

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Санфиоров И.А. Ярославцев А.Г. Опыт применения сейсморазведки ОГТ для решения инженерно-геологических задач // Геофизика. – 2004. – № 3. – С. 27-30.
2. Бобров В.Ю. Сравнительный анализ источников упругих колебаний для инженерных сейсморазведочных работ // XIX Уральская молодежная научная школа по геофизике: сб. науч. материалов / ИГФ УрО РАН [и др.]. – Екатеринбург, 2018. – С.27-29.
3. Geodevice: [сайт]. – URL: <https://geodevice.ru/main/seismic/sources/awd/awd33> (дата обращения 04.02.2020). – Текст электронный.
4. Сульфидные медно-никелевые руды норильских месторождений / [Генкин А.Д., Дистлер В.В., Гладышев Г.Д. и др.; отв. Ред. Т.Н. Шадлун]. – М.: Наука, 1981. – 234 с.: ил.
5. Никитин В.Н. Основы инженерной сейсмологии. – М.: Изд-во МГУ, 1981. – 176 с.: ил.
6. Санфиоров И.А. Рудничные задачи сейсморазведки МОГТ / ГИ УрО РАН. – Екатеринбург, 1996. – 168 с.
7. Горяинов Н.Н., Ляховицкий Ф.М. Сейсмические методы в инженерной геологии. – М.: Недра. 1979. – 143 с.: ил.
8. Фрейдин А.М., Неверов А.А., Неверов С.А. Подземная разработка рудных месторождений / Учебное пособие.- Под редакцией чл.-корр. РАН В.Н. Опарина. – Новосибирск.: ИГД СО РАН, 2010. – 372с.

УДК 550.83:004.42

DOI:10.7242/echo.2020.3.12

**СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ ЗАДАЧ ГРАВИРАЗВЕДКИ И МАГНИТОРАЗВЕДКИ**

И.В. Геник

Горный институт УрО РАН, г. Пермь

Аннотация: Возможность иметь современное программное обеспечение может быть обеспечена несколькими путями. Первый – это разработка собственных программ. Данный путь ограничен постоянными существенными затратами, которые должны окупаться тем или иным способом. Затраты включают в себя такие зачастую вынужденные меры, как поддержка устаревшего программного кода, разнообразных специфических узкоспециализированных библиотек, расширяющих возможность старого кода, невозможность и ограничения по использованию новых программных решений, что существенно увеличивает время разработки. Второй путь – покупка готовых программных пакетов и их обновлений, что почти полностью исключает собственную научно-исследовательскую деятельность и также требует затрат, пусть и не таких существенных, как в первом случае. Третий путь связан с использованием свободного программного обеспечения, подразумевая использование бесплатных программных продуктов, их самостоятельное изучение, использование и совершенствование.

В статье рассмотрено свободное программное обеспечение, достигшее значительных успехов за последние 10-15 лет. Главное внимание уделено программам и библиотекам, предназначенным для решения задач гравиметрии и магнитометрии: PyGIMLi, Fatiando a Terra, PyGMI, SimPEG. Указанное программное обеспечение реализовано на языке программирования Python и использует продвинутое математические и прочие библиотеки этого языка. Программы и библиотеки дают возможности решать прямые и обратные задачи гравиразведки, магниторазведки, а также некоторых других геофизических методов, выполнять операции с данными, визуализировать поля и модели. Возможности языка программирования Python позволяют наращивать необходимую функциональность интерпретации гравиметрии и магнитометрии на пути комплексирования с другими геофизическими методами, обработки больших данных, интеграции с геоинформационными системами и реализации геофизических вебсервисов.

Ключевые слова: геофизика, гравиразведка, магниторазведка, интерпретация, программное обеспечение, свободное программное обеспечение.

