

Лабораторна робота №2

Об'єктно-орієнтоване програмування в C#

План

1. Постановка задачі.
2. Приклад розв'язання.
3. Завдання.

1. На практиці часто виникає задача *інтерполяції* або *екстраполяції*, коли за наявним набором значень деякої величини потрібно знайти її додаткові значення.

Наприклад, якщо на деякій території розташовано n метеостанцій з датчиками температури, то рано чи пізно виникне задача обчислення температури в деякій точці місцевості, яка знаходиться між датчиками (Рис. 1).

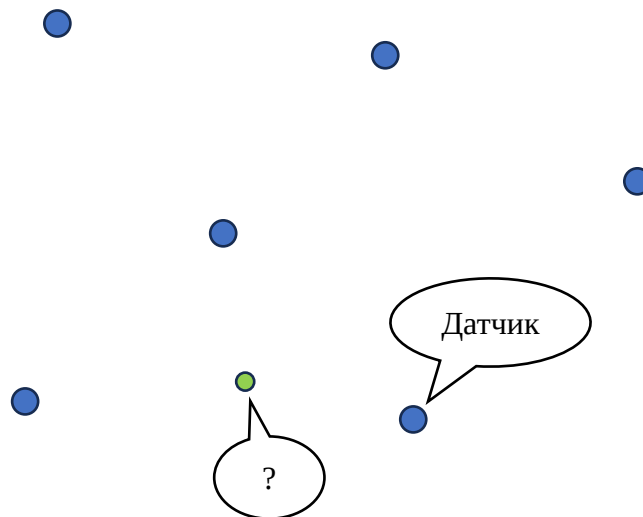


Рис. 1. Схема розташування метеостанцій

Одним із можливих варіантів розв'язання цієї задачі є побудова деякого інтерполяційного многочлена. Так як у загальному вигляді побудувати систему лінійно незалежних базисних функцій для геометричної області довільної форми практично неможливо, то на практиці зазвичай використовують такий спосіб. Геометричну область, утворену множиною опорних точок (місць, де розташовані датчики), розбивають на геометричні фігури простої форми, які називаються скінченними елементами (СЕ). У разі трикутних СЕ такий процес називається триангуляцією (Рис. 2).

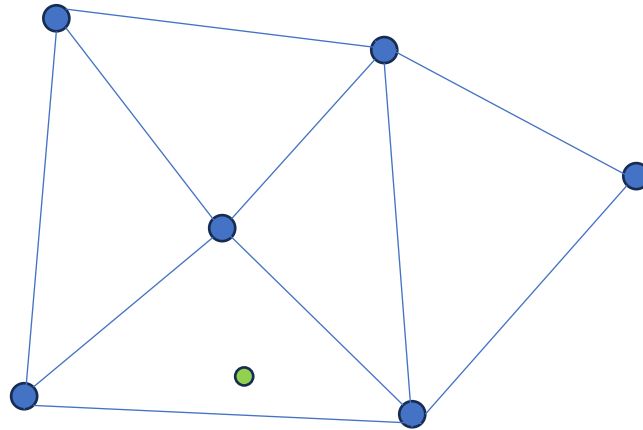


Рис. 2. Тріангуляція множини опорних точок

Після цього для кожного трикутника будується множина його базових функцій, які називаються функціями форми. Знаючи значення деякої величини (температури) у вершинах трикутника обчислити потрібне значення у будь-якій точці поточного СЕ можна за допомогою інтерполяційного багаточлена, побудованого на базі функцій форми. Наприклад, для трикутного елемента, лінійні функції форми мають такий вигляд:

$$N_i(x, y) = c_0^i + c_1^i x + c_2^i y, \quad i = 0, 1, 2,$$

де коефіцієнти c_j ($j = 0, 1, 2$) залежать від координат вершин трикутника і обчислюються за допомогою наступного співвідношення:

$$N_i(x_j, y_j) = \begin{cases} 1, & i = j; \\ 0, & i \neq j, \end{cases} \quad i, j = 0, 1, 2.$$

