

Въведение

Практикум по приложение на графични програмни среди

Владимир Димитров

Катедра Силова Електроника
Факултет по Електронна Техника и Технологии

29 септември 2021 г.



- 1 За дисциплината
 - Организация на курса
 - План на семинарните упражнения
 - План на лабораторните упражнения
- 2 Въведение в Matlab
 - Работа в команден ред
 - Скриптове
 - Променливи и операции с тях
 - Функции
 - Live scripts and functions
- 3 Литература

За дисциплината

- 1 За дисциплината
 - Организация на курса
 - План на семинарните упражнения
 - План на лабораторните упражнения
- 2 Въведение в Matlab
 - Работа в команден ред
 - Скриптове
 - Променливи и операции с тях
 - Функции
 - Live scripts and functions
- 3 Литература

Цел на дисциплината

Запознаване с Matlab, Simulink и Labview за решаване на инженерни задачи.

Цел на дисциплината

Запознаване с Matlab, Simulink и Labview за решаване на инженерни задачи.

- ▶ Лабораторни упражнения в 1208 от втора седмица - 8 компютъра (Matlab 2017b, Labview 2017)
 - ▶ Ако си носите лаптопи, парола за WiFi: [power_lab1208](#)

Цел на дисциплината

Запознаване с Matlab, Simulink и Labview за решаване на инженерни задачи.

- ▶ Лабораторни упражнения в 1208 от втора седмица - 8 компютъра (Matlab 2017b, Labview 2017)
 - ▶ Ако си носите лаптопи, парола за WiFi: [power_lab1208](#)
- ▶ Семинарни упражнения - тук, или в 1208 при малко хора (as usual)

Цел на дисциплината

Запознаване с Matlab, Simulink и Labview за решаване на инженерни задачи.

- ▶ Лабораторни упражнения в 1208 от втора седмица - 8 компютъра (Matlab 2017b, Labview 2017)
 - ▶ Ако си носите лаптопи, парола за WiFi: [power_lab1208](#)
- ▶ Семинарни упражнения - тук, или в 1208 при малко хора (as usual)
- ▶ Курсова работа - на лабораторни упражнения от асистента
 - ▶ група 29 - Цвети Хранов
 - ▶ група 30 - Християн Кънчев
 - ▶ група 31,32а - Теодора Тодорова
 - ▶ групи 32б - Владимир Димитров

Информация за дисциплината

Цел на дисциплината

Запознаване с Matlab, Simulink и Labview за решаване на инженерни задачи.

- ▶ Лабораторни упражнения в 1208 от втора седмица - 8 компютъра (Matlab 2017b, Labview 2017)
 - ▶ Ако си носите лаптопи, парола за WiFi: [power_lab1208](#)
- ▶ Семинарни упражнения - тук, или в 1208 при малко хора (as usual)
- ▶ Курсова работа - на лабораторни упражнения от асистента
 - ▶ група 29 - Цвети Хранов
 - ▶ група 30 - Християн Кънчев
 - ▶ група 31,32а - Теодора Тодорова

No	ДИСЦИПЛИНА	Седмичен хорариум						Контрол				Код на дисциплините	Кредити по ЕСТК
		Л	СУ	ЛУ	Аудит. общ	Само подг.	Общо	И	ТО	КП	КР		
55	Практикум по приложение на графични програмни среди		1	2	3	3	6				1	BE55	3

За дисциплината

План на семинарните упражнения

План на семинарните упражнения

- ▶ Какво мога да науча?
 - 1 Въведение в Matlab - операции, променливи, скриптове и функции

План на семинарните упражнения

▶ Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)

План на семинарните упражнения

▶ Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR

► Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak

► Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.

► Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab

► Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab
- 7 Въведение в Labview - основни библиотеки

► Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab
- 7 Въведение в Labview - основни библиотеки
- 8 Основни алгоритми в Labview и работа с хардуер

План на семинарните упражнения

▶ Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab
- 7 Въведение в Labview - основни библиотеки
- 8 Основни алгоритми в Labview и работа с хардуер

▶ А как да го науча?

План на семинарните упражнения

▶ Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab
- 7 Въведение в Labview - основни библиотеки
- 8 Основни алгоритми в Labview и работа с хардуер

▶ А как да го науча?

- ▶ Влизане в акаунта на компютъра user и pass: `.\student`

План на семинарните упражнения

▶ Какво мога да науча?

- 1 Въведение в Matlab - операции, променливи, скриптове и функции
- 2 Интерполация и Оптимизационни задачи с Matlab (основно регресия)
- 3 Цифрови филтри с Matlab - FIR, IIR
- 4 Internet of Things (IoT) - протоколи, анализ на данни с Matlab и Thingspeak
- 5 Символно и числено решаване на уравнения, интеграли и ДУ.
- 6 Въведение в Simulink - библиотеки и интеграция с Matlab
- 7 Въведение в Labview - основни библиотеки
- 8 Основни алгоритми в Labview и работа с хардуер

▶ А как да го науча?