Введение

Современное программное обеспечение должно отвечать многим требованиям: функциональность, гибкость и расширяемость, постоянная модернизация, дешевизна использования. Попытки реализовать указанное при собственной разработке программного обеспечения, несмотря на кажущуюся простоту, упираются в значитель-

ные затраты на улучшение, окупающиеся или при массовом (низкоприбыльном, по типу Windows), или эксклюзивном (высокоприбыльном, по типу систем автоматизированного проектирования) использовании. Покупка программного обеспечения, особенно специального, также имеет ряд минусов. В российском сегменте оно зачастую недостаточно функционально или не используется за рубежом, а иностранное программное обеспечение (ПО) дорого для потребителя в РФ, что особенно касается специализированного геофизического программного обеспечения. Все это приводит к необходимости рассмотрения свободного программного обеспечения, как альтернативы традиционному подходу в виде полностью собственной разработки или покупки.

Методы и подходы

Целью работы является анализ общих вопросов, связанных с разработкой программного обеспечения, их применение к программам, связанным с обработкой гравиметрических и магнитометрических данных; определение основных проблем и путей их решения, нацеленных на минимизацию затрат, повышение гибкости и скорости разработки, а также на наиболее полное использование всех возможностей современных информационных технологий.

Результаты и обсуждение

Проблемы создания и использования программного обеспечения

В современных условиях для начала разработки программного обеспечения имеются благоприятные возможности: языки программирования имеют развитые возможности, включая разработку в объектно-ориентированном и функциональном стиле; хорошие стандартные и дополнительные библиотеки, хорошие математические библиотеки как для отдельных языков (C/C++, Python, Fortran, Java, Julia), так и многоязыковые (ALGLIB, IMSL Numerical Libraries, NAG Library и др.)¹; очень удобные интегрированные среды разработки (Integrated Development Environment, IDE)²; обширные сообщества пользователей, которые могут дать ответы на разные вопросы. Все это потенциально позволяет создавать многофункциональные, гибкие и расширяемые программы.

Проблемы начинаются на следующих этапах, связанных с постоянной модернизацией, поддержкой использования и сопровождения. Имеются два пути решения.

Первое направление связано с отказом от создания единого продукта и концентрация усилий на отдельных программах. Это значительно уменьшает затраты на архитектуру и целостность решений, но с другой стороны приводит к невозможности получить интеграционный эффект от единой обрабатывающей системы, множество разрозненных программ останется зачастую сделанными в первом приближении и не имеющими дальнейшего развития.

Второй путь, связанный с постоянной разработкой собственной системы, приводит к значительным затратам на улучшения, окупающиеся или при массовом (низкоприбыльном, по типу Windows или MS Office), или эксклюзивном (высокоприбыльном, по типу разного рода систем автоматического проектирования, инженерии, управления данными и информацией и т.д.) использовании. Обеспечение постоянства процесса разработки требует постоянных маркетинговых усилий или на массовом рынке, или на узкоспециализированных рынках. Отметим, что специализированные рынки могут быть очень малыми по емкости, допуская рентабельное финансовое существование не более семи-девяти, а часто трех-четырех программных решений.

¹ https://en.wikipedia.org/wiki/List_of_numerical_libraries

² https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments

Попытки уйти от собственной разработки и общее развитие ситуации в направлении стандартизации ПО приводят к использованию покупных проприетарных³ (частных, патентованных, имеющих собственника) программ. В российских условиях это может быть оправдано для массового сегмента (офисные и бухгалтерские программы), но в специализированных случаях все не так однозначно. Отечественные программы, ввиду малого внутреннего рынка, могут быть недостаточно функциональны, развиваясь весьма медленно или только в направлении, интересном персонально авторам. Зарубежные программы несут более развитые средства, но имеют значительную цену. Дополнительный фактор, появившийся в последние годы – санкции и импортозамещение, поэтому не очевидны долговременные возможности использования иностранного покупного ПО. Общей проблемой для покупного программного обеспечения является жестко заданная функциональность, закрытость системы, невозможность (или весьма не дешевая возможность) расширить возможности программы. Зачастую пользователи приходят к комбинации проприетарных программ с собственными модулями, выполняющими недостающие обработки. Существенным недостатком и ограничением проприетарного обеспечения является то, что пользователь является фактическим бесплатным тестировщиком, рекламщиком и даже в некотором роде разработчиком, предлагая изменения в программные функции. Взамен пользователь получает локальную улучшенную версию «черного ящика», привязанную с помощью системы защиты от копирования к ключу или компьютеру, а все улучшения, научные и торговые плюсы, достаются собственникам ПО. В некотором приближении это можно характеризовать как превращение пользователя в узкого потребителя («кнопочника») – рекламиста ПО. Такой результат может быть приемлемым для крупных геофизических сервисных организаций, занятых чисто производственной тематикой, но крайне нежелателен для научных и учебных организаций, устраняя всю научно-исследовательскую часть и возможности ее окупить.

