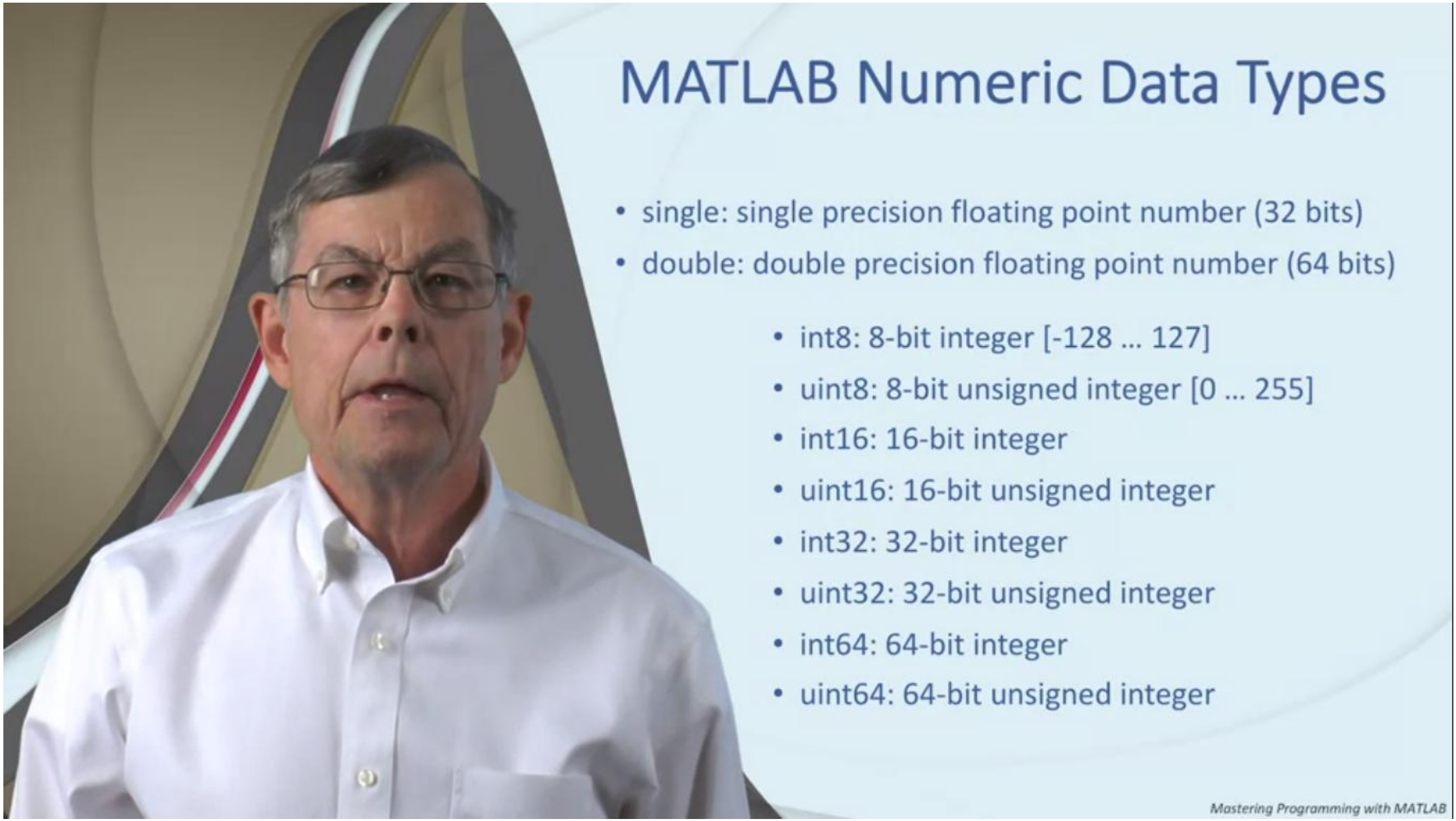

MATLAB Numeric Data Types
• single: single precision floating point number (32 bits)
• double: double precision floating point number (64 bits)
• int8: 8-bit integer [-128 … 127]
• uint8: 8-bit unsigned integer [0 … 255]
• int16: 16-bit integer
• uint16: 16-bit unsigned integer
• int32: 32-bit integer
• uint32: 32-bit unsigned integer
• int64: 64-bit integer
• uint64: 64-bit unsigned integer
Mastering Programming with MATLAB

