DLTI Convolution (1A)

Young Won Lim
3/25/19

Copyright (c) 2013 - 2018 Young W. Lim.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.
This document was produced by using LibreOffice.

Based on

Introduction to Signal Processing

S. J. Ofranidis

The necessities in DSP C Programming

Filtering C codes (A.pdf) 20190307

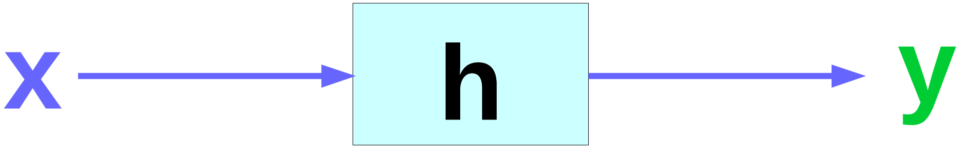
conv

```
#include <stdlib.h>       // to use max, min
/* conv.c - convolution of x[n] with h[n], resulting in y[n] */
/* h : filter array, M : filter order */
/* x : input array, L : input length */
/* y : output array  with length of L+M */

void conv(int M, double *h, int L, double *x, double *y)
{
        int n, m;

        for (n = 0; n< L+M-1; n++)
                for (y[n] = 0, m = max(0, n-L+1); m <= min(n, M-1); m++)
                        y[n] += h[m] * x[n-m];
}
```

Index Variable Constraints



x[0..L-1]
input array
L input length

h[0..M-1]
filter array,
M (filter order)

y[0..L+M-2]
output array
(M+L-1) output length

Assume
M < L

Case A

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

n $\in$ **[0, L+M-2]**
m $\in$ **[0, M-1]**
n-m $\in$ **[0, L-1]**

Case B

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$

n $\in$ **[0, L+M-2]**
m $\in$ **[0, L-1]**
n-m $\in$ **[0, M-1]**

- Flipped and shifted functions
  - Case **A**: **h[n-m]**
  - Case **B**: **x[n-m]**
- Range partitions for **n**
- Effective index ranges for **n**, **m**, **n-m**

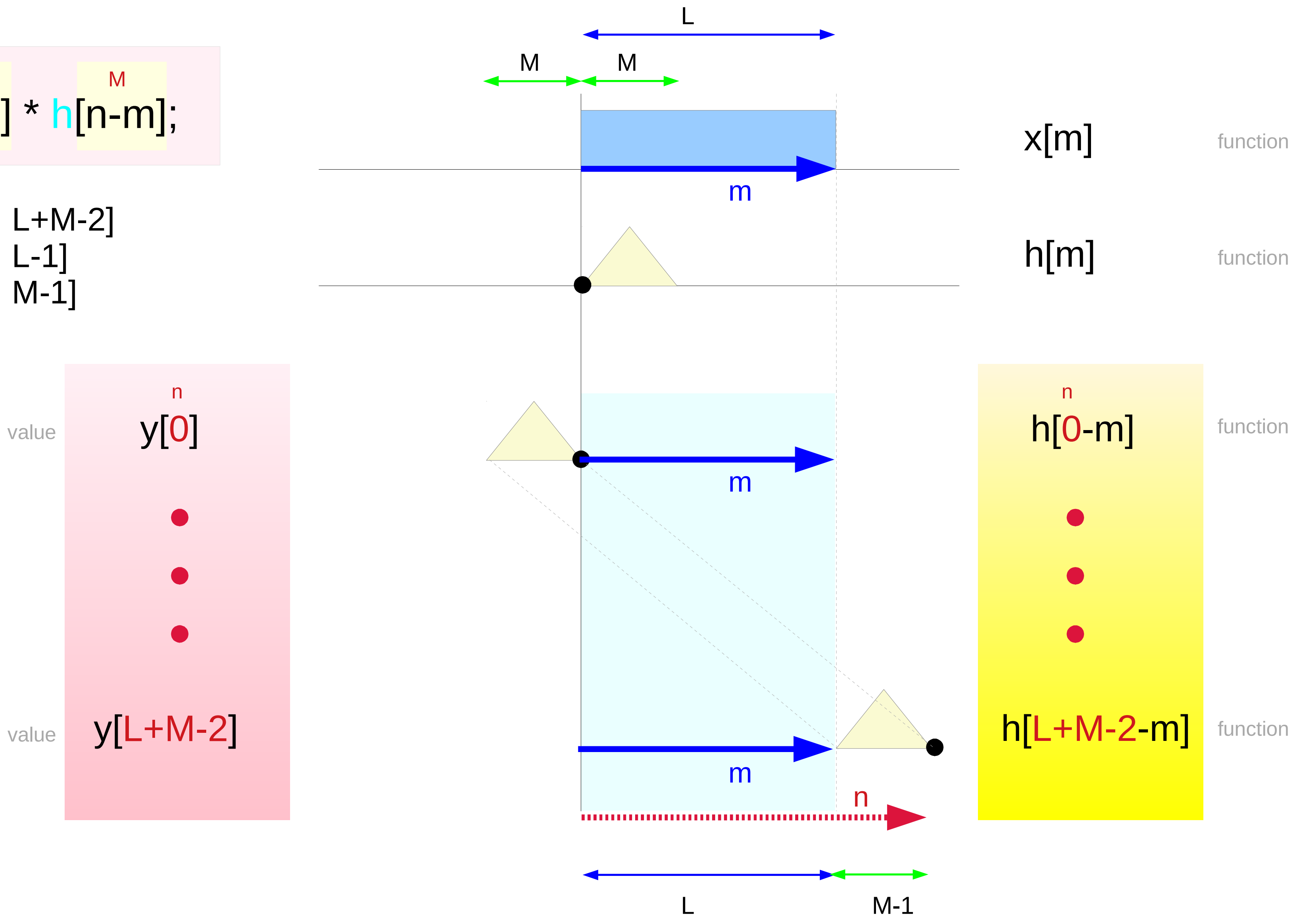
Flipped and shifted function of **h[n-m]**

Case A

M+L-1 L M

y[n] += x[m] * h[n-m];

n $\in$ [0, L+M-2]
m $\in$ [0, L-1]
n-m $\in$ [0, M-1]



L

M M

x[m] function

m

h[m] function

value n y[0]

●
●
●

value y[L+M-2]

n h[0-m] function

m

●
●
●

h[L+M-2-m] function

m

n

L M-1

Flipped and shifted function of **x[n-m]**

Case B

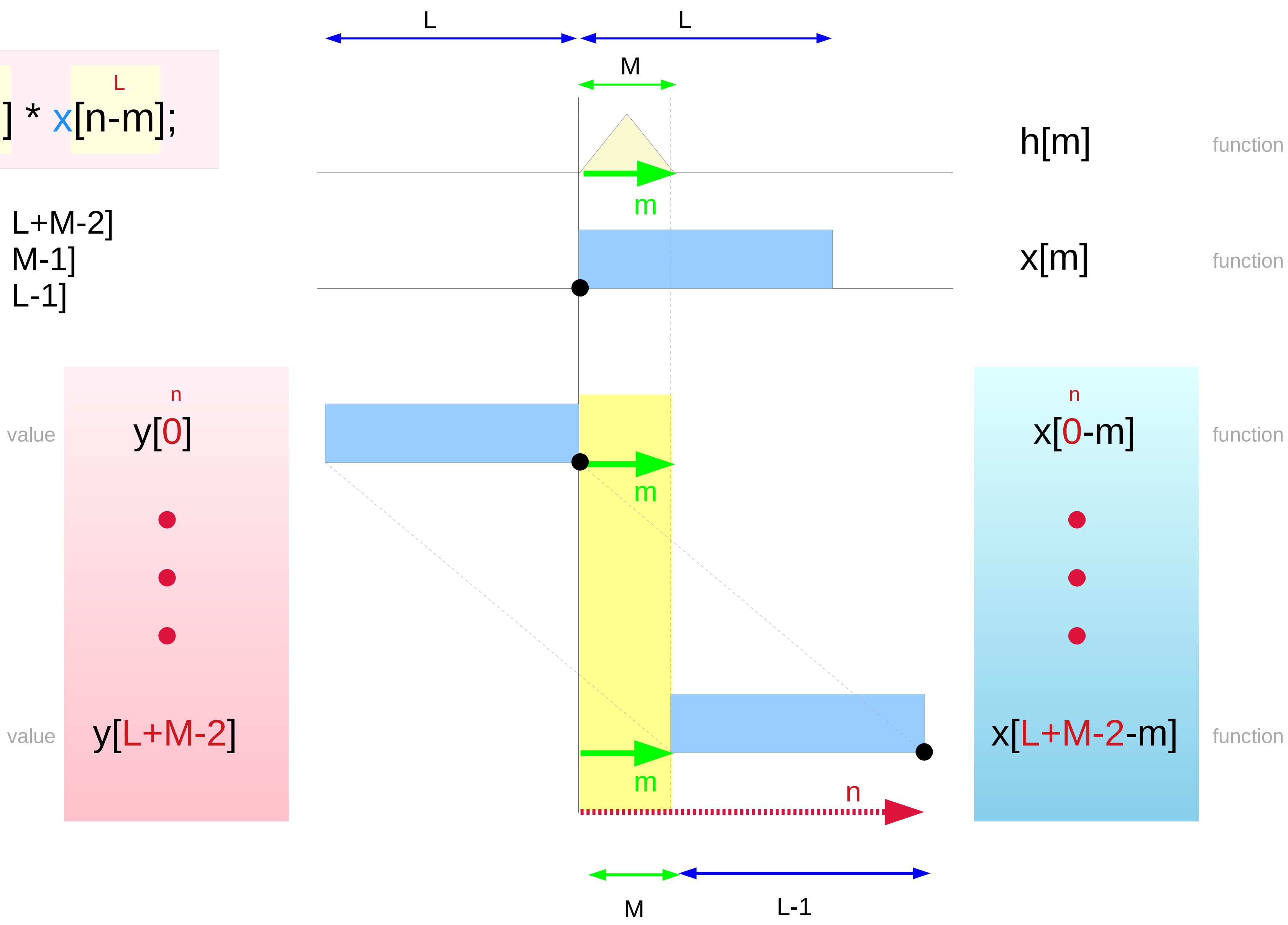
M+L-1 M L

```
y[n] += h[m] * x[n-m];
```

$n \in [0, L+M-2]$

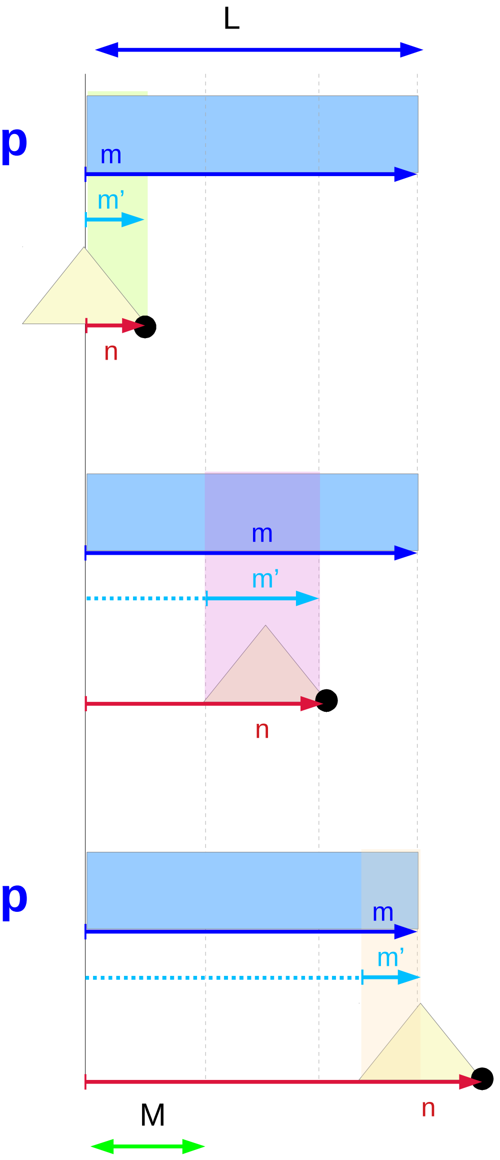
$m \in [0, M-1]$

$n-m \in [0, L-1]$



Range partitions for **n** (1)

Case A



partial overlap

x[m]

h[n-m]

$n \in [0, M-2]$

full overlap

x[m]

h[n-m]

$n \in [M-1, L-1]$

partial overlap

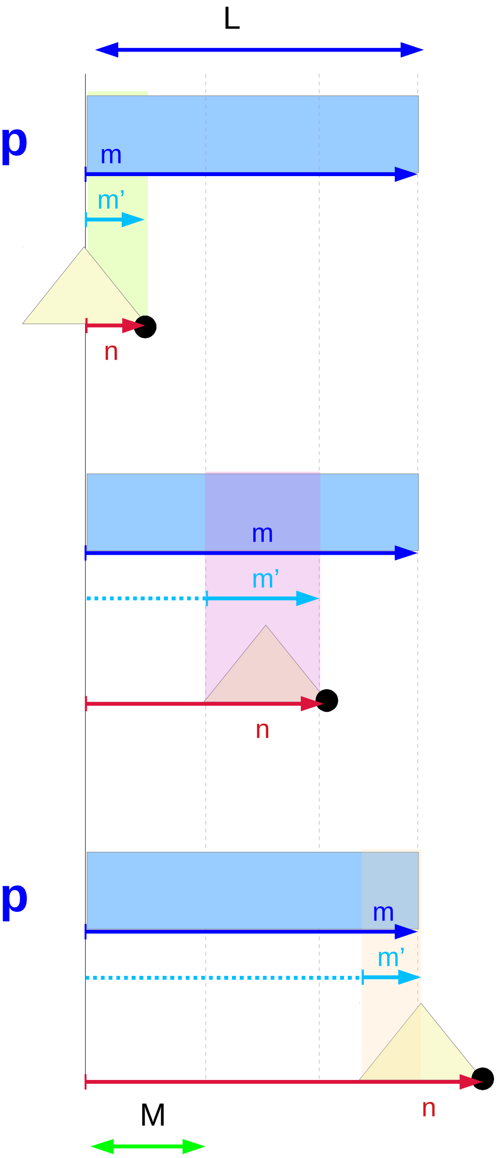
x[m]

h[n-m]

$n \in [L, L+M-2]$

Effective index for **x[m]** (2)

Case A



partial overlap

x[m]

h[n-m]

m' ∈ [0, n]

for a given **n**

n ∈ [0, M-2]

full overlap

x[m]

h[n-m]

m' ∈ [n-M+1, n]

for a given **n**

n ∈ [M-1, L-1]

partial overlap

x[m]

h[n-m]

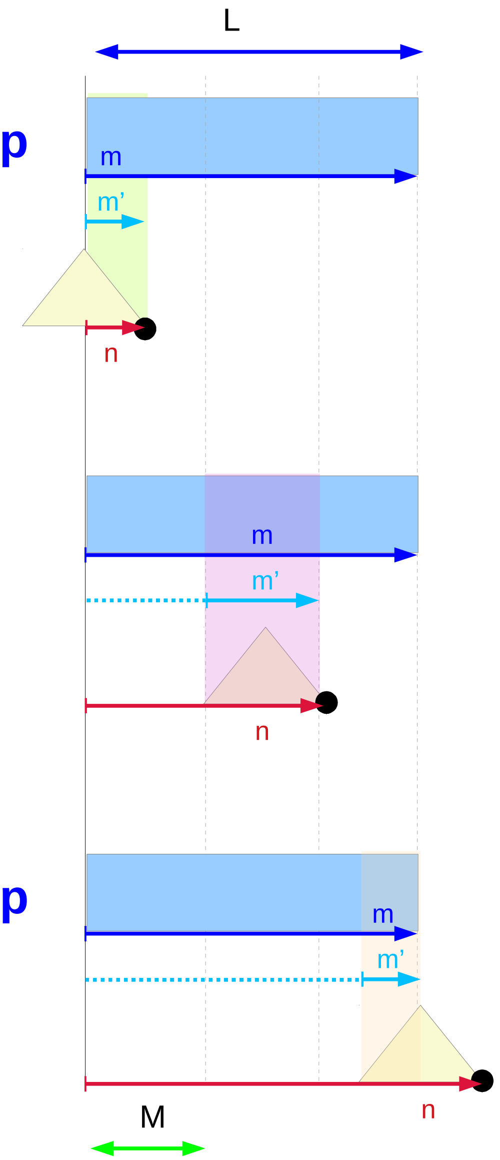
m' ∈ [n-M+1, L-1]

for a given **n**

n ∈ [L, L+M-2]

Effective index for **h[n-m]** (3)

