

Компьютерные технологии в научных исследованиях и образовании

Юрий Анатольевич Богоявленский, заведующий кафедрой Информатики и математического обеспечения, к.т.н., доцент, ybgv@cs.petrso.ru,

Лекция 1. Введение

Алгоритмы для ЭВМ

Числа с плавающей точкой

Классы и технологии математического ПО

Свободные математические пакеты GNU

Научная библиотека GSL - GNU Scientific Library

Замечание о функциях GSL для работы с векторами и матрицами.

Предметные области алгоритмов библиотеки GSL

Примеры использования библиотеки GSL

Технологические характеристики библиотеки GSL

Математики внесли основополагающий вклад в создание ЭВМ. Достаточно вспомнить авторов теории алгоритмов (стандартизации понятия алгоритма) английского математика Алана Тьюринга, американских математиков Эмиля Поста и Алонзо Черча. Большой вклад в эту теорию также внес Андрей Андреевич Марков (младший), который ввел и обосновал понятие нормального алгоритма.

Нельзя не отметить и математика с мировым именем Джона фон Неймана (при рождении Янош Лайош Нейман, при работе в Берлинском университете - Иоганн фон Нейман). Он является ведущим автором работы: Burks A. W., Goldstine H. H., Neumann J. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. —Institute for Advanced Study,

Princeton, N. J., July 1946. В этой работе были сформулированы основные принципы логической структуры ЭВМ, названные позже «архитектура фон Неймана» и реализованные в десятках ЭВМ по всему миру.

Обратим внимание, что в этой работе речь идет об «электронном вычислительном инструменте», который, очевидно, предназначается для реализации численных методов математики. Например ныне покойный профессор З. Л. Рабинович в конце 40-х - начале 50-х годов двадцатого века под руководством академика С. А. Лебедева участвовал в разработке ламповой ЭВМ МЭСМ. На организованной в 2006 г ПетрГУ конференции SoRuCom 2006 в своем интервью он отметил, что эта ЭВМ решала баллистические задачи. Таким образом первые ЭВМ применялись, как правило, для решения задач численных методов математики.

Очевидно, что наблюдаемый с конца 40-х годов XX века по настоящее время активный рост применений ЭВМ в самых разных областях невозможен без наличия соответствующих алгоритмов, например сервисы сети Интернет реализуются сотнями сетевых алгоритмов, таких, в частности, как взаимодействующие алгоритмы протокола TCP — Slow Start, Congestion Avoidance, Fast Retransmit и Fast Recovery.

Это означает, что также активно растут исследования и разработки собственно алгоритмов, опирающиеся на теорию их сложности, которая, как самостоятельная научная дисциплина, начала развиваться в 1960-х гг. XX века. Параллельно развиваются и методы и технологии реализации и тестирования алгоритмов, формирования их коллекций (например библиотек по предметным областям) и предоставления их пользователям.

Таким образом можно сказать, что именно в XX веке начался бурный рост алгоритмической базы, хотя использование алгоритмов прослеживается со времен шумерской цивилизации, самой древней из известных.

Сам термин «алгоритм» происходит от латинских слов «Dixit Algorizmi» (Сказал Аль-Хорезми), встречающихся в выполненных начиная с XII века латинских переводах написанных в первой половине IX века трудов персидского математика Мухаммеда ибн Муса аль-Хорезми (англ.

Muhammad ibn Musa al-Khwarizmi). В них автор, в частности, основываясь на работах индийских математиков, излагает алгоритмы действий в десятичной системе счисления и нахождения решений линейных и квадратных уравнений.

Примеры:

- перевод XII века на латынь под названием «De numero Indorum» (Индийский счет) не сохранившегося оригинала о десятичной системе счисления;
- перевод на латынь книги «Китаб аль-Джебр ва-ль-Мукабала» (Краткая книга восполнения и противопоставления), сделанный в 1145 г. английским востоковедом Робертом Честерским (Robert of Chester) под названием «The Compendious Book on Calculation by Completion and Balancing».

