Notes

Notes

Notes

Notes

Boolean algebra, conditional statements, loops.

Eugeniy E. Mikhailov



Lecture 03

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03

Variable of boolean type can have only two values

- true
- false

□ > < @ > < 注 > 〈注 > 注 > 注 → 久 (> Lecture 03 2 / 19

Lecture 03

Eugeniy Mikhailov (W&M) Boolean algebra

Variable of boolean type can have only two values

• true (Matlab use 1 to indicate it, actually everything but zero)

Practical Computing

false

Eugeniy Mikhailov (W&M) Boolean algebra

Variable of boolean type can have only two values

• true (Matlab use 1 to indicate it, actually everything but zero)

Practical Computing

• false (Matlab uses 0)

Eugeniy Mikhailov (W&M)

Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

Notes

Notes

Eugeniy Mikhailov (W&M) Boolean algebra

Variable of boolean type can have only two values

• true (Matlab use 1 to indicate it, actually everything but zero)

Prostinal Computin

Lecture 03

Lecture 03

Lecture 03

Lecture 03

• false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

• ¬ - logic **not**, Matlab

¬true = false

¬false = true

Eugeniy Mikhailov (W&M) Boolean algebra

Variable of boolean type can have only two values

• true (Matlab use 1 to indicate it, actually everything but zero)

Practical Computing

• false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

 $\bullet \ \neg$ - logic not, Matlab

 \neg true = false \neg false = true

• - logic and, Matlab &

$$A \wedge B = \begin{cases} \text{true, if } A = \text{true and } B = \text{true,} \\ \text{false, otherwise} \end{cases}$$

Practical Computing

Eugeniy Mikhailov (W&M) Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

 $\bullet \ \neg$ - logic not, Matlab

¬true = false

 $\neg false = true$

• <- logic and, Matlab &</p>

 $A \wedge B = \begin{cases} \text{true, if } A = \text{true and } B = \text{true,} \\ \text{false, otherwise} \end{cases}$

• V - logic or, Matlab

 $A \lor B = \begin{cases} \text{false, if } A = \text{false and } B = \text{false,} \\ \text{true, otherwise} \end{cases}$

Notes

Boolean operators precedence in Matlab

If A = false, B = true, C = true

Eugeniy Mikhailov (W&M)

Eugeniv Mikhailov (W&M)

If A = false, B = true, C = true

If A = false, B = true, C = true

A|∼*B*&*C*

Practical Computing

 $A|\sim B\&C$

Practical Computing

A|∼*B*&*C*

 $A|\left(({\sim}B)\&C\right)$

Boolean operators precedence in Matlab

 \sim has highest precedence, then &, and then

Boolean operators precedence in Matlab

 \sim has highest precedence, then &, and then

Lecture 03

Lecture 03

Lecture 03

Notes

Notes

Notes

Notes

Boolean operators precedence in Matlab

If A = false, B = true, C = true

Eugeniy Mikhailov (W&M)

A|∼*B*&*C*

Practical Computing

 \sim has highest precedence, then &, and then |

 $A|((\sim B)\&C)$

Thus

 $A|{\sim}B\&C=false$

Boolean operators precedence in Matlab

If A = false, B = true, C = true

 $A|\sim B\&C$

 \sim has highest precedence, then &, and then |

A|((∼*B*)&*C*)

Thus

 $A|{\sim}B\&C=false$

"Cat is an animal and cat is not an animal"

		() <) <) <) <) <) <) <) <) <)	nac
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	3 / 19
Boolean operators	s precedence in I	Matlab	
If $A = false$, $B = true$,	C = true		
	A ~B&C		
\sim has highest precede	nce, then &, and then		
	$A ((\sim B)\&C)$		
Thus	$A {\sim}B\&C=\mathit{false}$		

"Cat is an animal and cat is not an animal" is false statement

Boolean operators precedence in Matlab

If A = false, B = true, C = true

Eugeniv Mikhailov (W&M)

A|∼*B*&*C*

Practical Computing

Lecture 03

Lecture 03

Lecture 03

 \sim has highest precedence, then &, and then

 $A|((\sim B)\&C)$

Thus

 $A|{\sim}B\&C=false$

"Cat is an animal and cat is not an animal" is false statement

 $\sim Z\&Z =$

Boolean operators precedence in Matlab

If A = false, B = true, C = true

Eugeniv Mikhailov (W&M)

 $A|\sim B\&C$

Practical Computing

 \sim has highest precedence, then &, and then |

A|((∼*B*)&*C*)

Thus

 $A|\sim B\&C = false$

"Cat is an animal and cat is not an animal" is false statement

 $\sim Z\&Z = false$

Notes

Notes

Notes

Boolean logic examples

Notes

Notes

Notes

There is an island, which is populated by two kind of people: liars and truthlovers.