Case A



partial overlap

x[m] $m' \in [0, n]$

h[n-m] $n-m' \in [n, 0]$

for a given n

$n \in [0, M-2]$

full overlap

x[m] $m' \in [n-M+1, n]$

h[n-m] $n-m' \in [M-1, 0]$

for a given n

$n \in [M-1, L-1]$

partial overlap

x[m] $m' \in [n-M+1, L-1]$

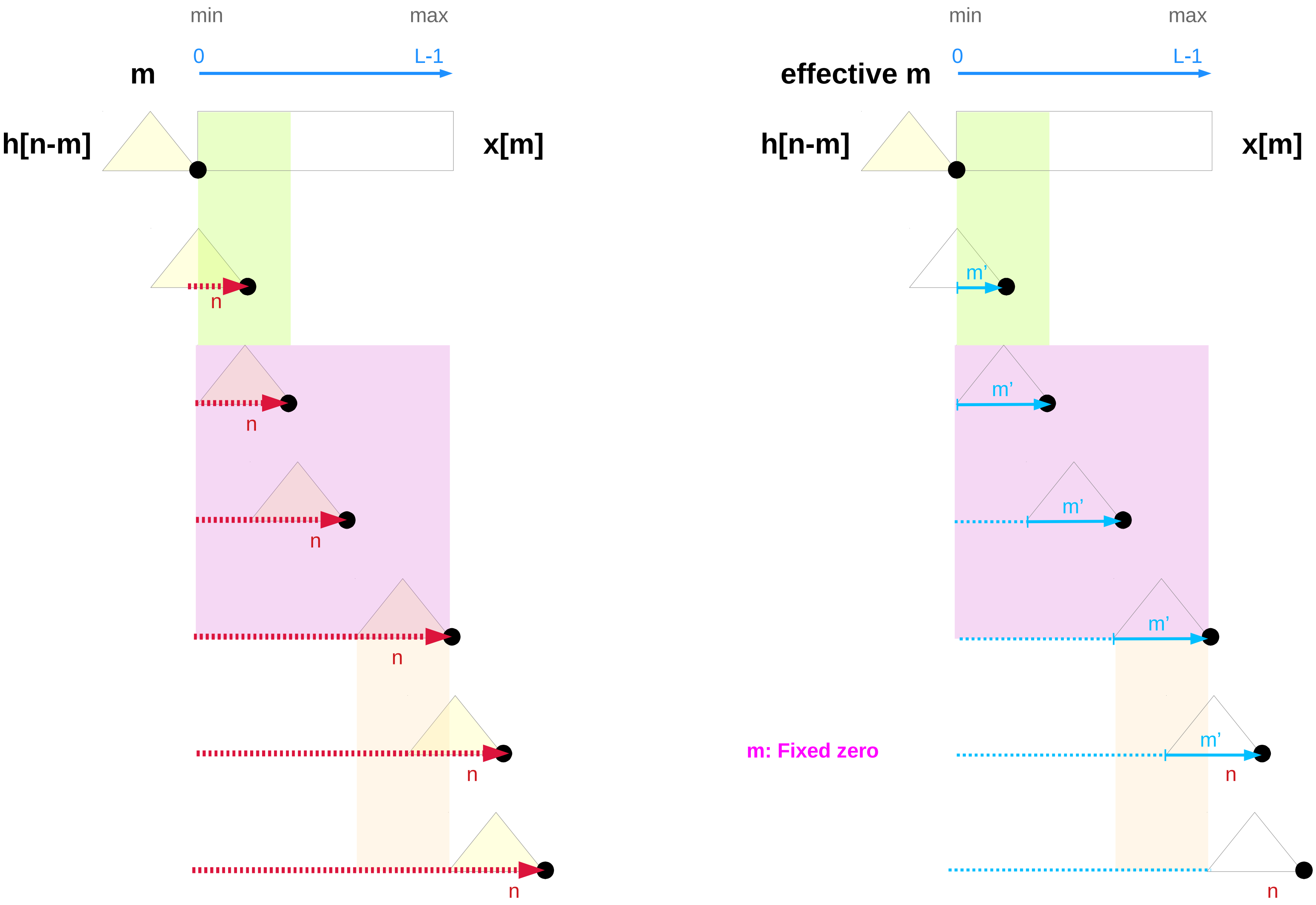
h[n-m] $n-m' \in [M-1, n-L+1]$

for a given n

$n \in [L, L+M-2]$

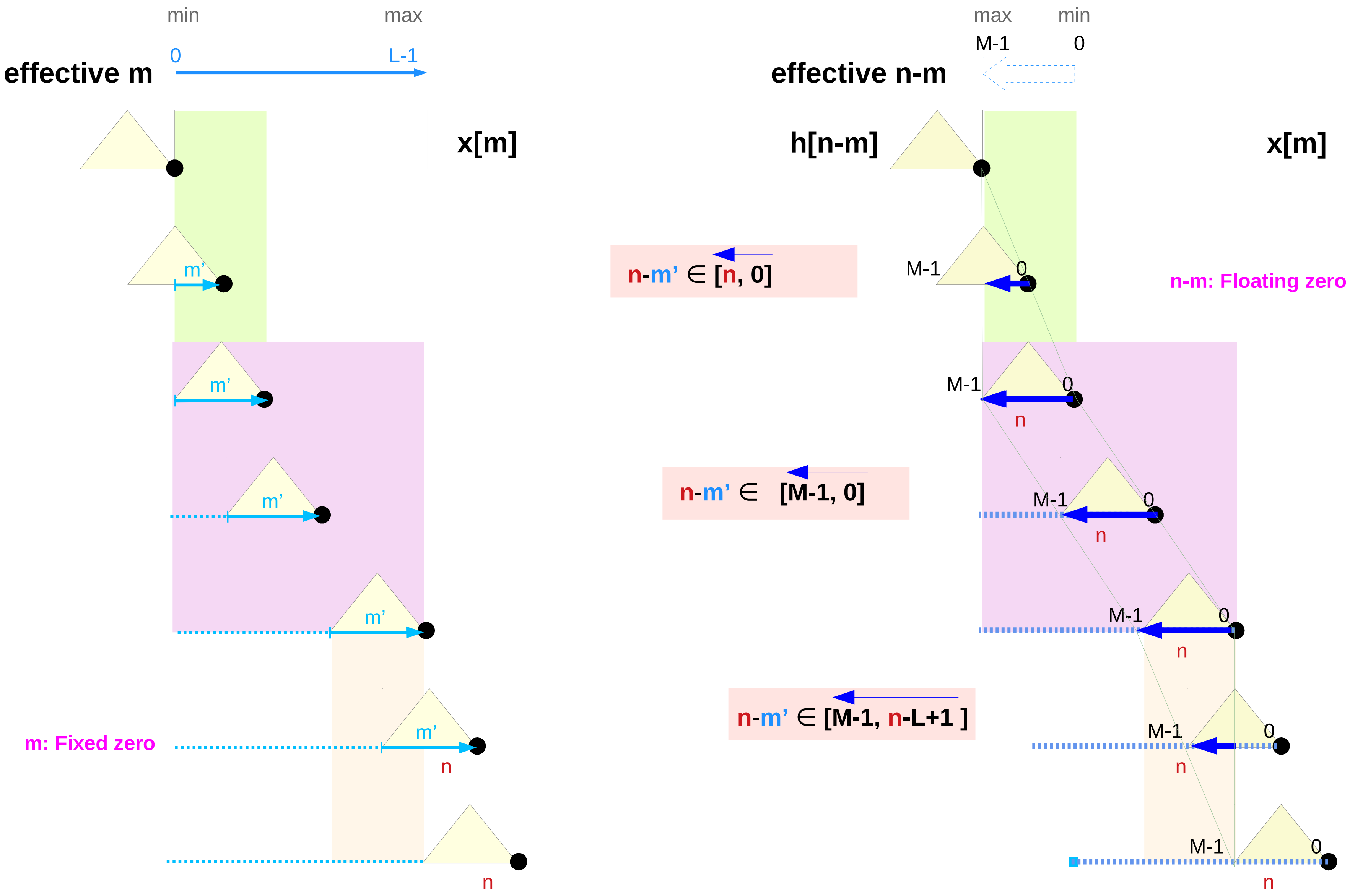
Index **n** and **m**

Case A



Index **m** and **n-m**

Case A



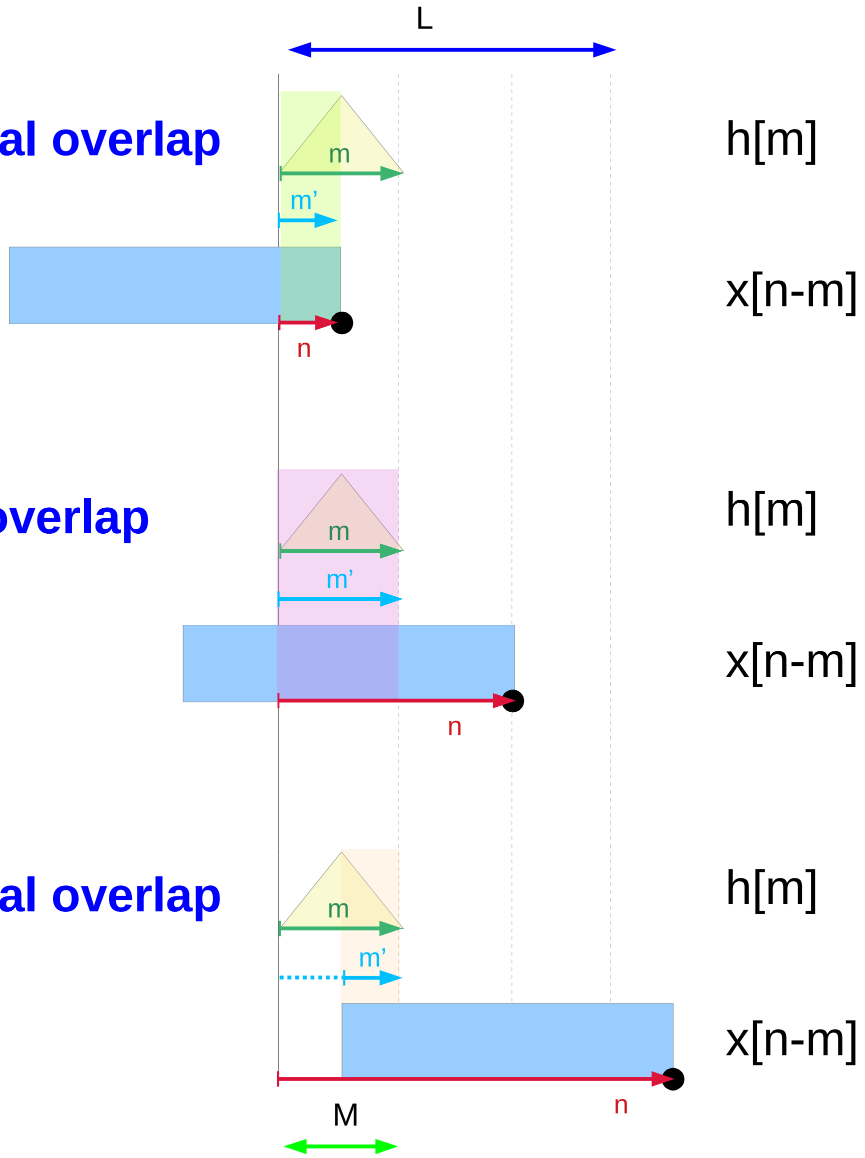
n-m' ∈ [n, 0]

n-m' ∈ [M-1, 0]

n-m' ∈ [M-1, n-L+1]

Range partitions for **n** (1)

Case B



partial overlap

$n \in [0, M-2]$

full overlap

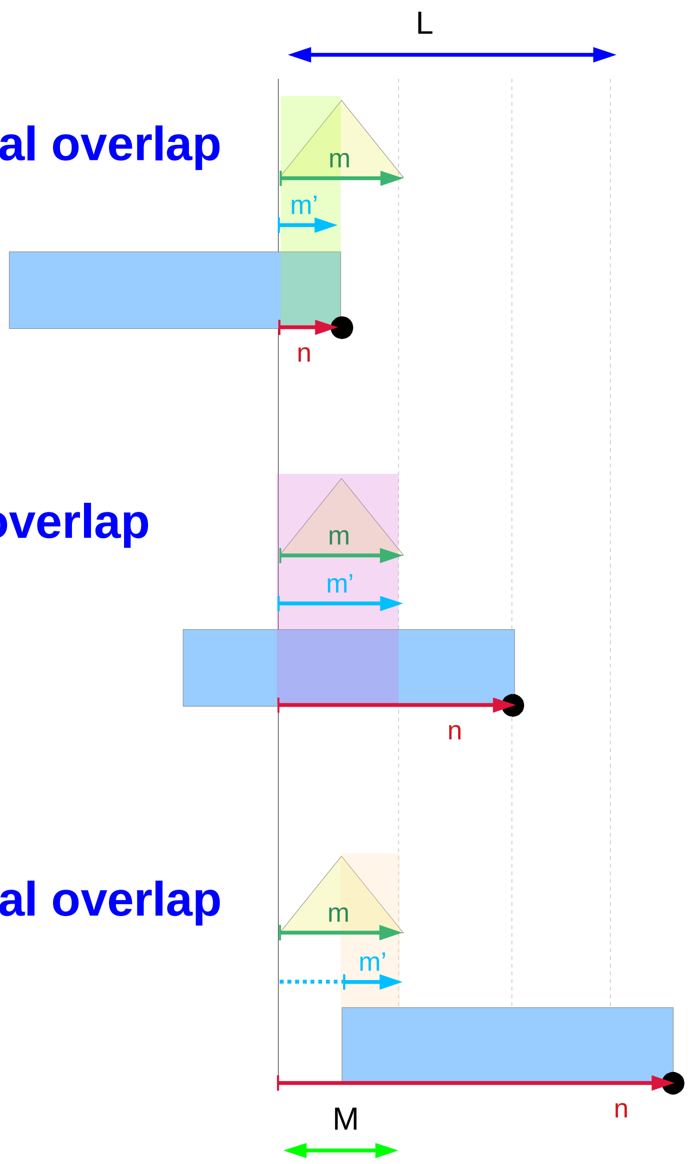
$n \in [M-1, L-1]$

partial overlap

$n \in [L, L+M-2]$

Effective index for **h[m]** (2)

Case B



partial overlap

h[m]

x[n-m]

m' ∈ [0, n]

for a given **n**

n ∈ [0, M-2]

full overlap

h[m]

x[n-m]

m' ∈ [0, M-1]

for a given **n**

n ∈ [M-1, L-1]

partial overlap

h[m]

x[n-m]

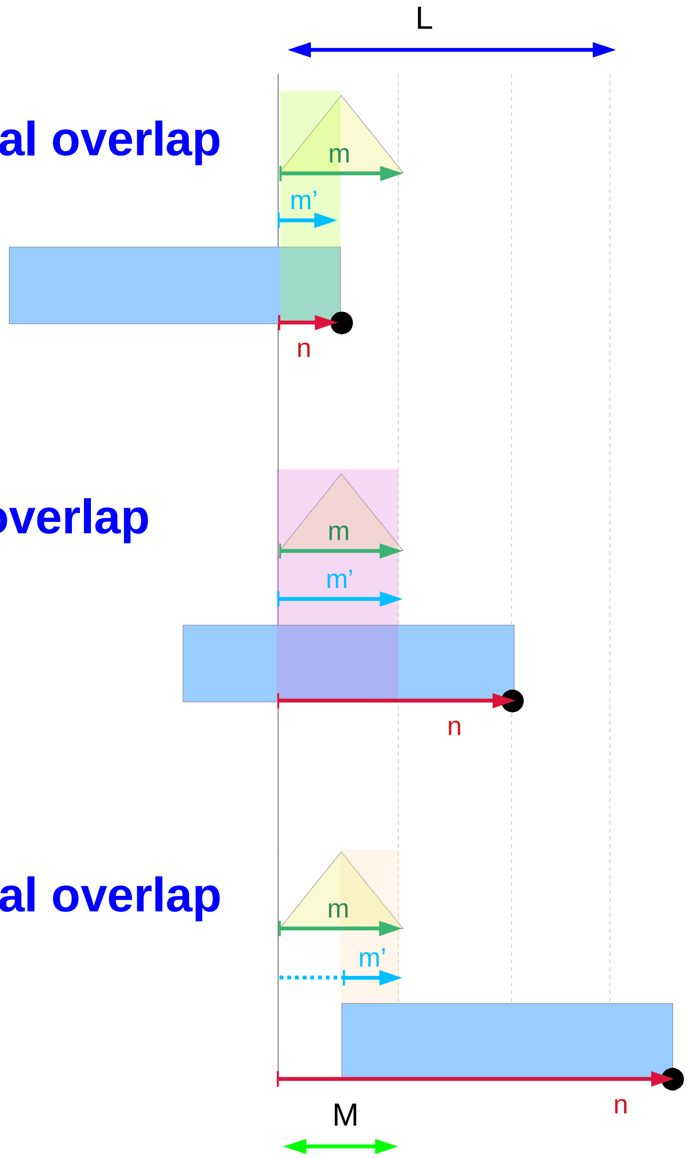
m' ∈ [n-L+1, M-1]

for a given **n**

n ∈ [L, L+M-2]

Effective index for **x[n-m]** (3)

Case B



partial overlap

h[m] $m' \in [0, n]$

x[n-m] $n-m' \in \overleftarrow{[n, 0]}$

for a given n

$n \in [0, M-2]$

full overlap

h[m] $m' \in [0, M-1]$

x[n-m] $n-m' \in \overleftarrow{[n, n-M+1]}$

for a given n

$n \in [M-1, L-1]$

partial overlap

h[m] $m' \in [n-L+1, M-1]$

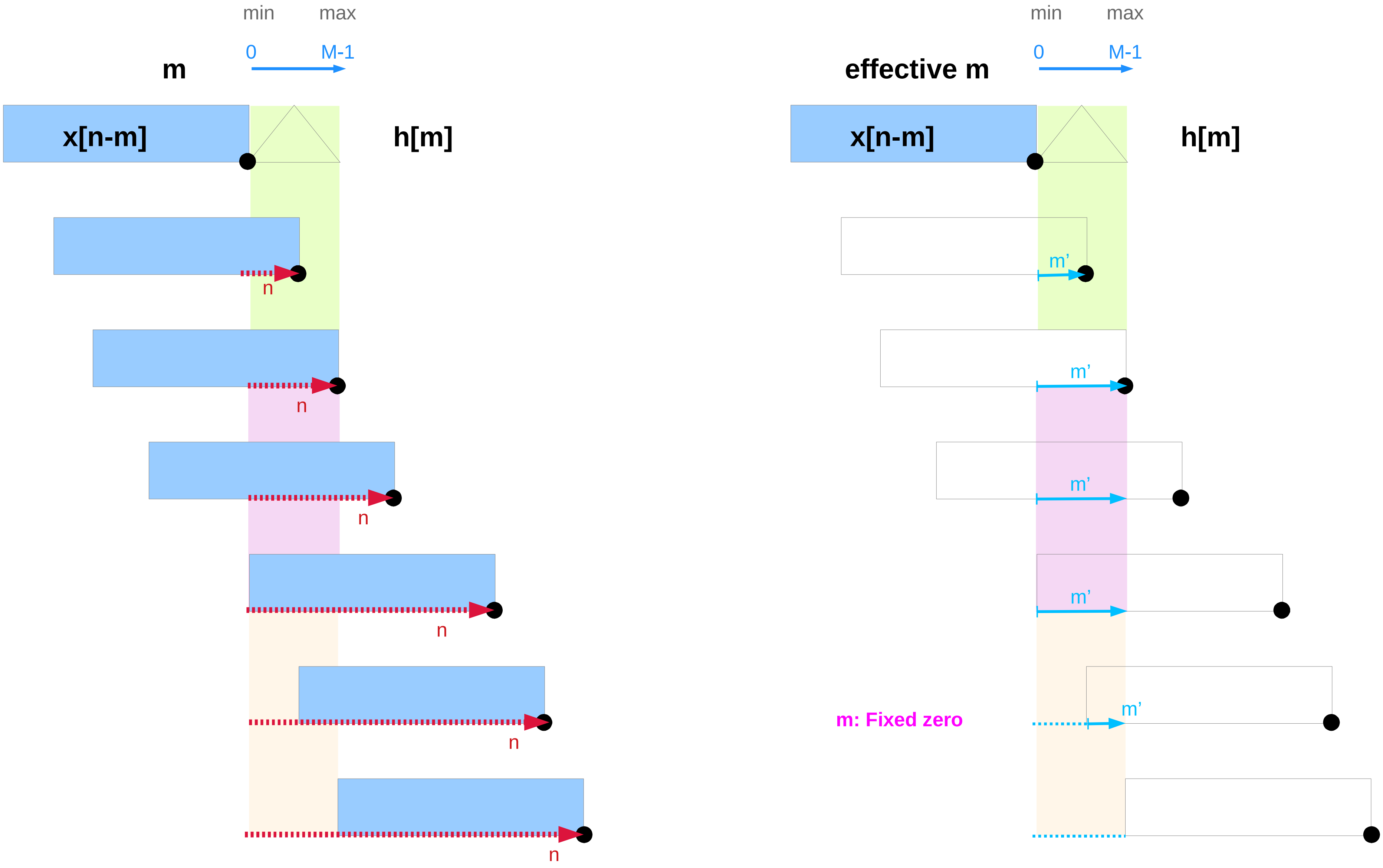
x[n-m] $n-m' \in \overleftarrow{[L-1, n-M+1]}$

for a given n

$n \in [L, L+M-2]$

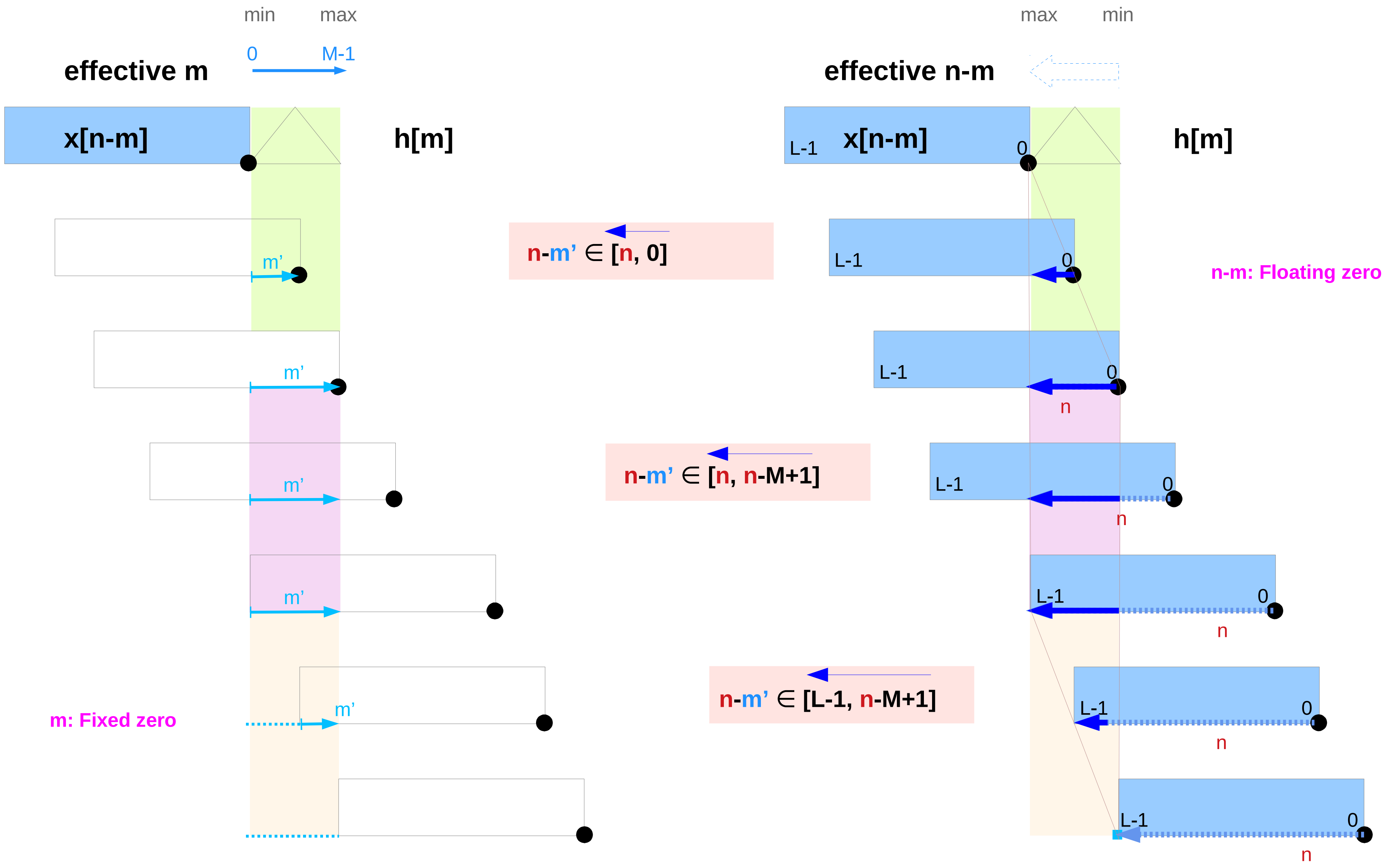
Index **n** and **m**

Case B



Index **m** and **n-m**

Case B



n-m' ∈ [n, 0]