- ▶ Влизане в акаунта на компютъра user и pass: `.\student`
- ▶ Вероятно да внимавате, питате и да пробвате и във въкщи сами..

За дисциплината

План на лабораторните упражнения

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегране с Matlab

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели
- 9 Stateflow - съставяне на диаграми на състоянията за цифрови системи за управление

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели
- 9 Stateflow - съставяне на диаграми на състоянията за цифрови системи за управление
- 10 Моделиране на хибридни и електромобили

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели
- 9 Stateflow - съставяне на диаграми на състоянията за цифрови системи за управление
- 10 Моделиране на хибридни и електромобили
- 11 Labview - Въведение, елементарни операции

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели
- 9 Stateflow - съставяне на диаграми на състоянията за цифрови системи за управление
- 10 Моделиране на хибридни и електромобили
- 11 Labview - Въведение, елементарни операции
- 12 Елементарни алгоритми, визуализации

План на лабораторните упражнения

- 1 Въведение в Matlab - Математически операции, работа с функции
- 2 Интерполация и регресионен анализ
- 3 Оптимизационни проблеми с Matlab
- 4 Цифрови филтри и регулатори
- 5 Символно решаване на линейни ДУ и интеграли
- 6 Числено решаване на нелинейни ДУ (динамични системи) - фрактали, хаос и бифуркационни диаграми
- 7 Въведение в Simulink и интегриране с Matlab
- 8 Моделиране и симулиране на dc/dc преобразуватели
- 9 Stateflow - съставяне на диаграми на състоянията за цифрови системи за управление
- 10 Моделиране на хибридни и електромобили
- 11 Labview - Въведение, елементарни операции
- 12 Елементарни алгоритми, визуализации
- 13 DAQ устройства I - Workshop
- 14 DAQ устройства II - Workshop

Когато чуя плана...

Въведение в Matlab

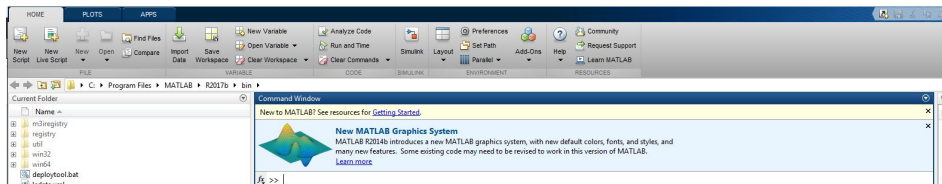
Въведение в Matlab

- 1 За дисциплината
 - Организация на курса
 - План на семинарните упражнения
 - План на лабораторните упражнения

- 2 Въведение в Matlab
 - Работа в команден ред
 - Скриптове
 - Променливи и операции с тях
 - Функции
 - Live scripts and functions

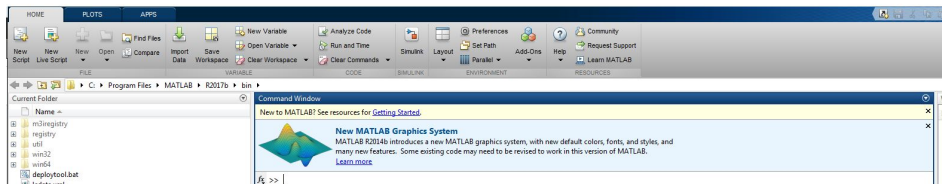
- 3 Литература

Начини за ползване



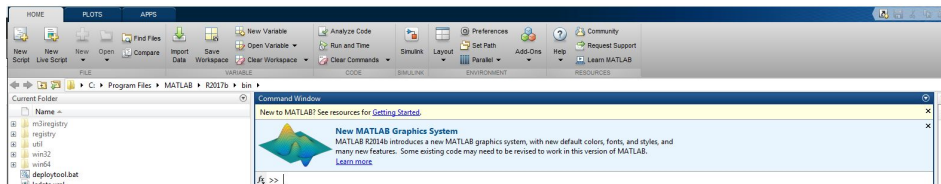
- ▶ Калкулатор - директно в командния ред ▶▶

Начини за ползване



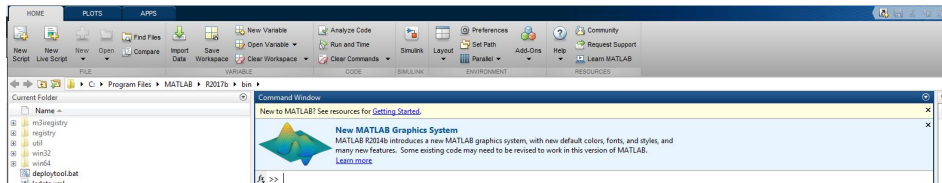
- ▶ Калкулатор - директно в командния ред »
- ▶ Писане на скриптове и функции **Script** или **New →Function**