Разработчики алгоритмов, не ставящие себе целью их реализацию на ЭВМ, описывают и публикуют их используя т. н. псевдокоды - неформальные языки, опускающие несущественные для понимания подробности и специфический синтаксис. Псевдокоды описаны в ресурсах:

— <https://en.wikipedia.org/wiki/Pseudocode>

— [https://ru.wikipedia.org/wiki/Псевдокод_\(язык_описания_алгоритмов\)](https://ru.wikipedia.org/wiki/Псевдокод_(язык_описания_алгоритмов))

Псевдокод для LaTeX реализован в пакете algorithmx и описан в ресурсе:

<http://mirrors.ctan.org/macros/latex/contrib/algorithmicx/algorithmicx.pdf>

а сам пакет LaTeX algorithmx доступен на ресурсе:

— <https://ctan.org/tex-archive/macros/latex/contrib/algorithmicx>

Пример псевдокода из: <https://en.wikipedia.org/wiki/Pseudocode>

```
algorithm ford-fulkerson is
  input: Graph  $G$  with flow capacity  $c$ ,
         source node  $s$ ,
         sink node  $t$ 
  output: Flow  $f$  such that  $f$  is maximal from  $s$  to  $t$ 

  (Note that  $f_{(u,v)}$  is the flow from node  $u$  to node  $v$ , and  $c_{(u,v)}$  is the flow
  capacity from node  $u$  to node  $v$ )

  for each edge  $(u, v)$  in  $G_E$  do
     $f_{(u, v)} \leftarrow 0$ 
     $f_{(v, u)} \leftarrow 0$ 

  while there exists a path  $p$  from  $s$  to  $t$  in the residual network  $G_f$  do
    let  $c_f$  be the flow capacity of the residual network  $G_f$ 
```

```
 $c_f(p) \leftarrow \min\{c_f(u, v) \mid (u, v) \text{ in } p\}$   
for each edge  $(u, v)$  in  $p$  do  
     $f(u, v) \leftarrow f(u, v) + c_f(p)$   
     $f(v, u) \leftarrow -f(u, v)$ 
```

```
return  $f$ 
```

Алгоритмы на языках программирования публикуются в печатном виде, например:

- Агеев М.И., Алик В.П., Галис Р.М. Алгоритмы : (1-50) / Под общ. ред. М.И. Агеева; АН СССР. Вычислит. центр. - Москва : Б. и., 1966. - 106 с. ; 22 см. - (Общие вопросы программирования ; Вып. 2). - Библиогр.: с. 102-103 (22 назв.)

АКАДЕМИЯ НАУК СССР
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

ОБЩИЕ ВОПРОСЫ ПРОГРАММИРОВАНИЯ

ВЫПУСК 2

М.И.АГЕЕВ, В.П.АЛИК, Р.М.ГАЛИС

А Л Г О Р И Т М Ы
(1-50)

Под общей редакцией
М.И.АГЕЕВА

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР АН СССР
МОСКВА - 1966

АЛГОРИТМ 13а
РЕКУРСИВНОЕ ВЫЧИСЛЕНИЕ ПОЛИНОМА
ЛЕЖАНДРА $P_n(x)$ [S 16]

Процедура leg вычисляет полином Лежандра $P_n(x) = (1/2^n * n!) \times d^n/dx^n(x^2 - 1)^n$ для любого действительного аргумента x и любого порядка n по рекурсивной формуле.

```
real procedure leg(n,x);
  value n,x; real x; integer n;
begin real a,b,c; integer i;
  a:=1; b:=x;
  if n=0 then c:=a else
    if n=1 then c:=b else
      for i:=1 step 1 until n-1 do
        begin c:=b*x+(i/(i+1))*(b*x-a);
          a:=b; b:=c;
        end;
      leg:=c;
end leg;
```

Свидетельство к алгоритму 13а

Алгоритм 13а получен в результате исправления и ординарной переработки алгоритма 13 (Galler G. M. "CACM", 1960, № 6). Была исправлена опечатка — отсутствие точки с запятой перед последним оператором тела процедуры.

Рекурсивная формула в алгоритме была сверена с аналогичной формулой, приведенной в [10].

Подтверждение к алгоритму 13

Дж. Херндон (Herndon J. "CACM", 1961, № 4)

Алгоритм был переведен на БАЛГОЛ и проверен на машине Burroughs 220. Процедура leg(n,x) дала точность более 7 знаков для $n=0; 1; 4; 9$ и для $x=0; 0,1; 0,2; 0,7; 1,9; 5,0$.