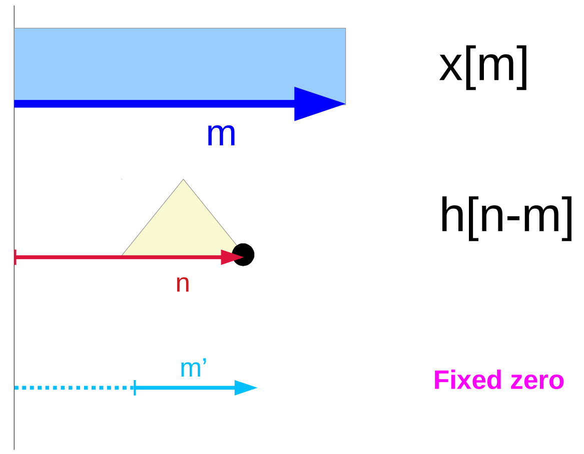
n-m' ∈ [n, n-M+1]

n-m' ∈ [L-1, n-M+1]

- Summary

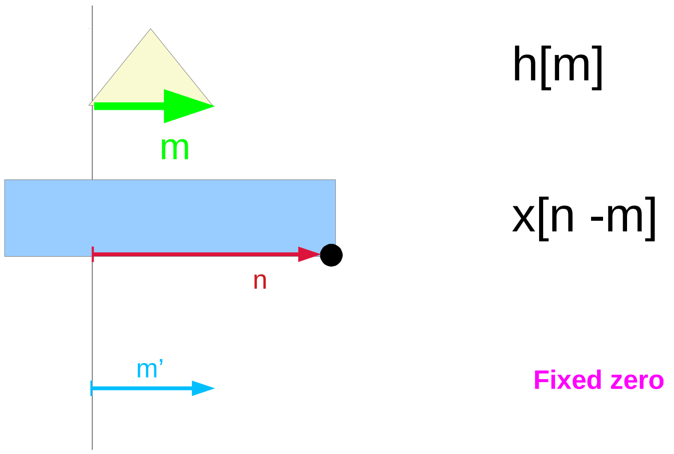
Summary (1) : effective ranges for **m**

Case A



x[m]	**m’ ∈ [0, n]**
x[m]	**m’ ∈ [n-M+1, n]**
x[m]	**m’ ∈ [n-M+1, L-1]**

Case B

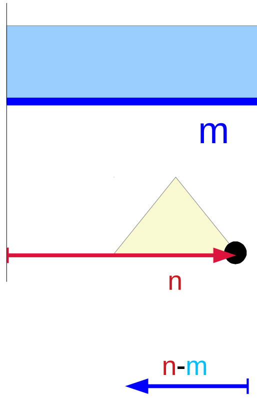


h[m]	**m’ ∈ [0, n]**
h[m]	**m’ ∈ [0, M-1]**
h[m]	**m’ ∈ [n-L+1, M-1]**

Summary (2) : effective ranges for **n-m**

Case A, B

Case A



x[m]

h[n-m]

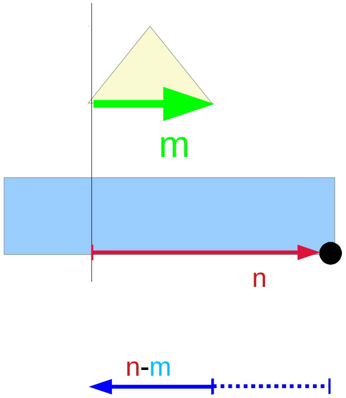
Floating zero

h[n-m] $n-m' \in [\; n, \quad 0]$

h[n-m] $n-m' \in [\; M-1, \quad 0]$

h[n-m] $n-m' \in [\; M-1, \; n-L+1]$

Case B



h[m]

x[n -m]

Floating zero

x[n -m] $n-m' \in [\; n, \quad 0]$

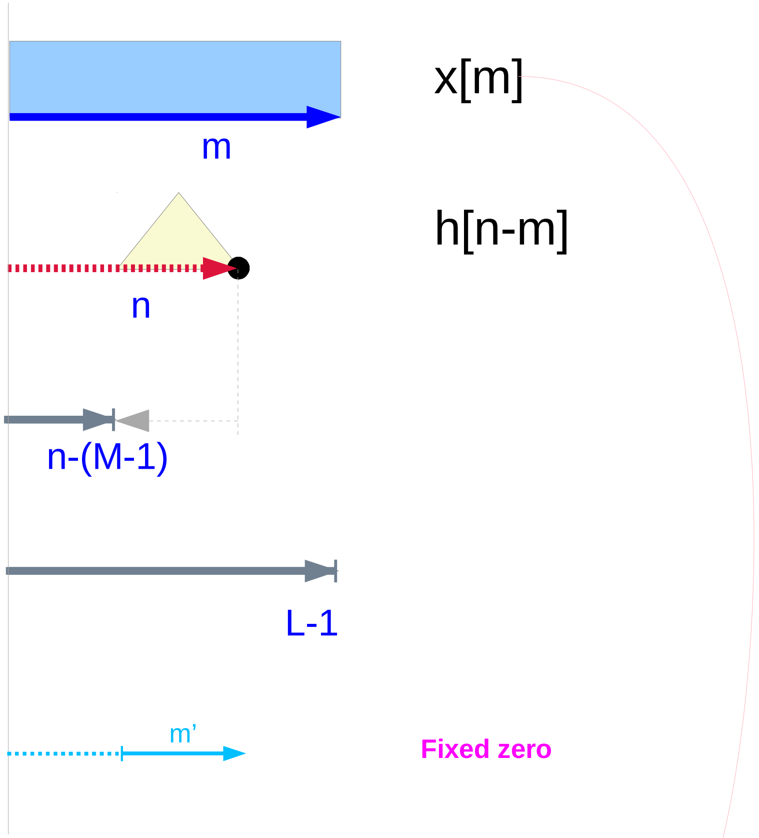
x[n -m] $n-m' \in [\; n, \; n-M+1]$

x[n -m] $n-m' \in [L-1, \; n-M+1]$

Summary (3) : memorizing effective ranges for **m**

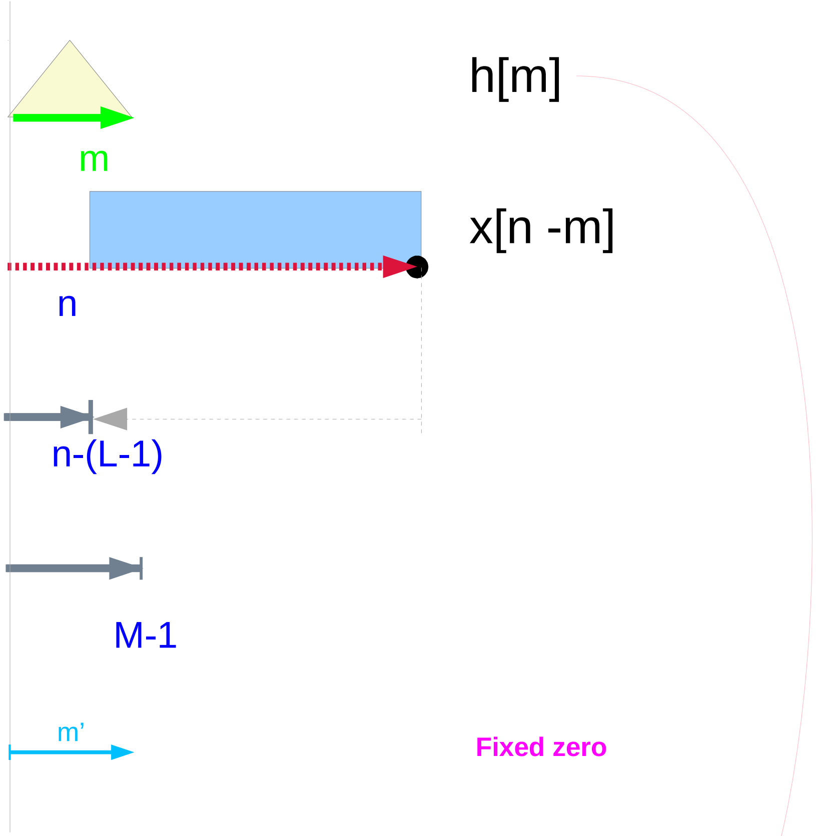
Case A, B

Case A



[max(0, n-(M-1)), min(n, L-1)]

Case B

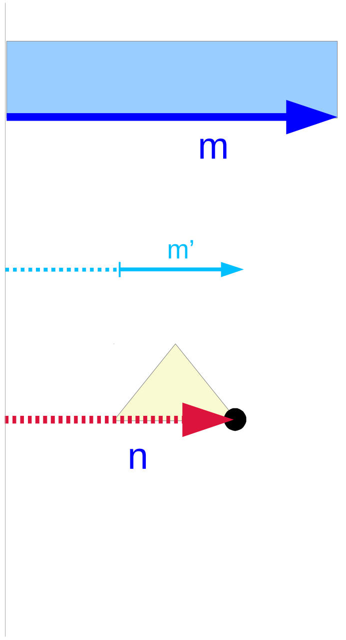


[max(0, n-(L-1)), min(n, M-1)]

Summary (4) : lower and upper bounds for **m**

Case A, B

Case A



upper bound

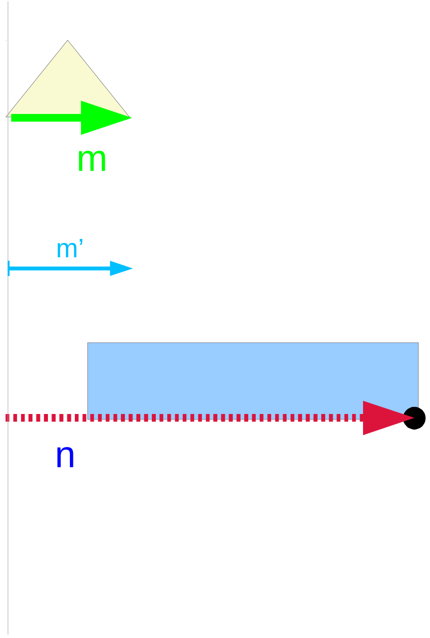
min(n, L-1)

Fixed zero

lower bound

max(0, n-(M-1))

Case B



upper bound

min(n, M-1)

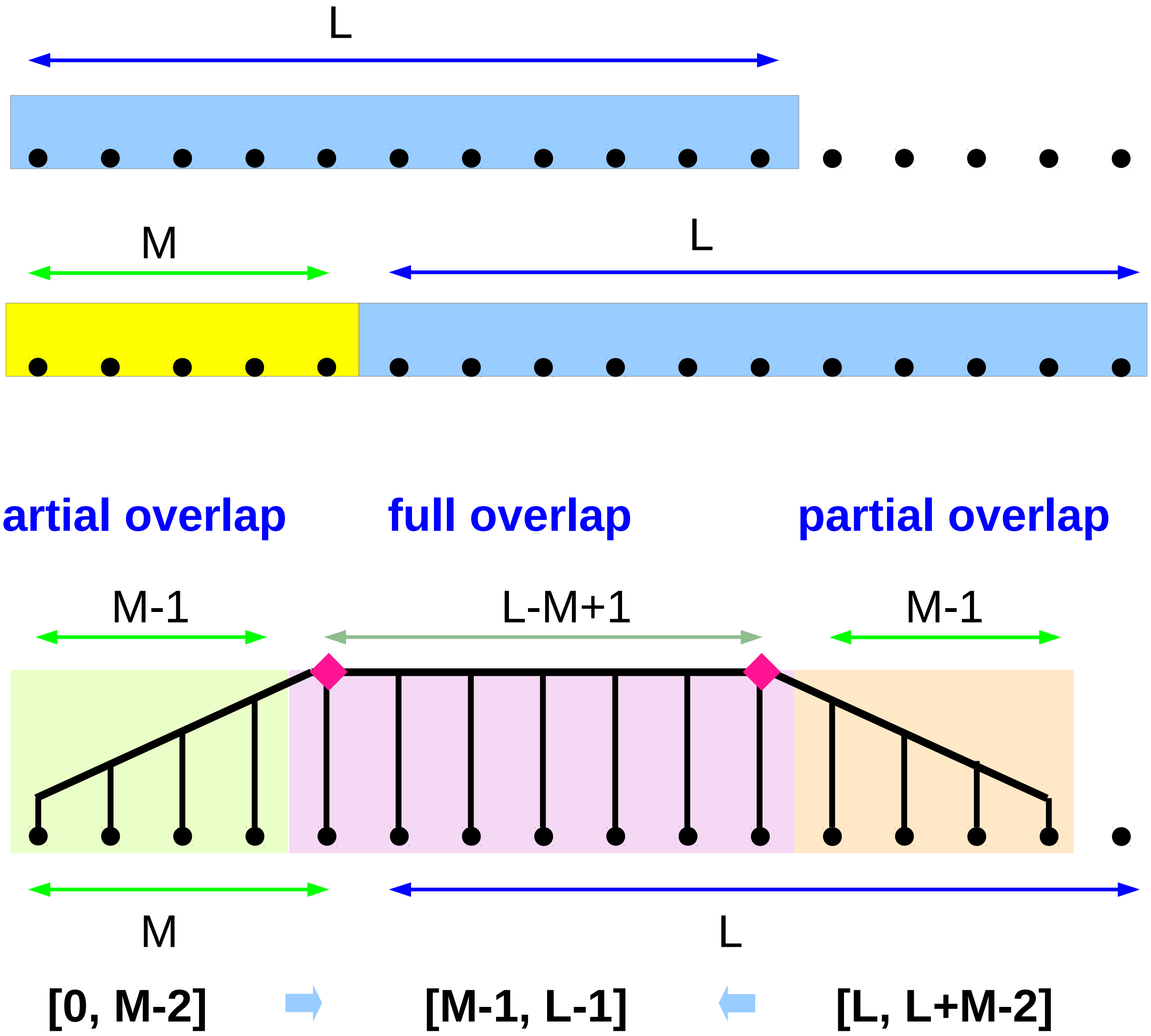
Fixed zero

lower bound

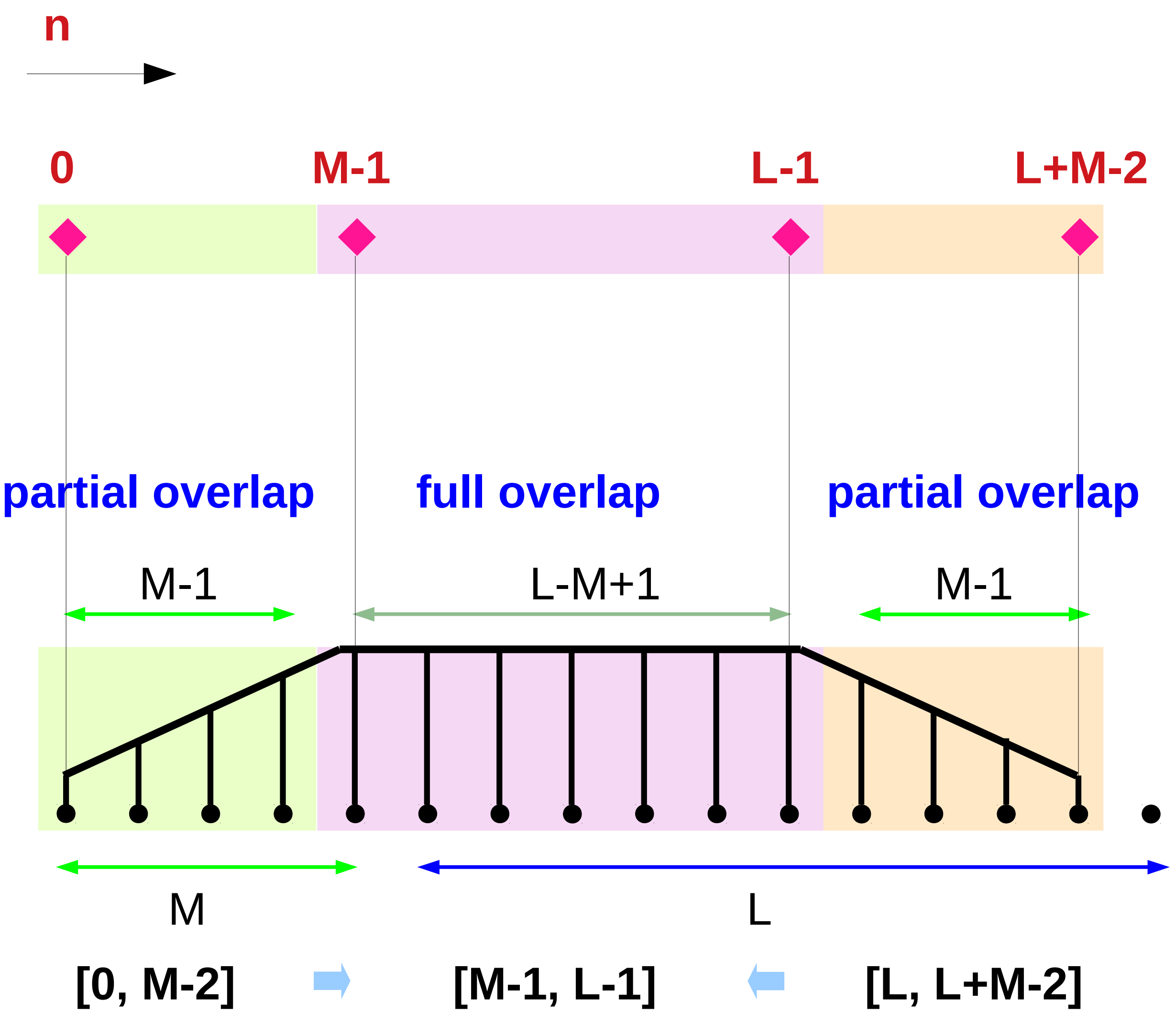
max(0, n-(L-1))

- Range Partitions for **n**

Sizes of overlapped index regions



Four boundary points



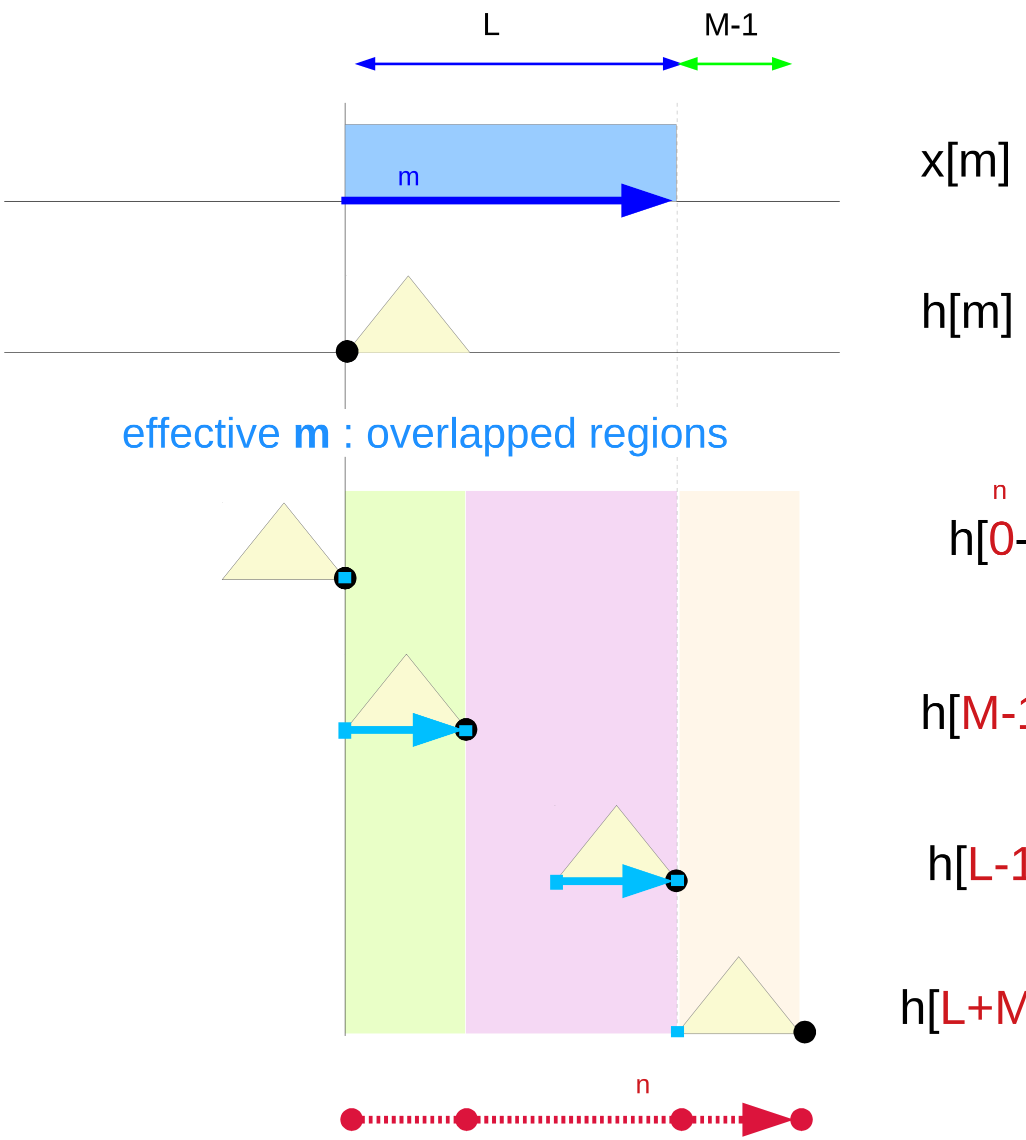
h[n-m] at the boundary points (1)