Начини за ползване



- ▶ Калкулатор - директно в командния ред »
- ▶ Писане на скриптове и функции **Script** или **New → Function**
- ▶ Ако имате RAM и >2016а пробвайте **Live Script**

Начини за ползване



- ▶ Калкулатор - директно в командния ред »
- ▶ Писане на скриптове и функции **Script** или **New → Function**
- ▶ Ако имате RAM и >2016а пробвайте **Live Script**
- ▶ Използване на готови приложения за определени задачи - меню **Apps**

Въведение в Matlab

Работа в команден ред

Елементарни операции (за елементарни хора) в командния ред:

Пример

Да се извършат следните операции в командния ред

▶ $x = 6 + 7$

▶ $x = 6 * 7$

▶ $x = \sqrt{3}$

▶ $x = 5^6$

Въведение в Matlab

Скриптове

- ▶ New → Script или Ctrl+N

- ▶ **New** → **Script** или **Ctrl+N**
- ▶ Скриптът изпълнява записаните команди последователно

- ▶ **New** → **Script** или **Ctrl+N**
- ▶ Скриптът изпълнява записаните команди последователно
- ▶ Всички променливи в скрипта са видими и в работното пространство и могат да се модифицират допълнително извън него

- ▶ **New** → **Script** или **Ctrl+N**
- ▶ Скриптът изпълнява записаните команди последователно
- ▶ Всички променливи в скрипта са видими и в работното пространство и могат да се модифицират допълнително извън него
- ▶ Коментирание на код
 - ▶ Всичко започващо с **%**

- ▶ **New** → **Script** или **Ctrl+N**
- ▶ Скриптът изпълнява записаните команди последователно
- ▶ Всички променливи в скрипта са видими и в работното пространство и могат да се модифицират допълнително извън него
- ▶ Коментирание на код
 - ▶ Всичко започващо с **%**
 - ▶ Създаване на help file - първият непрекъснат коментар се появява като help (**F1**)

Пример 1

Да се напише скрипт за изобразяване на Wanna beer? на екрана.

Скриптове в Matlab

Пример 1

Да се напише скрипт за изобразяване на Wanna beer? на екрана.



Променливи в скриптите

- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double

Променливи в скриптите

- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете де

Променливи в скриптите

- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете до
- ▶ Наименованията са case sensitive (`i` и `I` са две различни)

Променливи в скриптите

- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете де
- ▶ Наименованията са case sensitive (`i` и `I` са две различни)
- ▶ Вградени променливи - не им присвоявайте други стойности
 - ▶ `i` и `j` - за комплексни числа
 - ▶ `pi` си е ясно
 - ▶ `ans` резултатът от последното изчисление - като калкулатор
 - ▶ `Inf` и `-Inf` плюс и минус безкрайност
 - ▶ `NaN` представя променлива, която не е число

Променливи в скриптите

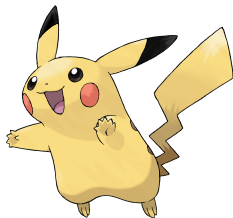
- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете де
- ▶ Наименованията са case sensitive (`i` и `I` са две различни)
- ▶ Вградени променливи - не им присвоявайте други стойности
 - ▶ `i` и `j` - за комплексни числа
 - ▶ `pi` си е ясно
 - ▶ `ans` резултатът от последното изчисление - като калкулатор
 - ▶ `Inf` и `-Inf` плюс и минус безкрайност
 - ▶ `NaN` представя променлива, която не е число
- ▶ Могат да се задават директно в скрипта `a=10`

Променливи в скриптите

- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете де
- ▶ Наименованията са case sensitive (`i` и `I` са две различни)
- ▶ Вградени променливи - не им присвоявайте други стойности
 - ▶ `i` и `j` - за комплексни числа
 - ▶ `pi` си е ясно
 - ▶ `ans` резултатът от последното изчисление - като калкулатор
 - ▶ `Inf` и `-Inf` плюс и минус безкрайност
 - ▶ `NaN` представя променлива, която не е число
- ▶ Могат да се задават директно в скрипта `a=10`
- ▶ Като функция на съществуващи променливи `b=5*a+3`

Променливи в скриптите

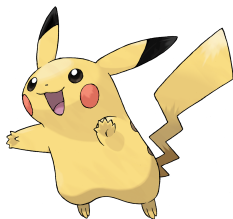
- ▶ Променливи не се инициализират, като най-често срещаните са 'a'
- 16 bit char, 3.84 - 64bit double
- ▶ Други видове: complex, symbolic, 8bit (`int8(3.84)`) и другите стандартни - ще ги видите на упражнения, стига да дойдете до
- ▶ Наименованията са case sensitive (`i` и `I` са две различни)
- ▶ Вградени променливи - не им присвоявайте други стойности
 - ▶ `i` и `j` - за комплексни числа
 - ▶ `pi` си е ясно
 - ▶ `ans` резултатът от последното изчисление - като калкулатор
 - ▶ `Inf` и `-Inf` плюс и минус безкрайност
 - ▶ `NaN` представя променлива, която не е число
- ▶ Могат да се задават директно в скрипта `a=10`
- ▶ Като функция на съществуващи променливи `b=5*a+3`
- ▶ Ако се постави `;` накрая не излизат в командния ред при старт на скрипта `b=5*a+3;`



Проблем 1

Да се изчисли колко надалеч може да се хвърли Пикачу? Известно е:

- ▶ Ние сме атлетични $\rho = 20 \left[\frac{W}{kg} \right]$, високи $th = 1.9[m]$, мускулести $tw = 80[kg]$, умни (??) и красиви (??).



