Notes

Notes

Boolean algebra, conditional statements, loops.

Eugeniy E. Mikhailov



Lecture 03

∎ → Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 Boolean algebra

Variable of boolean type can have only two values

- true
- false

Lecture 03

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

Boolean algebra

Eugeniy Mikhailov (W&M)

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)

Notes

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

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

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

 \bullet \wedge - 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

• ¬ - logic **not**, Matlab

¬true = false

- ¬false = true
- $\bullet~\wedge$ logic and, Matlab &

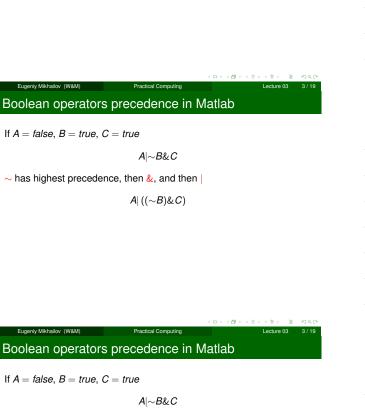
 $A \wedge B = \begin{cases} \text{true, if } A = \text{true and } B = \text{true,} \\ \text{false, otherwise} \end{cases}$

• 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

Notes



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

Boolean operators precedence in Matlab

A|∼*B*&*C*

Practical Computing

A|∼*B*&*C*

Boolean operators precedence in Matlab

 \sim has highest precedence, then &, and then

→ + Ø + + 3

Lecture 03

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

Eugeniy Mikhailov (W&M)

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

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

Thus

Eugeniy Mikhailov (W&M)

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

Practical Computing

Notes

Notes

Notes

Boolean operators precedence in Matlab

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

A|∼*B*&*C*

 \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"

		() <	A & C
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	3 / 19
Boolean operators	precedence in I	Matlab	
If $A = false, B = true, C$	C = true		
	<i>A</i> ∼ <i>B</i> & <i>C</i>		
\sim has highest preceder	nce, then &, and then	L	
	$A ((\sim B)\&C)$		
Thus			

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

"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

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

"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

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

Thus

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

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

 $\sim Z\&Z = false$

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?"

		< = > < @ > < 문 > < 문 > _ 문
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 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.

Practical Computing

Lecture 03

Lecture

- 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=

 Eugeniy Mikhalov (W&M)
 Practical Computing
 Lecture 03
 5/19

 Matlab boolean logic examples

 • 123.3 & 12=1

 • ~ 1232e-6 =

Practical Computing

Matlab boolean logic examples

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

Notes

Notes

Notes

Matlab boolean logic examples

• 123.3 & 12=**1**

Eugeniy Mikhailov (W&M)

• ~ 1232e-6 = **0**

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

Notes

Lecture 03

Matlab boolean logic examples

۰	123	. 3	&	12 =	1

• ~ 1232e-6 = 0

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

~B

				$+ \Box \mapsto + \Box \to + \Xi$		
Eugeniy Mikhailo	v (W&M)	Practical	Computing		Lecture 03	5 / 19
Matlab bo	olean logi	c exam	ples			
• 123.3 • ~ 123	& 12= 1 32e-6 = 0					
>> B=[1.2 B = 1.2231 34.3430		4.343,	12]			
~B						
ans = 0 1 0 0						

				 I □ ► I □ ► I Ξ 	> 제품 > 드 문.	9 Q (P
Eugeniy Mikhai	lov (W&M)	Practical	Computing		Lecture 03	5 / 19
Matlab bo	oolean logi	c exam	ples			
	5					
• 123.3	& 12= 1					
• ~ 12	32e-6 = 0					
>> B=[1.2 B =	22312, 0; 3	4.343,	12]			
1.2231	0					
34.3430						
D						
~B						
ans =						
0 1						
0 0						
B ~B						

Eugeniy Mikhailov (W&M)	Practical	Computing	(1)	K ≥ > ≥ Lecture 03	
Matlab boolean log					
• 123.3 & 12=1 • ~ 1232e-6 = 0					
>> B=[1.22312, 0; B = 1.2231 0 34.3430 12.0000	34.343,	12]			
~B					
ans = 0 1 0 0					
B ~B					
"To be or not to be"		ans = 1 1	1 1		
Fugeniy Mikhailov (W&M)	Practical	Computing	(a) < (b) < (b) < (c)	Lecture 03	•) ९ (? 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		

Eugenly Mikhalov (WBM) 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
>> B=[1.22312, 0; 34.343, 12] B = 1.2231 0
B = 1.2231 0
54.5450 12.0000
>> A=[56, 655; 0, 24.4] A = 56.0000 655.0000 0 24.4000

B&A

< 🗗 > Lecture 03 6 / 19 Eugeniy Mikhailov (W&M) Practical Computing 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

В

ans = 1 0 0 1

Eugeniy Mikhailov (W&M)