Case A

M+L-1 L M

y[n] += x[m] * h[n-m];

n ∈ [0, L+M-2]
m ∈ [0, L-1]
n-m ∈ [0, M-1]



effective **m** : overlapped regions

Pt 1	value	y[0]	h[0-m]	function
partial overlap		⋮		
Pt 2	value	y[M-1]	h[M-1-m]	function
full overlap		⋮		
Pt 3	value	y[L-1]	h[L-1-m]	function
partial overlap		⋮		
Pt 4	value	y[L+M-2]	h[L+M-2-m]	function

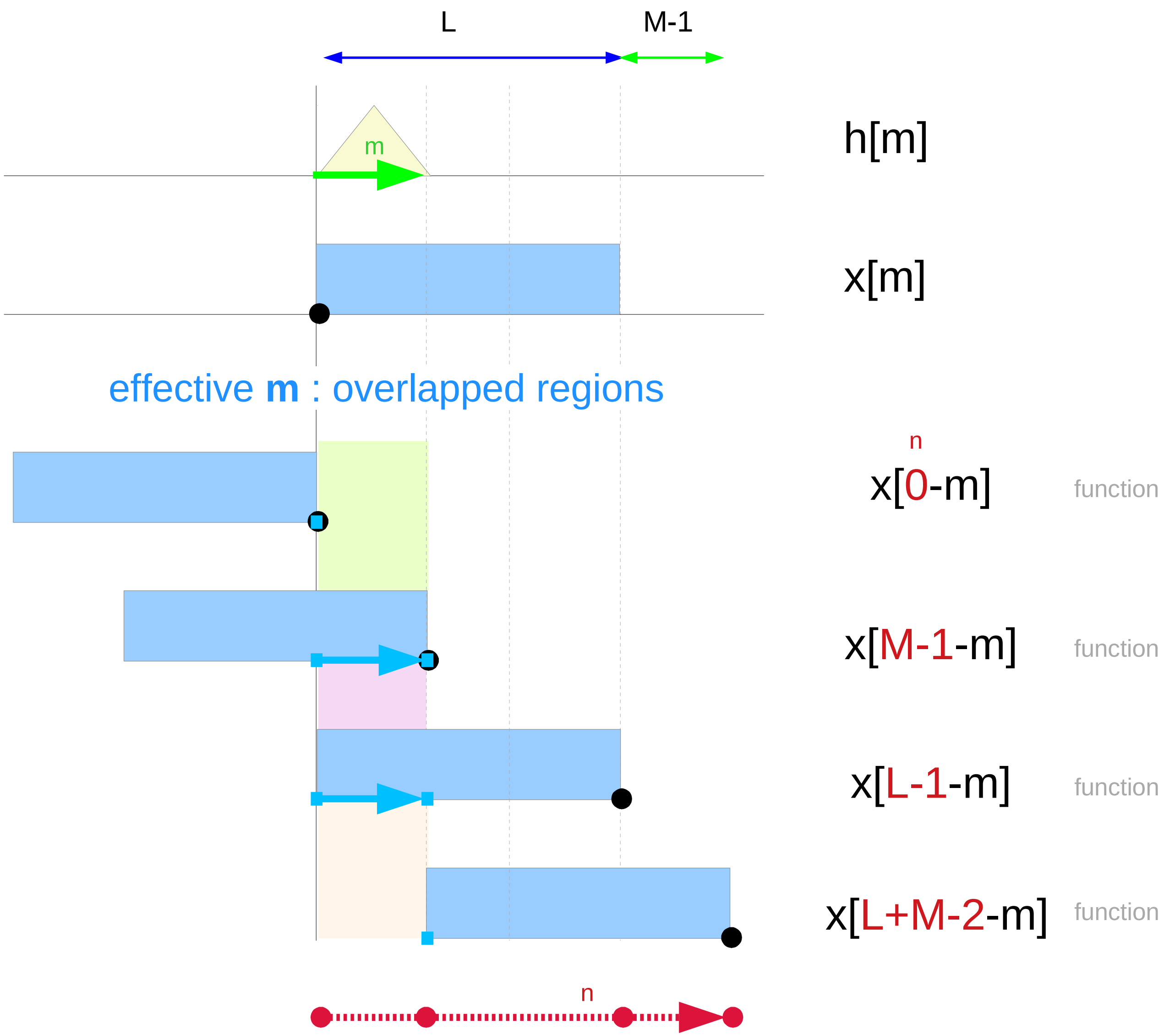
x[n-m] at the boundary points (2)

Case B

M+L-1　M　L

```
y[n] += h[m] * x[n-m];
```

n　∈ [0, L+M-2]
m　∈ [0, M-1]
n-m　∈ [0, L-1]



effective **m** : overlapped regions

Pt 1 value y[0] — x[0-m] function

partial overlap

Pt 2 value y[M-1] — x[M-1-m] function

full overlap

Pt 3 value y[L-1] — x[L-1-m] function

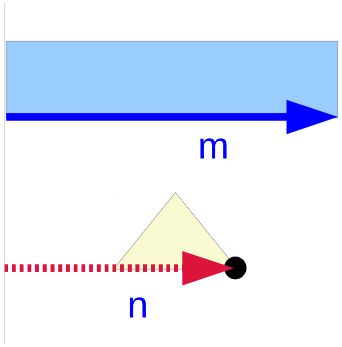
partial overlap

Pt 4 value y[L+M-2] — x[L+M-2-m] function

Effective ranges of **m** and **n-m**

Case A, **B**

Case A

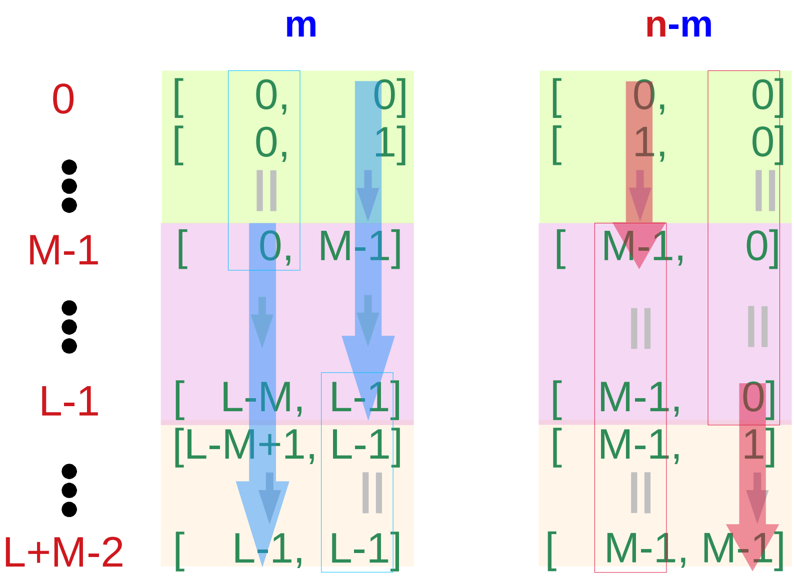


upper bound

min(n, L-1)

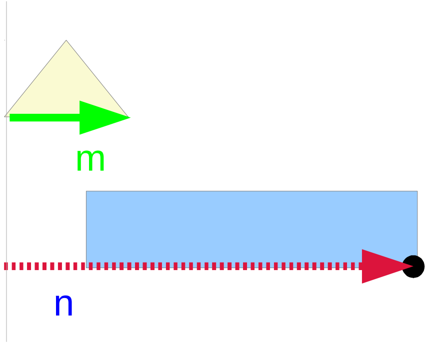
lower bound

max(0, n-(M-1))



	m	**n-m**
0	[0, 0]	[0, 0]
	[0, 1]	[1, 0]
⋮	‖	‖
M-1	[0, M-1]	[M-1, 0]
⋮		‖ ‖
L-1	[L-M, L-1]	[M-1, 0]
	[L-M+1, L-1]	[M-1, 1]
⋮	‖	‖
L+M-2	[L-1, L-1]	[M-1, M-1]

Case B

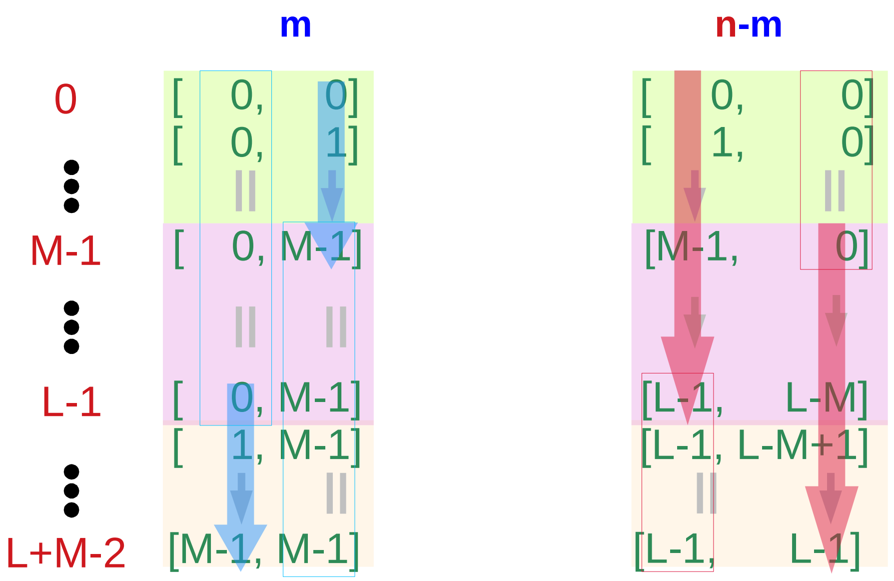


upper bound

min(n, M-1)

lower bound

max(0, n-(L-1))



	m	**n-m**
0	[0, 0]	[0, 0]
	[0, 1]	[1, 0]
⋮	‖	‖
M-1	[0, M-1]	[M-1, 0]
⋮	‖ ‖	
L-1	[0, M-1]	[L-1, L-M]
	[1, M-1]	[L-1, L-M+1]
⋮	‖	‖
L+M-2	[M-1, M-1]	[L-1, L-1]

Effective ranges of **m** in **x[m]** (1)

Case A

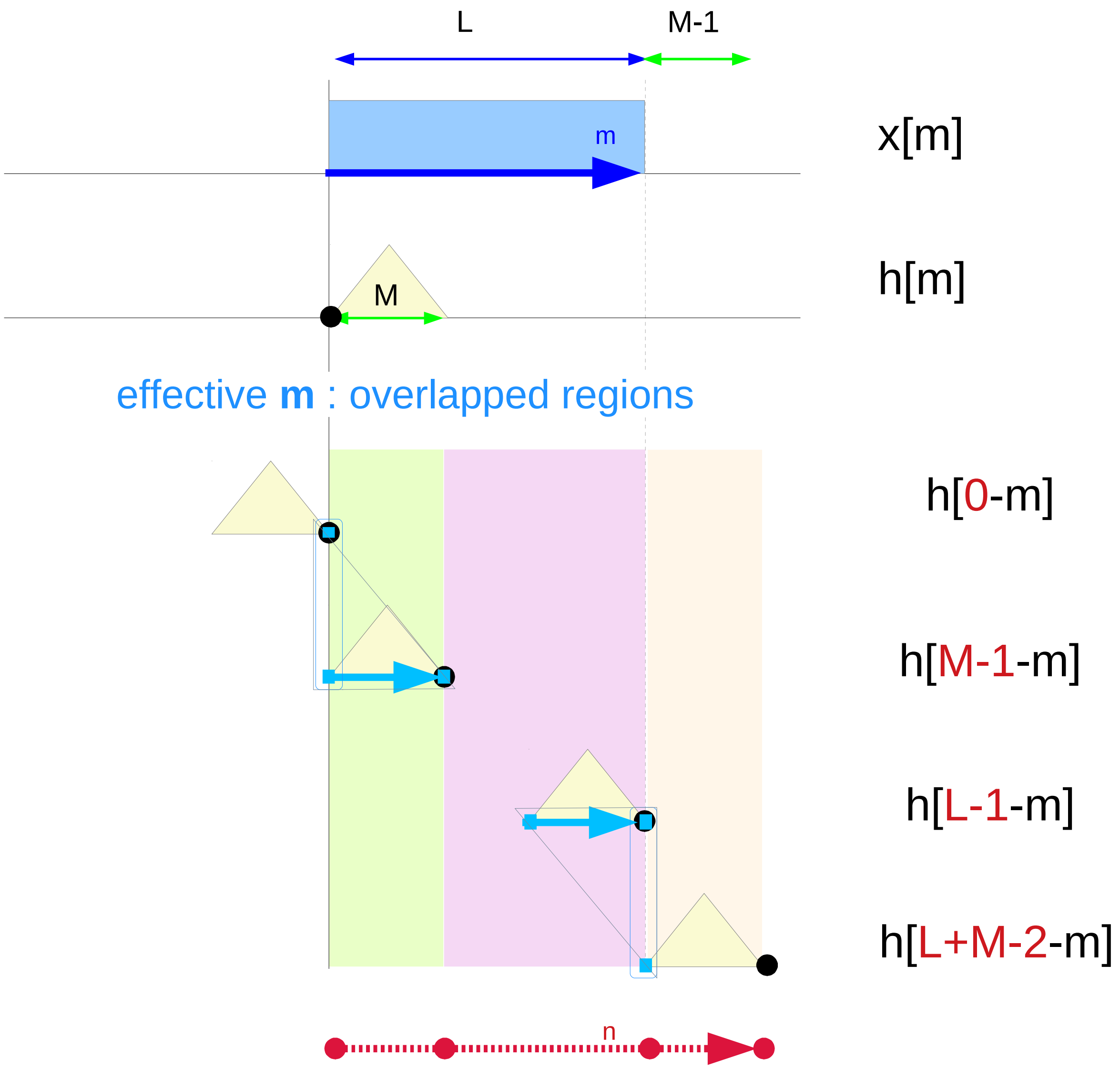
M+L-1 L M

y[n] += x[m] * h[n-m];

n ∈ [0, L+M-2]
m ∈ [0, L-1]
n-m ∈ [0, M-1]

	max(0, n-(M-1)) lower bound	min(n, L-1) upper bound
y[0]	[0,	0]
	[0,	1]
⋮	‖	↓
y[M-1]	[0,	M-1]
⋮	↓	↓
y[L-1]	[L-M,	L-1]
	[L-M+1,	L-1]
⋮	↓	‖
y[L+M-2]	[L-1,	L-1]

n n-m



effective **m** : overlapped regions

Effective ranges of **m** in **h[m]** (2)

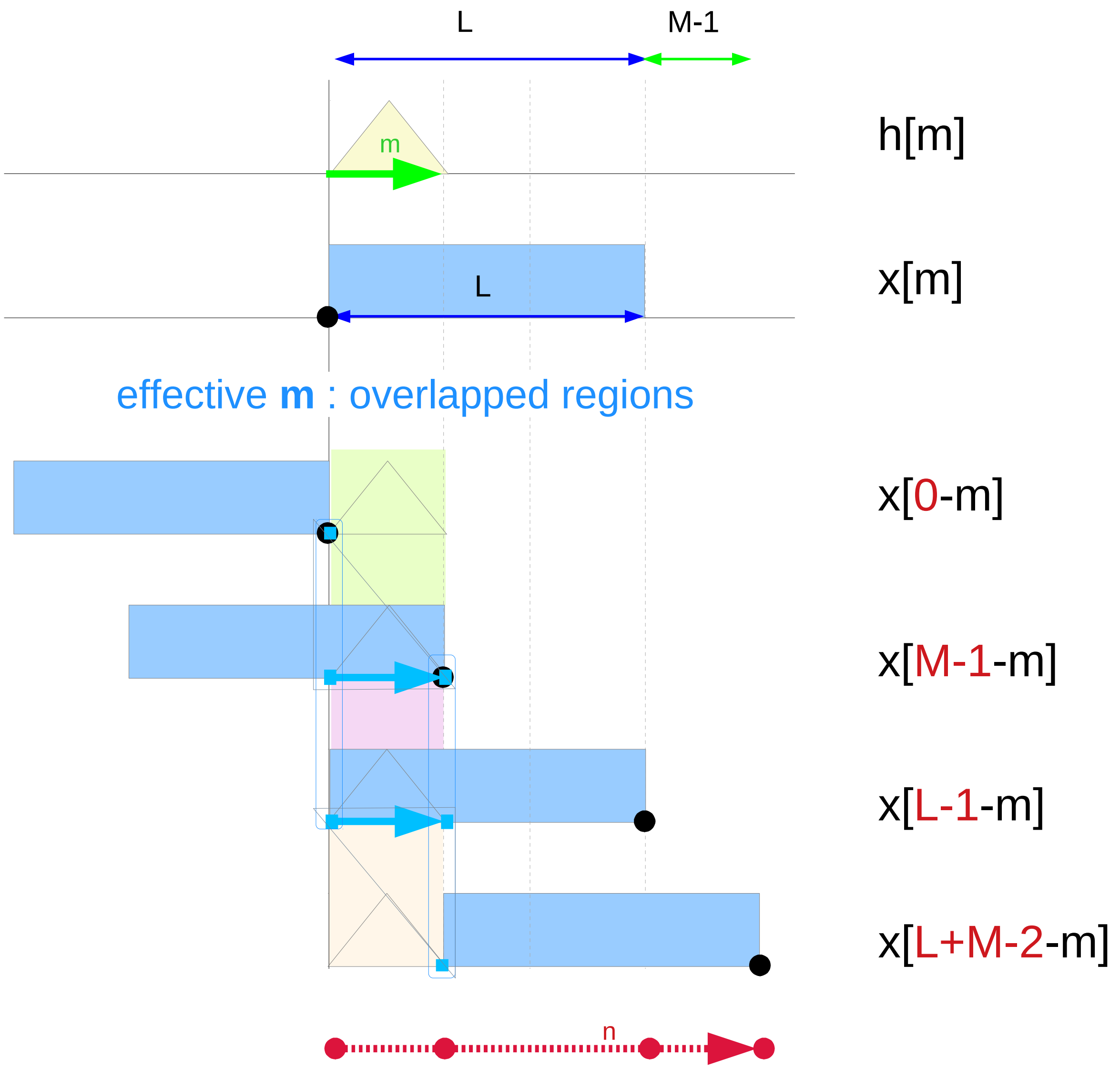
Case B

M+L-1 M L

y[n] += h[m] * x[n-m];

n ∈ [0, L+M-2]
m ∈ [0, M-1]
n-m ∈ [0, L-1]

	max(0, n-(L-1)) lower bound	min(n, M-1) upper bound
y[0]	[0,	0]
	[0,	1]
⋮	‖	↓
y[M-1]	[0,	M-1]
⋮	‖	‖
y[L-1]	[0,	M-1]
	[1,	M-1]
⋮	↓	‖
y[L+M-2]	[M-1,	M-1]
n	n-m	



effective **m** : overlapped regions

Effective ranges of **n-m** in **h[n-m]** (1)