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

		(a) (a) (b) (a) (b) (a) (b) (b) (b) (b) (b) (b) (b) (b) (b) (b	10
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture	e 03
Boolean logic exa	mples		

There is an island, which is populated by two kind of people: liars and truthlovers.

• Liars always lie and never speak a word of truth.

• Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Boolean logic examples

Eugeniv Mikhailov (W&M

There is an island, which is populated by two kind of people: liars and truthlovers.

ractical Computing

Lecture 03

Lecture 03

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Now you see a person who answers to your question. "I am a liar." Is it possible?

Boolean logic examples

ugeniv Mikhailov (W&M

Notes

There is an island, which is populated by two kind of people: liars and truthlovers.

Practical Computing

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Now you see a person who answers to your question. "I am a liar." Is it possible?

• This makes a paradox and should not ever happen on this island.

• 123.3 & 12=

Notes

Notes

$\Box \mapsto \neg \Box \to \neg \equiv \flat$ Eugeniy Mikhailov (W&M) Lecture 03 5 / 19 Practical Computing Matlab boolean logic examples • 123.3 & 12=**1** • ~ 1232e-6 =

Lecture 03 Eugeniy Mikhailov (W&M) Practical Computing 5/19 Matlab boolean logic examples

Practical Computing

• 123.3 & 12=**1** • ~ 1232e-6 = **0**

Notes

Matlab boolean logic examples

• 123.3 & 12=**1**

Eugeniy Mikhailov (W&M)

• ~ 1232e-6 = 0

>> B=[1.22312, 0; 34.343, 12] в = 1.2231 0 34.3430 12.0000

Notes

 (a) Lecture 03 5 / 19 Practical Computing

Eugeniy Mikhailov (W&M)

Lecture 03

Matlab boolean logic examples

•	123.	2	c	12-	1
•	123.	5	à	12=	

• ~ 1232e-6 = 0

>> B=[1.22312, 0; 34.343, 12] B = 1.2231 0 34.3430 12.0000

~B

Eugeniy Mikhailo		Duration	0	$\leftarrow \Box \mapsto \leftarrow \Box P \mapsto \leftarrow \Xi$	► < E > E	୬ ଏ.୧୦ 5 / 19
					Lecture 03	5719
Matlab bo	olean lo	gic exam	ipies			
• 123.3	& 12= 1					
• ~ 123	32e-6 = 0					
>> B=[1.2	2312, 0;	34.343,	12]			
в =						
1.2231	0					
34.3430	12.0000					
~B						
ans =						
0 1						
0 0						

Eugeniy Mikhailo	v (W&M)	Practica	I Computing	< < < < < < < < < < < < < < < < < < <	► < E > E Lecture 03	- ୬ ୯.୦ 5 / 19
Matlab bo	olean lo	gic exarr	ples			
• 123.3 • ~ 123	& 12= 1 32e-6 = 0					
>> B=[1.2 B = 1.2231 34.3430	0	34.343,	12]			
~B						
ans = 0 1 0 0						
B ~B						

Eugeniy Mikhailov (W&M)	Practica	I Computing		< ≥> < ≥> ≥ Lecture 03	- ୬ ୯.୦ 5 / 19
Matlab boolean lo					
• 123.3 & 12=1 • ~ 1232e-6 = 0)				
>> B=[1.22312, 0; B = 1.2231 0 34.3430 12.0000		12]			
~B					
ans = 0 1 0 0					
B ~B					
"To be or not to be" The answer is to be		ans = 1 1	1	127127 2	*) Q (P
Eugeniy Mikhailov (W&M)	Practica	I Computing		Lecture 03	5/19

Notes

Notes

Notes

Matlab boolean logic examples

>> B=[1.2	2312,	0;	34.343,	12]
в =				
1.2231	0			
34.3430	12.00	000		
>> A=[56,	655;	Ο,	24.4]	
A =				
56.0000	655.00	000		
0	24.400	00		