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		

Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	6 / 19
	-		୬୯୯
0 1			
1 0			
ans =			
B&A	A ~B		

Notes

Notes

Notes

Notes

Lecture 03

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
```

Even all Million (MOMAN)	Department Operations	1 t 00	0.14
		$(\Box \rightarrow \forall B \rightarrow \forall \Xi \rightarrow \forall \Xi \rightarrow \Box = \Xi$	e se a
0 1	0 1		
1 0	1 1		
ans =	ans =		
B&A	A ~B		

Comparison operators

x =

1 2 3 4 5

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

		$\leftarrow \Box \rightarrow$	 (1) (2) (3) (4) (4)	<	50
Eugeniy Mikhailov (W&M)		Practical Computing		Lecture 03	7/1
Comparison op	erato	rs			
	Math	Matlab]		
	=	== double equal sign!]		
	\neq	~=			
	<	<			
	< <> >	<=			
	>	>			
	\geq	>=			
			-		
x=[1,2,3,4,5]					

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 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

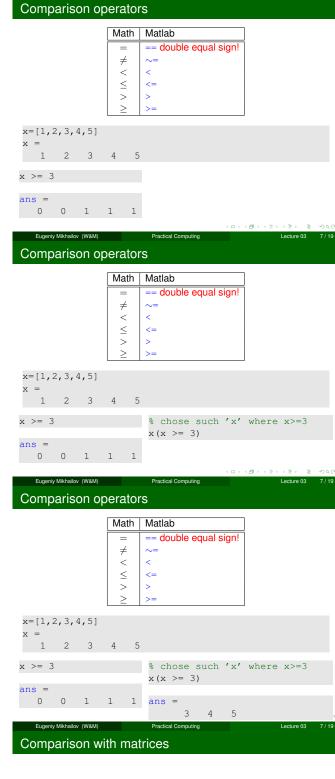
Notes

Notes

Notes

Notes

x >= 3



>> A=[1,2;3,4]	>>	B=[3	33,11;53,42]
A =		в	=	
1	2	22		11
3	4	53		42

Eugeniy Mikhailov (W&M)

Notes

Notes

Notes

Comparison with matrices

>> I	A=[1,2;3,4]	>> B=	[33,11;53,42]	
A =		в =		
1	2	22	11	
3	4	53	42	
A>=2				

	< • • •	- <問> < 言> < 言> 、言 のQで
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 8 / 19
Comparison with r	natrices	
>> A=[1,2;3,4]	>> B=[33,1	11:53.421
A =	B =	,,,
1 2	22 11	
3 4	53 42	

ans = 0 1 1 1

A>=2

		・ロト・(部)・・ミト・ミト 三臣	4
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	8
Comparison with	matrices		
>> A=[1,2;3,4]	>> B=	[33,11;53,42]	

A =			в =		
1	2		22	11	
3	4		53	42	
A>=2		A (A>=2))		
ans =					
0	1				
1	1				
0 1	-				

Comparison with matrices

Eugeniy Mikhailov (W&M)

>> A=[1,2;3,4]		>> B=[33,11;53,42]
A =		в =	
1 2		22	11
3 4		53	42
A>=2	A(A>=2)		
ans =	ans =		
0 1	3		
1 1	2		
	4		

Practical Computing

Notes

Notes

Notes

8/19

Notes

0

Lecture 03

Comparison with matrices

>> A=[1,2;3,4]		>> B=[33,11;53,42]
A =		в =	
1 2		22	11
3 4		53	42
A>=2	A(A>=2)		B(A>=2)
ans =	ans =		Chose such elements of B where
0 1	3		elements of A>2
1 1	2		
	4		

Eugeniy Mikhailov (W&M)	Practical (Computing	<ロ><部><部><き><き><き><き><き><き<の<()>
Comparison with		Somputing	
>> 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		Chose such elements of B where elements of A \geq 2
	4		ans = 53 11 42
Eugeniy Mikhailov (W&M)	Practical	Computing	<ロト < 個 > < 言 > < 言 > 、 言 > く 言 > 、 言 > の へ () Lochura 02 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

Lecture 03

з.

9 / 19

Lecture 03

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

< 🗇 >

Lecture 03

Lecture 03

Lecture 03

10/19

0/10

Eugeniy Mikhallov (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)

Eugeniy Mikhailov (W&M)

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

		$\leftarrow \Box \rightarrow \leftarrow \Box D \rightarrow \leftarrow \Xi \rightarrow \leftarrow \Xi \rightarrow - \Xi$
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03
Short form of 'if-e	nd' statement	

if expression this part is executed only if expression is true	if <i>won a million</i> go party end
end	

Short form of 'if-end' statement

if expressionthis part is executed if won a million only if expression is go party true end

Eugeniy Mikhailov (W&M)

end

Practical Computing

Practical Computing

if (deviation<=0)</pre> exit; end

Lecture 03

11/19

The 'while' statement

Eugeniy Mikhailov (W&M)

while expression this part is executed while expression is true end

Notes

Notes

Notes

Notes

Lecture 03

The 'while' statement

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

end

keep eating end

		・ロト・個ト・ミト・モー モークへの
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 12 / 19
The 'while' state	ement	
while expression this part is executed while expression is true end	while hungry keep eating end	<pre>i=1; while (i<=10) c=a+b; z=c*4+5; i=i+2; end</pre>

Eugeniy Mikhailov (W&M)	Practical Computing	<ロト < 团ト < 臣ト < 臣ト < 臣ト 三 の Q Lecture 03 12/1	
The 'while' state			
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.

Eugeniy Mikhailov (W&M)	Practical Computing	<ロ>< (四) < (四) < (回) < (O) <
The 'while' state	ement	
while expression this part is executed while expression is true end	while hungry keep eating end	<pre>i=1; while (i<=10) c=a+b; z=c*4+5; i=i+2; end</pre>
For a bit more compl	nely useful but they an licated conditional stat if the loop will finish.	e not guaranteed to finish. ement and loop it is