АЛГОРИТМ 14а
КОМПЛЕКСНАЯ ИНТЕГРАЛЬНАЯ ПОКАЗАТЕЛЬНАЯ ФУНКЦИЯ [S13]

Процедура ekz вычисляет $w(k,z) = u+iv = z^k e^z \int_z^\infty \frac{e^{-t}}{t^k} dt$ путем представления в виде непрерывной дроби, взятой из работы Уалла

- Уилкинсон Дж.Х., Райнш С. Справочник алгоритмов на языке Алгол. Линейная алгебра, М., 1976.

Для применения этих алгоритмов их нужно было перенести на носитель для ввода в ЭВМ и отладить.

С 1975 г. издается ежеквартальный журнал:

«ACM Transactions on Mathematical Software» (TOMS) [Электронный ресурс] — URL: <https://dl.acm.org/journal/toms> (31.07.2022).

Цель журнала — «распространение существенных результатов по теоретическим основам численных, символьных, алгебраических и геометрических вычислительных приложений. Основное внимание уделяется анализу и построению алгоритмов и программ, а также взаимодействию программ и архитектуры. Алгоритмы, представленные в TOMS, доступны в виде коллекции алгоритмов ACM» (1025 алгоритмов на 31.07.2022) по адресу на ресурсе:

ACM Collected Algorithms [Электронный ресурс]

— URL: <https://calgo.acm.org/> (31.07.2022)

Алгоритм в коллекции представляется на некотором языке программирования (часто одна из версий Fortran) вместе со ссылкой на соответствующую статью в TOMS, описанием и контрольными тестами.

Существует много журналов по этим проблемам, например организация «Society for Industrial and Applied Mathematics» (SIAM) издает:

«SIAM JOURNAL ON SCIENTIFIC COMPUTING» [Электронный ресурс]

— URL: <https://www.siam.org/publications/journals/siam-journal-on-scientific-computing-sisc> (31.07.2022).

Редакция так описывает его содержание: ««SIAM Journal on Scientific Computing» (SISC) содержит исследовательские статьи по численным методам и методам научных вычислений. Статьи в этом журнале посвящены вычислительным вопросам, связанным с решением научных или инженерных задач, и включают результаты вычислений, демонстрирующие эффективность предлагаемых методов».

Проблемам вычислительной математики также посвящены журналы:

- «Журнал вычислительной математики и математической физики»;

- «Сибирский журнал вычислительной математики»;
- «Вычислительные методы и программирование»;
- «Математическое моделирование и численные методы»;
- «Numerische Mathematik»; «Journal of Computation and Applied Mathematics»;
- «Computer Physics Communications»;
- «SIAM Journal of Numerical Analysis».

Оценку существующих классов алгоритмов можно получить в материале:

https://en.wikipedia.org/wiki/List_of_algorithms.

С развитием сфер применения ЭВМ и программных систем сформировался класс систем, предназначенный для решения различных задач математики — специализированных пакетов в которых алгоритмы уже реализованы и отлажены. Для их применения необходимо знать свойства и ограничения алгоритма, язык взаимодействия с пакетом, задать исходные данные и интерпретировать и оценить основные свойства полученного решения (например его точность, границы устойчивости, зависимость от начальных параметров).

NBNB. Применение пакета освобождает пользователя от программирования и отладки алгоритма, но не освобождает его от понимания специфики и ограничений алгоритма.

Числа с плавающей точкой

Целые числа и диапазоны их представления в ОП не подходят для многих задач, поэтому было введено представление чисел с «плавающей точкой», соответствующее записи вида:

$Q = m \cdot e^p$, где m — значение мантиссы, e — архитектурно зафиксированное основание, p — значение порядка. Такая запись распространена, пример — число Авогадро — $NA = 6.0214076 \cdot 10^{23}$ моль⁻¹, где $m = 6.0214076$, $e = 10$, $p = 23$

Важно понимать, что это представление является приближенной моделью множества вещественных чисел. До 1985 г. разные производители процессоров по разному реализовывали представления и операции над этими числами (компания Intel выпускала отдельную микросхему FPU — Floating Point Unit, которая позже стала встраиваться в микросхему процессора), что затрудняло перенос программ на другие платформы. Например, программы итерационных алгоритмов, прекращающие работу по достижении некоторого малого значения при переносе с ЭВМ ЕС 1022 (минимальное значение числа около 10^{-78}) на ЭВМ БЭСМ-6 (минимальное значение числа около 10^{-19}) требовали коррекции.