		$\leftarrow \Box \rightarrow \rightarrow \Box D \rightarrow \rightarrow \Xi \rightarrow \rightarrow \Xi \rightarrow -\Xi$	D 2 C
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	6 / 19
Matlab boolean lo	gic examples		
>> B=[1.22312, 0; B = 1.2231 0 34.3430 12.0000	, ,		
>> A=[56, 655; 0, A = 56.0000 655.0000 0 24.4000	-		

B&A

 Eugenity Mikhallov (W&M)
 Practical Computing
 Lecture 03
 6/19

 Matlab boolean logic examples
 >> B=[1.22312, 0; 34.343, 12]
 B
 =
 1.2231 0
 34.3430
 12.0000

 >> A=[56, 655; 0, 24.4]
 A
 =
 56.0000 655.0000
 0
 24.4000

Practical Computing

B&A

ans = 1 0 0 1

Eugeniy Mikhailov (W&M)

Eugeniy Mikhailov (W&M)

Matlab boolean logic examples

>> B=[1.2	22312,	0;	34.343,	12]
в =				
1.2231	0			
34.3430	12.00	000		
>> A=[56,	655;	Ο,	24.4]	
A =				
56.0000	655.00	000		
0	24.400	00		
BEA				AL~B

B&A		A ∼B	
ans =			
1	0		
0	1		
			10116

Practical Computing

Notes

Notes

Notes

Notes

Lecture 03

Lecture 03 6 / 19

6 / 19

Matlab boolean logic examples

```
>> B=[1.22312, 0; 34.343, 12]
B =
1.2231 0
34.3430 12.0000
>> A=[56, 655; 0, 24.4]
A =
56.0000 655.0000
0 24.4000
```



Comparison operators

Math	Matlab
=	== double equal sign!
\neq	~=
<	<
\leq	<=
>	>
\geq	>=

		4			৩৫
Eugeniy Mikhailov (W&M)		Practical Computing		Lecture 03	
Comparison op	erato	rs			
	Math	Matlab			
	=	== double equal sig	jn!		
	\neq	~=			
	<	<			
	≤ >	<=			
	>	>			
	\geq	>=			

x=[1,2,3,4,5]

x = 1 2 3 4 5

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 7/19 Comparison operators Math Matlab == double equal sign! = \neq $\sim =$ $\langle \langle \langle \rangle \rangle$ < <= > > >= x=[1,2,3,4,5] х = 1 2 3 4 5 x >= 3

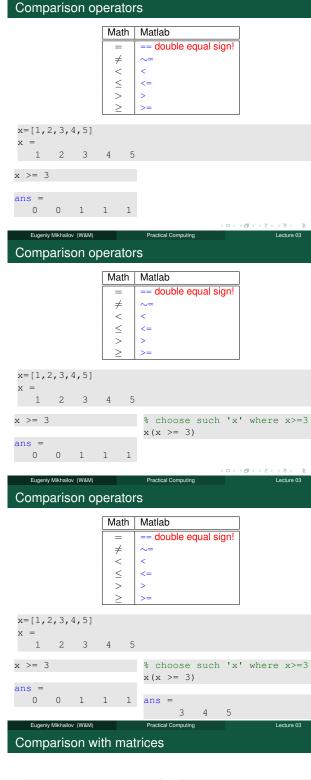
Lecture 03 7 / 19

Eugeniy Mikhailov (W&M) Practical Computing

Notes

Notes





>> A=[1,2;3,4]	>> B=[33,11;53,42]
A =	В =
1 2	33 11
3 4	53 42

Notes

Notes

7/10

Notes





>> A=[1,2;3,4]	>> B=[33,11;53,42]
A =	В =
1 2	33 11
3 4	53 42
A>=2	

		・ロ> < 個> < 回> < 回> < 回> < 回
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 8 / 19
Comparison with r	natrices	
>> A=[1,2;3,4]	>> B=[33,11;53,42]
A =	в =	
1 2	33	11
3 4	53	42

A>=2	2	
ans	=	
0	1	

0	1
1	1

Eugeniy Mikhailov (W&M)

Comparison with matrices					
>> A=[1,2;3,4]	>> B=[33,11;53,42]				
A =	в =				
1 2	33 11				
3 4	53 42				

A>=2 ans = 0 1 1 1

Eugeniy Mikhailov (W&M)