Practical Computing

Yet another common mistake is

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

Eugeniy Mikhailov (W&M)

Notes

Notes

Notes

The 'while' statement

while expression
this part is executed while hungry
while expression is keep eating
true end end
end
i=1;
while (i<=10)
c=a+b;
z=c*4+5;
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 Today Eugenly Mikhalov (W&M) Practical Computing Lecture 03 12/19

Practical Computing

The 'for' statement

Notes

Notes

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

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

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



Lecture 03

Lecture 03

13/19

>> sum sum = 15

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

for variable = *expression* do something end

In this case variable is assigned concequently 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*).

Practical Computing

Lecture 03

Notes

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

		$+ \Box \rightarrow + d D \rightarrow + \Xi \rightarrow + \Xi \rightarrow - \Xi$	500
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

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

Practical Computing

Practical Computing

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

Eugeniy Mikhailov (W&M)

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

ø

ð

৮ ব ≣ ৮ া≣ পি ৫.৫ Lecture 03 14 / 19

2

14 / 19

Lecture 03

Example

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

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

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 15/19

Notes

Notes

Notes

Example

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

Until 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

Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 15 /	19
Example			
	$S = \sum_{k=1}^{\infty} a_k$		
Until k<=100 and $a_k \ge$	10^{-5} , where $a_k = k^{-k}$.		
<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	<ロト < 回 > < 臣 > < 臣 > < 臣 > < 臣 > < 臣 > < 臣 > < 臣 > < 日 > 15 / 19
Example		
	$S = \sum_{k=1}^{k} a_k$	
Until k<=100 and $a_k \ge$	n=1	
<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;</pre>
>> S S = 1.2913		<pre>end S=S+a_k; k=k+1; end</pre>

Eugeniy Mikhailov (W&M)	Practical Computing	<ロ><部><2><2><2><2><2><2><2><2><2><2><2><2><2>
Example		
	$S = \sum_{k=1} a_k$	
Until 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 / 1

Notes

Notes

Notes

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

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

Notes

$+ \Box \mapsto + \partial \!\!\!/ \to + \Xi$ Lecture 03 Eugeniy Mikhailov (W&M) Practical Computing 16/19 Same example with 'for' loop and use of matrix ops $S = \sum_{k=1}^{\infty} a_k$

Until 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) break; end S=S+a_k; end

Eugeniv Mikhailov (W&M)

S=0;

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

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

Until k<=100 and $a_k \ge 10^{-5}$, where $a_k = k^{-k}$.
S=0;
Often it is more elegant to use

built in Matlab matrix operators for k=1:100 a_k=k^-k; >> k=1:100; **if** (a_k < 1e-5) >> a_k=k.^-k; break; >> S=sum(a_k(a_k>=1e-5)) end S = S=S+a_k; 1.2913 end Note >> S • use of the choose elements construct S = 1.2913 • built in sum function

Practical Computing

Interest rate related example

Eugeniy Mikhailov (W&M)

Eugeniy Mikhailov (W&M)

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

Practical Computing

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

Lecture 03

16/19

16/19

Lecture 03



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 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

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

• 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 you money growth (M_N) depends on the number of payments per year

Practical Computing

Lecture 03

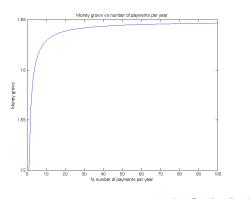
x=.5; N_max=100; N=1:N_max; M=0*(N); % since N is vector M will be a vector too for i=N M(i)=(1+x/i)^i; end plot(N,M,'-'); xlabel('N, number of payments per year'); ylabel('Money grows'); title('Money grows vs number of payments per year');

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 Lecture 03 18 / 19

Interest rate related example

Eugeniy Mikhailov (W&M)



Notes

Notes

Notes

Notes