Подобные проблемы исчезли после того, как форматы чисел с плавающей точкой были зафиксированы в стандарте IEEE 754-1985 и затем уточнены в стандартах IEEE 754-2008 и IEEE 754-2019. Отметим, что стандарт определяет несколько форматов числе разной битовой длины и при активной работе с числами с плавающей точкой следует его изучить. Для иллюстрации покажем на рисунке как определяется число в 32 битовом представлении.

Стандарт *IEEE 754*...

$$F = (-1)^S 2^{(E-127)} (1 + M/2^{23})$$

F – десятичное число

- S – бит знака (31-й бит)
- E – смещение экспоненты (30–23 биты)
- M – остаток от мантиссы (22–0 биты)



23

Примерный диапазон — от $1.2 \cdot 10^{-38}$ до $3.4 \cdot 10^{+38}$.

Классы и технологии математического ПО

Программное обеспечение (ПО) для решения математических задач интенсивно развивалось как по классам самих задач, так и по технологиям реализации ПО. Не претендуя на полноту систематизации приведем основные этапы этого развития.

По классам задач — стали развиваться системы:

- реализации численных методов;
- компьютерной алгебры, автоматизирующие выполнение символьных преобразований (например получение формулы производной из формулы функции);
- решающие задачи математической статистики;
- визуализации математических объектов;

- автоматизации доказательств.

Сейчас существуют сотни, если не тысячи, разнообразных систем. Более детальная классификация систем математического ПО приведена в материалах:

— Математическое программное обеспечение.

[Электронный ресурс] — URL: https://translated.turbopages.org/proxy_u/en-ru.ru.c16c6dd7-62d94de3-a2650a1e-74722d776562/https/en.wikipedia.org/wiki/Mathematical_software#Evolution_of_mathematical_software (21.07.22).

— Mathematical software. [Электронный ресурс] — URL:

https://en.wikipedia.org/wiki/Mathematical_software#Evolution_of_mathematical_software (21.07.22).

Представление о разнообразии и применениях математического ПО можно получить в материалах:

— Oberwolfach References on Mathematical Software. [Электронный ресурс] — URL: <https://orms.mfo.de/index.html> (21.07.22).

— swMATH an information service for mathematical software. [Электронный ресурс] — URL: <https://www.swmath.org/> (21.07.22).

По технологиям реализации ПО можно отметить следующие этапы развития:

- программы на машинном языке и языке ассемблера;
- программы на языках высокого уровня (Фортран, С, другие);
- библиотеки программ в виде объектных модулей для одной платформы;
- переносимые (мобильные относительно платформы) библиотеки программ;
- диалоговые системы с интерпретируемыми языками;
- комбинированные системы (обеспечивается одновременный доступ, например, к программам численных методов и символьных преобразований. Пример - система MathConnex пакета Mathcad версии 7.0 PRO , имеющая 16 компонентов; возможность использования функций

других систем (Excel, Axum, Matlab и др.) и фактическая интеграция с ними;

- веб системы — работа идет через веб обозреватель без необходимости загрузки и установки системы, например система PARI/GP.

Математические пакеты подразделяются на платные, наиболее популярные из которых Mathcad, Matlab, Maple, Mathematica, Statistica и пакты со свободными лицензиям (далее свободные пакеты), которые позволяют успешно решать многие классы задач.

Лицензии на свободные пакеты, как правило, на требуют платы за их применение, но предполагают, что пользователь должен выполнять некоторые требования, например после публикации модифицированного исходного кода программы предоставлять доступ к нему всем желающим, как это определено в лицензии GNU General Public License (GPL). Отметим, что существует много различных свободных лицензий. Тексты лицензий, их переводы на русский язык, ссылки на официальные сайты и др. представлены на ресурсе:

Заглавная страница. Категории статей на LicenseIT. [Электронный ресурс] — URL: http://licenseit.ru/wiki/index.php/Заглавная_страница (08.08.2022).