В российских условиях значительная ориентация на проприетарное программное обеспечение уходит корнями в особенности советского и постсоветского развития.

Советская специфика связана со следующими обстоятельствами.

Во-первых, разделение образования и научных исследований по разным организациям, в отличие от зарубежных университетов, являющихся лидерами научных инноваций.

Во-вторых, намного меньшие возможности для собственных разработок и сообществ вокруг них, связанные с советским решением конца 1960-х годов о копировании зарубежной компьютерной техники и использованием готового ПО. Это было проявлением намного меньшего финансирования разработки программного обеспечения, ставшей уже тогда в США значимой отраслью, работавшей с банками, страховыми компаниями и множеством небольших платежеспособных потребителей.

В-третьих, намного большая информационная закрытость.

Постсоветские особенности напрямую вытекают из опыта 1990-х годов.

Во-первых, «дыра» в финансировании науки и образования, что чрезвычайно сузило возможности собственных исследований.

Во-вторых, отсутствие крупных корпораций, в первую очередь информационных, подобных IBM, Sun Microsystems и др., тесно связанных с университетами и заинтересованных в общей информационной и программной среде и финансирующих её.

В-третьих, весьма либеральное (в 1990-х) отношение к пиратскому программному обеспечению, что опять подрывало собственные разработки.

В-четвертых, намного меньшее, чем в зарубежных университетах, внимание к свободному программному обеспечению. Примечательный факт: если за рубежом в иссле-

³ https://ru.wikipedia.org/wiki/Проприетарное_программное_обеспечение

довательских организациях использовали в основном операционные системы на основе UNIX, а потом Linux, то в РФ в 1990-е годы перешел почти тотальный переход на «бесплатный» Windows. В дальнейшем, начавшееся с 2000-х постепенное ужесточение требований к лицензионности ПО привело к тому, что после изначальной привязки к Windows и проприетарным программам, был затруднен переход на Linux и бесплатное программное обеспечение, имеющееся для него.

В-пятых, ползучая приватизация программного обеспечения, сделанного в советское время в НИИ и ВУЗах, приведшая к тому, что разработки, сделанные на государственные деньги, перешли к частникам, которые не желали появления каких-либо конкурентов и превращали ВУЗы в площадки для продвижения своей новой собственности, предоставляя бесплатные версии программ для учебных целей. Ситуация усугублялась тем, что российский частник был крайне мелким и не выдвигал каких-либо общезначимых проектов, в отличие, например, от IBM, вкладывающей миллиарды долларов в Linux, Canonical, создавшей Ubuntu (дистрибутив Linux), или других компаний, поддерживающих Free Software Foundation (Фонд свободного программного обеспечения)⁴.

В-шестых, возможности свободного программного обеспечения (СПО) значительно расширились, начиная только с 2000-х годов, когда Linux стали поддерживать крупные корпорации, а Microsoft начала поворачиваться лицом к Linux. Особенно большие изменения произошли в последние 5-10 лет, когда, например, появились или были переведены в свободный доступ многие ныне популярные библиотеки (TensorFlow, Pandas, PyTorch).

