

28/02/2025

#1лаба

bash

1. Открываем ОС и запускаем терминал
Ставим без GUI
Открываем консоль
SSH клиент

```
ssh student@os.noc-simplex.ru -p 30022
```

Ходим по структуре каталогов

Создаём папку, к ней "подключаем диски".

Символические ссылки - ссылки на папки, можно использовать вместо адресов
имя пользователя@имя компьютера

Корень - /

В каждой папке есть . и ..

. - символическая ссылка на эту папку

.. - папка, в которой находится эта папка.

Базовый набор команд:

1. ls
[Директория_name]
[-a] - скрытые файлы
[-l] - показывает атрибуты
2. cd
3. pwd - если потерялся
4. whoami - например, для использования программой
5. mkdir - создает папку
Правила:
Пользователь с маленькой
1 буква имени и фамилия
Перенаправление потоков

```
ls // > out.txt #- поток вывода в файл  
ls // > another.txt #- поток ошибок в файл  
ls //
```

cat - просматривает файл

1. clear

```
ls /usr/bin | less #- q выход
ls /usr/bin -la > out.txt
less < out.txt
```

Сборка программы

Создать файл

touch

```
touch maxim.cpp
```

открываем файл в [neo]vim или nano

```
#include <iostream>

int main()
{
    std::cout << "Hello world!" << std::endl;
    return 0;
}
```

Ctrl+X - сохраняем

Собираем с помощью g++

mv - перемещение/переименование

```
g++ FILENAME -o maxim
[Полный путь]
./maxim
../lab1/maxim
```

Существует 3 вида объектов, которые хранятся в накопителе:

папки

файл

исполняемый файл

С точки зрения накопителя - всё это куча

У каждого файла 10 регистров

[d/-] - папка

3 группы по 3 бита

[r/-] - право чтения (и видения)

[w/-] - право записи

[x/-] - право запуска

владелец - создатель файл

shown - поменять владельца

1 группа - права владельца

2 группа - права группы

3 группа - права остальных
Рут может всё
магические числа связанные с chmod
chmod - меняет атрибуты

```
chmod 700 me # меняет численную характеристику атрибутов, отбирая у всех права кроме owner  
chmod -x me #
```

Создать программу
Stake

Установка програм,
Переключение пользователя

14/03/2025

#2лаба

Входные данные:

google -> boeing sparse matrix -> *.mtx
200000 на 200000

Метод сопряжённых градиентов (симметричные, + определённые матрицы)

Обусловленность матрицы

Плохая обусловленность - определитель близок к 0

Метод бисопряжённых градиентов

Метод сопряжённых градиентов с предобуславливателем

Метод бисопряжённых градиентов с предобуславливателем

LU разложение

Метод Холецкого

Метод Ланцоша

LDLT разложение

метод Гаусса

Как свести решение этого уравнения к экстремальной задачке?

$Ax=b$

Создать матрицу для проверки:

1. Создаём симметричную матрицу
2. Положительно определённая ($A \cdot A^T$)

$Ax=b$

$Ax-b \sim 0$

$\|Ax - b\| < \epsilon$

Если $\dim > 10000$ критерий плохо подходит

$$\frac{\|Ax - b\|}{\|b\|} < \epsilon$$

Закон Амдала

$$S_p = \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

$$S_p = \frac{t_1}{t_p}$$

Р - количество потоков	t	$S = \frac{t_1}{t_p}$	α - доля последовательных операций

График в Excel е
от 1 до 10 через 1
от 10 до 100 через 10

Задачи
у нас 4

Метод бисопряжённых градиентов с предобуславливателем

Полезная информация:

Что это?

Что такое метод бисопряжённых градиентов?

Это метод решения $Ax = b$.

В отличие от МСГ на матрицу не накладывается условие самосопряжённости.

Зачем нужен предобуславливатель?

$$Ax = b \Leftrightarrow AP^{-1}Px = b \Leftrightarrow \begin{cases} AP^{-1}y = b \\ Px = y \end{cases} \text{ или } P^{-1}(Ax - b) = 0$$

Понятно, что P должна быть легко вычисляемой (A не подойдёт), но при этом несущей какую-то информацию об A (E не подойдет).

Задача предобуславливателя - уменьшить число обусловленности κ .

$\kappa(A) = \|A\| \cdot \|A^*\| \geq 1$. $\kappa = 1 \Leftrightarrow A$ – симметричная

Сам метод:

Дана система $M^{-1}AP^{-1}x = M^{-1}b$ (где $P^{-1}x$ - это на самом деле y)

https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B1%D0%B8%D1%81%D0%BE%D0%BF%D1%80%D1%8F%D0%B6%D1%91%D0%BD%D0%BD%D1%8B%D1%85_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%BE%D0%B2#Алгоритм_для_предобусловленной_системы

Остальная информация:

Что такое предобуславливатель?

Предобуславливатель - это результат операции предобуславливания.

Предобуславливание P матрицы A - это матрица, такая что у $P^{-1}A$ число обусловленности меньше,

чем у A .