Основные лицензии на свободные ПО и предоставляемые ими пользователям права описаны в работе:

П. В. Кормина Свободная лицензия ПО. [Электронный ресурс] — URL: https://zakon.ru/blog/2017/4/8/svobodnaya_licenziya_po (08.08.2022).

Отметим, что перед использование конкретного пакета нужно ознакомиться с его лицензией и следовать ее требованиям.

Свободные математические пакеты GNU

Определение из: <http://www.gnu.org/>

GNU — операционная система типа Unix. Это значит, что она представляет собой собрание множества программ: приложений, библиотек, средств разработки и даже игр. Многие программы в GNU выпускаются под эгидой проекта GNU; их мы называем пакетами GNU.

Название “GNU” — это рекурсивное сокращение фразы “GNU's Not Unix!” (“GNU — не Unix!”). “GNU” произносится как гну, одним слогом, как глагол “гнуть” в первом лице, единственном числе настоящего времени изъявительного наклонения.

В системе типа Unix программа, которая выделяет машинные ресурсы и общается с аппаратурой, называется “ядром”. GNU, как правило, применяется сегодня с ядром, называемым “Linux”. Эта комбинация является операционной системой GNU/Linux. Миллионы людей пользуются GNU/Linux, хотя многие ошибочно называют ее “Linux”.

На ресурсе: *Операционная система GNU [Электронный ресурс] — URL: <https://www.gnu.org/software/software.html> (09.08.2022)* дан список 384 активных пакетов GNU и некоторого количества пакетов выведенных из эксплуатации. Также списки свободных систем ПО, в том числе пакетов GNU, по категориям даны на:

The Free Software Foundation (FSF). Welcome to the Free Software Directory. [Электронный ресурс] — URL:

https://directory.fsf.org/wiki/Main_Page (02.08.2022)

В этом документе пакеты математического ПО представлены в категориях «mathematics» и (частично) «science».

Перечислим некоторые математические пакеты GNU.

- GLPK (GNU Linear Programming Kit) предназначен для решения задач линейного и смешанного целочисленного программирования, а также родственных задач большой размерности и реализован как набор подпрограмм, написанных на языке ANSI C. Пакет разработан доцентом МАИ А. О. Махориным и имеет привязки к 12 языкам и системам, в т.ч. к Python, Java, Matlab, Octave, см.: GLPK [Электронный ресурс] — URL: <https://en.wikibooks.org/wiki/GLPK> (09.08.2022)
- GMP (GNU Multiple Precision Arithmetic Library) библиотека для арифметики произвольной точности, работающая с целыми числами со знаком, рациональными числами и числами с плавающей запятой. Применяется в криптографии, системах безопасности, вычислительной алгебре.
- R — это язык и среда для статистических вычислений и графики. R предоставляет широкий спектр статистических методов (линейное и нелинейное моделирование, классические статистические тесты, анализ временных рядов, классификация, кластеризация, другое) и графических инструментов.

Научная библиотека GSL - GNU Scientific Library

GNU (GSL) это библиотека программ численных методов для программистов на языках C и C++. На середину 2022 г. реализовано более 1000 функций с обширным набором тестов.

Разработки библиотек программ для численных методов (например Netlib, IMSL, NG) начались в 70-е годы XX века чтобы решить весьма сложную проблему обеспечения многократного использования уже отлаженного алгоритма и, в том числе, повторного использования его кода при разработке нового ПО. Актуальность этой проблемы определяется двумя факторами:

1. Отладка алгоритмов численных методов и определение границ и условий из применимости представляют собой весьма трудоемкую задачу.
2. Архитектуры аппаратно-программных платформы, на которых выполняются алгоритмы быстро развиваются (например системы с параллельными возможностями).

Библиотека GSL является, в некотором смысле, библиотекой «второго поколения» т. к. она разрабатывалась на основе принципов, позволивших решать проблему многократного использования на существенно более высоком уровне.