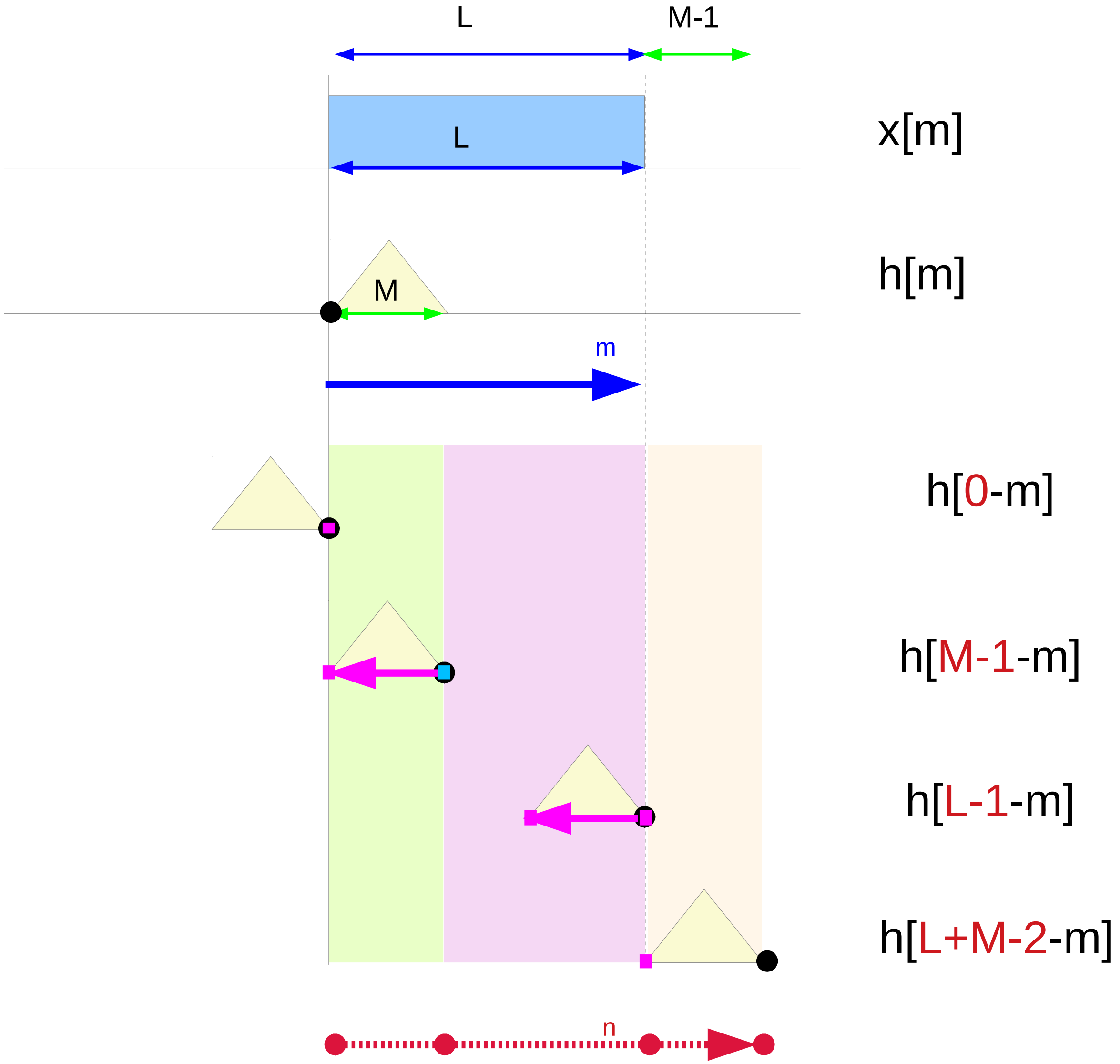
Case A

M+L-1 L M

y[n] += x[m] * h[n-m];

n ∈ [0, L+M-2]
m ∈ [0, L-1]
n-m ∈ [0, M-1]

	max(0, n-(M-1)) lower bound	min(n, L-1) upper bound
y[0]	[0,	0]
	[1,	0]
⋮	↓	‖
y[M-1]	[M-1,	0]
⋮	‖	‖
y[L-1]	[M-1,	0]
	[M-1,	1]
⋮	‖	↓
y[L+M-2]	[M-1,	M-1]
n	n-m	



Effective ranges of **n-m** in **x[n-m]** (2)

Case B

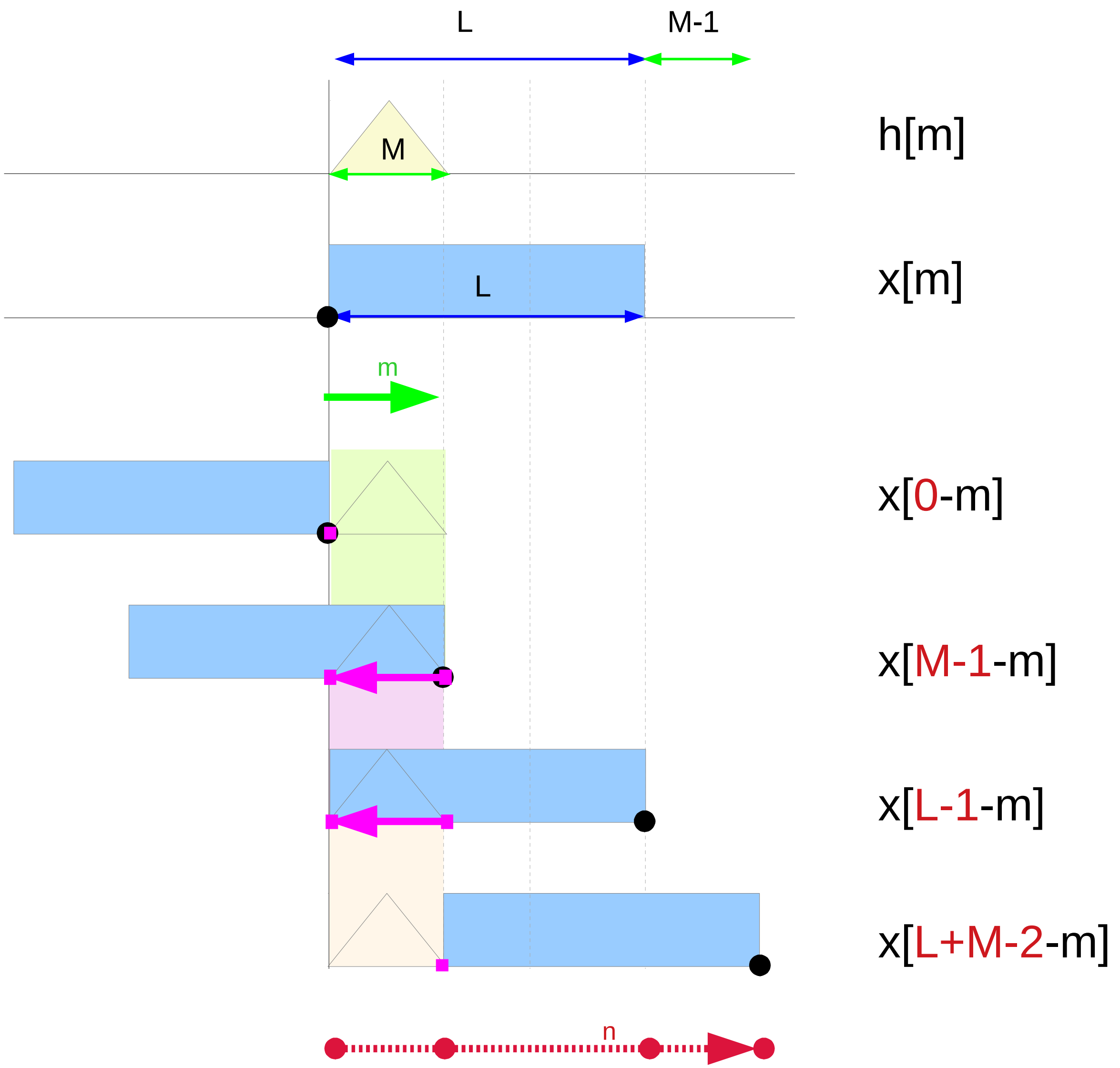
M+L-1 M L

`y[n] += h[m] * x[n-m];`

n ∈ [0, L+M-2]
m ∈ [0, M-1]
n-m ∈ [0, L-1]

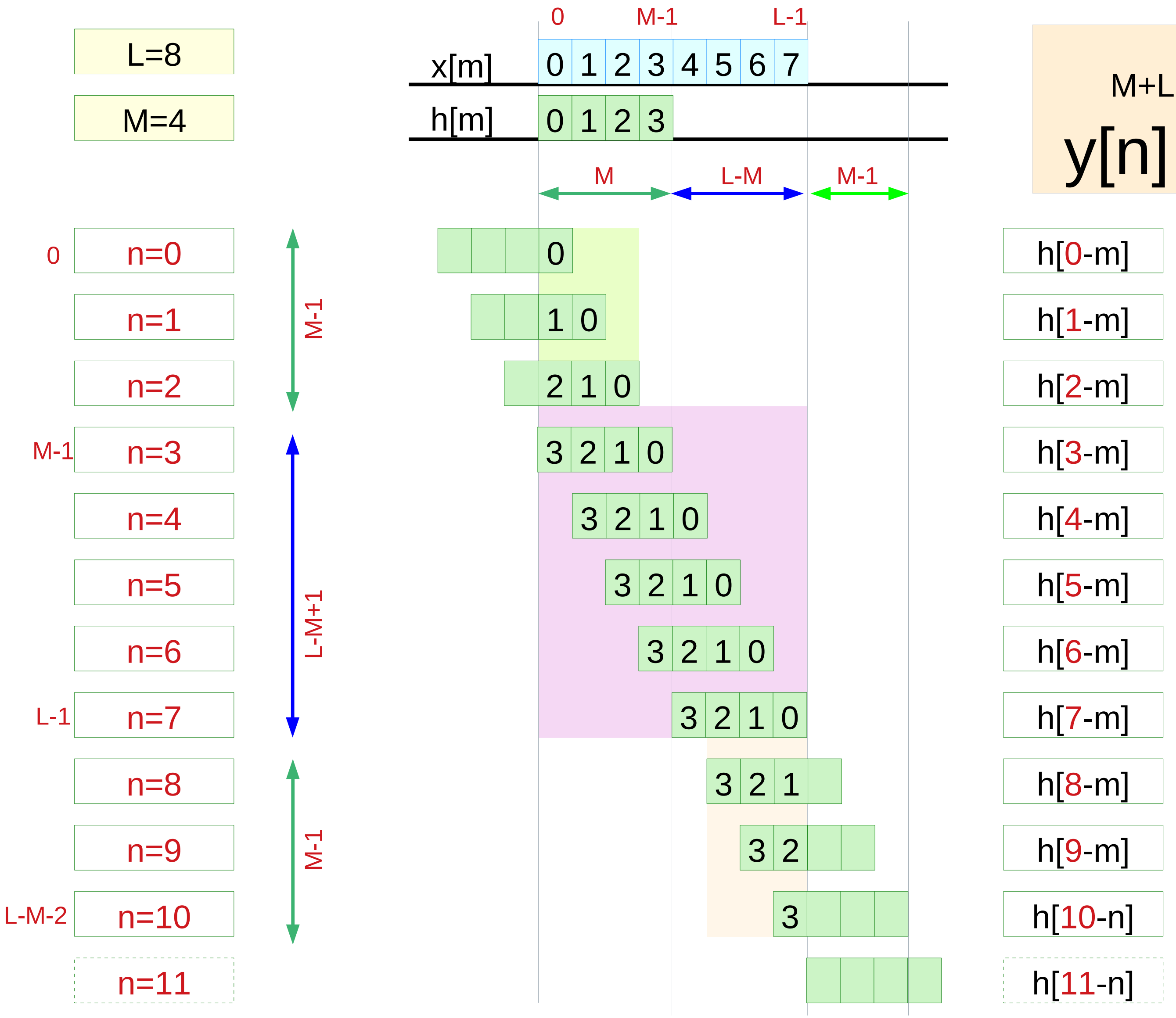
	max(0, n-(L-1)) lower bound	min(n, M-1) upper bound
y[0]	[0,	0]
	[1,	0]
y[M-1]	[M-1,	0]
y[L-1]	[L-1,	L-M]
	[L-1,	L-M+1]
y[L+M-2]	[L-1,	L-1]

n n-m



Index **n-m** value example

Case A



L=8

M=4

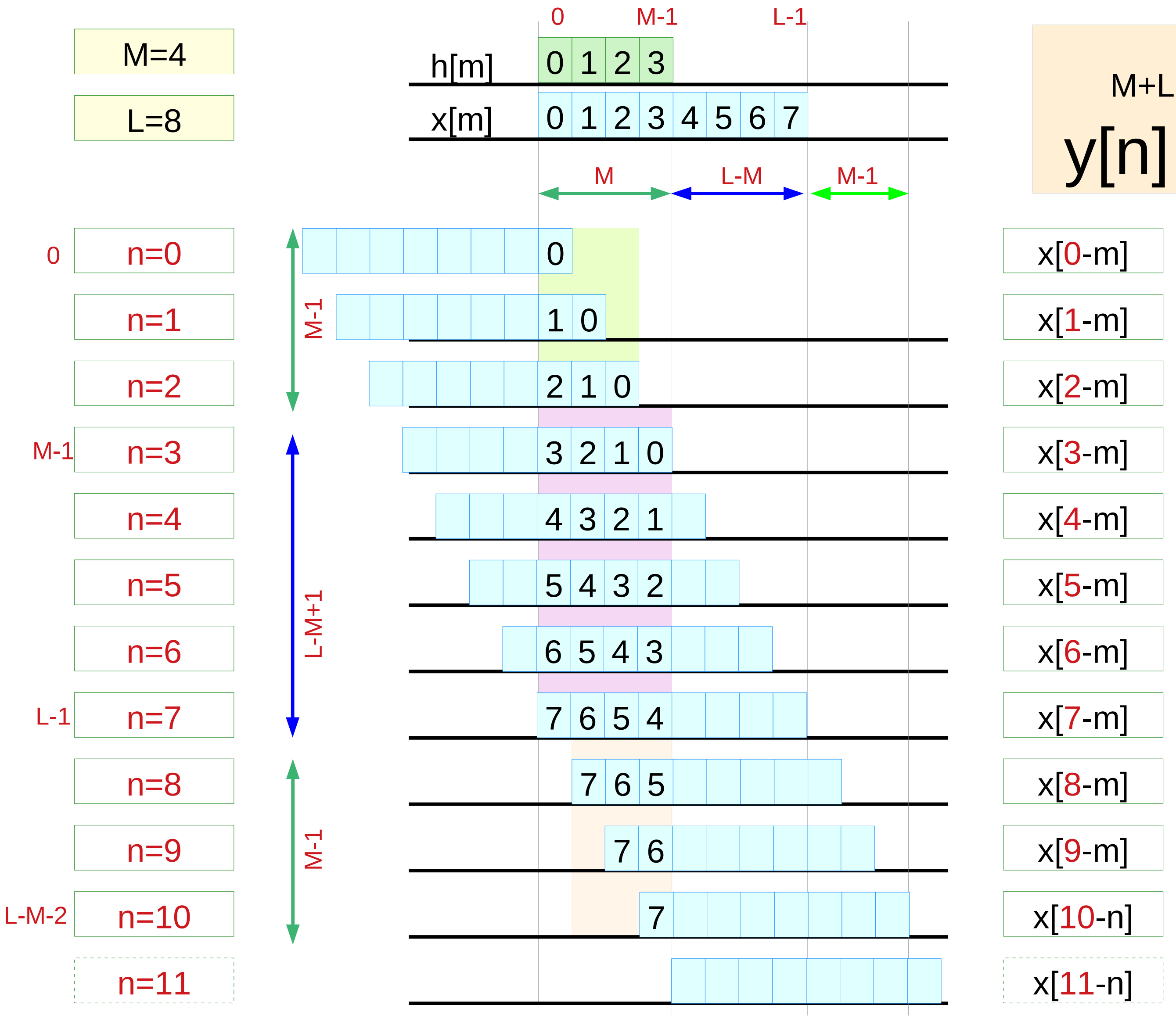
	x[m]	0 1 2 3 4 5 6 7	
	h[m]	0 1 2 3	
0	n=0	0	h[0-m]
	n=1	1 0	h[1-m]
	n=2	2 1 0	h[2-m]
M-1	n=3	3 2 1 0	h[3-m]
	n=4	3 2 1 0	h[4-m]
	n=5	3 2 1 0	h[5-m]
	n=6	3 2 1 0	h[6-m]
L-1	n=7	3 2 1 0	h[7-m]
	n=8	3 2 1	h[8-m]
	n=9	3 2	h[9-m]
L-M-2	n=10	3	h[10-n]
	n=11		h[11-n]

M+L-1 L M

y[n] += x[m] * h[n-m];

Index **n-m** value example

Case B



M+L-1 M L

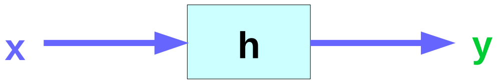
y[n] += h[m] * x[n-m];

- Reasoning about lower and upper bounds of **m**

Index Variable Constraints

Case A

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$



Constraint 1 : **n $\in$ [0, L+M-2]**

Constraint 2 : **n-m $\in$ [0, M-1]**

Constraint 3 : **m $\in$ [0, L-1]**

y[ ] : array with size of **L+M-1**

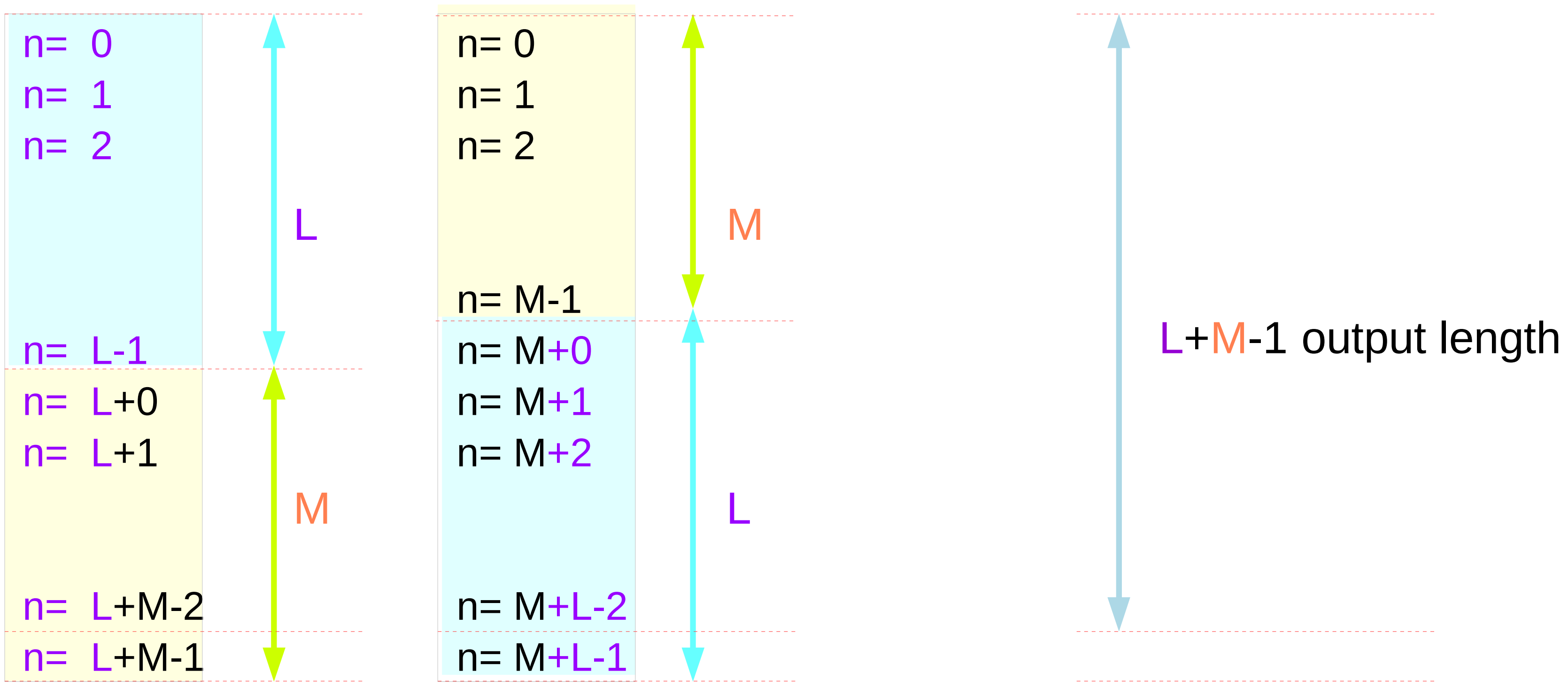
h[ ] : array with size of **M**

x[ ] : array with size of **L**

Constraint 1 – counting **n**

Case A

Constraint 1 :**n ∈ [0, L+M-2]**



n= 0
n= 1
n= 2
n= L-1
n= L+0
n= L+1
n= L+M-2
n= L+M-1

L
M

n= 0
n= 1
n= 2
n= M-1
n= M+0
n= M+1
n= M+2
n= M+L-2
n= M+L-1

M
L

L+M-1 output length

Counting 1

Counting 2

M < L is assumed

len(filter) **<** len(input)

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$

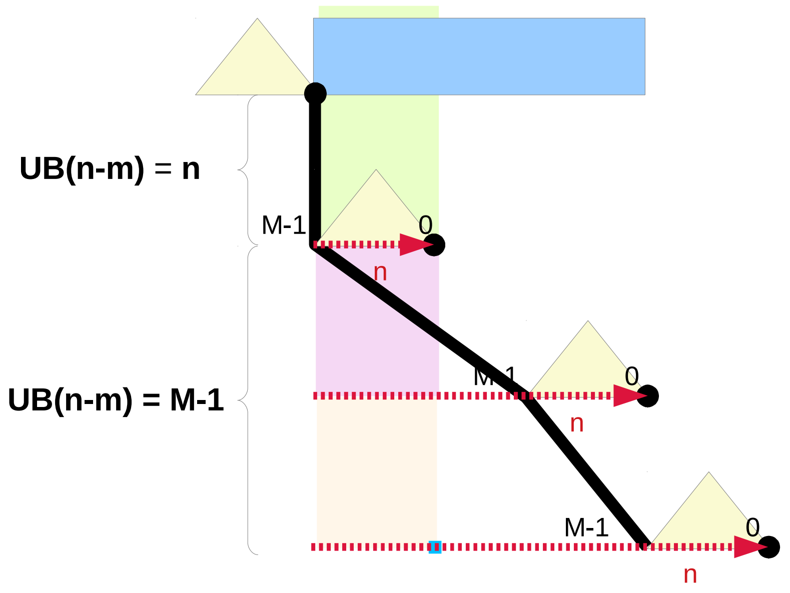
Lower Bound of **m** = max(**0**, n-(M-1))