Свободное программное обеспечение

Термин свободное программное обеспечение (СПО)⁵ описывает следующие права («свободы»): программу можно использовать с любой целью; изучать работу программы и адаптировать её, условием этого является доступность исходного текста программы; свободно распространять копии; улучшать и публиковать свою улучшенную версию, условием этого является доступность исходного текста программы и возможность внесения в него модификаций и исправлений.

Начиная с 2000-х годов произошли значительные изменения в сфере свободного программного обеспечения как общего назначения, так и специализированных. Графический интерфейс Linux в значительной степени подтянулся к возможностям Windows, созданы и стали удобными в использовании новые интегрированные среды разработки (JetBrains, 2000; Eclipse, 2001; Lazarus, 2001; Qt Creator, 2009). Особо примечателен начавшийся примерно 10-15 лет назад поворот Microsoft в сторону СПО⁶. Из числа крупных шагов в этом направлении – выпуск .NET Core (2016) – модульной платформы для разработки программного обеспечения, потенциально являющейся серьезным конкурентом Java⁷. Появилось много других программ, ставших качественными продуктами в своих нишах – офисные пакеты (LibreOffice, 2010) графические редакторы (Inkscape, 2003; Krita, 2005), браузеры (Chromium, 2008), библиотеки (OpenCV, 2006) др.

Последние 10-15 лет также стали временем бурного развития свободного геофизического программного обеспечения. Появились крупные платформы для анализа данных (Madagascar, 2006) и пакеты для отдельных геофизических методов, в первую очередь сейсморазведки (ObsPy, 2008; PySIT, 2011; JavaSeis, 2015 и другие). Сейсморазведка

⁴ <https://www.fsf.org>

⁵ https://ru.wikipedia.org/wiki/Свободное_программное_обеспечение

⁶ https://www.cnews.ru/news/top/microsoft_povorachivaetsya_v_storonu

⁷ <https://habr.com/ru/company/jugru/blog/327492/>

имеет несколько десятков свободных пакетов и крупных библиотек, намного превосходя другие геофизические методы.

Свободное программное обеспечение для гравиразведки и магниторазведки

Решение задач гравиразведки и магниторазведки возможно с использованием всего четырех крупных пакетов и библиотек, что отражает как меньшую распространенность этих методов (региональные работы, детальные работы на металлические полезные ископаемые) по сравнению с сейсмикой, так и достаточно устоявшиеся алгоритмы обработки, намного менее переменчивые, чем сейсморазведка. С другой стороны пакеты новые: PyGIMLi (2009), Fatiando a Terra (2010), PyGMI (2014), SimPEG (2015), и написаны на языке программирования Python, что упрощает их применение как по причине простоты языка, так и вследствие возможности использования библиотек разных пакетов. Все указанные программные комплексы и библиотеки реализуют обработку не только данных гравиметрии и магнитометрии, но и ряда других геофизических методов.

Использование одного языка в новых пакетах программ является значительным преимуществом по сравнению с более старым ПО для других методов, где используют несколько языков: обычно это C/C++, Fortran, но иногда Java. Важным достоинством языка Python является наличие для него больших специально разработанных библиотек математических программ (NumPy, SciPy), визуализации (Matplotlib, Plotly и др.), анализа данных (TensorFlow, Pandas, PyTorch), что делает в современных условиях этот язык главным средством обработки больших массивов данных. Отметим, что Python также является встроенным языком программирования в свободной геоинформационной системе (ГИС) QGIS и на нем можно создавать отдельные приложения на основе данной ГИС. Это открывает возможность создавать высококачественные геофизические приложения и в перспективе вебсервисы, реализующие обработку и интерпретацию данных геопотенциальных и других геофизических полей.

Рассмотрим главные возможности упомянутых выше программ и библиотек.

