, методи сіток - чисельні методи розв'язку інтегро-диференціальних рівнянь алгебри, диференціального, інтегрального числення, основані на заміні диференціальних операторів різницевими операторами, інтегралів - сумами, а функцій неперервного аргументу - функціями дискретного аргументу. Така заміна приводить до системи, взагалі кажучи, нелінійних алгебраїчних рівнянь <http://uk.wikipedia.org/w/index.php?title=%D0%9D%D0%B5%D0%BB%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D1%96\_%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F&action=edit&redlink=1>, які зрештою зводяться до лінійної системи <http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\_%D0%BB%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B8%D1%85\_%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D1%8C> деяким ітераційним <http://uk.wikipedia.org/wiki/%D0%86%D1%82%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F> методом. Якщо початкова задача має вигляд



,



де  - циліндрова область інтеграції  - межа області ,  - її основа, - шукана вектор-функція і - задані вектор-функції,  - просторовий векторний аргумент,  і  - оператори (не обов'язково обмежені), то найпростіша схема інтеграції початкового рівняння має вигляд:

,



Тут  - сіткова функція, що є розв'язком різницевого рівняння, - різницеві оператори, залежні від параметрів ,  сітки, ,  - сіткова область, що апроксимує деяким чином область , - її границя і - сіткові функції, що апроксимують функції і  відповідно. Окремим випадком схеми (2) є схема з вагами, коли  ,  - ваговий коефіцієнт.

Схема (2) називається двошаровою, оскільки вона зв'язує між собою значення  і  різницевого розв'язку на двох тимчасових шарах ; можливі також і багатошарові схеми. Якщо оператор , де  - одиничний оператор, оборотний, то схема (2) може бути представлена у вигляді, що розв'язується  де оператор  називаєстья оператором кроку різницевої схеми і враховує крайові умови, а  - функція, залежна від і . Кажуть, що оператор , залежний від параметра , апроксимує (приблизно) оператор , якщо  при . Тут  - деяке еталонне сімейство функцій, на якому перевіряється апроксимація (наприклад, сімейство достатньо гладких функцій <http://uk.wikipedia.org/wiki/%D0%93%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0\_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F>).

Схема (2) називаєтсья коректною, або стійкою, якщо , де  означає норму оператора  в деякому просторі. (див. простір абстрактний у функціональному аналізі <http://uk.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%96%D1%80\_%D0%B0%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%B8%D0%B9\_%D1%83\_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC%D1%83\_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7%D1%96&action=edit&redlink=1>), яке може залежати від .

Схема (2) апроксимує рівняння <http://uk.wikipedia.org/wiki/%D0%A0%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F> (1), якщо  і . Для лінійних систем рівнянь встановлені теореми збіжності, що стверджують, що збіжність різницевого рішення до розв'язку початкового рівняння виходить з апроксимації і коректності (стійкості) різницевої схеми.

Якщо властивості апроксимації, стійкості і збіжності мають місце лише при деякому співвідношенні між параметрами сітки , , де , то їх називають умовними. Якщо ж ці властивості справедливі при будь-якому співвідношенні між  і , то їх називають абсолютними.

Схема (2) називається явною, якщо , і неявною, якщо .

Схеми, що абсолютно збіжні, існують лише в класі неявних схем. Як правило, при відповідному виборі параметрів схеми (наприклад вагових коефіцієнтів) неявні схеми є абсолютно стійкими, вони допускають скільки завгодно великий крок . Але перетворення оператора  ускладнює алгоритм <http://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC>. У разі одновимірних задач неявні схеми реалізують методом факторизації <http://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%8F>; вони є достатньо економічними. Для багатовимірних задач неявні економічні схеми одержують за допомогою дробових кроків методу <http://uk.wikipedia.org/w/index.php?title=%D0%94%D1%80%D0%BE%D0%B1%D0%BE%D0%B2%D0%B8%D1%85\_%D0%BA%D1%80%D0%BE%D0%BA%D1%96%D0%B2\_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4&action=edit&redlink=1>, який зводить багатовимірні задачі до послідовності одномірних або простіших задач.

Для розв'язання стаціонарних задач застосовують метод стаціонування (встановлення), в якому стаціонарний розв'язок розглядається як межа нестаціонарного розв'язку із стаціонарним (або що встановлюються) краєвими умовами. Відповідно до цього стаціонарну задачу вирішують ітераційним методом, аналогічним різницевому методу інтеграції (2). На відміну від нестаціонарного випадку, оператор  для ітераційного процесу повинен бути сильно стійкий, тобто повинен задовольняти умові , . При розв'язку нелінійних задач, особливо в механіці суцільного середовища, застосовують комбінації схем інтеграції з ітераційними методами (т.з. ітерації по нелінійності).