Эти принципы представлены на ресурсе:

Mark Galassi, James Theiler, Brian Gough GNU Scientific Library -- Design document[Электронный ресурс] — URL:

https://www.gnu.org/software/gsl/design/gsl-design_toc.html

В частности, в разделе Motivation этого документа авторы декларируют необходимость для ученых и инженеров в наличии библиотеки численных методов, которая:

- является свободной в смысле лицензии GPL;
- реализована на языке C с использованием современных соглашений о кодировании, соглашений о вызовах, областях видимости т. п.;
- ясно и педагогически задокументирована, предпочтительно с помощью инструмента TeXinfo, чтобы обеспечить онлайн представление и вывод в форматах WWW и TeX;
- использует высококачественные современные алгоритмы;
- является переносимой на другие платформы и настраиваемой с помощью инструментов autoconf и automake;
- в основном корректно поддерживает принципы GNU.

Отметим, что выбор языка C очень удачен т. к. его компилятор и базовая системная библиотека libc реализованы для подавляющего большинства современных ОС, в т.ч. Windows и Android.

Затем авторы приводят краткий анализ достоинств и недостатков 20 известных библиотек, 11 из которых не лицензированы под GPL. Многие из них реализованы на языке Fortran, роль которого в развитии программ численных методов невозможно переоценить, но который не имеет технологических преимуществ инструментальной среды языка C.

Далее излагаются технические требования к реализации.

Работа над библиотекой GSL была начата в мае 1996 г. и выполнялась регулярно на протяжении 26 лет от первого выпуска gsl-0.0 в 1996 г. до последнего стабильного выпуска gsl-2.7 в июне 2021 г. На середину 2022 г. библиотека содержит более 1000 функций с обширным набором тестов, реализованных на языках C и C++.

GSL разработана на платформе GNU/Linux с компилятором gcc, сообщалось, что она компилируется на следующих платформах:

- SunOS 4.1.3 & Solaris 2.x (Sparc)
- Alpha GNU/Linux, gcc
- HP-UX 9/10/11, PA-RISC, gcc/cc
- IRIX 6.5, gcc
- m68k NeXTSTEP, gcc
- Compaq Alpha Tru64 Unix, gcc
- FreeBSD, OpenBSD & NetBSD, gcc
- Cygwin
- Apple Darwin 5.4
- Hitachi SR8000 Super Technical Server, cc
- Microsoft Windows

Характерные особенности

- Библиотека использует объектно-ориентированный дизайн. Различные алгоритмы могут быть легко подключены или изменены во время выполнения без перекомпиляции программы.

- Она предназначена для обычных ученых. Любой, кто немного разбирается в программировании на языке Си, сможет сразу же начать пользоваться библиотекой.
- Интерфейс был разработан таким образом, чтобы его было просто связать с языками очень высокого уровня, такими как GNU Guile или Python.
- Все функции библиотеки за счет специальных мер при реализации можно использовать в многопоточных программах.
- Там, где это было возможно, процедуры были основаны на надежных общедоступных пакетах на языке Fortran, таких как FFTPACK и QUADPACK, которые разработчики GSL реализовали на языке C с использованием современных соглашений о кодировании.
- Библиотека проста в компиляции и не имеет никаких зависимостей от других пакетов.

Отметим также, что внутри самой библиотеки в принципе не предусмотрена поддержка параллельных аппаратных архитектур и их элементов.

Замечание о функциях GSL для работы с векторами и матрицами.

В 1979 г. опубликована первая версия развивающегося до сих пор фактического стандарта BLAS (Basic Linear Algebra Subprograms — базовые подпрограммы линейной алгебры), который является API (Application Programming Interface — интерфейс программирования приложений). Он предназначен для стандартизации разработок библиотек, выполняющих основные операции линейной алгебры, такие как умножение векторов и матриц.

GSL требует наличия библиотеки BLAS для векторных и матричных операций. Библиотека CBLAS, по умолчанию поставляемая с GSL, может быть заменена настраиваемой библиотекой ATLAS (Automatically Tuned Linear Algebra Software) для повышения производительности при больших

размерностях матриц. Последняя использует эмпирический подход для оптимизации скорости выполнения функций путем учета системных и аппаратных особенностей платформ. Например может использоваться расширение системы команд SSE (англ. Streaming SIMD Extensions, потоковое SIMD-расширение процессора) — это SIMD- (англ. Single Instruction, Multiple Data, одна инструкция — множество данных). SSE — набор инструкций параллельной обработки данных , разработанный Intel и впервые представленный в процессорах серии Pentium III.