Binary Arithmetic Operations, x op y

Shape Compatibility Rules

A.	`+ -` `.op`	`x` and `y` have the same size and shape
B.	`*`	`size(x,2)` equals `size(y,1)`
C.	`\`	`size(x,1)` equals `size(y,1)`
D.	`/`	`size(x,2)` equals `size(y,2)`
E.	`^`	`x` and `y` are square and at least one is scalar

Here we have five rules for
a binary operation,

Binary Arithmetic Operations, x op y

Shape Compatibility Rules

A.	`+ - .op`	`x` and `y` have the same size and shape or at least one is scalar
B.	`*`	`size(x,2)` equals `size(y,1)` or at least one is scalar
C.	`\`	`size(x,1)` equals `size(y,1)` or at least `x` is scalar
D.	`/`	`size(x,2)` equals `size(y,2)` or at least `y` is scalar
E.	`^`	`x` and `y` are square and at least one is scalar

The special cases are that
if one operand is a scaler,

Binary Arithmetic Operations, x op y

Shape Compatibility Rules

A.	`+ -` `.op`	`x` and `y` have the same size and shape or at least one is scalar
B.	`*`	`size(x,2)` equals `size(y,1)` or at least one is scalar
C.	`\`	`size(x,1)` equals `size(y,1)` or at least `x` is scalar
D.	`/`	`size(x,2)` equals `size(y,2)` or at least `y` is scalar
E.	`^`	`x` and `y` are square and at least one is scalar
F.	`+ -` `.op`	array and vector with equal row or column length

Binary Arithmetic Operations, x op y

Shape Compatibility Rules

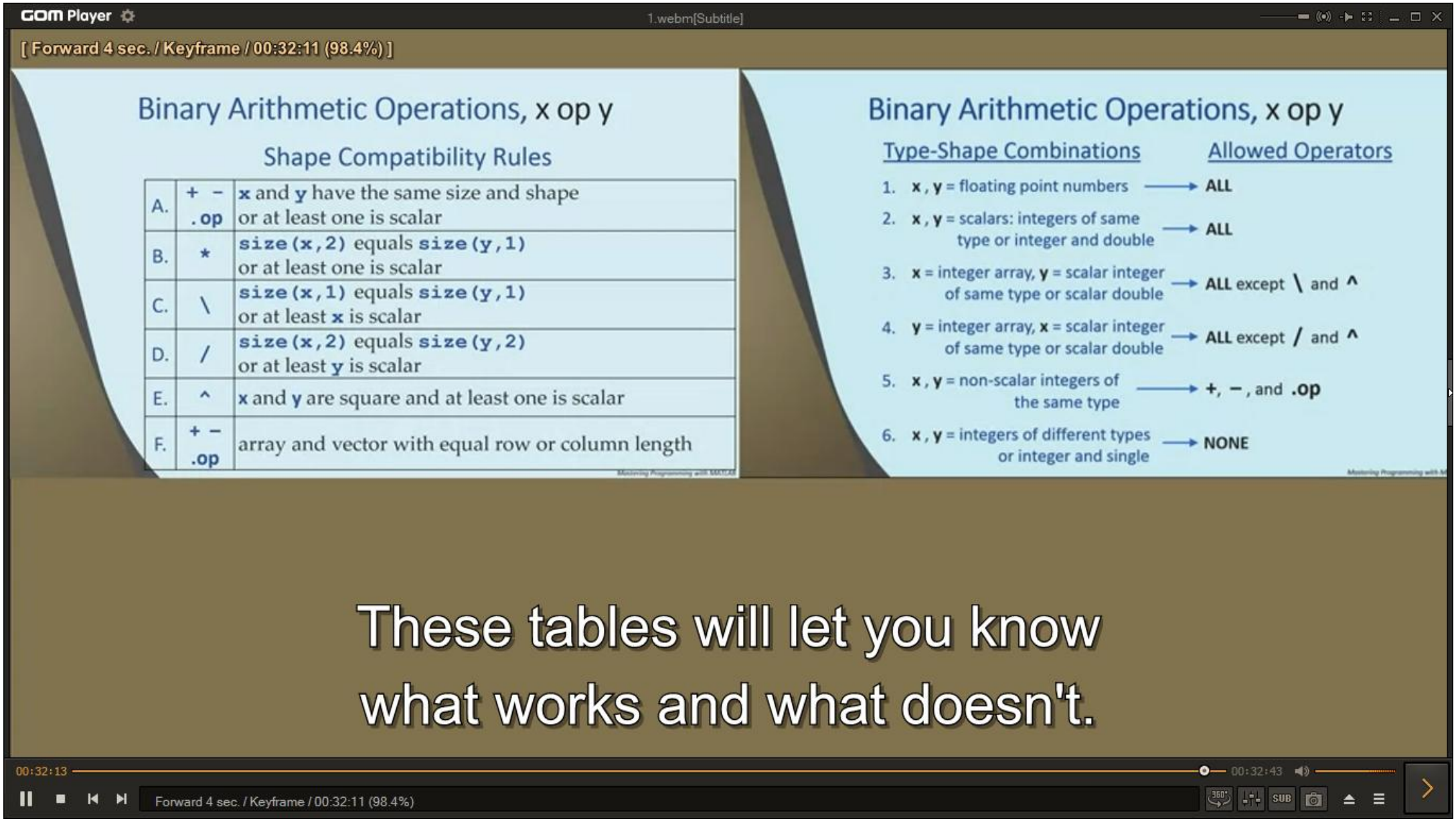
A.	`+ -` `.op`	`x` and `y` have the same size and shape or at least one is scalar
B.	`*`	`size(x,2)` equals `size(y,1)` or at least one is scalar
C.	`\`	`size(x,1)` equals `size(y,1)` or at least `x` is scalar
D.	`/`	`size(x,2)` equals `size(y,2)` or at least `y` is scalar
E.	`^`	`x` and `y` are square and at least one is scalar
F.	`+ -` `.op`	array and vector with equal row or column length