Проблем 1

Да се изчисли колко надалеч може да се хвърли Пикачу? Известно е:

- ▶ Ние сме атлетични $ro = 20 \left[\frac{W}{kg} \right]$, високи $th = 1.9 [m]$, мускулисти $tw = 80 [kg]$, умни (??) и красиви (??).
- ▶ Пикачу е сфера с маса $bm = 6 [kg]$, сечение $cs = 0.1256 [m^2]$, челно съпротивление $drag = 0.3$. Пикачу ще лети над Земята $g = 9.81 \left[\frac{m}{s^2} \right]$, във въздуха с плътност $\rho = 1.1455 \left[\frac{kg}{m^3} \right]$



Проблем 1

Да се изчисли колко надалеч може да се хвърли Пикачу? Известно е още:

- ▶ Начална пикачува скорост: $v = \sqrt{\frac{3*th*tw*po}{bm + \frac{tw}{1000}}}$



Проблем 1

Да се изчисли колко надалеч може да се хвърли Пикачу? Известно е още:

- ▶ Начална пикачува скорост: $v = \sqrt{\frac{3 \cdot th \cdot tw \cdot po}{bm + \frac{tw}{1000}}}$
- ▶ Максимална скорост на развивана от Пикачу: $v_t = \sqrt{\frac{2 \cdot bm \cdot gravity}{cs \cdot \rho \cdot drag}}$

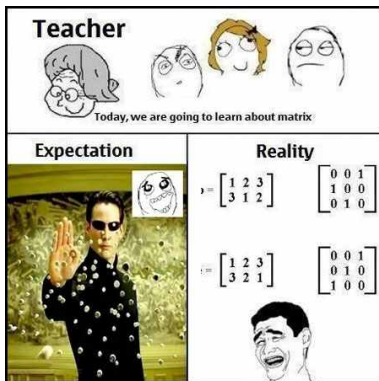


Проблем 1

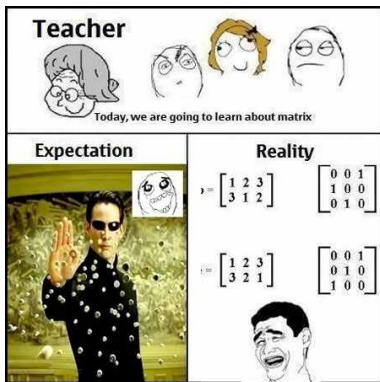
Да се изчисли колко надалеч може да се хвърли Пикачу? Известно е още:

- ▶ Начална пикачува скорост: $v = \sqrt{\frac{3 \cdot h \cdot t_w \cdot p_0}{b m + \frac{t_w}{1000}}}$
- ▶ Максимална скорост на развивана от Пикачу: $v_t = \sqrt{\frac{2 \cdot b m \cdot \text{gravity}}{c s \cdot \rho \cdot \text{drag}}}$
- ▶ Разстояние $d = \frac{v^2 \sqrt{2}}{g \sqrt{\frac{4}{5} \frac{v^4}{v_t^4} + 3 \frac{v^2}{v_t^2} + 2}}$

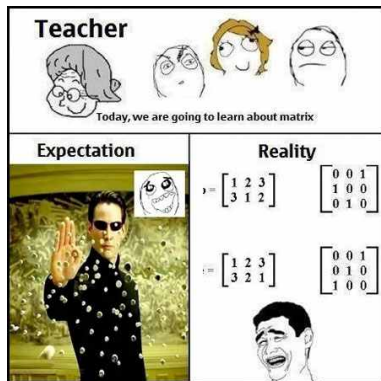
Матрици



- ▶ Матрица от числа или обекти

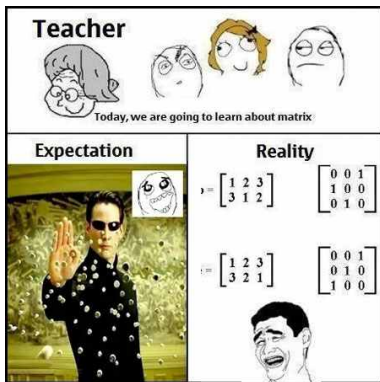


- ▶ Матрица от числа или обекти
- ▶ Вектор (ред)- разделя се със запетайки $row=[1, 2, 3]$