Case A

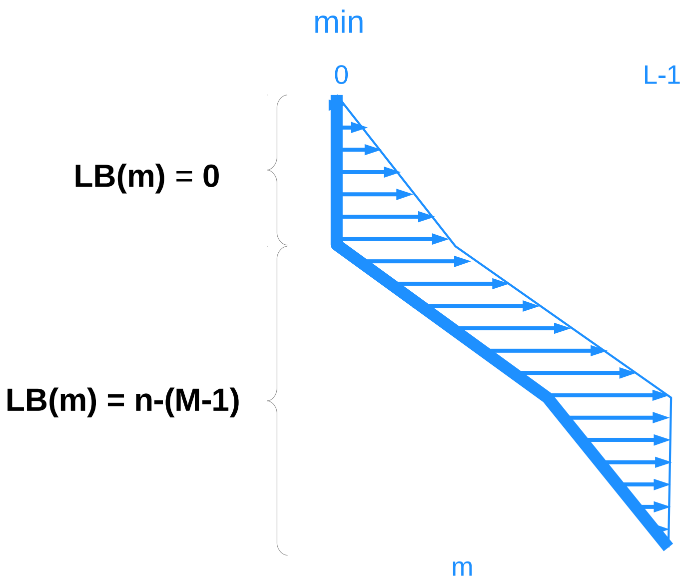
UB(n-m)

LB(m) = max(0, n-(M-1))

LB(m)



Floating zero



Fixed zero

```
M+L-1         L         M
y[n] += x[m] * h[n-m];
```

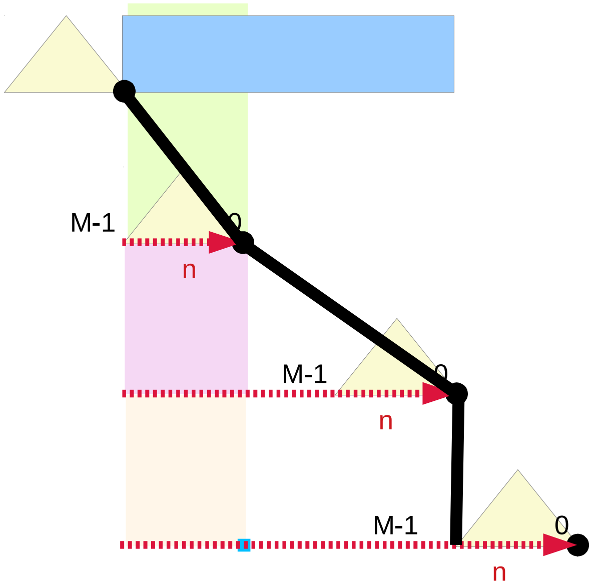
Upper Bound of **m** = min(n, **L-1**)

Case A

LB(n-m)

UB(m) = min(n, L-1)

UB(m)



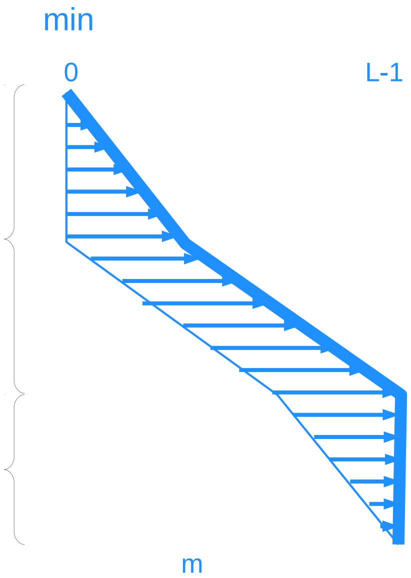
LB(n-m) = **0**

LB(n-m) = n-(L-1)

Floating zero

UB(m) = **n**

UB(m) = (L-1)



Fixed Zero

M+L-1 L M

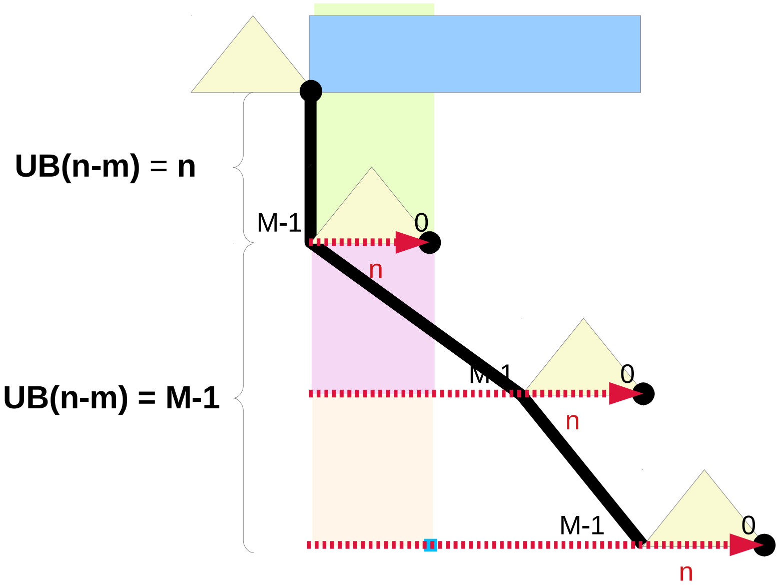
y[n] += x[m] * h[n-m];

Constraint 1 & 2 – UB(**n-m**) ➡ LB(**m**)

Case A

Constraint 2 : **n-m** ∈ **[0, M-1]** ➡ n - m : <u>max</u> **M-1** ... 0 min   (0, n+1-M)

for **UB(n-m)** values
m should be <u>least</u> possible



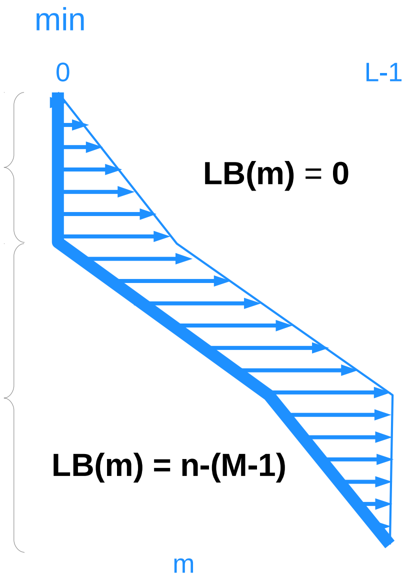
Floating zero

0 ≤ (n-m) ≤ M-1

Case A) n ≤ M-1
➡UB(**n-m**) **= n**
➡LB(**m**) = **0**

Case B) n ≥ M
➡UB(**n-m**) = **M-1**
➡LB(**m**) = **n-(M-1)**

LB(m) = max(0, n-(M-1))



Fixed zero

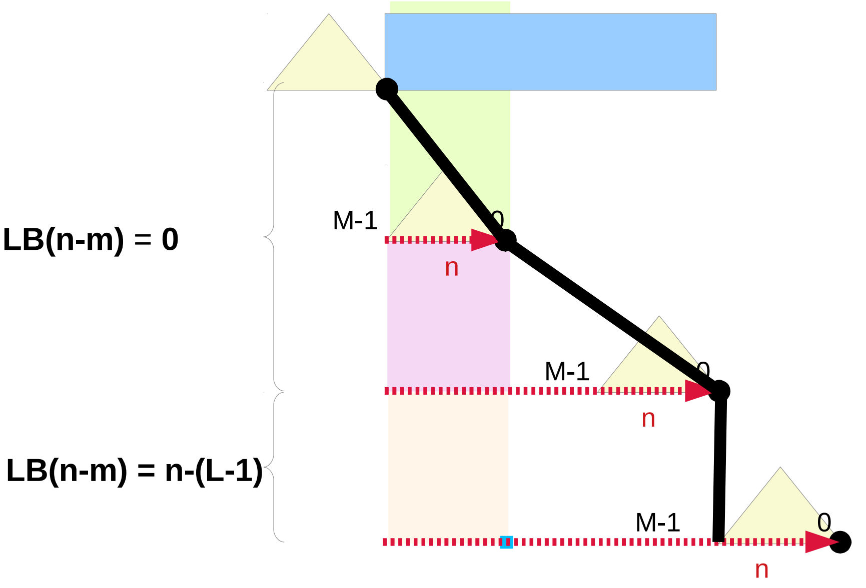
M+L-1   L   M
y[n] += x[m] * h[n-m];

Constraint 1 & 3 – UB(**m**) ➡ LB(**n-m**)

Case A

Constrain 3 : **m** ∈ **[0, L-1]** ➡ n - m : max M-1 ... **min 0** (n, M-1)

for **LB(n-m)** values
m should be greatest possible



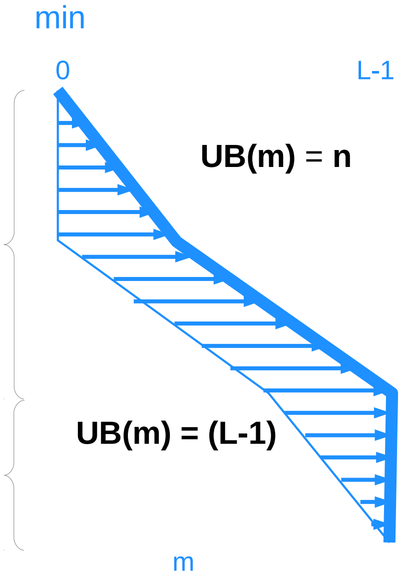
Floating zero

0 ≤ (n-m) ≤ M-1

Case A) n ≤ M-1
➡ UB(**m**) = **n**
➡ LB(**n-m**) = **0**

Case B) n ≥ M
➡ UB(m) = **L-1**
➡ LB(**n-m**) = **n-(L-1)**

UB(m) = min(n, L-1)



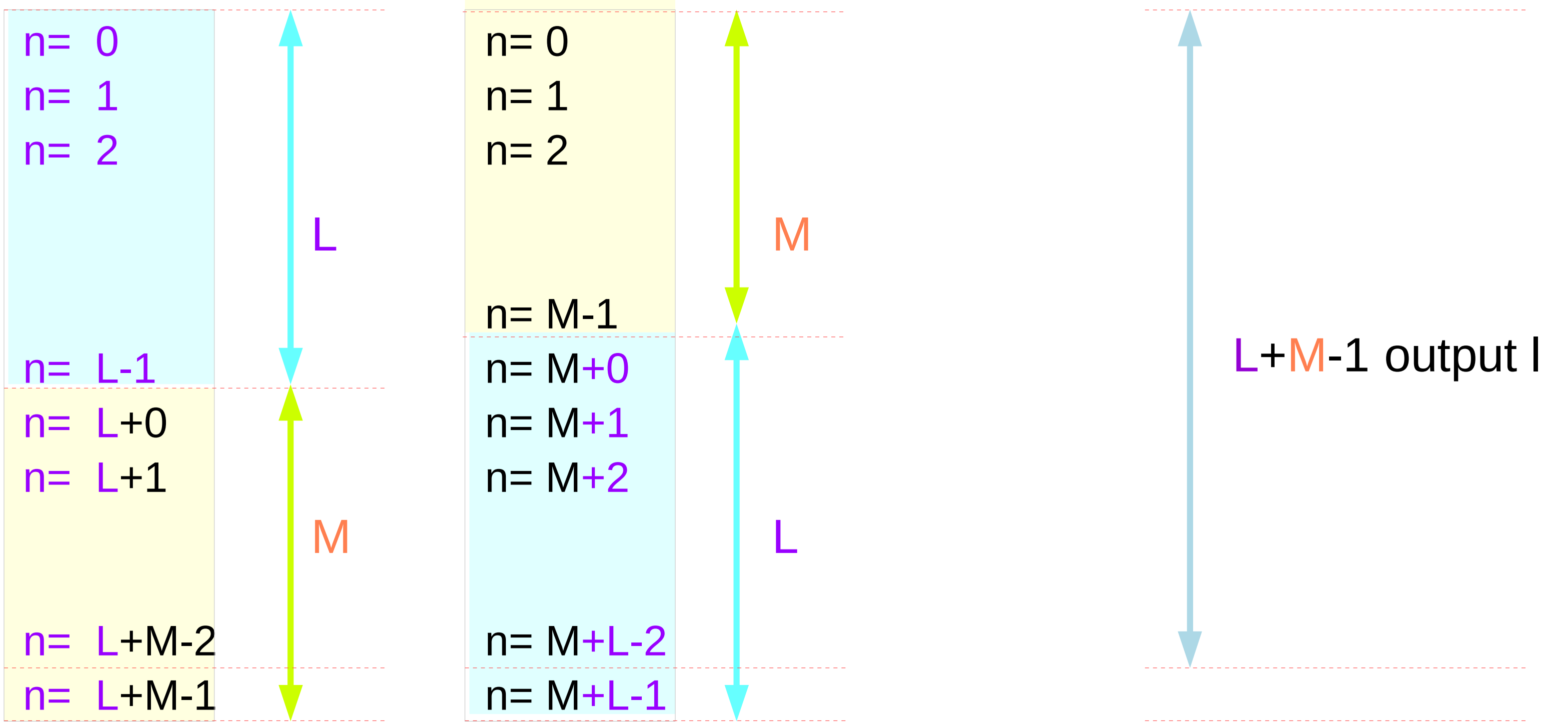
Fixed zero

$$y[n] \mathrel{+}= x[m] * h[n-m];$$

M+L-1 L M

Case A

Constraint 1 : **$n \in [0, L+M-2]$**



n= 0
n= 1
n= 2
n= L-1
n= L+0
n= L+1
n= L+M-2
n= L+M-1

Counting 1

n= 0
n= 1
n= 2
n= M-1
n= M+0
n= M+1
n= M+2
n= M+L-2
n= M+L-1

Counting 2

M < L is assumed

len(filter) **<** len(input)

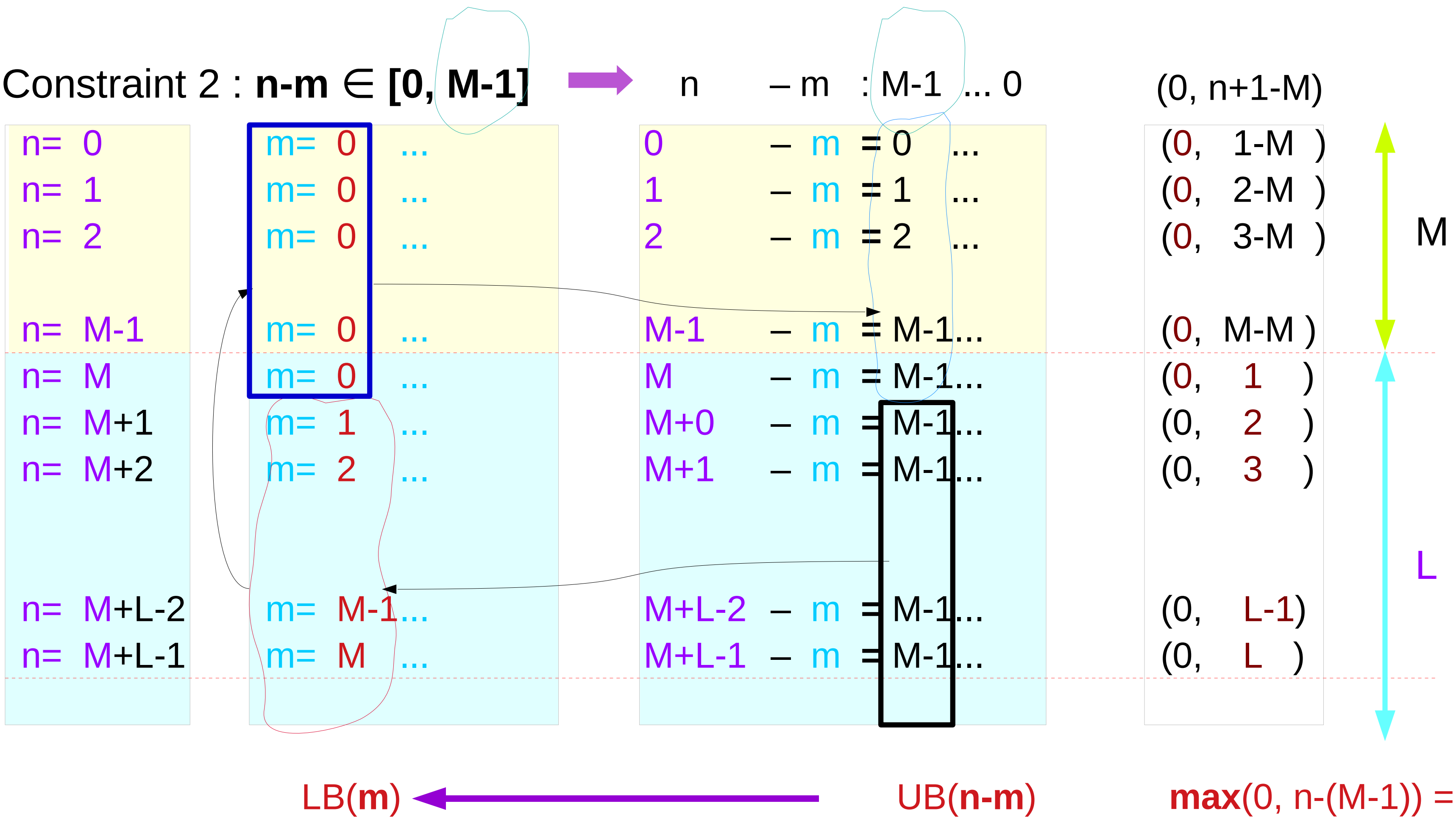
M+L-1 L M

y[n] += x[m] * h[n-m];

Constraint 1 & 2 – UB(**n-m**) ➡ LB(**m**)

Case A

Constraint 2 : **n-m** ∈ **[0, M-1]** ➡ n – m : M-1 ... 0   (0, n+1-M)



n= 0	m= 0 ...	0 – m = 0 ...	(0, 1-M)
n= 1	m= 0 ...	1 – m = 1 ...	(0, 2-M)
n= 2	m= 0 ...	2 – m = 2 ...	(0, 3-M)
n= M-1	m= 0 ...	M-1 – m = M-1...	(0, M-M)
n= M	m= 0 ...	M – m = M-1...	(0, 1)
n= M+1	m= 1 ...	M+0 – m = M-1...	(0, 2)
n= M+2	m= 2 ...	M+1 – m = M-1...	(0, 3)
n= M+L-2	m= M-1...	M+L-2 – m = M-1...	(0, L-1)
n= M+L-1	m= M ...	M+L-1 – m = M-1...	(0, L)

LB(**m**) ← UB(**n-m**)   **max**(0, n-(M-1)) = LB(**m**)

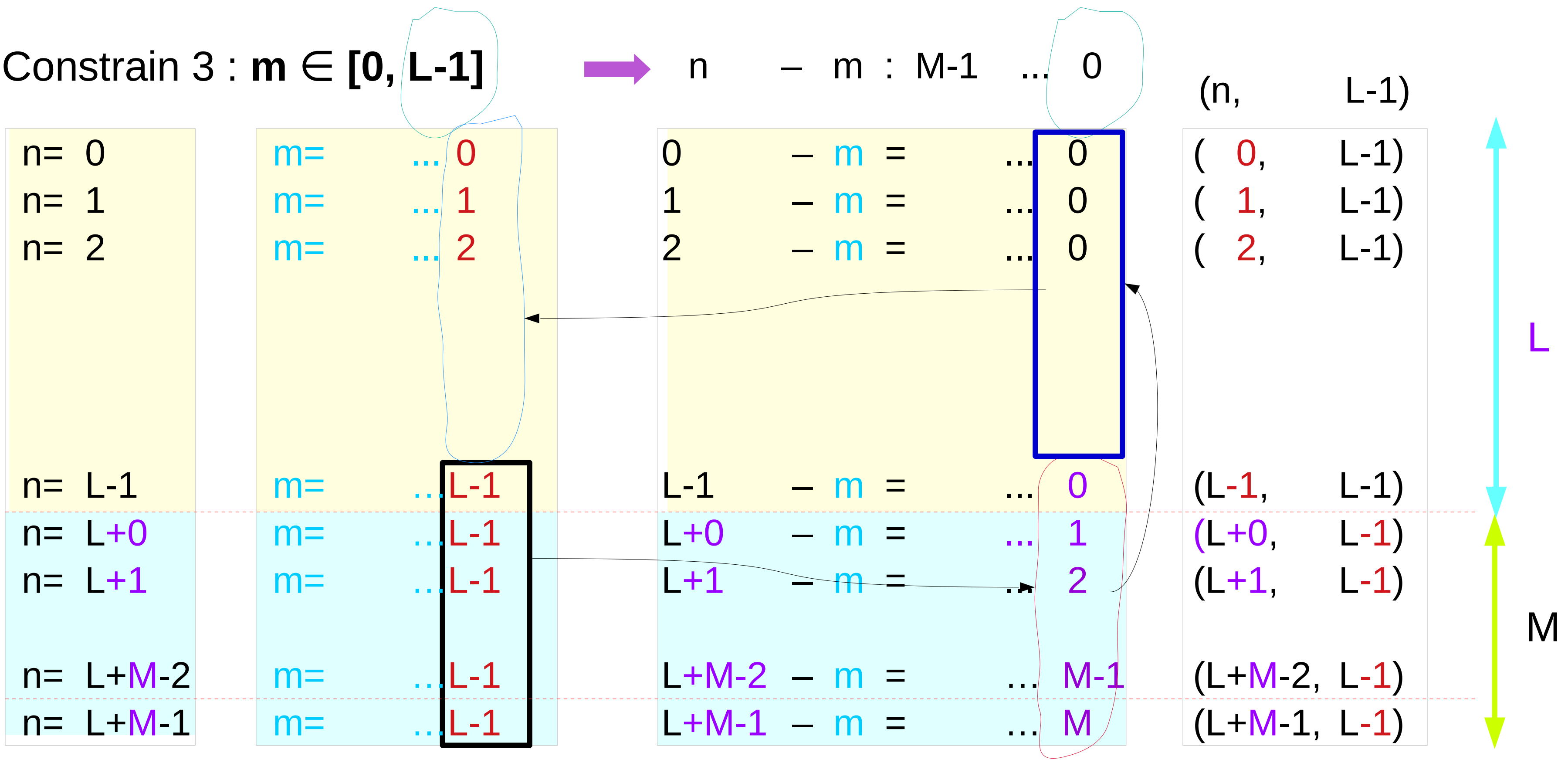
m = **[max**(0, n-(M-1)), **min**(n, L-1)**]**