Тобто i -та функція форми дорівнює 1 в i -й вершині трикутника і 0 у інших. Для трикутного СЕ таким чином потрібно вирішити три системи лінійних алгебраїчних рівнянь (СЛАР). Наприклад, для відшукування коефіцієнтів функції форми N_0 необхідно вирішити таку СЛАУ:

$$\begin{cases} c_0^0 + c_1^0 \cdot x_0 + c_2^0 \cdot y_0 = 1; \\ c_0^0 + c_1^0 \cdot x_1 + c_2^0 \cdot y_1 = 0; \\ c_0^0 + c_1^0 \cdot x_2 + c_2^0 \cdot y_2 = 0. \end{cases}$$

Або в матричній (розширеній) формі:

$$\begin{pmatrix} 1 & x_0 & y_0 & 1 \\ 1 & x_1 & y_1 & 0 \\ 1 & x_2 & y_2 & 0 \end{pmatrix} \quad (1)$$

Розв'язуючи цю систему, отримаємо коефіцієнти c_0^0 , c_1^0 і c_2^0 функції форми N_0 . Аналогічним чином для трикутного СЕ знаходяться коефіцієнти функцій форми N_1 і N_2 .

Знаючи вузлові значення температури у вершинах, інтерполювати його значення в будь-якій точці поточного трикутника можна на підставі наступного співвідношення:

$$T(x, y) = U_0 \cdot N_0(x, y) + U_1 \cdot N_1(x, y) + U_2 \cdot N_2(x, y),$$

де T_i ($i = 0, 1, 2$) – відомі значення температури (або іншої функції) у вершинах СЕ.

Очевидно, що $T(x_0, y_0) \equiv T_0$, так як $N_0(x_0, y_0) = 1$, $N_1(x_0, y_0) = 0$ та $N_2(x_0, y_0) = 0$. І так далі. Таким чином у межах СЕ з використанням вищенаведеної апроксимації значення фазового параметра тотожно дорівнює відомим значенням температури у вершинах трикутника, а в інших точках у середині трикутника певним чином лінійно апроксимуватимуться.

2. Для розв'язання задачі пошуку коефіцієнтів функцій форми трикутного СЕ створимо два класи: **ShapeFunction** і **ShapeTriangle**.

1) Абстрактний базовий клас **ShapeFunction** описує найбільш загальні властивості функції форми: її розмірність (одно-, двох- або тривимірна), кількість коефіцієнтів, а також матрицю коефіцієнтів функцій форми, (кожний рядок містить коефіцієнти однієї функції) і матрицю координат вузлів СЕ. Крім властивостей, цей клас містить також ряд методів: конструктор, а також методи, призначені для ініціалізації координат СЕ і обчислення коефіцієнтів його функцій форм.