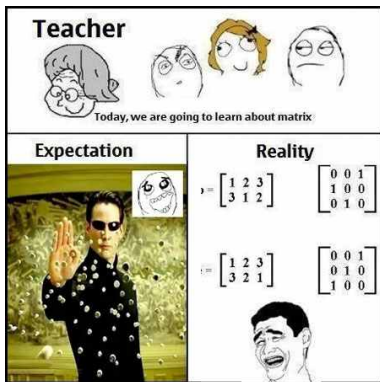


- ▶ Матрица от числа или обекти
- ▶ Вектор (ред)- разделя се със запетайки $row=[1, 2, 3]$
- ▶ Вектор (колона)- разделя се с точка и запетая $column=[1; 2; 3]$

Матрици



- ▶ Матрица от числа или обекти
- ▶ Вектор (ред)- разделя се със запетайки $row=[1, 2, 3]$
- ▶ Вектор (колона)- разделя се с точка и запетая $column=[1; 2; 3]$
 - ▶ Брой редове и колони $size(size)$



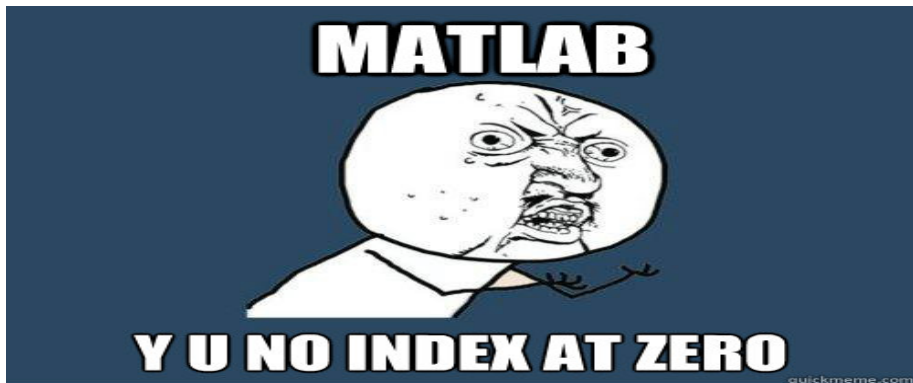
- ▶ Матрица от числа или обекти
- ▶ Вектор (ред)- разделя се със запетайки $row=[1, 2, 3]$
- ▶ Вектор (колона)- разделя се с точка и запетая $column=[1; 2; 3]$
 - ▶ Брой редове и колони $size(size)$
 - ▶ Най-голям размер $length(row)=max(size(row))$

Матрици

- ▶ Дефиниция $\text{mat}=[1, 2, 3; 1, 2, 3]$

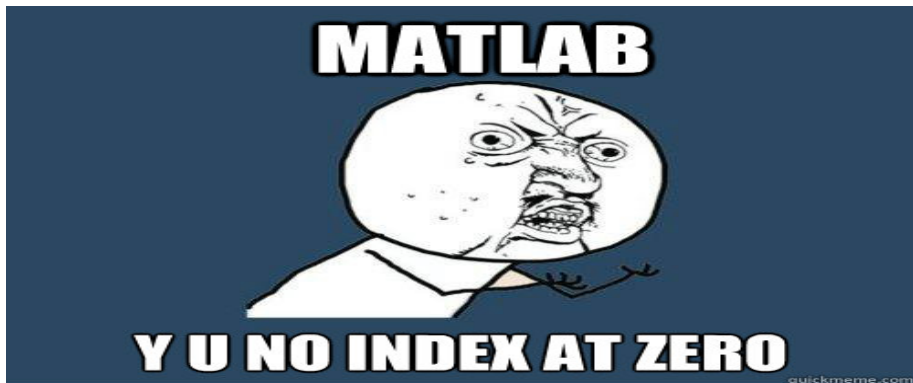
Матрици

- ▶ Дефиниция $\text{mat}=[1, 2, 3;1,2,3]$
- ▶ Първи елемент $\text{mat}(1,1)$



Матрици

- ▶ Дефиниция `mat=[1, 2 ,3;1,2,3]`
- ▶ Първи елемент `mat(1,1)`
- ▶ Може и няколко елемента - `mat(1:end-1,1:end)`
 - ▶ Колко е `size(mat)`?



