

## **РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ МОДЕЛИРОВАНИЯ ГЕОДИНАМИЧЕСКИХ ЯВЛЕНИЙ МЕТОДОМ ХРИСТОВА**

**П.С. Долгий**

*аспирант кафедры геодезии и геоинформационных систем факультета  
информационных технологий Полоцкого государственного университета,  
Новополоцк, p.dolgi@psu.by*

**Г.А. Шароглазова**

*доцент, заведующая кафедрой геодезии и геоинформационных систем факультета  
информационных технологий Полоцкого государственного университета,  
Новополоцк.*

В статье отражена методика моделирования геодинамических процессов методом Христова на основе результатов повторного высокоточного нивелирования. Сделано обоснование выбора языка программирования для разработки программного продукта. Разработка велась на языке python с использованием фреймворка интерфейса PyQt5, что позволяет работать на любой операционной системе, а также оставляет широкие возможности для редактирования исходного кода. Полученный продукт снабжен модулями анализа полученных результатов и графического представления в виде пространственно-временных графиков в двумерном и трехмерном виде, графиков накопления разностей превышений, графиков поведения каждого из реперов в отдельности. Отладка продукта выполнена на основе реальных данных Полоцкого геодинамического профиля. Результаты, полученные для 2015 года, подтвердили полученные ранее для 2013 г.

**Ключевые слова:** геодинамика; моделирование; метод Христова; Python; Полоцкий геодинамический профиль.

Геодинамические явления происходят непрерывно на всей планете и подразделяются на глобальные (движение полюсов Земли), крупномасштабные (движение литосферных плит), а также региональные и локальные, связанные с изменением местоположения и вариациями силы тяжести относительно небольших участков земной коры [1].

Цель данной работы – разработка программного продукта для моделирования геодинамических процессов по результатам повторного нивелирования. В работе будет уделено внимание исследованию локальных современных вертикальных движений земной коры.

Для исследования локальных движений земной коры используют такие геодезические методы как повторное нивелирование, повторные линейно-угловые измерения, повторные спутниковые наблюдения. Способом, предоставляющим наиболее точные сведения о вертикальных движе-

ниях точек земной поверхности, остается повторное высокоточное нивелирование [2].

Современные исследования в геодинимике требуют комплексного междисциплинарного подхода. Важнейшие инструменты исследования в геодинимике – методы математического моделирования и информационные технологии [3].

Метод, используемый для моделирования современных вертикальных движений земной коры во времени основан на алгебраических полиномах и назван в честь болгарского геодезиста Владимира Христова.

Функция алгебраического полинома для линейной аппроксимации приобретает следующий вид

$$H_k^i = H_k^0 + V_k(T_i - T_0) \quad (1)$$

Для квадратичной аппроксимации:

$$H_k^i = H_k^0 + V_k(T_i - T_0) + A_k(T_i - T_0)^2 \quad (2)$$

Для кубической аппроксимации:

$$H_k^i = H_k^0 + V_k(T_i - T_0) + A_k(T_i - T_0)^2 + B_k(T_i - T_0)^3 \quad (3)$$

Для определения коэффициентов  $H(0)$ ,  $V$ ,  $A$ ,  $B$  составляется система уравнений – одно уравнение для каждого измеренного превышения в каждую эпоху. Этим уравнениям придают веса  $P$ , определяемые как величины, обратно пропорциональные длине секций:

$$p_m = \frac{c}{L_m}; \quad (4)$$

$C$  – постоянная, равная отношению априорной погрешности единицы веса к погрешности на 1 км хода;  $L_m$  – длина линии в километрах [4].

В результате получают систему нормальных уравнений, которую решают по формуле

$$X = (A^T P A)^{-1} A^T P L \quad (5)$$

Прибавляя полученные значения поправок к приближенным высотам пунктов, получают уравненные высоты на начальную эпоху

$$H_0 = H^\circ + \delta H^\circ \quad (6)$$

Подставляя полученные значения скорости, ускорения и производной ускорения в уравнения, получают модельные значения высот на каждую

эпоху. Вычитая модельные высоты смежных пунктов хода, получают модельные превышения. Из вычисленных модельных значений высот вычитают полученные по результатам измерений. Среднеквадратическую ошибку модели вычисляют по формуле

$$\sigma = \sqrt{\frac{\sum \Delta h_{nk}}{n \cdot k}}, \quad (7)$$

$\Delta h_i$  – разность вычисленных и измеренных значений высот для каждого пункта за каждую эпоху измерений;  $n$  – номер пункта,  $k$  – номер эпохи измерений [4].

На данный момент для моделирования современных вертикальных движений земной коры используется программный продукт ДК СВДЗК, разработанный в Полоцком государственном университете в 2008 году [5]. Этот продукт имеет высокую скорость вычислений, однако не имеет возможности выполнения кубической аппроксимации, модулей анализа и графического представления результатов.

Исходный код продукта к настоящему времени утерян, поэтому было решено выполнить разработку подобного программного продукта заново.