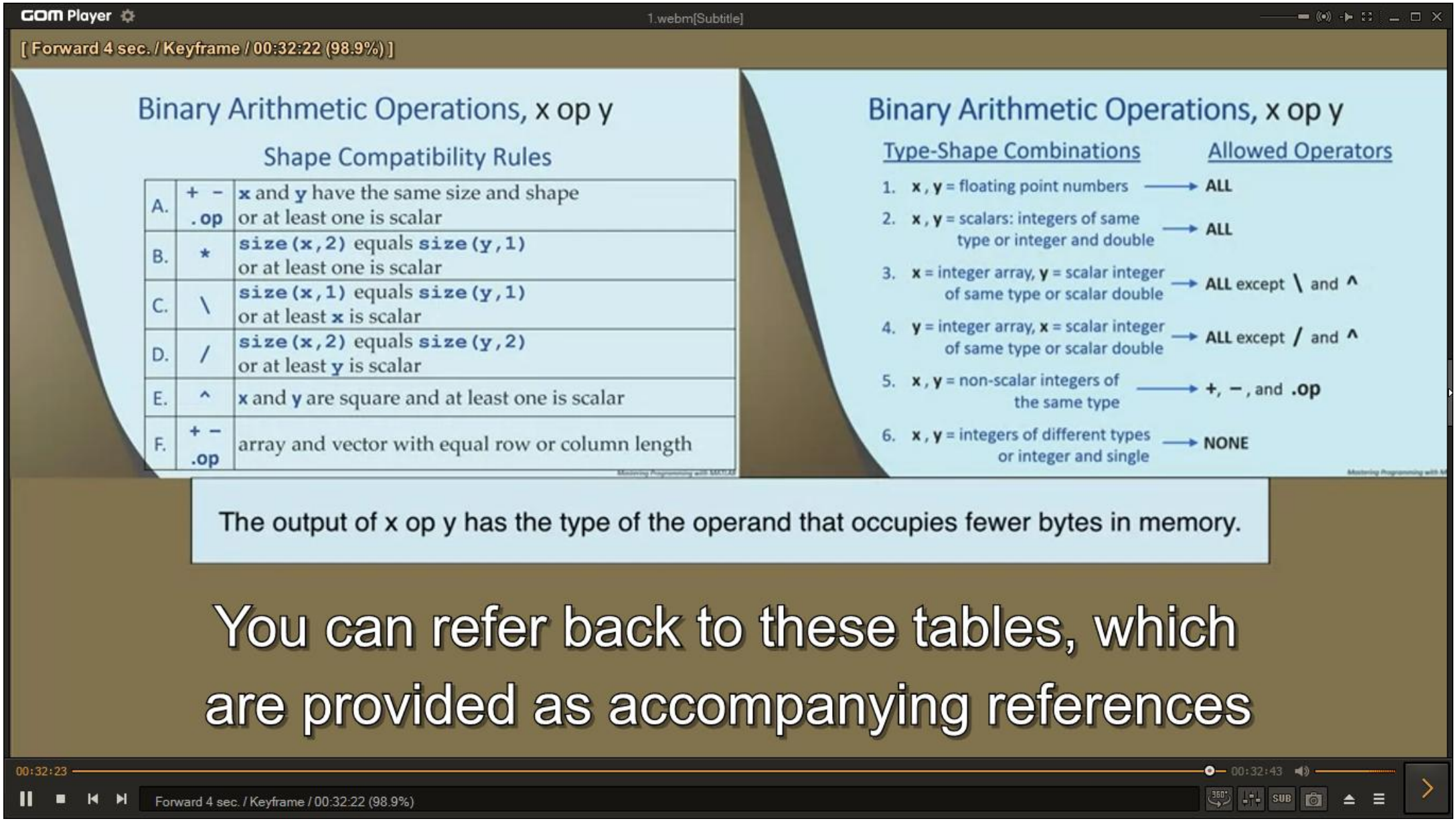
Binary Arithmetic Operations, x op y

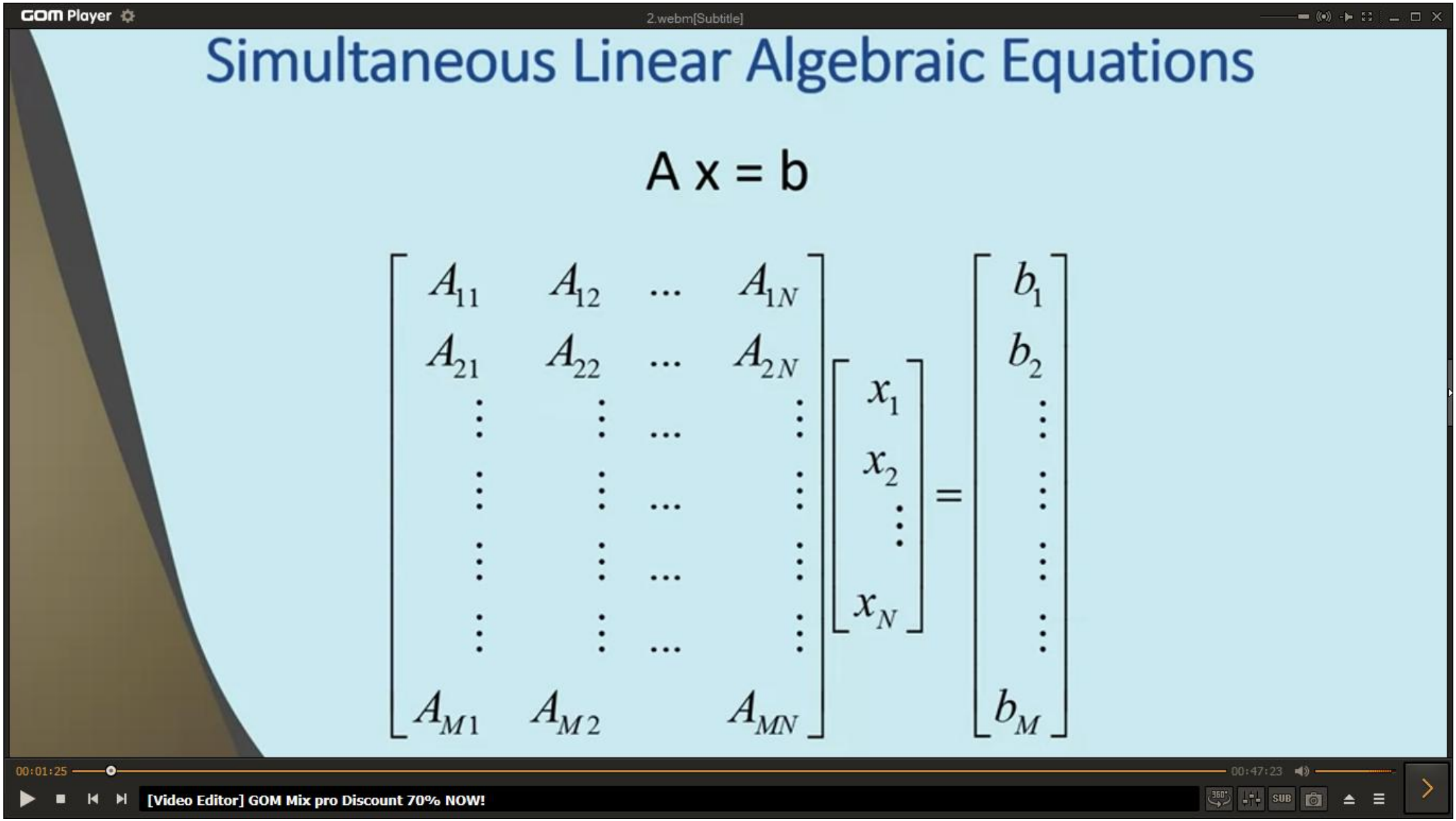
Type-Shape Combinations	Allowed Operators
1. **x** , **y** = floating point numbers	**ALL**
2. **x** , **y** = scalars: integers of same type or integer and double	**ALL**
3. **x** = integer array, **y** = scalar integer of same type or scalar double	**ALL** except **** and **^**
4. **y** = integer array, **x** = scalar integer of same type or scalar double	**ALL** except **/** and **^**
5. **x** , **y** = non-scalar integers of the same type	**+**, **−** , and **.op**
6. **x** , **y** = integers of different types or integer and single	**NONE**