PyGIMLi⁸ – библиотека для решения прямых и обратных геофизических задач, имеющая следующие возможности: операции с 2D и 3D сетками, реализация методов конечных элементов и объемов⁹, моделирование с различными граничными условиями, решение обратных задач методом Гаусса-Ньютона (нелинейный метод наименьших квадратов) с гибкими возможностями регуляризации [2]. PyGIMLi является именно библиотекой и не имеет графического интерфейса, хотя может работать с блокнотами Jupyter¹⁰. Библиотека позволяет работать со всеми основными геофизическими методами (гравиметрия, магнитометрия, сейсмика, электрометрия), производить совместное решение обратных задач для нескольких геофизических методов, а также выполнять гидродинамическое моделирование.

Fatiando a Terra¹¹ – ранее единый пакет, а теперь совокупность пакетов-библиотек (*Verde, Pooch, Harmonica, Boule, RockHound*) для геофизики, в связи с этим охарактеризуем кратко каждый пакет. *Verde* – библиотека, написанная на языке Python, для обработки пространственных данных и их интерполяции на регулярных сетках; для переноса данных на сетку используют подход функций Грина [3]. *Pooch* – пакет, предназначенный для управления файлами (загрузка данных с сервера, хранение их в локальном каталоге, обработка обновлений), используемый другими библиотеками [4].

⁸ <https://www.pygimli.org>

⁹ https://ru.wikipedia.org/wiki/Метод_конечных_объемов

¹⁰ <https://jupyter-notebook.readthedocs.io/en/latest/notebook.html#notebook-documents>

¹¹ <https://www.fatiando.org>

Harmonica – библиотека для обработки и моделирования гравитационных и магнитных данных, включая поправку Буге и поправку за рельеф, продолжение поля вверх и вниз, трехмерные обратные задачи, моделирование эффектов для основных геометрических фигур, таких как сферы, призмы, многоугольные призмы и др. *Boole* – сервисная библиотека для работы с земным эллипсоидом, выполнения расчета нормального гравитационного поля, сферического гравитационного моделирования и преобразования координат. *RockHound* – библиотека для работы с геофизическими моделями и наборами данных (PREM – preliminary reference Earth model; CRUST 1.0 – global model of Earth's crustal structure; ETOPO1 – Global Relief Model) и загрузки их в структуры данных Python.

PyGMI¹² – пакет для моделирования и интерпретации, предназначенный для обработки магнитометрической, гравиметрической, сейсмологической информации и других данных; пакет разработан Советом по геонаукам (Геологическая служба Южной Африки) [1]. Реализованы: алгоритмы манипуляции с данными, решение прямых задач гравитационного и магнитного 3D моделирования, а также работа с данными SEISAN (анализ землетрясений) и дистанционного зондирования Земли. PyGMI (в отличие перечисленных выше библиотек) является готовой интерпретируемой программой, запускаемой под управлением Python.

SimPEG¹³ – пакет, предназначенный для решения задач гравirazведки, магниторазведки, электроразведки. Реализованы методы моделирования различной размерности. Решение обратных задач имеет блоки регуляризации, оптимизации, подбора данных, совместной инверсии. В состав пакета входят дополнительные библиотеки, развиваемые параллельно: *discretize* – решение дифференциальных уравнений в частных производных и визуализация; *pymatsolver* – решение уравнений с разреженными матрицами; *geoana* – библиотека аналитических функций, наиболее часто используемых в геофизике.

Все рассмотренные выше свободные геофизические пакеты, реализующие решение задач гравиметрии и магнитометрии, в настоящее время быстро эволюционируют, привлекая к себе внимание геофизиков разных стран, что объясняется следующим образом. Во-первых, в плане свободного программного обеспечения геофизика значительно отстает от других наук, что объясняется значительной историей развития с привычкой к проприетарному ПО, расходы на которое покрывали доходы от геофизических услуг добывающим организациям в условиях благоприятной сырьевой конъюнктуры. Во-вторых, период бурного роста поисков и разведки месторождений полезных ископаемых прошел во всем мире, разведано много потенциальных месторождений, вопросы связаны только с финансированием их разработки, примером тому в РФ судьбы Удоканского медного и Штокмановского газоконденсатных месторождений. В-третьих, значительным прогрессом в свободном программном обеспечении и сервисов, с ним связанных (GitHub), буквально за последнее десятилетие. Все это открывает хорошие возможности для лидеров в перспективе выйти на устойчивую работу по модели RedHat¹⁴. Дополнительным стимулом к этому является увеличение численности геофизиков в развивающихся странах, которые могут стать клиентской базой для свободных проектов.