53 42 A(A>=2)

Practical Computing

Practical Computing

ø

10

Lecture 03

8/19

Lecture 03

8/19

Comparison with matrices

Eugeniy Mikhailov (W&M)

>> A=[1,2;3,4] A = 1 2 3 4		>> B= B = 33 53	[33,11;53,42] 11 42
A>=2	A(A>=2)		
ans = 0 1 1 1	ans = 3 2 4		

Notes

Notes

Notes





Comparison with matrices

>> A=[1,2;3,4]		>> B=[33,11;53,42]
A =		в =	
1 2		33	11
3 4		53	42
A>=2	A(A>=2)		B(A>=2)
ans =	ans =		Choose such elements of B where
0 1	3		elements of A>2
1 1	2		
	4		

Eugeniy Mikhailov (W&M)	Durantia	al Computing	<ロ> < 部> くき> くき> き Lecture 03	-୬୦.୦ 8/19
Comparison with			Lecture 03	8/19
>> A=[1,2;3,4] A = 1 2 3 4		в =	33,11;53,42] 11 42	
A>=2	A(A>=2)		B(A>=2)	
ans = 0 1 1 1	ans = 3 2		Choose such elements of B who elements of A≥2	əre
	4		ans = 53 11 42	
Eugeniy Mikhailov (W&M)	Practica	al Computing	< 다 > < 문 > < 문 > < 문 > Lecture 03	- ୬ ୯.୦ 8 / 19

if-else-end statement

if expressionthis part is executed only if expression is true else this part is executed only if expression is false end

if-else-end statement

Eugeniy Mikhailov (W&M)

if expression this part is executed only if expression is true else this part is executed keep working only if expression is end false end

Eugeniy Mikhailov (W&M)

if hungry buy some food else

Practical Computing

Notes

Notes

Notes

Notes

(**a**) (2) Lecture 03

Lecture 03

9/19

9/19

Notes

Notes

Notes

if expressionthis part is executedonly if expression istrueelsethis part is executedonly if expression isfalse

end

if (x>=0)
 y=sqrt(x);
else
 error('cannot do');
end

< Ø⇒

Lecture 03

Lecture 03

Lecture 03

10/19

0/10

Eugeniy Mikhailov (W&M) Practical Computing Common mistake in the 'if' statement

if (x=y)			
D=4;			
Z=45;			
C=12;			
else			
D=2;			
end			

Practical Computing

Common mistake in the 'if' statement

Eugeniv Mikhailov (W&M)

Eugeniy Mikhailov (W&M)

if ((x=y)			
D=	=4;			
Z=	=45;			
C=	=12;			
else	e			
D=	=2;			
end				

Practical Computing

the value of 'D' is always 4, except the case when y=0

Common mistake in the 'if' statement

Notes

if (x=y)
 D=4;
 Z=45;
 C=12;
else
 D=2;
end

the value of 'D' is always 4, except the case when y=0 someone used assignment operator (=) instead of comparison (==)

Short form of 'if-end' statement

if expression this part is executed only if expression is true end

Eugeniy Mikhailov (W8M) Practical Computing Lecture 03 11/19 Short form of 'if-end' statement

Practical Computing

Practical Computing

if expressionthis part is executedif won a milliononly if expression isgo partytrueendend

Short form of 'if-end' statement

if expression this part is executed only if expression is true end

Eugeniv Mikhailov (W&M)

if won a million go party end if (deviation<=0)
 exit;
end</pre>

Lecture 03

Lecture 03

The 'while' statement

Eugeniy Mikhailov (W&M)

while *expression* this part is executed while *expression* is true end

Eugeniy Mikhailov (W&M)

Notes

Notes

Notes

Notes

12/19

The 'while' statement

while expression this part is executed while hungry while expression is true

Eugeniy Mikhailov (W&M)

while expression

The 'while' statement

this part is executed while hungry

end

keep eating end

Notes

Notes

while <i>expression</i> is true end	keep eating end	<pre>c=a+b; z=c*4+5; i=i+2; end</pre>

Practical Computing

< @ →

while (i<=10)

c=a+b;

i=1;

Lecture 03 12 / 19

Eugeniy Mikhailov (W&M)	Practical Computing	(ロ)(個)(主)(主)(主)(の) Lecture 03 12/19
The 'while' state	ement	
while expression this part is executed while expression is true end	while <i>hungry</i> keep eating end	<pre>i=1; while (i<=10) c=a+b; z=c*4+5; i=i+2; end</pre>

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

		(미)(西)(관)(관)(관) 원 (오)	10
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 12 /	
The 'while' state	ment		
while <i>expression</i> is	while <i>hungry</i> keep eating end	<pre>i=1; while (i<=10) c=a+b; z=c*4+5; i=i+2; end</pre>	
while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.			