Binary Arithmetic Operations, x op y

Type-Shape Combinations	Allowed Operators
least 1. **x** , **y** = floating point numbers	**ALL**
2. **x** , **y** = scalars: integers of same type or integer and double	**ALL**
3. **x** = integer array, **y** = scalar integer of same type or scalar double	**ALL** except **** and **^**
4. **y** = integer array, **x** = scalar integer of same type or scalar double	**ALL** except **/** and **^**
5. **x** , **y** = non-scalar integers of the same type	**+**, **−** , and **.op**
most 6. **x** , **y** = integers of different types or integer and single	**NONE**


GOM Player
1.webm[Subtitle]
[Forward 4 sec. / Keyframe / 00:32:11 (98.4%)]
Binary Arithmetic Operations, x op y
Shape Compatibility Rules
A. + - .op x and y have the same size and shape or at least one is scalar
B. * size(x,2) equals size(y,1) or at least one is scalar
C. \ size(x,1) equals size(y,1) or at least x is scalar
D. / size(x,2) equals size(y,2) or at least y is scalar
E. ^ x and y are square and at least one is scalar
F. + - .op array and vector with equal row or column length
Binary Arithmetic Operations, x op y
Type-Shape Combinations
Allowed Operators
1. x , y = floating point numbers → ALL
2. x , y = scalars: integers of same type or integer and double → ALL
3. x = integer array, y = scalar integer of same type or scalar double → ALL except \ and ^
4. y = integer array, x = scalar integer of same type or scalar double → ALL except / and ^
5. x , y = non-scalar integers of the same type → +, -, and .op
6. x , y = integers of different types or integer and single → NONE
These tables will let you know what works and what doesn't.
00:32:13
00:32:43
Forward 4 sec. / Keyframe / 00:32:11 (98.4%)
SUB


GOM Player
1.webm[Subtitle]
[Forward 4 sec. / Keyframe / 00:32:22 (98.9%)]
Binary Arithmetic Operations, x op y
Shape Compatibility Rules
A. + - .op x and y have the same size and shape or at least one is scalar
B. * size(x,2) equals size(y,1) or at least one is scalar
C. \ size(x,1) equals size(y,1) or at least x is scalar
D. / size(x,2) equals size(y,2) or at least y is scalar
E. ^ x and y are square and at least one is scalar
F. + - .op array and vector with equal row or column length
Binary Arithmetic Operations, x op y
Type-Shape Combinations
Allowed Operators
1. x , y = floating point numbers → ALL
2. x , y = scalars: integers of same type or integer and double → ALL
3. x = integer array, y = scalar integer of same type or scalar double → ALL except \ and ^
4. y = integer array, x = scalar integer of same type or scalar double → ALL except / and ^
5. x , y = non-scalar integers of the same type → +, −, and .op
6. x , y = integers of different types or integer and single → NONE
The output of x op y has the type of the operand that occupies fewer bytes in memory.
You can refer back to these tables, which are provided as accompanying references
00:32:23
00:32:43
Forward 4 sec. / Keyframe / 00:32:22 (98.9%)
SUB


GOM Player
2.webm[Subtitle]
Simultaneous Linear Algebraic Equations
A x = b
$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ A_{M1} & A_{M2} & & A_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ b_M \end{bmatrix}$$
00:01:25
00:47:23
[Video Editor] GOM Mix pro Discount 70% NOW!

Simultaneous Linear Algebraic Equations

$$A\,x = b$$

$$\begin{aligned}
A_{11}x_1 &+ A_{12}x_2 &+ \ldots &+ A_{1N}x_N &= b_1 \\
A_{21}x_1 &+ A_{22}x_2 &+ \ldots &+ A_{2N}x_N &= b_2 \\
\vdots & \vdots & & \vdots & \vdots \\
A_{M1}x_1 &+ A_{M2}x_2 &+ \ldots &+ A_{MN}x_N &= b_M
\end{aligned}$$

or some other permutation
of these three adjectives,

Summary of Error Handling

- Previous error handling
  - MATLAB error messages to help us debug our code
    - not informative enough
    - change from one MATLAB version to another
  - If-statements to examine input for errors
    - too many possibilities to be feasible