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$

Constraint 1 & 3 – UB(**m**) ➡ LB(**n-m**)

Case A

Constrain 3 : **m** ∈ **[0, L-1]** ➡ n – m : M-1 ... 0



n= 0	m= ... 0	0 – m = ... 0	(0, L-1)
n= 1	m= ... 1	1 – m = ... 0	(1, L-1)
n= 2	m= ... 2	2 – m = ... 0	(2, L-1)
n= L-1	m= ... L-1	L-1 – m = ... 0	(L-1, L-1)
n= L+0	m= ... L-1	L+0 – m = ... 1	(L+0, L-1)
n= L+1	m= ... L-1	L+1 – m = ... 2	(L+1, L-1)
n= L+M-2	m= ... L-1	L+M-2 – m = ... M-1	(L+M-2, L-1)
n= L+M-1	m= ... L-1	L+M-1 – m = ... M	(L+M-1, L-1)

UB(**m**) → LB(**n-m**) **min**(n, L-1) = UB(**m**)

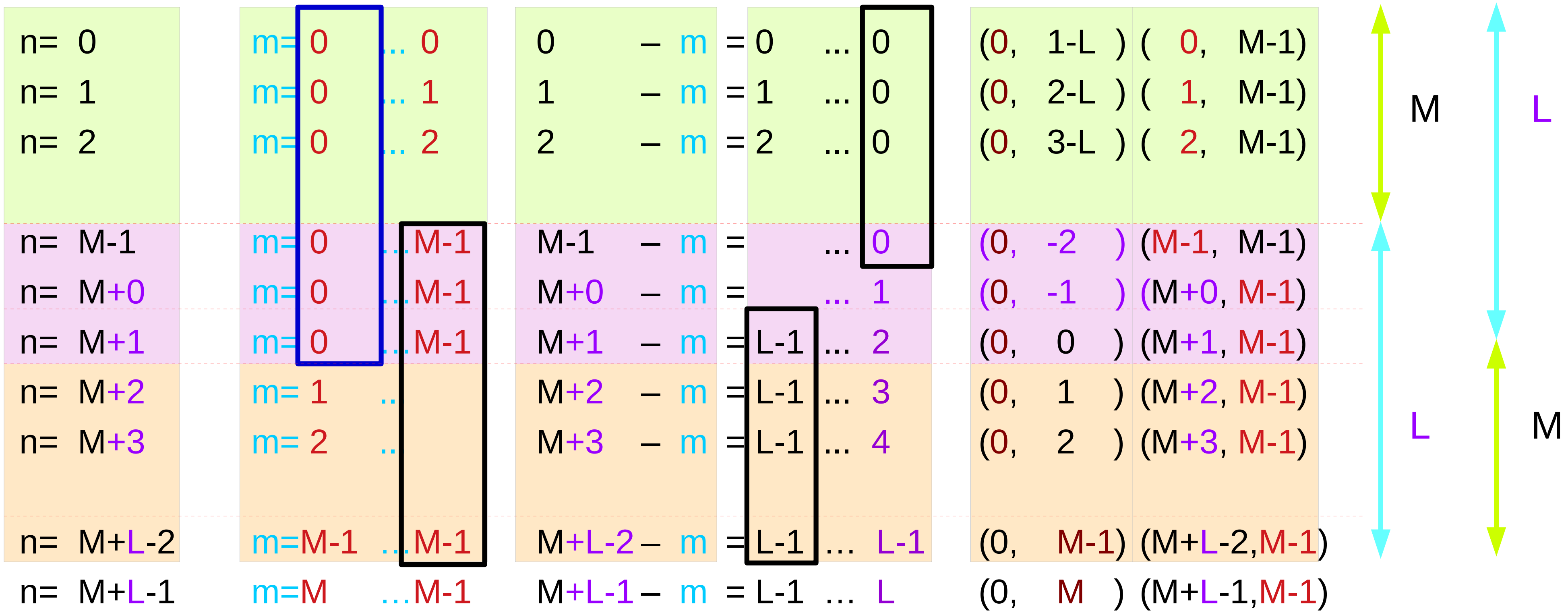
$$y[n] \mathrel{+}= \overset{M+L-1}{} x[m]^{L} * h[n-m]^{M};$$

m = **[max**(0, n-(M-1)), **min**(n, L-1)**]**

Constraint 1, 2, 3 – max **m** and min **m**

Case B

Constrain 3 : **m ∈ [0, M-1]** ➡ n – m : L-1 ... 0 (0, n+1-L) (n, M-1)



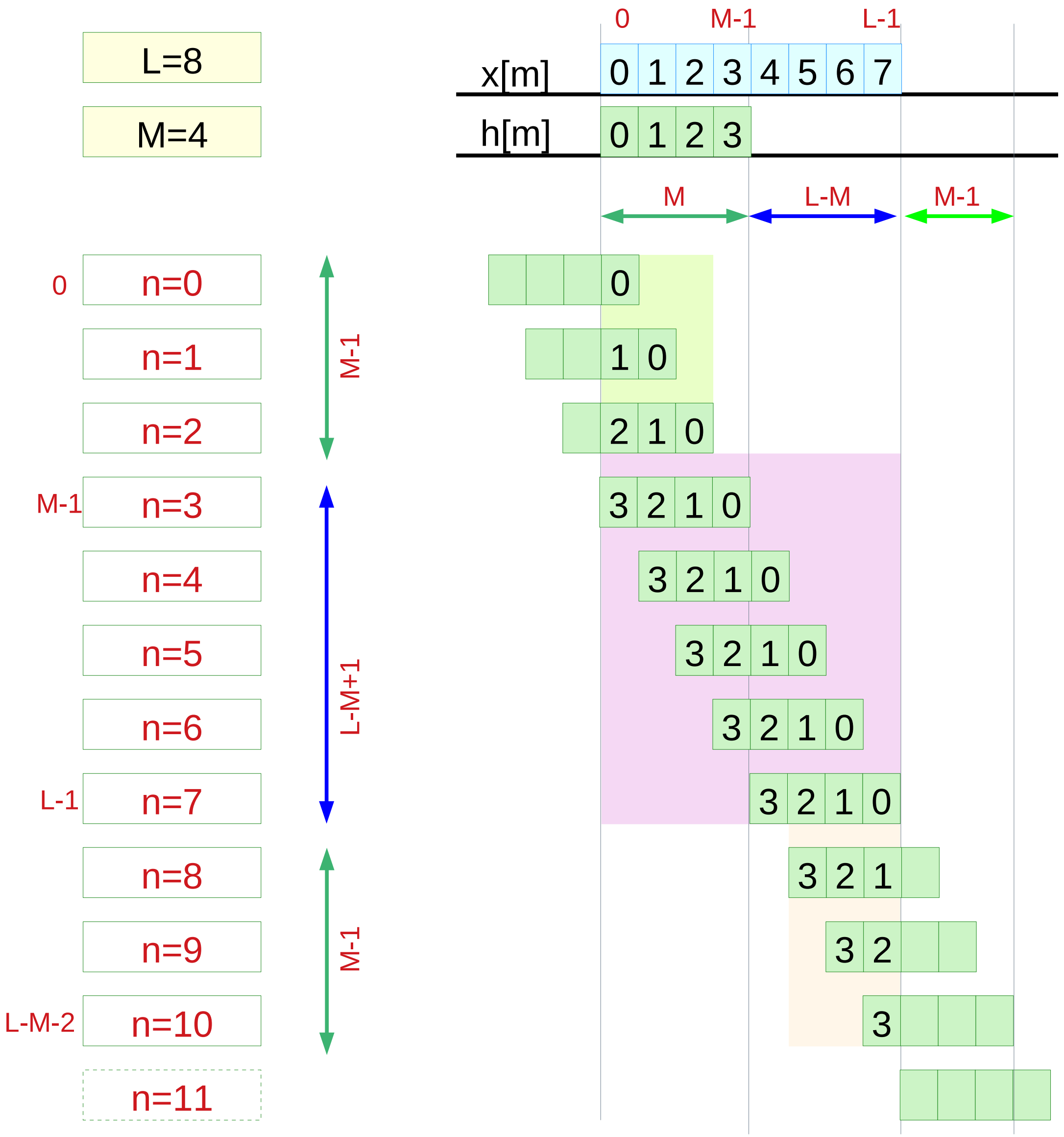
n= 0	m= 0	... 0	0 – m	= 0 ... 0	(0, 1-L)	(0, M-1)
n= 1	m= 0	... 1	1 – m	= 1 ... 0	(0, 2-L)	(1, M-1)
n= 2	m= 0	... 2	2 – m	= 2 ... 0	(0, 3-L)	(2, M-1)
n= M-1	m= 0	...M-1	M-1 – m	= ... 0	(0, -2)	(M-1, M-1)
n= M+0	m= 0	...M-1	M+0 – m	= ... 1	(0, -1)	(M+0, M-1)
n= M+1	m= 0	...M-1	M+1 – m	= L-1 ... 2	(0, 0)	(M+1, M-1)
n= M+2	m= 1	..	M+2 – m	= L-1 ... 3	(0, 1)	(M+2, M-1)
n= M+3	m= 2	..	M+3 – m	= L-1 ... 4	(0, 2)	(M+3, M-1)
n= M+L-2	m=M-1	...M-1	M+L-2 – m	= L-1 ... L-1	(0, M-1)	(M+L-2,M-1)
n= M+L-1	m=M	...M-1	M+L-1 – m	= L-1 ... L	(0, M)	(M+L-1,M-1)
	LB(m)	UB(m)			max(0, n+1-L)	min(n, M-1)

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$

m = max(0, n-L+1) .. min(n, M-1)

Valid index set example

Case A



$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{L}{x[m]} * \overset{M}{h[n-m]};$$

n=0 m=0
----
n=1 m=0
n=1 m=1
----
n=2 m=0
n=2 m=1
n=2 m=2
----
n=3 m=0
n=3 m=1
n=3 m=2
n=3 m=3
----

n=4 m=0
n=4 m=1
n=4 m=2
n=4 m=3
----
n=5 m=0
n=5 m=1
n=5 m=2
n=5 m=3
----
n=6 m=0
n=6 m=1
n=6 m=2
n=6 m=3
----

n=7 m=0
n=7 m=1
n=7 m=2
n=7 m=3
----
n=8 m=1
n=8 m=2
n=8 m=3
----
n=9 m=2
n=9 m=3
----
n=10 m=3

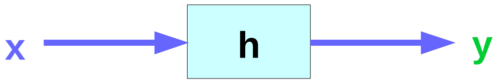
$$\mathbf{m \in [0, L-1]}$$
$$\mathbf{n-m \in [0, M-1]}$$

$$\mathbf{n \in [0, L+M-2]}$$

Index Variable Constraints

Case B

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$



Constraint 1 : **n** ∈ **[0, L+M-2]**

Constraint 2 : **n-m** ∈ **[0, L-1]**

Constraint 3 : **m** ∈ **[0, M-1]**

y[ ] : array with size of **L+M-1**

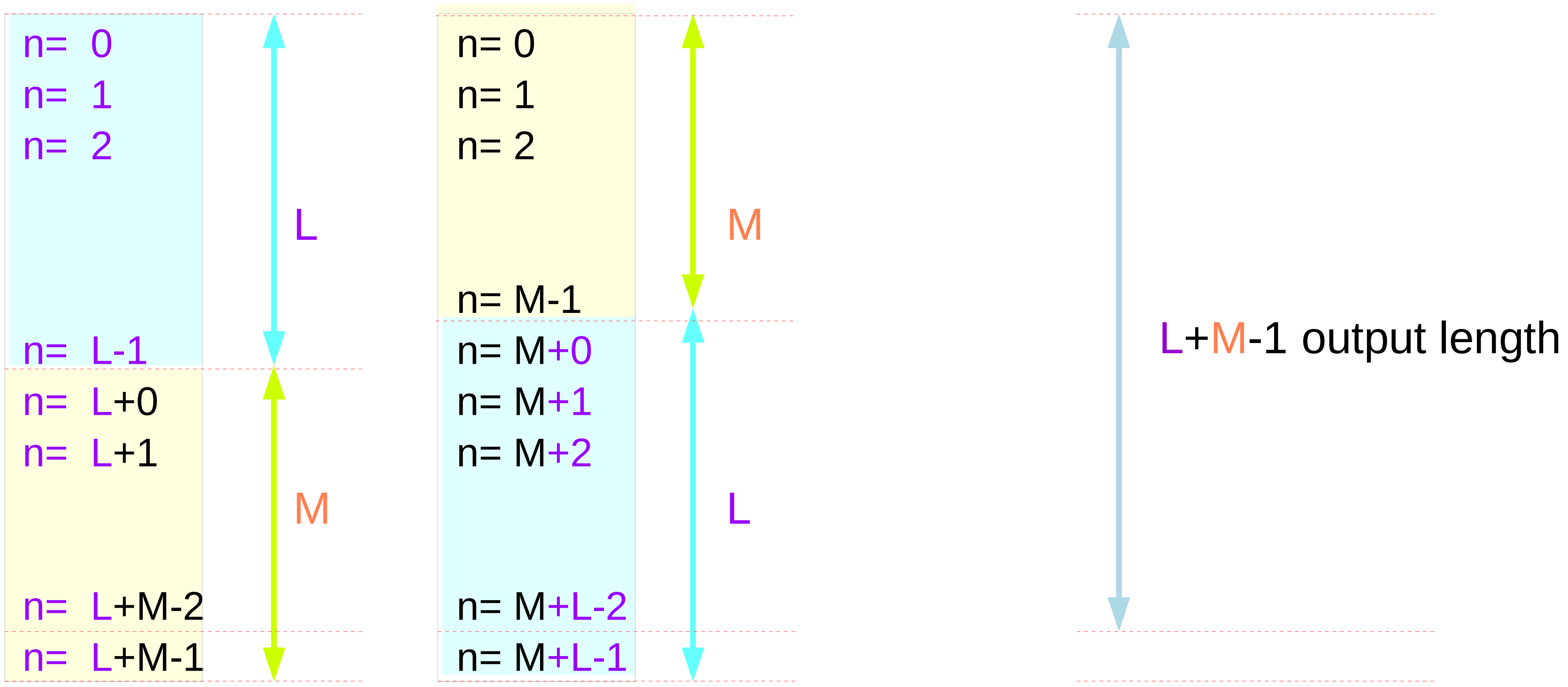
x[ ] : array with size of **L**

h[ ] : array with size of **M**

Constraint 1 – counting **n**

Case B

Constraint 1 :**n ∈ [0, L+M-2]**



Counting 1

Counting 2

M < L is assumed

len(filter) **<** len(input)

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

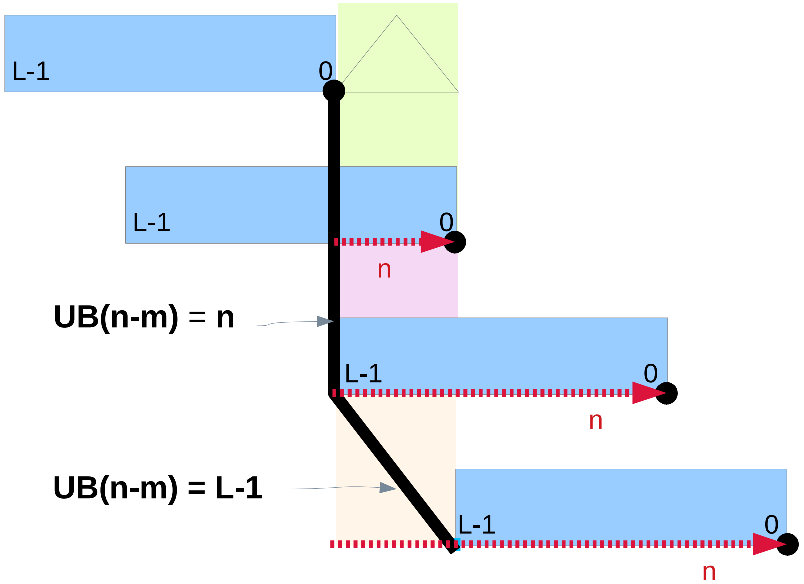
Lower Bound of **m** = max(**0**, n-(L-1))

Case B

UB(n-m)

LB(m) = max(0, n-(L-1))

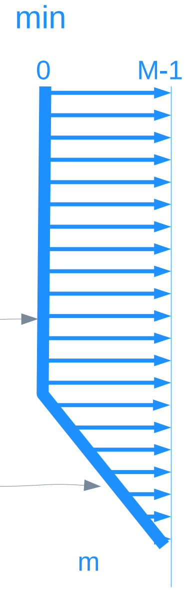
LB(m)



UB(n-m) = n

UB(n-m) = L-1

Floating zero



LB(m) = 0

LB(m) = n-(L-1)

Fixed zero

```
 M+L-1        M          L
y[n] += h[m] * x[n-m];
```

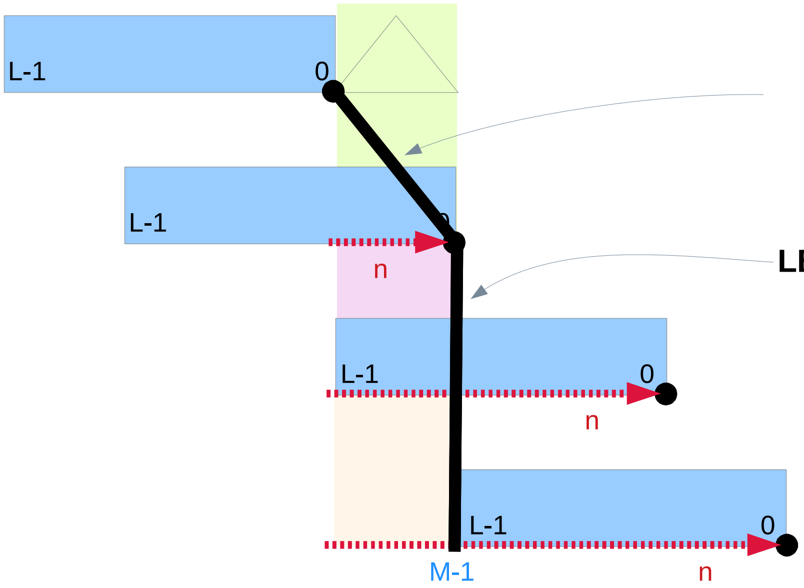
Upper Bound of **m** = min(n, **M-1**)

Case B

LB(n-m)

UB(m) = min(n, M-1)

UB(m)



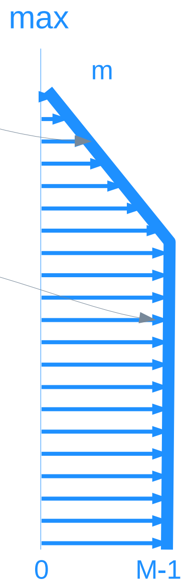
LB(n-m) = 0

LB(n-m) = n-(M-1)

Floating zero

UB(m) = n

UB(m) = M-1



Fixed zero

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

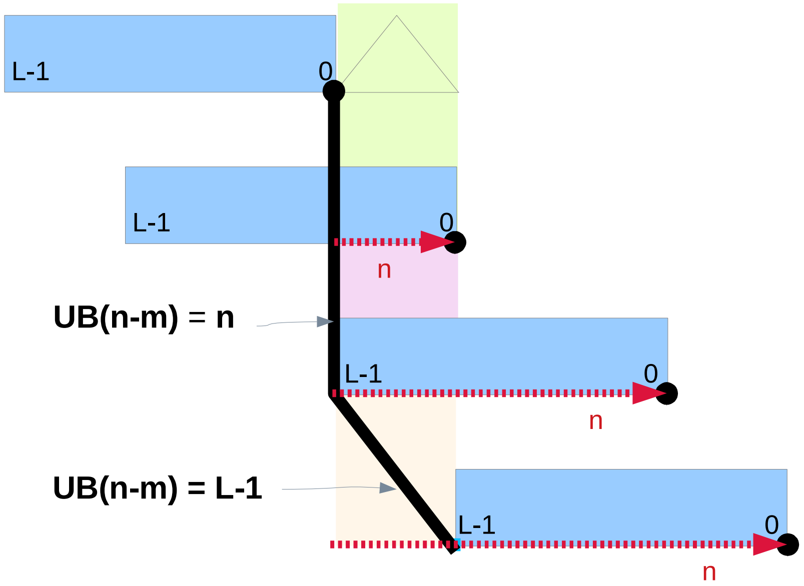
Constraint 1 & 2 – UB(**n-m**) ➡ LB(**m**)