Yet another common mistake is

i=1; while (i<=10) c=a+b; end

(a)

з Lecture 03 12 / 19

Notes

The 'while' statement

i=1; while expression while (i<=10) this part is executed while hungry c=a+b; while expression is keep eating z=c*4+5; true end i=i+2; end end

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

Yet another common mistake is

i=1;	
while	(i<=10)
c=a+	b;
end	

not updating the term leading to fulfillment of the while condition Eugeniy Mikhailov (W&M) Practical Computing Lecture 03

Practical Computing

Practical Computing

The 'for' statement

Notes

Notes

for variable = expression do something end In this case variable is assigned consequently with columns of the expression, and then statements inside of the loop are executed

Eugeniv Mikhailov (W&M) The 'for' statement

for variable = *expression* do something end In this case variable is assigned consequently with columns of the expression, and then statements inside of the loop are executed



Lecture 03

Lecture 03

>> sum sum = 15

ugeniy Mikhailov (W&M) The 'for' statement

for variable = *expression* do something end

In this case variable is assigned consequently with columns of the expression, and then statements inside of the loop are executed

sum=0; x=[1,3,5,6] for v=x sum=sum+v; end

>> sum sum = 15

for loops are guaranteed to complete after predictable number of iterations (the amount of columns in expression).

Lecture 03

13/19

Notes

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

		+ □ > + ∅ > + ≥ > + ≥ > - ≥	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	14 / 19
Example			

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

S=0; i=1; while(i<=100) S=S+i; i=i+1; end

Eugeniy Mikhailov (W&M Example

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

Practical Computing

Practical Computing

Practical Computing

S=0; i=1; while(i<=100) S=S+i; i=i+1; end

Eugeniy Mikhailov (W&M)

Eugeniy Mikhailov (W&M)

S=0; for i=1:100 S=S+i; end

Example

$$S = \sum_{k=1}^{k} a_k$$

ø

While k<=100 and $a_k \ge 10^{-5}$, where $a_k = k^{-\kappa}$.

Notes

Notes

Lecture 03 14 / 19

2

14 / 19

Lecture 03

Notes

Example

 $S = \sum_{k=1}^{\infty} a_k$

While k<=100 and $a_k \ge 10^{-5}$, where $a_k = k^{-k}$. S=0; k=1; while((k<=100) & (k^-k >= 1e^-5)) S=S+k^-k; k=k+1; end

		$\rightarrow \Box \rightarrow \rightarrow \Box $	9 Q (P
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	15 / 19
Example			
	$S = \sum_{k=1}^{k} a_k$		
While k<=100 and $a_k \ge$	10^{-5} , where $a_k = k^{-1}$	^k .	
<pre>S=0; k=1; while((k<=100) & S=S+k^-k; k=k+1; end</pre>	(k^-k >= 1e-5))		
>> S S = 1.2913			

Eugeniy Mikhailov (W&M)	Practical Computing	<ロ> < () > < () > < () > < () > < () > < () > < () > < () > < () > < () < ()
Example		
	$S = \sum_{k=1}^{k} a_k$	
While k<=100 and $a_k \ge$	$\geq 10^{-5}$, where $a_k = k^{-1}$	k
<pre>S=0; k=1; while((k<=100) & S=S+k^-k; k=k+1; end</pre>	(k^-k >= 1e-5))	<pre>S=0; k=1; while(k<=100) a_k=k^-k; if (a_k < 1e-5) break; end</pre>
>> S S = 1.2913		S=S+a_k; k=k+1; end

Eugeniy Mikhailov (W&M)	Practical Computing	<ロ>(ロ)(部)(き)(き)(き) Lecture 03 15/19
Example	Tradical Comparing	
	$S = \sum_{k=1} a_k$	
While k<=100 and $a_k \ge 1$		-k.
<pre>S=0; k=1; while((k<=100) & (S=S+k^-k; k=k+1; end >> S S = 1.2913</pre>	(k^-k >= 1e-5)	<pre>S=0; k=1; while(k<=100) a_k=k^-k; if (a_k < 1e-5) break; end S=S+a_k; k=k+1; end</pre>
		>> S S = 1.2913
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 15 / 19