Математически операции в Matlab

- ▶ Аритметичните операции в Matlab биват два вида:

Математически операции в Matlab

- ▶ Аритметичните операции в Matlab биват два вида:
 - ▶ Върху матрици - съгласно законите в Линеината Алгебра

Математически операции в Matlab

- ▶ Аритметичните операции в Matlab биват два вида:
 - ▶ Върху матрици - съгласно законите в Линеината Алгебра
 - ▶ Върху масиви (поелементно върху всеки елемент от матриците) -
Чрез добавяне на точка пред операцията

Математически операции в Matlab

- ▶ Аритметичните операции в Matlab биват два вида:
 - ▶ Върху матрици - съгласно законите в Линеината Алгебра
 - ▶ Върху масиви (поелементно върху всеки елемент от матриците) -
Чрез добавяне на точка пред операцията

Оп.	Предназначение	Масиви	Матрично	Бележки
+	Събиране	$A.+B$	$A+B$	Едно и също
-	Изваждане	$A.-B$	$A-B$	Едно и също
*	Умножение	$A.*B$	$A*B$	Динемсиите на матрицата
/	Ляво деление	$A./B$	A/B	Динемсиите на матрицата
\	Дясно деление	$A.\ B$	$A\ B$	$=B/A$
^	Степен	$A.^2$	A^2	$A^2=A*A$
'	Транспониране	$A.'$	A'	комплексно спрегната

- ▶ Транспониране - превръща редовете на матрица в колони и обратно - чрез $\text{transpose}(A)$, A' или $A.'$

- ▶ Транспониране - превръща редовете на матрица в колони и обратно - чрез $\text{transpose}(A)$, A' или $A.'$
 - ▶ За комплексна матрица A' - комплексно спрегнато транспониране

Операции с матрици

- ▶ Транспониране - превръща редовете на матрица в колони и обратно - чрез $\text{transpose}(A)$, A' или $A.'$
 - ▶ За комплексна матрица A' - комплексно спрегнато транспониране
- ▶ Сумиране и изваждане - поелементно и матрично е едно и също, но двете матрици трябва да са с еднакъв размер или да се сумира с константа

Операции с матрици

- ▶ Транспониране - превръща редовете на матрица в колони и обратно - чрез `transpose(A)`, `A'` или `A.'`
 - ▶ За комплексна матрица `A'` - комплексно спрегнато транспониране
- ▶ Сумиране и изваждане - поелементно и матрично е едно и също, но двете матрици трябва да са с еднакъв размер или да се сумира с константа
 - ▶ За поелементно събиране на вектори `s=sum(row)`
 - ▶ За поелементно умножение на вектори `s=prod(row)`

Операции с матрици

- ▶ Транспониране - превръща редовете на матрица в колони и обратно - чрез `transpose(A)`, A' или $A.'$
 - ▶ За комплексна матрица A' - комплексно спрегнато транспониране
- ▶ Сумиране и изваждане - поелементно и матрично е едно и също, но двете матрици трябва да са с еднакъв размер или да се сумира с константа
 - ▶ За поелементно събиране на вектори `s=sum(row)`
 - ▶ За поелементно умножение на вектори `s=prod(row)`
- ▶ Всички функции работят и за скалари и за матрици - например, ако $t = [1, 2, 3]$ и $f = \exp[t]$, то $f = [\exp(1), \exp(2), \exp(3)]$

Умножение, деление и вдигане на степен

- ▶ За да се приложи действието поелементно $.*$, $./$, $.^{\wedge}$, освен ако не е скалар единият от операндите

Умножение, деление и вдигане на степен

- ▶ За да се приложи действието поелементно $.*$, $./$, $.^{\wedge}$, освен ако не е скалар единият от операндите
- ▶ Умножение на матрици $*$ - вътрешните размери трябва да са еднакви, например $(3, 4) * (4, 3) = (3, 3)$

Умножение, деление и вдигане на степен

- ▶ За да се приложи действието поелементно $.*$, $./$, $.^{\wedge}$, освен ако не е скалар единият от операндите
- ▶ Умножение на матрици $*$ - вътрешните размери трябва да са еднакви, например $(3, 4) * (4, 3) = (3, 3)$
- ▶ Вдигането на степен $^{\wedge}$ - само за квадранти матрици и със скалар

Умножение, деление и вдигане на степен

- ▶ За да се приложи действието поелементно $.*$, $./$, $.^{\wedge}$, освен ако не е скалар единият от операндите
- ▶ Умножение на матрици $*$ - вътрешните размери трябва да са еднакви, например $(3, 4) * (4, 3) = (3, 3)$
- ▶ Вдигането на степен $^{\wedge}$ - само за квадранти матрици и със скалар
- ▶ Деление на матрици $/$ или \backslash - ляво и дясно деление.

Пример

Да се приложат трите действия поелементно или матрично, когато е възможно към матриците:

- ▶ $a = [1, 2, 3]$ и $b = [3, 2, 1]$
- ▶ $a = [1, 2, 3]$ и $b = [3, 2, 1]^T$
- ▶ $a = [1, 2, 3; 1, 2, 3; 1, 2, 3]$ и $b = [1, 2, 3; 1, 2, 3; 1, 2, 3]$

Автоматични инициализации на матрици

- ▶ Вектор от нули, единици или произволни числа
 - ▶ `o=ones(1,10)`
 - ▶ Вектор (ред) със 10 елемента, които са 1.