Для реализации продукта выбран язык программирования Python версии 3.3. Язык является современным, в настоящий момент, согласно различным рейтингам [6–8], входит в пятерку наиболее популярных в мире. Разработку значительно упрощают лаконичность языка, широкое использование таких производительных языковых конструкций, как генераторы, лямбда-функции и другие, многофункциональная стандартная библиотека, а также возможность простой установки дополнительных модулей. Графический интерфейс пользователя разрабатывался на основе фреймворка PyQt5. По сравнению с другими фреймворками графического интерфейса для языка Python (PySide, Tkinter, wxWidgets) PyQt5 имеет наиболее полный набор графических элементов, хорошо описан в технической литературе. Разработчиком PyQt5 является компания Riberbank Computing. Продукт имеет лицензию GPL, которая предоставляет пользователю право копировать, модифицировать и распространять программы, а также гарантировать, что и пользователи всех производных программ получают вышеперечисленные права, при этом запрещая создавать на основе свободной программы под GPL другой проект, не предоставляя его исходники пользователям [9].

Для расчетов использовалась библиотека NumPy. Данная библиотека, написанная на C, предназначена для научных вычислений. Библиотека не позволяет создавать коллекции данных с различными типами, как в «сыром» Python, следовательно, не требуется проверка типов при операциях с

этими данными и производительность возрастает в разы [12]. В разработанном нами программном продукте библиотека NumPy использовалась не только для сложных операций, таких как умножение и обращение матриц, но и для более простых, например, вычисление суммы массива [11].

Для визуализации расчетов использована библиотека Matplotlib. Данная библиотека имеет возможность как отображать построенные изображения в отдельном окне, так и встраивать их в окна различных фреймворков графического интерфейса Python [12].

Для написания кода использовалась интегрированная среда разработчика JetBrains PyCharm edu, являющаяся самой совершенной средой разработки на языке Python, поддерживающая подсветку синтаксиса, автодополнение и т.д.

Интерфейс программного продукта реализован при помощи вкладок. На первой вкладке выполняется ввод либо импорт исходных данных с возможностью редактирования. Далее необходимо выбрать исходный пункт и ввести его высоту. После выполнения расчетов в окне программы отображаются высоты и превышения, вычисленные по методу Христова, разность между вычисленными и измеренными значениями, а также графическое представление результатов – пространственно-временные графики движения пунктов нивелирования в двумерном и трехмерном виде, графики накопления разностей превышений, а также графики поведения каждого из реперов в отдельности.

Отладка программного продукта выполнялась на основе реальных измерительных данных Полоцкого геодезического профиля, расположенного к северо-западу от г. Полоцк. В данной работе использовались данные 10 циклов (с 2004 по 2015 годы за исключением 2005 и 2014) высокоточного нивелирования на 11 нивелирных пунктах (1 исходный). Таким образом, исходные данные были представлены 100 измерениями.

Прибавив вычисленные поправки к полученным по результатам измерений высотам, получили уравненные высоты на эпоху 2003 года (формула 6). Подставив полученные высоты и значения коэффициентов в формулы 1–3), вычислили модельные значения высот. Вычитая модельные значения высот конечных пунктов для каждой линии из высот начальных пунктов, получили модельные превышения. Далее сравнили модельные значения высот с измеренными. По формуле 7 нашли среднюю квадратическую ошибку (СКО) для каждой модели. Значения, которые превышают двойную СКО, не согласуются с моделью с вероятностью 95%.

Можно заметить, что с увеличением степени аппроксимации уменьшается число несогласующихся значений (таблица 1). Также примечательно, что во всех трех моделях наибольшее количество несогласующихся значений приурочены к реперу 8372, что, скорее всего, объясняется су-

ществованием тектонических разломов по обе стороны от него. Это также подтверждается и на пространственно-временном графике (рис. 1).

Таблица 1

**Характеристики различных моделей, описывающих движение нивелирных пунктов Полоцкого геодинамического профиля**

Характеристика модели	Модель		
	Линейная	Квадратичная	Кубическая
СКО, мм	1,135	0,860	0,768
Число несогласующихся с вероятностью 95% значений	6	5	3

