Features of the numerical solution of boundary value problems on Python algorithmic programming language

Vladimir Mondrus 1, and Dmitrii Sizov1,*

1Moscow State University of Civil Engineering, Yaroslavskoe shosse, 26, Moscow, 129337, Russia

Abstract. The solution of most practical problems arising in the calculation of structures requires a numerical solution of boundary value problems. In the article, examples of solving specific boundary-value problems describe the capabilities of the modern object-oriented programming language Python.

1 Mathematical Python

The Python language is programming language, simple for studying, promoting writing of the high-structured programs. One of primary benefits of this language as means for solution of edge tasks is, first of all, abundance of the libraries allowing to use in the most short time already existing library modules in own programs. One of attractive lines of this language is openness of implementations, there are language interpreters for many platforms. The most known reference implementation of language is CPython which is written in the C. Jython language is based on the Java programming language, in the course of broadcasting there is broadcast of the source code of the Python language in the Java language that promotes still bigger portability of programs. The special attention is deserved by the translator IronPython intended for integration of the Python programs with the applications created for the Microsoft.NET Framework platform. Also there is implementation of the Python language in the Python language (the PyPy interpreter including Jit the compiler). The described software is in free access and can be downloaded from the official sites.

Important property of language is availability of the extensive literature mentioning as the description of language syntax, and the specific packets intended for solution of certain circle of tasks. The most known scopes of Python in the development area of scientific software is machine learning and data analysis, including possessing the discharged and badly structured structure. This language is obliged by the ability to processing of data bulks, first of all, to several libraries, specially created for fast data processing.

Basis, so-called, "mathematical" Python number of the packets, most known in community Python, makes: NumPy, SciPy and Matplotlib, and also Pandas [1-3]. Originally the Python language did not reflect as language for numerical calculations,

* Corresponding author: newfff@mail.ru
however availability of these packets allows to use it as means of calculations, somewhat similar to other packets of the matrix calculations - MATLAB and GNU Octave extending as on the basis of commercial, and free licenses.

The NumPy library allows to create arrays of any dimension and possesses set of the built-in functions operations for work with them. So, for example, transposing of matrix, or calculation of any function from all array cells does not require writing of the code including cycles that leads to considerable reduction as the most program code, and promotes bigger vectorizuyemost of calculations.

The packet of SciPy includes support of special functions, fast Fourier transform, operations of linear algebra, the subprogramme of solution of differential equations and search engine optimization, and also includes functions of interpolation and smoothing. Also inclusion of code C in the Python code of programs is supported.

Matplotlib allows to receive qualitative graphical representation of data, as in the form of two-dimensional, and three-dimensional diagrams. Further in article the results received with use of this packet in addition to which application of DISLIN library, or Chaco and Mayavi, Mayavi2 is possible are used.

In qualities of examples we will consider some solutions of boundary value problems with use of opportunities of the described libraries and features of the Python programming language. In article the solution of the following tasks is provided:

The solution of the equation of Laplace on flat rectangular area describing distribution of temperatures (display of results of calculation will be made by means of graphic library Matplotlib).

Search of minimum of functionality of Dirikhle is carried out by variation and differential method, at solution arrays of required value are represented by means of Numpy library.

**1.1 Solution of the equation of Laplace on flat rectangular area by method of finite differences**

It is known that Laplace's equation describes many stationary processes (heat conductivity, diffusions, electrostatics).

Let's consider two-dimensional rectangular area on which borders distribution of temperature Figure 1 is set. Laplace's equation and boundary conditions for rectangular area have appearance:

\[
\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, \tag{1}
\]

\[
\begin{align*}
T(x,0) &= 0^\circ C, & T(x, Y_o) &= 700^\circ C \\
T(0, y) &= 0^\circ C, & T(X_o, y) &= 700^\circ C
\end{align*}
\]

Fig. 1. The considered area with temperature set on borders
We will apply method of finite differences to solution of task, having written finite difference analog of the equation in the form of [4,5]:

\[
\frac{T_{i+1,j}^{k+1} - 2T_{i,j}^{k+1} + T_{i-1,j}^{k+1}}{dx^2} + \frac{T_{i,j+1}^{k+1} - 2T_{i,j}^{k+1} + T_{i,j-1}^{k+1}}{dy^2} = 0,
\]

where \( T_{i,j}^{k+1} \) — value of required function in \([i, j]\) to point on iteration \( k+1 \), \( dx, dy \) — step on coordinate \( x \) and \( y \). The enlarged flowchart of solution of problem of distribution of temperature is presented on Figure 2.