Автоматични инициализации на матрици

- ▶ Вектор от нули, единици или произволни числа
 - ▶ `o=ones(1,10)`
 - ▶ Вектор (ред) със 10 елемента, които са 1.
 - ▶ `z=zeros(23,1)`
 - ▶ Вектор (колона) със 23 елемента, които са 0.

Автоматични инициализации на матрици

- ▶ Вектор от нули, единици или произволни числа
 - ▶ `o=ones(1,10)`
 - ▶ Вектор (ред) със 10 елемента, които са 1.
 - ▶ `z=zeros(23,1)`
 - ▶ Вектор (колона) със 23 елемента, които са 0.
 - ▶ `r=rand(1,45)`
 - ▶ Вектор (ред) със 45 елемента, които са с произволни стойности в диапазона $[0, 1]$

Автоматични инициализации на матрици

- ▶ Инициализация на линеен вектор (например за абцисна ос на графика)
 - ▶ Инициализация с функция `t=linspace(0,10,20)`
 - ▶ Вектор (ред) започващ от 0, стигащ до 10(включително) и имащ 20 равно разпределени стойности

Автоматични инициализации на матрици

- ▶ Инициализация на линеен вектор (например за абцисна ос на графика)
 - ▶ Инициализация с функция `t=linspace(0,10,20)`
 - ▶ Вектор (ред) започващ от 0, стигащ до 10(включително) и имащ 20 равно разпределени стойности
 - ▶ Инициализация с оператор `b=0:0.5:10`
 - ▶ Вектор (ред) започващ от 0, увеличаващ се с 0.5 и свързващ преди или на 10.
 - ▶ Може да се напише и `b=10:-0.5:0`
 - ▶ Може да се напише и `b=0:10` - по подразбиране през 1.

Автоматични инициализации на матрици

- ▶ Инициализация на линеен вектор (например за абцисна ос на графика)
 - ▶ Инициализация с функция `t=linspace(0,10,20)`
 - ▶ Вектор (ред) започващ от 0, стигащ до 10(включително) и имащ 20 равно разпределени стойности
 - ▶ Инициализация с оператор `b=0:0.5:10`
 - ▶ Вектор (ред) започващ от 0, увеличаващ се с 0.5 и свързващ преди или на 10.
 - ▶ Може да се напише и `b=10:-0.5:0`
 - ▶ Може да се напише и `b=0:10` - по подразбиране през 1.
- ▶ Инициализация на логаритмично разпределени стойности `logspace`

Автоматични инициализации на матрици

- ▶ Инициализация на линеен вектор (например за абцисна ос на графика)
 - ▶ Инициализация с функция `t=linspace(0,10,20)`
 - ▶ Вектор (ред) започващ от 0, стигащ до 10(включително) и имащ 20 равно разпределени стойности
 - ▶ Инициализация с оператор `b=0:0.5:10`
 - ▶ Вектор (ред) започващ от 0, увеличаващ се с 0.5 и свързващ преди или на 10.
 - ▶ Може да се напише и `b=10:-0.5:0`
 - ▶ Може да се напише и `b=0:10` - по подразбиране през 1.
- ▶ Инициализация на логаритмично разпределени стойности `logspace`
- ▶ Инициализация на матрица за 2D или 3D графика `meshgrid`

Въвеждане и извеждане на данни от Matlab

- ▶ Функция `save` записва променливите във файл `.mat` в текущата директория `save myFile a b, save myenv`. Второто запазва всички променливи

Въвеждане и извеждане на данни от Matlab

- ▶ Функция `save` записва променливите във файл `.mat` в текущата директория `save myFile a b, save myenv`. Второто запазва всички променливи
- ▶ Функция `load` зарежда променливите от файл `.mat` в текущата директория

Въвеждане и извеждане на данни от Matlab

- ▶ Функция `save` записва променливите във файл `.mat` в текущата директория `save myFile a b, save myenv`. Второто запазва всички променливи
- ▶ Функция `load` зарежда променливите от файл `.mat` в текущата директория
- ▶ Функция `clear` изчиства всички променливи от работния плот

Въвеждане и извеждане на данни от Matlab

- ▶ Функция `save` записва променливите във файл `.mat` в текущата директория `save myFile a b, save myenv`. Второто запазва всички променливи
- ▶ Функция `load` зарежда променливите от файл `.mat` в текущата директория
- ▶ Функция `clear` изчиства всички променливи от работния плот
- ▶ Функция `Import Data` от основния екран на програмата - на упражнения

Пример

Да се въведе в Matlab цената на енергийните източници от Historical-prices-BG.xlsx. Да се напише скрипт, който да пита за месеца и годината и да показва цената на избраното гориво

Въведение в Matlab

Функции

- ▶ Стандартни - като на другите програмни езици. Може да са на отделен файл или няколко в един файл

- ▶ Стандартни - като на другите програмни езици. Може да са на отделен файл или няколко в един файл
- ▶ Анонимни функции - използват се основно като аргументи на математически функции за решаване на интеграли, диференциални уравнения и д.р

Стандартни функции

- ▶ Създава се с *New* → *Function* на отделен файл

Стандартни функции

- ▶ Създава се с *New* → *Function* на отделен файл
- ▶ Синтаксис:

```
function [x, y, z] = funName (in1, in2)
```

```
...
```

```
end
```

Стандартни функции

- ▶ Създава се с *New* → *Function* на отделен файл
- ▶ Синтаксис:

```
function [x, y, z] = funName (in1, in2)
```

```
...
```

```
end
```

- ▶ Функцията изпълнява командите все едно са написани последователно в командния ред.