Суть методу полягає в наступному. Область безперервної зміни аргументів, замінюється дискретним безліччю точок (вузлів), яке називається сіткою або гратами. Замість функції неперервного аргументу розглядаються функції дискретного аргументу, визначені у вузлах сітки і звані сітковими функціями. Похідні, що входять в диференціальне рівняння і граничні умови, замінюються різницевими похідними, при цьому крайова задача для диференціального рівняння замінюється системою лінійних або нелінійних алгебраїчних рівнянь (сіткових або різницевих рівнянь). Такі системи часто називають різницевими схемами. І ці схеми вирішуються відносно невідомою сіткової функції.

Застосування методу сіток до вирішення гіперболічного рівняння вимагає, щоб вибрані кроки сітки по осях х і у задовольняли певним співвідношенням.

Застосування методу сіток для дослідження стійкості просторово-періодичних рухів //Чисельні методи динаміки в'язкої рідини - Новосибірськ ІТПМ СВ АН СССP, 1979 - С.

Схема для моделювання рівняння. Розглянемо застосування методу сітки спрямованої дії для вирішення конкретних завдань розробки нафтових родовищ. Для визначення температурного поля при закачуванні гарячого агента в пласт потрібно інтегрування рівняння (2.1) або (2.3) при певних початкових і граничних умовах. Нехай початкова температура пласта дорівнює нулю, іншими словами, відлік температури ведеться від її початкового значення. На кордоні х 0 температура закачуваного агента приймається TK 100 З і передбачається, що на деякій відстані від нагнітальної галереї температура залишається рівною її початкового значення.

Аналіз питань, пов'язаних з застосуванням методу сіток для вирішення конкретних рівнянь, проведемо для хвильового рівняння, при вирішенні якого в достатній мірі проявляються властиві методу сіток особливості.

Електрична схема для моделювання рівняння. Слід зазначити, що при вирішенні практичних завдань із застосуванням методу сітки спрямованої дії необхідно з'єднати між собою точки з однаковим потенціалом. Отже, сітка спрямованої дії має більш універсальне призначення, ніж звичайна електрична сітка. Вона містить операційні підсилювачі аналогових обчислювальних машин для виконання різних математичних операцій.

У роботі В. Ф. Баклановской розглянуті теоретичні питання, пов'язані із застосуванням методу сіток для чисельного розв'язання рівнянь одновимірної нестаціонарної фільтрації газу.

Для лінійних рівнянь в приватних похідних одним з очевидних прийомів є застосування методу сіток. Тоді рівняння в приватних похідних замінюється системою лінійних алгебраїчних рівнянь з числом невідомих,залежних від числа взятих точок. Основною перешкодою тут є недостатній обсяг пам'яті існуючих машин. Для нелінійних рівнянь в приватних похідних застосування методу сіток призводить, як правило, до непереборним труднощам.

Згідно з цими рекомендаціями, розроблено метод розв'язання системи диференціальних рівнянь (3.1) і (3.4), суть якого полягає в застосуванні методу сіток для просторової координати.

Рішення відповідних пружно завдань про полях місцевих деформацій здійснено в основному або обчислювальними засобами, або експериментально із застосуванням методу сіток, чутливих покриттів або муару.

Розглянемо застосування методу сіток до вирішення внутрішньої задачі Діріхле для рівняння Лапласа - рівняння еліптичного типу.

Порівняємо числа арифметичних операцій, необхідних для вирішення розглянутих електростатичних задач як методом інтегральних рівнянь, так і методом сіток. При застосуванні методу сіток до вирішення польової задачі необхідно штучно обмежити безмежну область, зайняту полем (див. гл.

На підставі проведеного аналізу та накопиченого досвіду можна відзначити, що при розрахунках несталих неізотермічних режимів трубопровідного транспорту газів в одномірної постановці найбільш ефективний метод сіток, причому перевага дають неявні схеми. Крім того, застосування методу сіток доцільніше, якщо використовується схема інтегрування не нижче другого порядку.

Розрахунки газотранспортних систем для оперативного управління припускають використання одновимірних рівнянь енергії і теплопровідності. У цьому розділі зроблені висновки про доцільність застосування методу сіток з неявній схемою для спрощених задач, коли до співвідношень гідродинаміки і нестаціонарного, теплообміну для середовища додається одномірне рівняння теплопровідності для стінки трубопроводу. У цьому випадку доводиться вирішувати задачу відшукання функцій р, Т, Тст, М методом послідовних наближень.