Notes

Notes

Notes

$$S = \sum_{k=1} a_k$$

While k<=100 and $a_k > 10^{-5}$, where $a_k = k^{-k}$

Notes

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 167 19
Same example with 'for' loop and use of matrix ops
$$S = \sum_{k=1}^{n} a_k$$

While k<=100 and $a_k \ge 10^{-5}$, where $a_k = k^{-k}$.

S=0; for k=1:100 a_k=k^-k; if (a_k < 1e-5)</pre> break; end S=S+a_k; end

Eugeniv Mikhailov (W&M)

if (a_k <</pre> break;

end

S=0;

Practical Computing Lecture 03 Same example with 'for' loop and use of matrix ops

$$\begin{split} S &= \sum_{k=1} a_k \\ \text{While } k{<}=100 \text{ and } a_k \geq 10^{-5}, \text{ where } a_k = k^{-k}. \\ \text{S=0;} \\ \text{for } k{=}1:100 \\ a_k{=}k^{-k}; \\ \text{if } (a_k < 1e{-5}) \\ \text{break;} \end{split} \qquad \begin{array}{l} \text{Often it is more elegant to use} \\ \text{built in Matlab matrix operators} \\ \text{>> } k{=}1:100; \\ \text{>> } a_k{=}k, \cdot k; \\ \text{>> } S{=sum}(a_k(a_k{>}{=}{1e{-5}})) \\ \text{or } k(a_k) = 1e{-5}) \end{array} \end{split}$$

S =

Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put one dollar in.

How much would you get at the end of the year?

• one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

Notes

Notes

16/19

Notes

Lecture 03 17 / 19

Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put one dollar in.

How much would you get at the end of the year?

• one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

interest payment every half a year

$$M_2 = 1 * (1 + x/2) * (1 + x/2) = 1 * (1 + .5/2)^2 = 1.5625$$

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 1
Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put one dollar in.

- How much would you get at the end of the year?
- one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

• interest payment every half a year

$$M_2 = 1 * (1 + x/2) * (1 + x/2) = 1 * (1 + .5/2)^2 = 1.5625$$

interest payment every month

$$M_{12} = 1 * (1 + x/12)^{12} = 1.6321$$

Interest rate related example

Eugeniv Mikhailov (W&M)

Now let's find how your return on investment (M_N) depends on the number of payments per year

```
x=.5; N_max=100; N=1:N_max;
M=0*(N); % since N is vector M will be a vector too
for i=N
    M(i)=(1+x/i)^i;
end
plot(N,M,'-'); set(gca,'FontSize',24);
xlabel('N, number of payments per year');
ylabel('M_n, return on investment'); % note M_n use
title('Return on investment vs number of payments');
```

Of course we do not need computer to show that $M_{\infty} = e^x = 1.6487$ but we need it to calculate something like $M_{1001} - M_{1000} = 2.0572 \times 10^{-7}$

Practical Computing

Interest rate related example

Eugeniy Mikhailov (W&M)

Now let's find how your return on investment (M_N) depends on the number of payments per year

x=.5; N_max=100; N=1:N_max; M=0*(N); % since N is vector M will be a vector too for i=N M(i) = (1+x/i)^i; end plot(N,M,'-'); set(gca,'FontSize',24); xlabel('N, number of payments per year'); ylabel('M_n, return on investment'); % note M_n use title('Return on investment vs number of payments'); Of course we do not need computer to show that $M_{\infty} = e^x = 1.6487$ but we need it to calculate something like $M_{1001} - M_{1000} = 2.0572 \times 10^{-7}$ Bonus question: can you calculate M without use of loops?

Notes

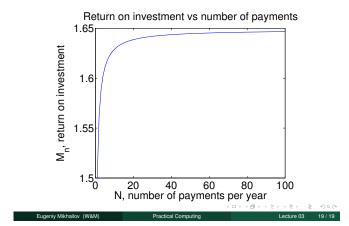
Notes

Notes

Lecture 03

Lecture 03

Interest rate related example



Notes

Notes

Notes