

# Информационные технологии в науке и образовании

М. А. Малахов

26 сентября 2017 г.

## 1 Расчет потока вектора через поверхность в пакете Matlab

Пусть в точке  $(0,0,0)$  находится заряд  $Q$ . Задача: найти поток вектора  $\mathbf{E}$  через поверхность, заданную функцией  $f(x, y, z) = z + ax^2 + by^2 - H = 0$  при  $z > 0$ .

Поток вектора задается формулой:

$$\Phi = \iint \mathbf{E} \mathbf{nd}S. \quad (1)$$

Напряженность задается формулой:

$$\mathbf{E} = \frac{Q}{r^3} \mathbf{r} \quad (2)$$

Вектор нормали можно рассчитать по формуле:

$$\mathbf{n} = \frac{\nabla f(x, y, z)}{|\nabla f(x, y, z)|} \quad (3)$$

Для начала построим поверхность, через которую будем искать поток напряженности.

```

1  %
2  clear % удалим предыдущие переменные
3  close % закроем все открытые графики
4  n=99; % число разбиений
5  global H a b % глобальные переменные видно в теле функций
6  a=1;
7  b=1;
8  H=2;
9  Q=1;
10
11 h=2/(n-1); % шаг
12 % строим сетку:
13 [x,y]=meshgrid(1:h:1,-1:h:1);
14 % строим график:
15 h1=figure(1) % номер графика
16 mesh(1:h:1,-1:h:1, H-a*x.^2-b*y.^2) % рисуем
17 title('Surface') % наводим красоту
18 xlabel('x')
19 ylabel('y')
20 zlabel('z')
21 % сохраняем
22 saveas(h1,strcat('Our_Figure', 'H=', num2str(H), '.png'),'png')

```

Чтобы вычислить градиент создадим цикл и специальную функцию в отдельном файле:

```

24
25
26 dfx=zeros(n,n); % prelocation
27 dfy=zeros(n,n); % резервируем место под массив
28 dfz=zeros(n,n); % иначе элементы массива будут разбросаны
29 ds=zeros(n,n); % в памяти компьютера и будет долго считать
30
31 z=H-a*x.^2-b*y.^2;
32
33 for ix=1:n
34     for iy=1:n
35
36         zi=z(ix,iy);
37         % func1(x(ix,ix)+h,y(iy,iy),zi)=0 т.к. это формула фигуры
38         dfx(ix,iy)=func1(x(ix,iy)+h,y(iy,iy),zi)/h; % производные
39         dfy(ix,iy)=func1(x(ix,iy),y(iy,iy)+h,zi)/h;
40         dfz(ix,iy)=func1(x(ix,ix),y(iy,iy),zi+h)/h;
41
42         dsx=sqrt(h^2+(h*dfx(ix,iy))^2);
43         dsy=sqrt(h^2+(h*dfy(ix,iy))^2);
44         ds(ix,iy)=dsx*dsy; % элемент площади
45
46     end
47 end
48
1 function [ f ] = func1( x,y,z )
2 global H a b
3 f=z+a*x.^2+b*y.^2-H;
4
5 end
6

```

теперь посчитаем нормальную компоненту вектора напряженности и тоже построим ее график:

```

51 -
52 - r=sqrt(x.^2+y.^2+z.^2); % радиус вектор
53 -
54 - %модуль градиента для вычисления единичного вектора нормали:
55 - absNablaf=sqrt(dfx.^2+dfy.^2+dfz.^2);
56 -
57 -
58 - % нормальная составляющая напряженности:
59 - En= (dfx.*x+dfy.*y+dfz.*z).*Q./r.^3;
60 - % точка в этом выражении означает поэлементное
61 - % перемножение матриц
62 -
63 - h2=figure(2) % номер графика
64 - mesh(-1:h:1,-1:h:1, En) % рисуем
65 - title('En') % наводим красоту
66 - xlabel('x')
67 - ylabel('y')
68 - zlabel('z')
69 - % сохраняем
70 - saveas(h1,strcat('En', 'H=', num2str(H), '.png'),'png')
71 -

```

Далее задание доделать самостоятельно:

- Выбрать самому форму поверхности и параметры задачи
- Задать правильно область интегрирования, чтобы учитывать только вклады где  $z > 0$ .
- Посчитать численно интеграл 1 по заданной поверхности и сравнить с аналитическим выражением

## 2 Разработка программ с использованием репозиторийев

Ни один серьезный проект невозможно разработать без двух очень важных моментов:

- Организация эффективного взаимодействия участников группы.
- Контроль версий проекта, с указанием что конкретно и когда было сделано.

Впринципе, можно справиться с этими проблемами уже известными вам способами. Можно вести часть документов в GoogleDocs, можно постоянно рассылать по электронной почте или в каком-то из мессенджеров новые версии проекта. Можно сохранять разные версии в разных папках на компьютере. Но, по мере усложнения проекта, делать

это становится все затруднительнее. Чтобы справиться была создана распределённая система управления версиями Git (произн. «гит»).

Скачать его можно здесь: <http://git-scm.com>. При установке пока просто нажимайте кнопку "далее".

Для работы с Git есть разные графические оболочки. Но мы будем работать без нее в командной строке linux/windows. На самом деле это будет не только быстрее, но и удобнее. Для многих это будет непривычно, но не так много команд, которые вам понадобятся для начала работы. Небольшой интерактивный tutorial можно пройти вот здесь: <https://try.github.io>. Русский язык работает с git bash плохо, поэтому следите, чтобы полный путь папок, где хранятся ваши файлы, не содержал русскоязычных имен папок (ваших папок пользователя это тоже касается).