![Flowchart of solution of problem of distribution of temperature](image)

**Fig. 2.** The enlarged flowchart of the program for problem of heat exchange (the block with calculation the boundary conditions depending on computation process allows to receive values of required function on those sections of border of area where boundary conditions are not set)

Having executed iterations of method of Zeydel [4,5], on grid of 50x50 points in size, we will receive field distribution of temperatures which can be presented means of Matplotlib library in the form of two and three-dimensional diagrams provided in Table 1. Apparently from drawings, boundary conditions tasks are carried out with high accuracy. The Matplotlib library allows to make setup of the prevailing flowers of graphic display in very wide limits and gives to the user rich opportunities of choice of color forms of design of diagrams from the existing set.
Table 1. Results of the solution of the equation of Laplace

| Isofield of temperatures on two-dimensional area | Three-dimensional schedule of a field of temperatures |
|------------------------------------------------|------------------------------------------------------|

| ![Isofield of temperatures on two-dimensional area](image1) | ![Three-dimensional schedule of a field of temperatures](image2) |

When performing calculations in the Python language it is necessary to avoid explicit record of cycles on isolated points of area, as much as possible using possibilities of index cuts of NumPy library that allows to accelerate as much as possible process of calculations [3].

1.2. Search of a minimum of functionality of Dirikhle by a variation and differential method

Potential energy of a membrane to within the multiplier expressing its physical properties is expressed by multiple integral on area [6,7]:

$$D[u] = \iint_{\Omega} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 \right] dx dy,$$

(3)

where:

- $\Omega$ — the area occupied by a membrane;
- $u$ — required function.

Position of balance of a membrane corresponds minimality condition of this functionality at the set boundary conditions. We will make the solution of a task by a variational-difference method with numerical calculation to each point of area of the decision, derivative of functionality. The finite difference analog of functionality has an appearance [6,7]:

$$D[u] = \sum_{i} \sum_{j} \left[ \left( \frac{u_{i+1,j} - u_{i,j}}{dx} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{dy} \right)^2 \right],$$

where $dx$ and $dy$ are step sizes in coordinates $x$ and $y$. The finite difference analog of the Laplace equation is:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{dx^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{dy^2} = 0, \quad \forall i, j.$$
\[ D[u] = \sum_{i=0}^{n} \sum_{j=0}^{m} \left[ \left( \frac{u_{i+1,j} - u_{i,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 \right] h_x h_y = \]

\[
\begin{align*}
& \left[ \ldots + \left( \frac{u_{i,j} - u_{i-1,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 \right] \\
& \left( \frac{u_{i+1,j} - u_{i,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 + \left( \frac{u_{i+1,j-1} - u_{i,j-1}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j-1}}{h_y} \right)^2 \ldots \right]
\end{align*}
\]

(4)

In Figure 3 the fragment of a finite difference grid in the vicinity of a point with indexes \(i, j\) is shown. It is visible that direct participation in forming of the equation in \([i, j]\) to a point accept only expressions for energy in \([i-1, j]\), \([i, j]\) and \([i, j-1]\) points in spite of the fact that five points will participate in each variation difference equation with indexes \([i-1, j]\), \([i, j]\), \([i, j-1]\), \([i+1, j]\), \([i, j+1]\). In the course of calculations on Zeydel's method [8] on points with indexes \([i+1, j]\), \([i, j+1]\) will extend influence from points \([i-1, j]\), \([i, j]\) and \([i, j-1]\).

In Figure 4 the enlarged flowchart of a variational-difference method taking into account use a hash table (a special form of the called array) of the Python [3] language is provided. Application a hash array allows to consider naturally change of a type of the equations on borders of the considered rectangular area.