На ресурсе:

GNU Operating System. GSL - GNU Scientific Library [Электронный ресурс] — URL: <https://www.gnu.org/software/gsl/>

Перечислены более двадцати использующих GSL расширений ее функций, десять использующих ее приложений и семнадцать оберток (wrapper) — интерфейсов для использования в популярных языках программирования (Java, Python, Vala, Ruby, и др).

Предметные области алгоритмов библиотеки GSL

Предметные области алгоритмов библиотек GSL из руководства:

Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Patrick Alken, Michael Booth, Fabrice Rossi, Rhys Ulerich GNU Scientific Library Release 2.7 [Электронный ресурс] — URL:

<https://www.gnu.org/software/gsl/doc/latex/gsl-ref.pdf>, Jun 18, 2021, P.728 (01.08.2022)

Complex Numbers	Roots of Polynomials	Special Functions
Vectors and Matrices	Permutations	Combinations
Sorting	BLAS Support	Linear Algebra
CBLAS Library	Fast Fourier Transforms	Eigensystems
Random Numbers	Quadrature	Random Distributions
Quasi-Random Sequences	Histograms	Statistics
Monte Carlo Integration	N-Tuples	Differential Equations
Simulated Annealing	Numerical Differentiation	Interpolation
Series Acceleration	Chebyshev Approximation	Root-Finding

Discrete Hankel Transforms	Least-Squares Fitting	Minimization
IEEE Floating-Point	Physical Constants	Basis splines
Wavelets	Sparse BLAS Support	Sparse Linear Algebra

Примеры использования библиотеки GSL

Вычисление значение функции Бесселя

```
#include <stdio.h>
#include <gsl/gsl_sf_bessel.h>
int main(void)
{
    double x = 5.0;
    double y = gsl_sf_bessel_J0(x);
    printf("J0(%g) = %.18e\n", x, y);
    return 0;
}
```

Компиляция:

```
gcc $(gsl-config --cflags) example.c $(gsl-config -libs)
```

Результат: J0(5) = -1.775967713143382642e-01

Умножение матриц

```
#include <stdio.h>
#include <gsl/gsl_cblas.h>
int
main (void)
{
    int lda = 3;
    float A[] = { 0.11, 0.12, 0.13,
                  0.21, 0.22, 0.23 };

    int ldb = 2;
    float B[] = { 1011, 1012,
                  1021, 1022,
                  1031, 1032 };

    int ldc = 2;
    float C[] = { 0.00, 0.00,
                  0.00, 0.00 };

    /* Compute C = A B */
```

```
cblas_sgemm (CblasRowMajor,
             CblasNoTrans, CblasNoTrans, 2, 2, 3,
             1.0, A, lda, B, ldb, 0.0, C, ldc);

printf (" [ %g, %g\n", C[0], C[1]);
printf ("  %g, %g ]\n", C[2], C[3]);
return 0;
}
Компиляция:
gcc -Wall demo.c -lgslcblas
Результат:
[ 367.76, 368.12
 674.06, 674.72 ]
```

Технологические характеристики библиотеки GSL

Существует много свободных пакетов разного качества. Уровень развития пакета и его основные характеристики можно получить на ресурсе:

Black Duck Open Hub. Discover, Track and Compare Open Source. [Электронный ресурс] — URL: <https://www.openhub.net/> (04.08.2022). Краткие характеристики пакета можно получить по ссылке с указанием имени пакета, например для пакета `gsl__gnu_scientific_library` нужно перейти по ссылке:

https://www.openhub.net/p/gsl__gnu_scientific_library

По данным на 09.08.2022

In a Nutshell, GSL - GNU Scientific Library..

...

has had 10,819 commits made by 20 contributors
representing 598,767 lines of code

...

is mostly written in C

with an average number of source code comments

...

has a well established, mature codebase
maintained by a small development team
with decreasing Y-O-Y commits

...

took an estimated 162 years of effort (COCOMO model)
starting with its first commit in July, 1996
ending with its most recent commit over 1 year ago