Його програмна реалізація мовою C# може мати, наприклад, такий вигляд:

```
// Абстрактный класс «Функция формы»
abstract class ShapeFunction
{
    protected int Size; // Кількість коефіцієнтів функції форми
    protected int Dim; // Розмірність
    protected double[,] C; // Матриця коефіцієнтів
    protected double[,] X; // Координати вершин СЕ
    protected ShapeFunction()
    {
        Size = Dim = 0;
    }
    protected void SetCoord(int psize, double[,] px)
    {
        Size = psize;
        X = new double[Size, Dim];
        for (int i = 0; i < Size; i++)
            for (int j = 0; j < Dim; j++)
                X[i, j] = px[i, j];
        Create();
    }
}
```

```

}
protected abstract double ShapeCoeff(int i, int j);
private bool Solve(double[,] matr, double[] result, double eps = 1.0E-10)
{
    double coeff;

    for (var i = 0; i < Size - 1; i++)
    {
        if (matr[i, i] < eps)
            continue;
        for (var j = i + 1; j < Size; j++)
        {
            if (Math.Abs(coeff = matr[j, i]) < eps)
                continue;
            for (var k = i; k < Size + 1; k++)
                matr[j, k] -= (coeff * matr[i, k] / matr[i, i]);
        }
    }
    if (Math.Abs(matr[Size - 1, Size - 1]) < eps)
        return false;
    result[Size - 1] = matr[Size - 1, Size] / matr[Size - 1, Size - 1];
    for (int k = 0; k < Size - 1; k++)
    {
        int i = Size - k - 2;
        var sum = matr[i, Size];

        for (int j = i + 1; j < Size; j++)
            sum -= result[j] * matr[i, j];
        if (Math.Abs(matr[i, i]) < eps)
            return false;
        result[i] = sum / matr[i, i];
    }
    return true;
}

public void Create()
{
    double[,] A = new double[Size, Size + 1];
    double[] res = new double[Size];

    C = new double[Size, Size];
    for (int i = 0; i < Size; i++)
    {
        for (int j = 0; j < Size; j++)
        {
            for (int k = 0; k < Size; k++)
                A[j, k] = ShapeCoeff(j, k);
            A[j, Size] = (i == j) ? 1.0 : 0.0;
        }
        if (!Solve(A, res))
            Console.WriteLine("Bad FE!");
        for (int j = 0; j < Size; j++)
            C[i, j] = res[j];
    }
}

```

```

    }
}
public void Print()
{
    for (int i = 0; i < Size; i++)
    {
        for (int j = 0; j < Size; j++)
            Console.Write("{0} ", C[i, j]);
        Console.WriteLine();
    }
}
}

```

У наведеному прикладі реалізації абстрактного класу ShapeFunction метод **SetCoord()** задає розмірність СЕ і значення координат його вершин; метод **Create()** реалізує процедуру обчислення коефіцієнтів функцій форм. Для цього використовуються метод **Solve()**, що реалізує розв'язання СЛАР методом Гауса, і абстрактний метод **ShapeCoeff()**, що повертає j -е значення i -го рядка матриці, описаної виразом (1). Вигляд цієї матриці залежить від типу СЕ, тому цей клас є абстрактним.

2) Похідний від ShapeFunction клас **ShapeTriangle** тоді може бути описаний, наприклад, таким чином:

```

// Функції форми трикутного елемента
class ShapeTriangle : ShapeFunction
{
    public ShapeTriangle()
    {
        Size = 3;
        Dim = 2;
    }
    public ShapeTriangle(double[,] px)
    {
        Size = 3;
        Dim = 2;
        SetCoord(Size, px);
    }
    protected override double ShapeCoeff(int i, int j)
    {
        double[] s = { 1.0, X[i, 0], X[i, 1] };

        return s[j];
    }
}

```

У даному класі перевизначено метод ShapeCoeff(), який повертає коефіцієнти матриці СЛАР, що відповідають трикутному СЕ, а також додано ще один конструктор, який реалізує весь функціонал побудови функцій форм СЕ за його координатами.

Практично застосувати таку систему класів можна, наприклад, таким чином:

```
class Program
{
    static void Main(string[] args)
    {
        double[,] x = { { 0, 0 }, { 1, 0 }, { 1, 1 } }; // Координати вершин CE
        ShapeTriangle fe = new ShapeTriangle(x);
        fe.Print();
    }
}
```

3. У Visual Studio створити новий консольний проект C#. Написати програму мовою C#, яка реалізує залежно від варіанта обчислення функцій форми скінченних елементів наступного виду:

- 1) чотирикутник;
- 2) тетраедр;
- 3) куб;
- 4) квадратичний (шестивузловий) трикутник;
- 5) квадратичний (десятивузловий) тетраедр.

Передбачити виведення на екран ПІБ та номери групи студента, який здає роботу.