---

**Table 1.** Results of the solution of the equation of Laplace

| Isofield of temperatures on two-dimensional area | Three-dimensional schedule of a field of temperatures |
|------------------------------------------------|--------------------------------------------------|
| When performing calculations in the Python language it is necessary to avoid explicit recording of cycles on isolated points of area, as much as possible using possibilities of index cuts of NumPy library that allows to accelerate as much as possible process of calculations [3]. |

1.2. Search of a minimum of functionality of Dirikhle by a variation and differential method

Potential energy of a membrane to within the multiplier expressing its physical properties is expressed by multiple integral on area [6,7]:

\[
\int_{\Omega} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \, dxdy
\]

where:

- \(\Omega\) – the area occupied by a membrane;
- \(u\) – required function.

Position of balance of a membrane corresponds minimality condition of this functionality at the set boundary conditions. We will make the solution of a task by a variational-difference method with numerical calculation to each point of area of the decision, derivative of functionality. The finite difference analog of functionality has an appearance [6,7]:

\[
\sum_{i=0}^{n} \sum_{j=0}^{m} \left[ \left( \frac{u_{i+1,j} - u_{i,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 \right] h_x h_y = 
\]

\[
\begin{align*}
& \left[ \ldots + \left( \frac{u_{i,j} - u_{i-1,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 \right] \\
& \left( \frac{u_{i+1,j} - u_{i,j}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j}}{h_y} \right)^2 + \left( \frac{u_{i+1,j-1} - u_{i,j-1}}{h_x} \right)^2 + \left( \frac{u_{i,j+1} - u_{i,j-1}}{h_y} \right)^2 \ldots \right]
\end{align*}
\]

(4)

In Figure 3 the fragment of a finite difference grid in the vicinity of a point with indexes \(i, j\) is shown. It is visible that direct participation in forming of the equation in \([i, j]\) to a point accept only expressions for energy in \([i-1, j]\), \([i, j]\) and \([i, j-1]\) points in spite of the fact that five points will participate in each variation difference equation with indexes \([i-1, j]\), \([i, j]\), \([i, j-1]\), \([i+1, j]\), \([i, j+1]\). In the course of calculations on Zeydel's method [8] on points with indexes \([i+1, j]\), \([i, j+1]\) will extend influence from points \([i-1, j]\), \([i, j]\) and \([i, j-1]\).
Fig. 4. The enlarged flowchart of a variational-difference method with numerical differentiation of functionality of energy

On area borders the number of the component's entering the equation of a variational-difference method changes, use a hash array allows to store indexes of those components of energy which make a nonzero contribution on border.

Results of the solution of a two-dimensional problem of minimization of functionality of Dirikhle are given in Table 2 under the set conditions on border.

Table 2. Results of the solution of a boundary value problem for Dirikhle's functionality

| The Isofield of \( u \) on a rectangular region | Three-dimensional graph of \( u \) |
|-----------------------------------------------|----------------------------------|
| ![Isofield of \( u \) on a rectangular region](image1) | ![Three-dimensional graph of \( u \)](image2) |
Fig. 4. The enlarged flowchart of a variational-difference method with numerical differentiation of functionality of energy

On area borders the number of the component's entering the equation of a variational-difference method changes, use a hash array allows to store indexes of those components of energy which make a nonzero contribution on border.

Results of the solution of a two-dimensional problem of minimization of functionality of Dirikhle are given in Table 2 under the set conditions on border.

Table 2. Results of the solution of a boundary value problem for Dirikhle's functionality

The Isofield of u on a rectangular region

Three-dimensional graph of u

References

1. N. A. Prohorenok, *Python 3 and PyQt5. Applications programming* (BHV-Petersburg, Saint-Petersburg, 2016)
2. W. McKinney, *Python and the analysis of data* (DKM Press, Moscow, 2015)
3. L. Ramalho, *Fluent Python Python* (DKM Press, Moscow, 2016)
4. E. E. Tyrtyshnikov, *Numerical analyses* (Academy, Moscow, 2007)
5. G. W. Stewart, *Introduction to Matrix Computations* (Academic Press, New York, 1973)
6. K Vasidzu, *Variation methods in the theory of elasticity and the plasticity* (Mir, Moscow, 1987)
7. S. P. Tymoshenko, D. Gudyer, *The theory of elasticity* (Nauka, Moscow, 1975)
8. D. Ortega., U. Pul, *At. Introduction to numerical methods of solution of differential equations* (Nauka, Moscow, 1986)