Зміна величини етах характеризує поступове зниження із зменшенням температури граничної пластичної деформації. Розв’язання відповідних пружно-пластичних задач про полях місцевих деформацій здійснюється в основному обчислювальними засобами або експериментально із застосуванням методу сіток, тензо-чутливих покриттів або муару. З наведених даних випливає, що з пониженням температури спостерігається більш раннє падіння величини бк, ніж руйнівних напружень. Це пояснюється чутливістю розкриття тріщини до пониження температури, пов'язаної з граничною пластичною деформацією і полегшенням проростання тріщин.

# 1.1 Постановка задачі

Постановка задачі: необхідно знайти неперервну функціюu=u(x,y),

що задовольняє всередині деякої кінцевої областіD еліптичному рівнянню

Лапласа:



діріхле задача лаплас

а на границі Г цій області - умові:(x,y)/Г= f(x,y)

деf(x,y) - задана неперервна на Г функція.

Вважаємо, що Г- простий кусково-гладкий контур.

Задачу пошуку розв’язання еліптичного рівняння 2-го порядку, за умовою, що розв’язок на границі приймає наперед задане значення, називається задачею Діріхле або першою крайовою задачею.

Для розв’язання:

а) оберемо квадратну сітку з кроком h;

б) для укладання різницевого рівняння скористуємося формулою;

в) визначаємо шаблон сітки. Множина вузлів , значення сіткової функції uik яких входять в різницеве рівняння, називається шаблоном. В різницевий аналог рівняння :



наявні елементи, що розташовані справа, зліва, зверху та знизу від елементу uik (рис. 1), тобто, шаблон має форму хреста і носить назву шаблон-хрест;



Рисунок 1 - Шаблон „хрест” для еліптичного рівняння

г) перетворимо рівняння (2) до виду (3):



Отримали систему лінійних алгебраїчних рівнянь (3). Доведено, що система (3) завжди сумісна і має єдине рішення. Розв’язок системи можна отримати методом Гауса. При наявності великої кількості вузлів доцільно використовувати метод прогонки або ітераційні методи.

Специфічним ітераційним методом для розв’язку системи (3) є метод осереднення Лібмана, який є варіантом методу Гауса-Зейделя:

а) обираємо початкові наближення ;

б) послідовні наближення для внутрішніх вузлів сіточної області розраховуємо по формулі:



в) ітераційний процес продовжують, поки не стане виконуватися нерівність:



де ε - мале наперед задане число, що характеризує точність розрахунків.

Доведено, що для будь-якого кроку h процес Лібмана збігається до точного розв’язку незалежно від вибору початкового наближення.

Погрішність наближеного рішення має порядок О(h2).

Для вибору початкового наближення часто користуються таким правилом: оскільки значення шуканої функції uik завжди обмежено зверху і знизу:



розумно обирати за  середнє арифметичне величин u(x,y) у 4-х граничних точках, що розташовані на однієї горизонталі та вертикалі з точкою ui,k.

Доведено, що погрішність від заміни диференціального рівняння (1) різницевими рівняннями (3) має порядок O(h2) і оцінюється за допомогою нерівності:



Різницева схема (3) має властивості стійності та збіжності.

# 1.2 Розв’язання задачі Діріхле для рівняння Лапласа методом сіток у прямокутної області

У цьому підрозділі будуть розглянуті деталі розв’язання задачі Діріхле для рівняння Лапласа методом сіток.

Приклад

Нехай функція u(х,у) задовольняє всередині прямокутної області

D = {(х,у)/0 ≤ х ≤ 4; 0 ≤ у ≤ 3) рівняння Лапласа з граничними умовами:

u|х=0=0; u|у=0=0;

u|х=4= u|y=3 = ;

Необхідно розв’язати задачу крайову задачу Діріхле, обравши квадратну сітку з кроком h = 1

Розв’язання

Схема граничної області та сітки наведена на рис. 2. Оскільки в Scilab не передбачено використання нульового індексу елементів матриць, починаємо нумерацію індексів з номеру 1. На рис. 2 наведено нумерацію вузлів у відповідності з цим.