Выводы

Проанализированы вопросы создания и использования программного обеспечения, показано, что собственная постоянная существенная модификация программного обес-

¹² <https://pypi.org/project/pygmi>

¹³ <https://simpeg.xyz>

¹⁴ <https://www.forbes.ru/tehnologii/373927-kak-zarabotat-milliony-na-otkrytom-kode-ot-red-hat-do-nginx>

печения окупается только при ряде специфических условий. Приобретение лицензий проприетарных программ является не самым лучшим выходом: возникает постоянная зависимость от разработчика, которая приводит с одной стороны к деградации собственного научно-исследовательского потенциала, а с другой является бесплатной помощью другой стороне, продающей «черные ящики».

Одним из кардинальных решений является переход на свободное программное обеспечение, которое позволяет самостоятельно модифицировать и распространять программное обеспечение, эффективно взаимодействовать с другими научными группами, работающими в соответствующих направлениях по всему миру, с потенциальной возможностью монетизировать разработки, выполняя различный спектр услуг (обучение, консультации, разработку под заказ и др.).

Рассмотрены особенности четырех главных пакетов программ и библиотек, реализующих алгоритмы, созданные на языке программирования Python, необходимые для обработки и интерпретации данных гравиразведки и магниторазведки. Проанализированы главные функциональные блоки программного обеспечения и показаны их возможности. Поскольку гравиметрия и магнитометрия являются методами, использующими геопотенциальные поля и для них недостижима детальность, предоставляемая сейсморазведкой, то важным направлением развития грави- и магниторазведки является развитие технологий комплексирования, широкие возможности для которых предоставляют библиотеки языка Python, интеграция с геоинформационными системами, а также другие новые направления, которые становятся возможными с использованием современного свободного программного обеспечения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Leroux M. PyGMI: an open source Generic Measurement Interface for data acquisition and instrument control, in Python with graphics in Qt. 2015, August. – DOI: 10.13140/RG.2.1.2254.5126.
2. Rücker C., Günther T., Wagner F.M. pyGIMLi: An open-source library for modelling and inversion in geophysics // Computers and Geosciences. – 2017. –V. 109. – P.106-123. DOI: <https://doi.org/10.1016/j.cageo.2017.07.011>.
3. Uieda L. Verde: Processing and gridding spatial data using Green's functions. // Journal of Open Source Software. – 2018. – V. 3, № 29. –P. 957. – DOI:10.21105/joss.00957.
4. Uieda L., Soler S.R., Rampin R., van Kemenade H., Turk M., Shapero D., Banihirwe A., Leeman J. Pooch: A friend to fetch your data files // Journal of Open Source Software. – 2020. – V. 5, № 45. – P. 1943. DOI:10.21105/joss.01943

УДК 550.8.05

DOI:10.7242/echo.2020.3.13

ОЦЕНКА ИНФОРМАТИВНОСТИ СЕЙСМОРАЗВЕДОЧНЫХ ИССЛЕДОВАНИЙ В ЗОНАХ РАЗВИТИЯ СОЛЯНОГО КАРСТА НА ОСНОВЕ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

А.А. Жикин

Горный институт УрО РАН, г. Пермь

Аннотация: На основе математического моделирования волновых полей на участке развития соленого карста на примере Харюшенского поднятия, расположенного в пределах Верхнекамского месторождения калийных и магниевых солей, выполнена оценка информативности малоглубинной сейсморазведки высокого разрешения в комплексе с шахтными сейсмоакустическими исследованиями.

Ключевые слова: малоглубинная сейсморазведка, шахтная сейсморазведка, синтетический временной разрез