Книга по Git на русском языке: [habrahabr.ru/post/150673/](http://habrahabr.ru/post/150673/)

Далее мы создаем репозиторий для своего проекта. В принципе, можно создать репозиторий на любом сервере, но это неудобно. Для репозитория существуют специальные площадки, самые известные это [github.com](https://github.com) и [bitbucket.com](https://bitbucket.com). Далее мы будем использовать bitbucket.

Регистрируемся на сайте [bitbucket.com](https://bitbucket.com) (sign up for free), запоминаем логин и пароль. На сайте создаете свой репозиторий по образцу: **Ivanov06611** (номер группы сожмите без знаков препинания и пробелов). Если вдруг у вас в группе однофамильцы, вы можете добавить к фамилии первую букву имени. Галочка This is a private repository - должна быть выбрана. Repository type - выберите Git.

После создания репозитория, необходимо клонировать его к себе на компьютер. Правой кнопкой мыши нажимаем в папке, где собираемся вести проект и выбираем пункт Git bash. Для начала работы за университетским компьютером потребуется настройка прокси сервера. Для этого вводим команду:

**git config --global http.proxy http://proxy.kpfu.ru:8080**

Чтобы импортировать репозиторий с сайта, переходим снова на страницу bitbucket (пункт меню Overview) и нажимаем на I'm starting from scratch. На экране появится список команд, которые можно копировать и выполнять по очереди. Рассмотрим их:

**git clone https://LOGIN@bitbucket.org/LOGIN/Ivanov06611.git**  
клонировать репозиторий по ссылке. Если бы у нас там лежали файлы, то они бы скопировались на компьютер.

**cd Ivanov06611** все следующие команды будут происходить в указанной папке

**echo "# My project's README" > README.md** создаем

файл README.md, текст которого состоит из строчки # My project's README, можно попробовать ввести другой текст

**git add README.md** добавляем изменения в git (пока оффлайн, только на нашем компьютере.)

**git commit -m "Initial commit"** создаем "коммит т.е. комментарий того что сделали с нашим проектом. Initial commit - это текст коммита, можно написать что угодно. *Примечание:* Кириллицей не писать. Транслит тоже плохой тон, но если вы уверены, что все участники проекта говорят по русски, это лучше чем пытаться говорить на английском.

**git push -u origin master** команда push отправляет изменения на сервер. master это имя ветки проекта (у вас пока одна, но можно если очень надо "разветвить" его, чтобы адаптировать один код для разных задач, или проверить несколько методов для решения вашей задачи). origin - общепринятое имя для первичного централизованного репозитория.

Выполнив эти команды можно зайти на сайт репозитория и посмотреть изменения. Можно увидеть, как подсвечиваются измененные строчки.

Если мы хотим добавить файлы, то копируем перемещаем их в папку проекта и используем команду:

**git add .**

Эта команда добавляет в git все что есть в папке. Ну и снова создаем коммит:

**git commit -m "Program added"**

и отправляем все на сервер:

**git push -u origin master**

А если хотим скачать изменения, который внес кто-то другой, или вы сами на другом компьютере, то используем: **git pull -u origin master**

Теперь на сайте нажимаем иконку с шестеренкой справа, сверху. Перейдите слева в Access management. В разделе Users введите mmalakhov (этой мой аккаунт на bitbucket), выберите справа Write вместо Read, нажмите Add. Так вы даете мне доступ к этому репозиторию.

### 3 Оформление результата

Результаты научной работы удобнее всего оформлять при помощи системы LaTeX. При оформлении научных статей использование этой системы становится обязательным, большинство серьезных журналов принимают статьи только в формате TeX.

Чтобы создавать документы используя LaTeX не обязательно ставить что либо на свой компьютер. Очень удобно использовать онлайн-компиляторы [overleaf.com](https://overleaf.com), [sharelatex.com](https://sharelatex.com) и прочие. Далее мы именно ими будем пользоваться.

### 4 Метод Рунге-Кутты 4 степени.

В методе Рунге-Кутты 4 степени значение переменной в каждой следующей точке задается следующим соотношением:

$$\rho(\varphi + \Delta\varphi) = \rho(\varphi) + (f(\varphi_1, \rho_1) + 2f(\varphi_2, \rho_2) + 2f(\varphi_3, \rho_3) + f(\varphi_4, \rho_4))\Delta\varphi/6. \quad (4)$$

Здесь  $f = \pm \frac{\rho^2}{M} \sqrt{2m(E - U^{\text{eff}}(\rho))}$ . Значение параметров  $\varphi_i, \rho_i$  определяется выражениями:

$$\begin{aligned} \varphi_1 &= \varphi, \rho_1 = \rho(\varphi) \\ \varphi_2 &= \varphi + \Delta\varphi/2, \rho_2 = \rho(\varphi) + f(\varphi_1, \rho_1)\Delta\varphi/2 \\ \varphi_3 &= \varphi + \Delta\varphi/2, \rho_3 = \rho(\varphi) + f(\varphi_2, \rho_2)\Delta\varphi/2 \\ \varphi_4 &= \varphi + \Delta\varphi, \rho_4 = \rho(\varphi) + f(\varphi_3, \rho_3)\Delta\varphi \end{aligned}$$

Таким образом на каждом шаге мы вычисляем значение функции несколько раз, но и сходимость такой процедуры тоже более высокая. Не требуется считать много шагов. Теперь напомним программу, которая считала бы этим методом, но при этом учитывала бы проблему знака.

## 5 Зачетное задание.

-