Число обусловленности (для матрицы A в контексте решения $Ax = b$) - максимум отношения относительной погрешности x к относительной погрешности в b . Т.е.

$$\kappa(A) = \max \frac{\|\Delta x\|}{\|x\|} / \frac{\|\Delta b\|}{\|b\|}$$

Т.к. $Ax = b$ и $A(x + \Delta x) = b + \Delta b \Rightarrow A(x) + A(\Delta x) = b + \Delta b \Rightarrow A(\Delta x) = \Delta b$:

$$\begin{aligned} x &= A^{-1}b, \Delta x = A^{-1}\Delta b \\ \kappa(A) &= \max \frac{\|A^{-1}\Delta b\|}{\|A^{-1}b\|} / \frac{\|\Delta b\|}{\|b\|} = \\ &= \max \frac{\|A^{-1}\Delta b\|}{\|\Delta b\|} \cdot \max \frac{\|b\|}{\|A^{-1}b\|} = \\ &= \|A^{-1}\| \cdot \max \frac{\|Ax\|}{\|x\|} = \|A^{-1}\| \cdot \|A\| \end{aligned}$$

Т.е. $\kappa(A)$ можно считать как $\|A\|\|A^{-1}\|$, и из этого следует, что $\kappa(A) \geq 1$

Думаю, что под нормой вектора мы имеем в виду евклидову норму, то норма матрицы, это, наверное, норма, порождённая евклидовой нормой. Пример того, как её считать:

$$\begin{aligned} A &= \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \\ A^* &= \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \\ AA^* &= \begin{pmatrix} 1 \cdot 1 + 2 \cdot 2 & 1 \cdot 3 + 2 \cdot 4 \\ 3 \cdot 1 + 4 \cdot 2 & 3 \cdot 3 + 4 \cdot 4 \end{pmatrix} = \begin{pmatrix} 5 & 11 \\ 11 & 25 \end{pmatrix} \\ \begin{matrix} 5-x & 11 \\ 11 & 25-x \end{matrix} &= x^2 - 30x + 125 - 121 = x^2 - 30x + 4 = \\ &= x^2 - 2 \cdot 15x + 225 - 221 = (x - 15)^2 - 221 \\ x &= \pm 15 + \sqrt{221} \\ |15 + \sqrt{221}| &> |-15 + \sqrt{221}| \Rightarrow \|AA^*\| = 15 + \sqrt{221} \Rightarrow \\ \|A\| &= \|A^*\| = \sqrt{\|AA^*\|} = \sqrt{15 + \sqrt{221}} \end{aligned}$$

Сам метод:

1. Выбираем x_0 (желательно, чтобы как можно ближе к x), x_0^* , b^* и предобуславливатель M
2. $r_0 = b - Ax_0$
3. $r_0^* = b^* - x_0^* A^*$
4. $p_0 = M^{-1}r_0$
5. $p_0^* = r_0^* M^{-1}$
6. Для $k = 0 \dots n$:

$$\alpha_k = \frac{r_k^* M^{-1} r_k}{p_k^*}$$

$$x_{k+1} = x_k + \alpha_k \cdot p_k$$

$$x_{k+1}^* = x_k^* + \overline{\alpha_k} \cdot p_k^*$$

$$r_{k+1} = r_k - \alpha_k \cdot A p_k$$

$$r_{k+1}^* = r_k^* - \overline{\alpha_k} \cdot p_k^* A^*$$

$$\beta_k = \frac{r_{k+1}^* M^{-1} r_{k+1}}{r_k^* M^{-1} r_k}$$

$$p_{k+1} = M^{-1} r_{k+1} + \beta_k \cdot p_k$$

$$p_{k+1}^* = r_{k+1}^* M^{-1} + \overline{\beta_k} \cdot p_k^*$$

$$x_{k+1} = x_k + P_k A^{-1} (b - A x_k)$$

#1лаба

Навигация (

навигировать:

`cd` - перемещает

`ls` - позволяет посмотреть, что внутри. `ls-a` показывает скрытые файлы.

`pwd` - показывает, где мы

создать файл:

`touch` - буквально создать файл

`> [имя файла]` - записывает вывод в файл вместо консоли

Обычно, всё-таки, используют текстовые редакторы, если речь идёт о текстовом файле, или специальные приложения, если о бинарнике. Например, `nano [файл]` или `vi[m] [файл]`

`mkdir` создаёт директорию

удалить файл:

`rm . rm -r`

`sudo rm -rf /*`

скопировать:

`cp`

поменять файл:

`mv` - переместить/переименовать

Honourable mention:

`cat` - выводит содержимое файла в консоль

`less` - позволяет посмотреть содержимое без мышки

что-нибудь найти:

`find`

`grep` - ищет и печатает строки, удовлетворяющие промπτу

Например, создаём исходники в редакторе

Выводим атрибуты (отличаем папку от файла, файл от программы)

Ссылки

Собираем hello world приложение

Стаке файл

Работа с системными переменными (PATH поменять)

Поменять `num_threads` как системную переменную???

Создать пользователя

Задать права пользователю

Установить программу

Запустить

Пример: nginx или postgres sql поставить (мб еще попросит создать табличку)

#2лаба

$$Ax = b$$