Розв’язання задачі проведемо методом Лібмана. Алгоритм розв’язання буде таким:

а) задаємо вхідні дані:

) координати прямокутної області (нижнього лівого та крайнього правого кутів) ;

) кількість кроків по кожній зі змінних;

) eps - точність розв’язання системи рівнянь;



Рисунок 2 - Гранична область і крайові умови задачі Діріхле

б) створюємо 4 підпрограми-функції для розрахунків крайових умов. В разі переходу до задачі з іншими крайовими умовами у них необхідно вносити відповідні зміни;

в) задаємо початкові наближення для внутрішніх точок. За них обираємо середні значення точок крайових умов, що лежать на осях х і у, що проходять через точку, тобто:

;

г) реалізуємо ітераційний метод Лібмана, до тих пір, поки для крайньої з внутрішніх точок буде виконуватися умова збіжності:

|

д) друкуємо результати

Для розв’язання задачі укладаємо скрипт . У скрипті

наведені коментарі, що дозволяють зясувати зміст кожної дії. Лістинг 1

//\*\*\*\*\*\*\*\*РОЗВЯЗАННЯ ЗАДАЧІ ДІРІХЛЕ\*\*\*\*\*\*\*\* //\*\*\*\*\*\*\*Підпрограми-функції для крайових умов function y=Nach\_Niz(x) y= 0 endfunctiony=Nach\_Verx(x) y = x л 2 / 16 endfunction function z=Nach\_Lev(y) z = 0 endfunctionz=Nach\_Prav(y) z = y л 2 / 9 endfunction

//\*\*\*\*\*\*Початок розрахунків\*\*\*\*\*\*\*

// \*\*\*Уведення координат кутових точок платівки x0=0, y0=0; //Лівий нижній кут x1=4, y1=3; // Правий верхній кут nx=4; //Кількість кроків по осі х ny=3; //Кількість кроків по осі у dx=(x1-x0)/nx;// Крок по осі Х dy=(y1-y0)/ny;// Крок по осі у=zeros(ny+1,nx+1);//визначення й обнуління матриці u //\*\*\*\* Формування крайових умов на границі сітки\*\*\*\*\* for i = 2:nx+1= x0 + (i-1) \* dx; u(1, i) = Nach\_Niz(x); u(ny+1, i) = Nach\_Verx(x);i = 2:ny+1= y0 + (i-1) \* dy; u(i, 1) = Nach\_Lev(y); u(i,nx+1) = Nach\_Prav(y);i=2:ny; for j=2:nx;(i,j)=(u(i,1)+u(i,nx+1)+u(1,j)+u(ny+1,j))/4;

//Метод Лібмана=1d>1e-5 a = u(ny, nx) for i = 2:ny for k = 2:nx(i, k) = (u(i - 1, k) + u(i + 1, k) + u(i, k - 1) +(i, k + 1)) / 4; end= abs(a - u(ny, nx)); end

Після закінчення розрахунків розв’язок - вектори у , х, матрицю u можна вивести на екран Результати наведені у табл. 1.

Таблиця 1 - Результат розв’язання прикладу

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | х=0 | х=1 | х=2 | х=3 | х=4 |
| у=0 | 0 | 0 | 0 | 0 | 0 |
| у=і | 0 | 0,0405074 | 0,0898696 | 0,1331149 | 0,1111111 |
| у=2 | 0 | 0,0722196 | 0,1858920 | 0,3314878 | 0,4444444 |
| у=3 | 0 | 0,0625 | 0,25 | 0,5625 | 1 |

Для більш точного розв’язання можна розділити область на більш дрібну сітку. Для наочного представлення результатів доцільно користуватися побудовою трьохвимірного графіка за допомогою комп’ютерних засобів.

2. ТЕКСТ ПРОГРАМИ

# include < stdio.h >

# include < fstream.h >

# include < conio.h >

# include < iostream.h >

# include < math.h >i, j, k;// Variablesh, x, y, tmp, E1;point {xx;yy;BelongsToDh\_;BelongsToDh;F;F\_;

}, arrayP [13] [33];arrayX [13];arrayY [33];diff [500];CreateNet (void);// Procedure PrototypesIsLineFit (float Param);CrMtrD (void);RegArrayX ();RegArrayY ();CreateDh\_ ();IsFit (point Par);FillF ();CreateDh ();IsInner (int i, int j);FillF\_ ();CountDif ();MakeFile ();main (void)// MAIN

{();.xx = 3;.yy = 5;= 0.2;.BelongsToDh\_ = 1;.BelongsToDh = 1;();();();();\_ ();();();\_ ();();(E1> = 0.005) {(i = 0; i <13; i + +)(j = 0; j <33; j + +) arrayP [i] [j]. F = arrayP [i] [j]. F\_;\_ ();();

}<<(0-arrayP [7] [17]. F\_);();();

}// MAIN ENDIsLineFit (float par, char Axis)// does the line belong to the defined area

{(Axis) {'y': if ((par> 8.0) | | (par <2.0)) return 1;return 0;'x': if (par <1.9) return 1;if (par> 4.0) return 1;return 0;

}

}CreateNet (void)// Creation of Net (area D)

{= p0.xx;= 0;[i] = x;(! IsLineFit (x, 'x'))