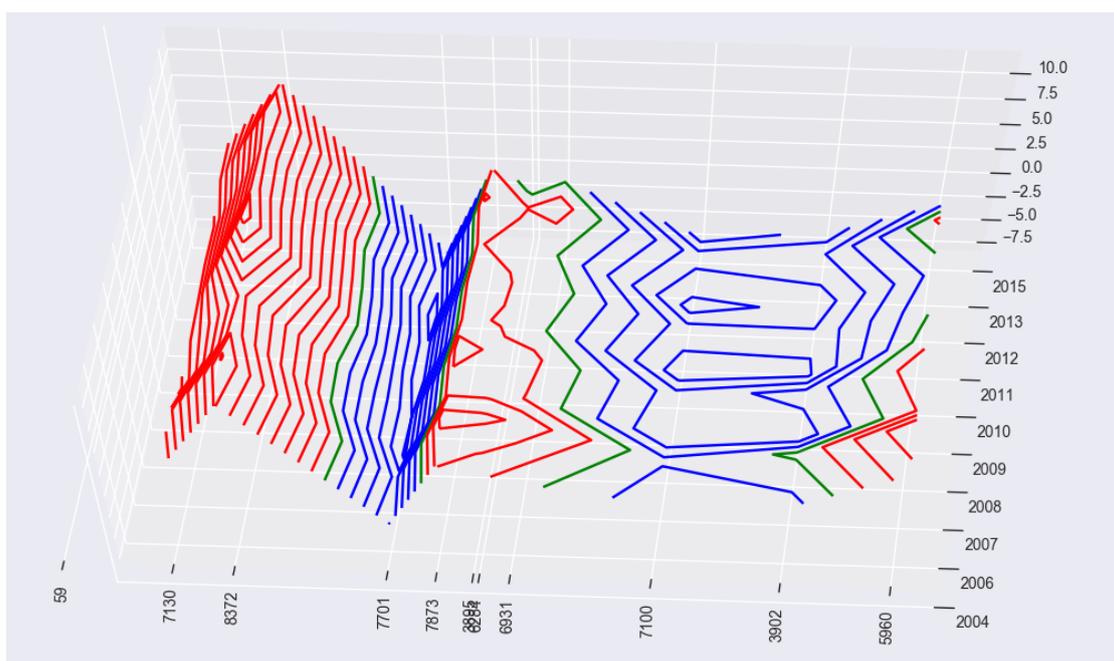


Рис. 1 – Пространственно-временной график движения нивелирных пунктов Полоцкого геодинамического профиля в трехмерном виде. Синим цветом обозначены отрицательные значения, красным – положительные, зеленым – нулевые

Далее мы рассмотрели накопление разностей превышений для различных моделей. В одних и тех же координатных осях строились два графика: для измеренных значений и для модельных. Рассматривая два графика, можно понять, насколько хорошо согласуется модель с измеренными значениями.

Изучив графики для разностей между различными эпохами, мы выяснили следующее: графики для более поздних эпох во всех моделях согласуются с измерениями лучше, чем для более ранних; графики для моделей с более высокими степенями аппроксимации согласуются с измерениями лучше, чем с более низкими.

Несмотря на то, что среднеквадратическая ошибка кубической аппроксимации наименьшая среди всех моделей, а графики накопления разностей превышений наилучшим образом согласуются с измеренными значениями, можно сделать вывод, что кубическая модель наилучшим образом описывает движение нивелирных пунктов. Однако такой вывод нельзя считать окончательным. Необходимо продолжить исследование при наличии большего количества эпох измерений.

#### Библиографические ссылки

1. Пеллинен Л.П. Высшая геодезия (теоретическая геодезия). М.: Недра, 1978.
2. Шароглазова Г.А. Применение геодезических методов в геодинимике. Новополоцк: ПГУ, 2002.
3. Шароглазова Г.А., Долгий П.С. Многодисциплинарный подход к моделированию геодинимических процессов // Вестник Полоцкого государственного университета. Сер. Ф. Строительство. Прикладные науки. 2018. № 8. С. 180–184.
4. Остач О.М., Дмитроченков В.Н. Методическое руководство по геодезическим работам на геодинимических полигонах. М.: ЦНИИГАИК, 1984.
5. Шароглазова Г.А., Коровкин В.Н., Соловьев А.Н. и др. Анализ применения метода Христова к исследованию вертикальных движений земной коры // Вестник Полоцкого государственного университета. Серия Ф: Строительство. Прикладные науки. 2008. №12. С. 99–105.
6. Топ-10 языков программирования в 2017 году по версии GitHub // Tproger – создано программистами для программистов [Электронный ресурс]. Режим доступа: <https://tproger.ru/articles/github-top-10-languages-2017/>. Дата доступа: 01.02.2018г.
7. PYPL PopularitY of Programming Language // PYPL PopularitY of Programming Language [Electronic resource]. Mode of access: <http://pypl.github.io/PYPL.html>. Date of access: 01.02.2018.
8. TIOBE Index | TIOBE – The Software Quality Company // TIOBE Index | TIOBE – The Software Quality Company [Electronic resource]. Mode of access: <https://www.tiobe.com/tiobe-index/>. Date of access: 01.02.2018.
9. Введение в PyQt5 [урок 1] [Электронный ресурс]. Режим доступа: <http://python-3.ru/page/into-pyqt5>. Дата доступа: 05.03.2018.
10. Вандер Плас Дж. Python для сложных задач. Наука о данных и машинное обучение. Санкт-Петербург: Питер, 2018.
11. NumPy – NumPy // NumPy – NumPy [Electronic resource]. Mode of access: <http://www.numpy.org/>. Date of access: 03.02.2018.
12. Matplotlib: Python plotting – Matplotlib 2.2.2 documentation [Electronic resource]. Mode of access: <https://matplotlib.org/>. Date of access: 05.08.2018.