Стандартни функции

- ▶ Създава се с *New* → *Function* на отделен файл
- ▶ Синтаксис:

```
function [x, y, z] = funName (in1, in2)
```

```
...
```

```
end
```

- ▶ Функцията изпълнява командите все едно са написани последователно в командния ред.
- ▶ Всички променливи, които са вътрешни във функцията не са видими в работното поле

Стандартни функции

- ▶ Създава се с *New* → *Function* на отделен файл
- ▶ Синтаксис:

```
function [x, y, z] = funName (in1, in2)
```

```
...
```

```
end
```

- ▶ Функцията изпълнява командите все едно са написани последователно в командния ред.
- ▶ Всички променливи, които са вътрешни във функцията не са видими в работното поле
- ▶ Функцията може да има различен брой входове и изходи (включително нула)
- ▶ Функциите се изпълняват по-бързо от скриптовете

Използване на готови функции

- ▶ Вариант 1: За научаване на синтаксиса `help` и името на функцията

Използване на готови функции

- ▶ Вариант 1: За научаване на синтаксиса `help` и името на функцията
- ▶ Вариант 2: `F1` или сайта на Matlab

Пример

Да се напише скрипт използващ функция `ezplot` за изобразяване на

- ▶ $f(x) = x^2$ в различни интервали

Аргумента на `ezplot` да се подаде като отделна функция

Анонимни функции

▶ Синтаксис:

$@(arg1, arg2, \dots)operations$

Анонимни функции

- ▶ Синтаксис:

$@(arg1, arg2, \dots)operations$

- ▶ Пример: $@(x, y, z)x^2 + y^2 + z^2$

Анонимни функции

- ▶ Синтаксис:

$@(arg1, arg2, \dots)operations$

- ▶ Пример: $@(x, y, z)x^2 + y^2 + z^2$

- ▶ Могат да се задават директно като аргумент на функция:

$q = integral(@(x)x.^2, 0, 1)$

Анонимни функции

- ▶ Синтаксис:

$@(arg1, arg2, \dots)operations$

- ▶ Пример: $@(x, y, z)x^2 + y^2 + z^2$

- ▶ Могат да се задават директно като аргумент на функция:

$q = integral(@(x)x.^2, 0, 1)$

- ▶ Може да дефинирате указател към тях

$pointer = @(x)x.^2$
 $integral(pointer, 0, 2)$

Анонимни функции

- ▶ Синтаксис:

$@(arg1, arg2, \dots)operations$

- ▶ Пример: $@(x, y, z)x^2 + y^2 + z^2$

- ▶ Могат да се задават директно като аргумент на функция:

$q = integral(@(x)x.^2, 0, 1)$

- ▶ Може да дефинирате указател към тях

$pointer = @(x)x.^2$

$integral(pointer, 0, 2)$

Пример

Да се напише скрипт използващ функцията `ezplot` за изобразяване на

- ▶ $f(x) = x^2$ в различни интервали

Аргумента на `ezplot` да се подаде като анонимна функция

Въведение в Matlab

Live scripts and functions

- ▶ Ново в Matlab >2016а, изисква java, RAM и търпение

- ▶ Ново в Matlab >2016а, изисква java, RAM и търпение
- ▶ Предимства
 - ▶ Много удобно за графики, писани на задачи (зададените ви курсови например)

- ▶ Ново в Matlab >2016а, изисква java, RAM и търпение
- ▶ Предимства
 - ▶ Много удобно за графики, писани на задачи (зададените ви курсови например)
 - ▶ Модифициране в реално време на графики и уравнения

Основни понятия за Live скиптове

- ▶ Ново в Matlab >2016a, изисква java, RAM и търпение
- ▶ Предимства
 - ▶ Много удобно за графики, писани на задачи (зададените ви курсови например)
 - ▶ Модифициране в реално време на графики и уравнения
- ▶ Оформяне на официални технически документи, поддържа директно L^AT_EX

Литература

- ▶ A. Gilat.
MATLAB: An Introduction with Applications.
John Wiley & Sons, Incorporated, 2017.
- ▶ E.B. Magrab, S. Azarm, B. Balachandran, J. Duncan, K. Herold, and G. Walsh.
An Engineers Guide to MATLAB.
Pearson Education, 2011.