{+ = h;+ +;[i] = x;

}= p0.xx-h;+ +;[i] = x;(! IsLineFit (x, 'x'))

{- = h;+ +;[i] = x;

}(i = 0; i <13; i + +) {printf ("% g", arrayX [i]);}("n");= p0.yy;= 0;[i] = y;(! IsLineFit (y, 'y'))

{+ = h;+ +;[i] = y;

}= p0.yy - h;+ +;[i] = y;(! IsLineFit (y, 'y'))

{- = h;+ +;[i] = y;

}(i = 0; i <33; i + +) {printf ("% g", arrayY [i]);}("n");

}// end CreateNetRegArrayX ()// Regulation of arrays X & Y

{LastUnreg = 13;(LastUnreg! = 0) {(i = 0; i(arrayX [i]> arrayX [i +1]) {double tmp = arrayX [i];[i] = arrayX [i +1];[i +1] = tmp;}}= LastUnreg-1;}(i = 0; i <13; i + +) {printf ("% g", arrayX [i]);

} printf ("n");

}RegArrayY ()

{LastUnreg = 33;(LastUnreg! = 0) {(i = 0; i(arrayY [i]> arrayY [i +1]) {tmp = arrayY [i];[i] = arrayY [i +1];[i +1] = tmp;}}= LastUnreg-1;}(i = 0; i <33; i + +) {printf ("% g", arrayY [i]);}("n");}// End of RegulationCrMtrD (void)// Create general Matrix

{(i = 0; i <13; i + +)(j = 0; j <33; j + +) {arrayP [i] [j]. BelongsToDh\_ = 0;[i] [j]. BelongsToDh = 0;}(i = 0; i <13; i + +)(j = 0; j <33; j + +) {[i] [j]. xx = arrayX [i];[i] [j]. yy = arrayY [j];

}

// printf ("% g% g", arrayP [12] [0]. xx, arrayP [12] [0]. yy);

// printf ("n");

}IsFit (point Par)// does point belong to area D?

{((Par.xx <= 4) && (Par.xx> = 1.99) && (Par.yy> = Par.xx)

&& (Par.yy <= Par.xx +4)) return 1;return 0;

}CreateDh\_ (void)// Create area Dh\_

{(i = 0; i <13; i + +)(j = 0; j <33; j + +)(IsFit (arrayP [i] [j])) arrayP [i] [j]. BelongsToDh\_ = 1;<<

}FillF (void)// calc function F (x, y) at area Dh\_

{(i = 0; i <13; i + +)(j = 0; j <33; j + +)(arrayP [i] [j]. BelongsToDh\_ == 1)[i] [j]. F = arrayP [i] [j]. xx \* pow (arrayP [i] [j]. yy, 2);arrayP [i] [j]. F = 0;

}IsInner (int i, int j)// Is point inner?

{((arrayP [i-1] [j]. BelongsToDh\_ == 1) &&

(arrayP [i +1] [j]. BelongsToDh\_ == 1) &&

(arrayP [i] [j +1]. BelongsToDh\_ == 1) &&

(arrayP [i] [j-1]. BelongsToDh\_ == 1)) return 1;return 0;

}CreateDh (void)// Create area Dh

{(i = 0; i <13; i + +)(j = 0; j <33; j + +)((arrayP [i] [j]. BelongsToDh\_ == 1) &&(i, j))[i] [j]. BelongsToDh = 1;

}FillF\_ () вЂ‹вЂ‹//calc new appr. values вЂ‹вЂ‹of F

{(i = 0; i <13; i + +)(j = 0; j <33; j + +) {(arrayP [i] [j]. BelongsToDh == 1)[i] [j]. F\_ = (arrayP [i-1] [j]. F + arrayP [i +1] [j]. F +[i] [j-1]. F + arrayP [i] [j +1]. F)/4;arrayP [i] [j]. F\_ = 0;}

}CountDif ()// find maximal difference abs (F-F\_)

{= 0;(i = 0; i <13; i + +)(j = 0; j <33; j + +)

{if (arrayP [i] [j]. BelongsToDh == 1) {[k] = fabs (arrayP [i] [j]. F\_-arrayP [i] [j]. F);+ +;}}= diff [0];(k = 1; k <500; k + +) {(diff [k]> E1) E1 = diff [k];}

}MakeFile ()

{f;\* f1 = fopen ("surf.dat", "w1");(f1);.open ("surf.dat", ios :: out, 0);(i = 0; i <13; i + +)(j = 0; j <33; j + +) {if (arrayP [i] [j]. BelongsToDh == 1) {<

"" <.close ();

}