Case B

Constraint 2 : **n-m** ∈ **[0, L-1]** ➡ n - m : **L-1** ... 0 (0, n+1-L)

max min

for **UB(n-m)** values
m should be least possible

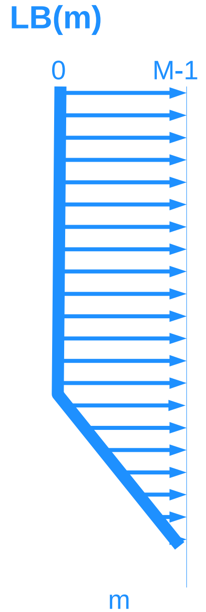


0 ≤ (n-m) ≤ L-1

Case A) n ≤ L-1
➡UB(**n-m**) = **n**
➡LB(**m**) = **0**

Case B) n ≥ L
➡UB(**n-m**) = **L-1**
➡LB(**m**) = **n-(L-1)**

LB(m) = max(0, n-(L-1))



Floating zero

Fixed zero

M+L-1 M L

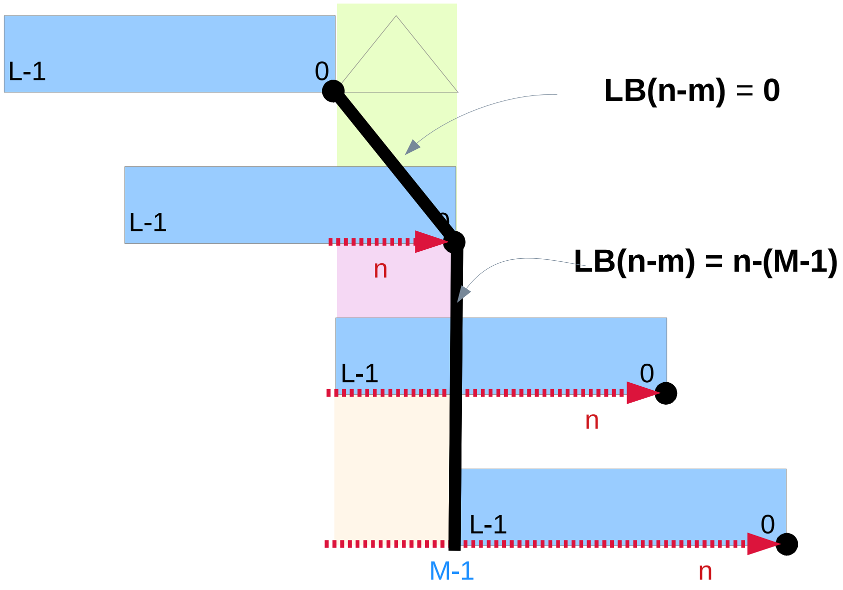
y[n] += h[m] * x[n-m];

Constraint 1 & 3 – UB(**m**) ➡ LB(**n-m**)

Case B

Constrain 3 : **m ∈ [0, M-1]** ➡ n - m : max L-1 ... **min 0** (n, M-1)

for **LB(n-m)** values
m should be greatest possible



LB(n-m) = **0**

LB(n-m) = n-(M-1)

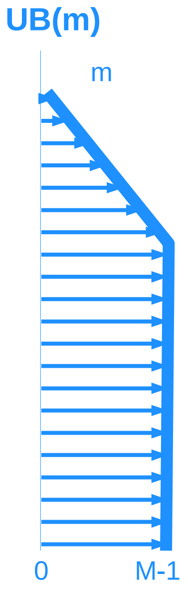
Floating zero

0 ≤ (n-m) ≤ L-1

Case A) n ≤ M-1
➡ UB(**m**) = **n**
➡ LB(**n-m**) = **0**

Case B) n ≥ M
➡ UB(m) = **M-1**
➡ LB(**n-m**) = **n-(M-1)**

UB(m) = min(n, M-1)

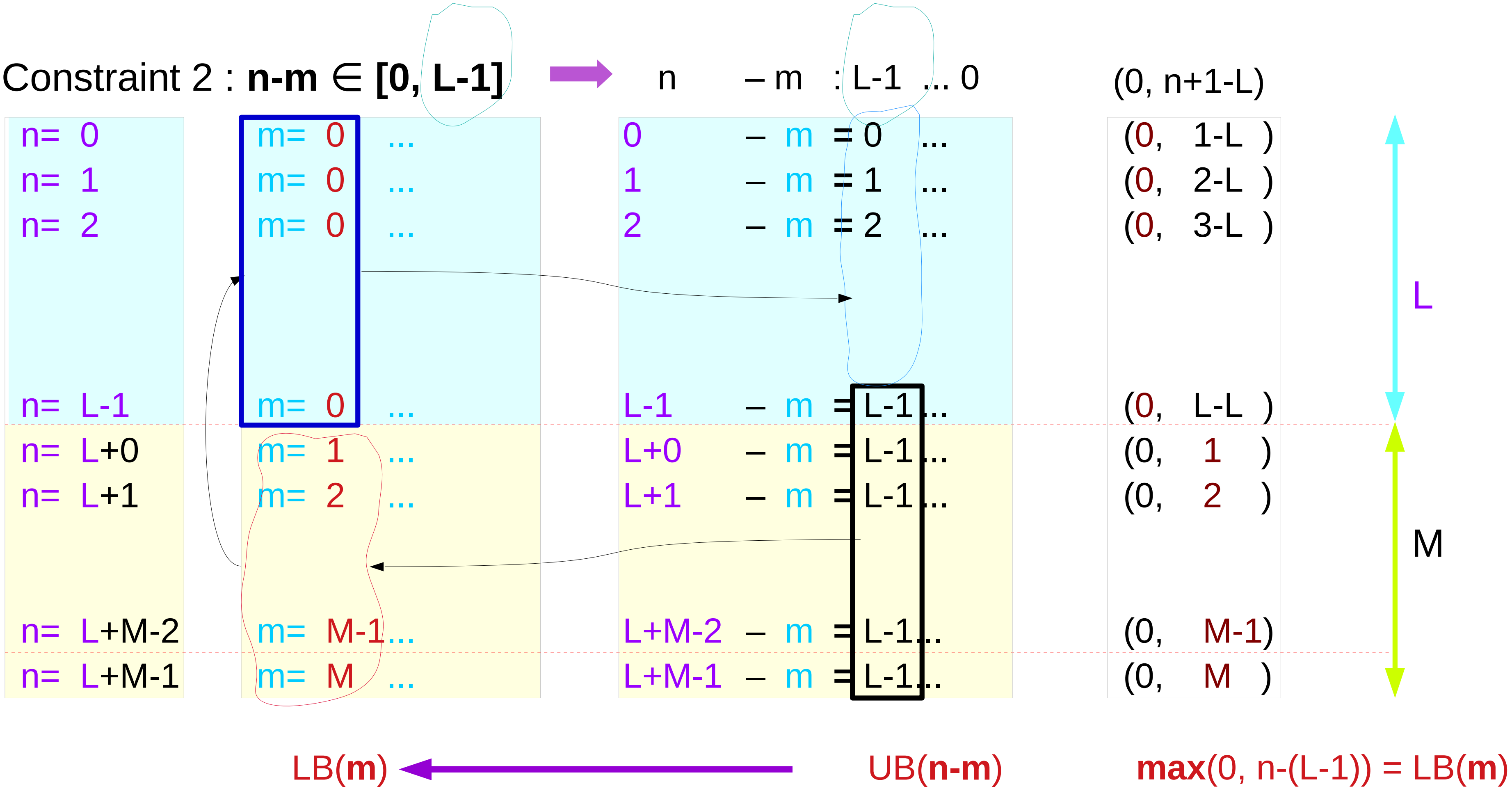


Fixed zero

M+L-1 M L
y[n] += h[m] * x[n-m];

Constraint 1 & 2 – UB(**n-m**) ➡ LB(**m**)

Case B



Constraint 2 : **n-m** ∈ **[0, L-1]** ➡ n – m : L-1 ... 0 (0, n+1-L)

n= 0	m= 0 ...	0 – m = 0 ...	(0, 1-L)
n= 1	m= 0 ...	1 – m = 1 ...	(0, 2-L)
n= 2	m= 0 ...	2 – m = 2 ...	(0, 3-L)
n= L-1	m= 0 ...	L-1 – m = L-1 ...	(0, L-L)
n= L+0	m= 1 ...	L+0 – m = L-1 ...	(0, 1)
n= L+1	m= 2 ...	L+1 – m = L-1 ...	(0, 2)
n= L+M-2	m= M-1 ...	L+M-2 – m = L-1 ...	(0, M-1)
n= L+M-1	m= M ...	L+M-1 – m = L-1 ...	(0, M)

LB(**m**) ← UB(**n-m**)

max(0, n-(L-1)) = LB(**m**)

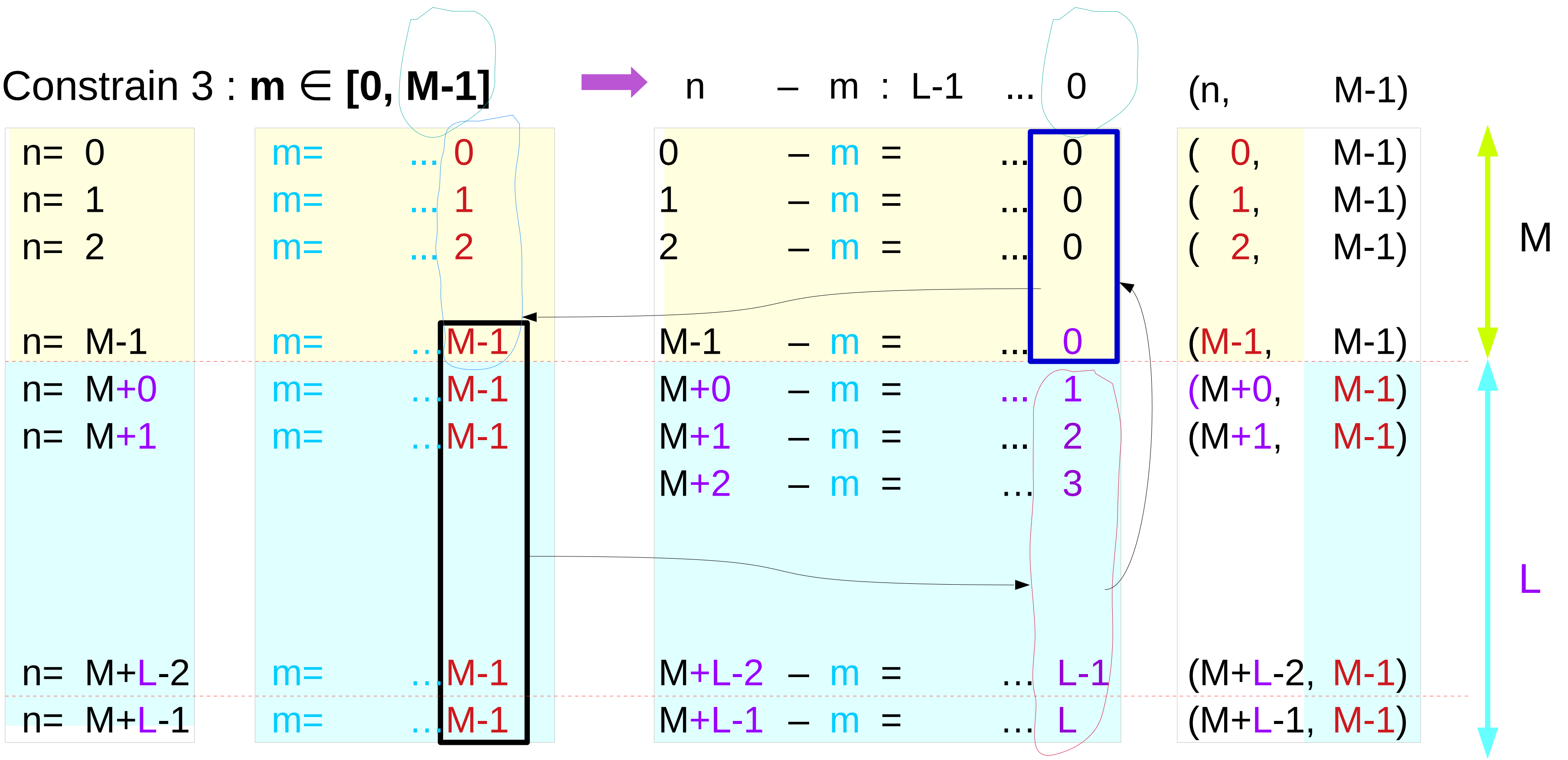
m = **[max**(0, n-(L-1)), **min**(n, M-1)**]**

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

Constraint 1 & 3 – UB(**m**) ➡ LB(**n-m**)

Case B

Constrain 3 : **m** ∈ **[0, M-1]** ➡ n – m : L-1 ... 0 (n, M-1)



n= 0	m= ... 0	0 – m = ... 0	(0, M-1)
n= 1	m= ... 1	1 – m = ... 0	(1, M-1)
n= 2	m= ... 2	2 – m = ... 0	(2, M-1)
n= M-1	m= ... M-1	M-1 – m = ... 0	(M-1, M-1)
n= M+0	m= ... M-1	M+0 – m = ... 1	(M+0, M-1)
n= M+1	m= ... M-1	M+1 – m = ... 2	(M+1, M-1)
		M+2 – m = … 3	
n= M+L-2	m= ... M-1	M+L-2 – m = … L-1	(M+L-2, M-1)
n= M+L-1	m= ... M-1	M+L-1 – m = … L	(M+L-1, M-1)

UB(**m**) ⟶ LB(**n-m**) **min**(n, M-1) = UB(**m**)

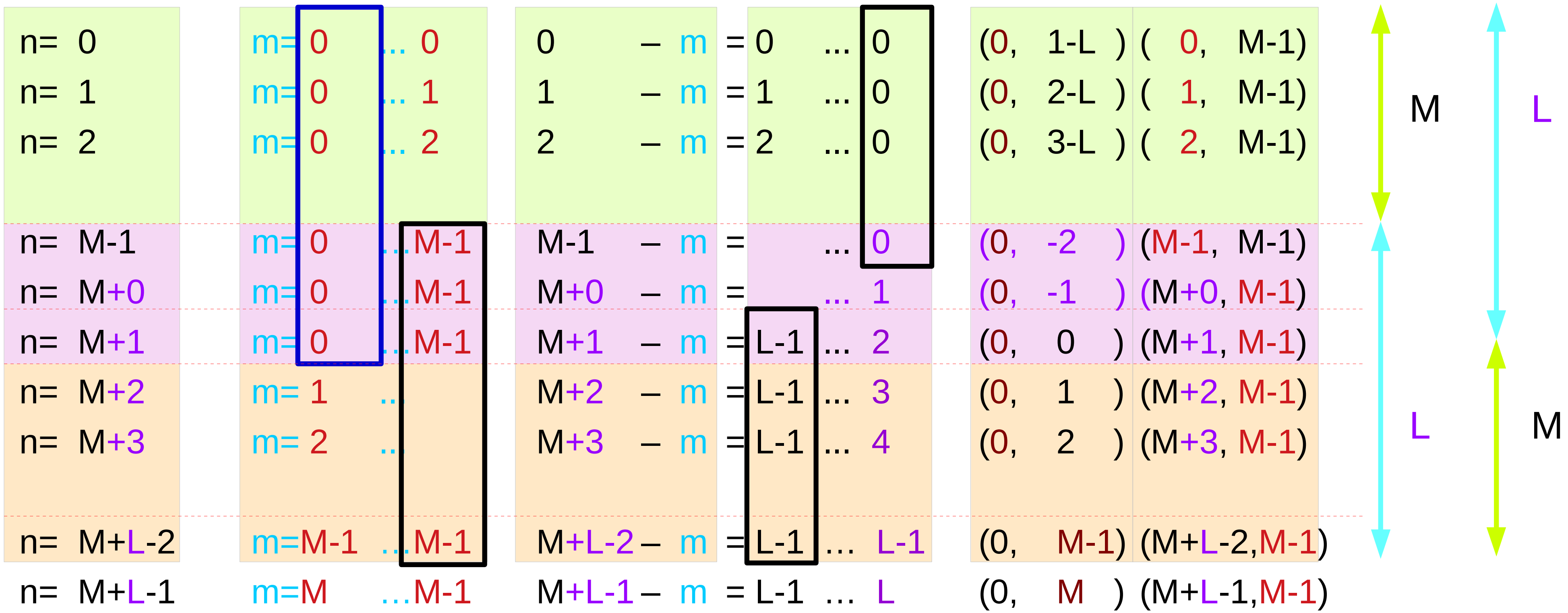
m = **[max**(0, n-(L-1)), **min**(n, M-1)**]**

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

Constraint 1, 2, 3 – max **m** and min **m**

Case B

Constrain 3 : **m ∈ [0, M-1]** ➡ n – m : L-1 ... 0 (0, n+1-L) (n, M-1)



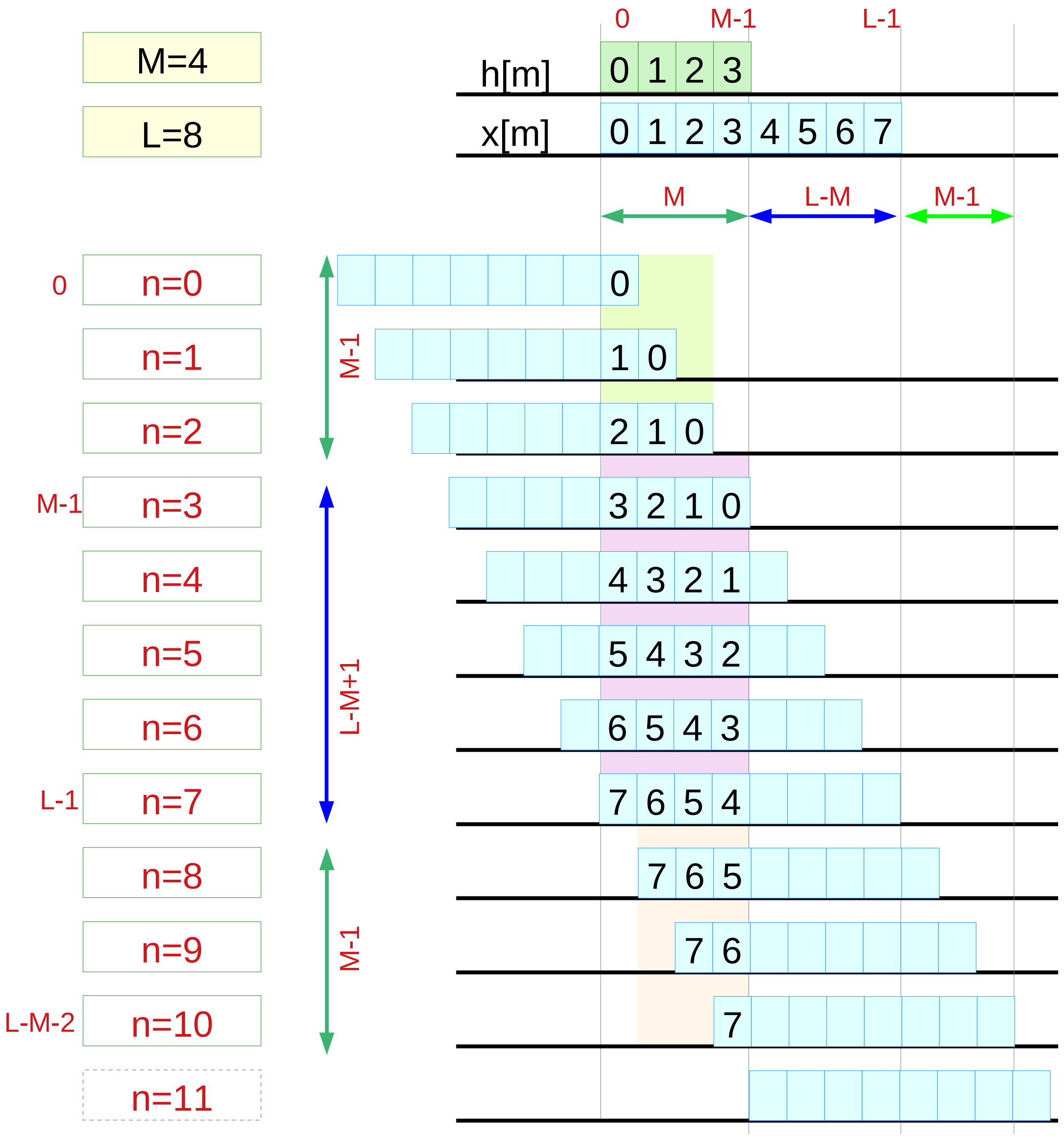
n= 0	m= 0	... 0	0	– m = 0	... 0	(0, 1-L)	(0, M-1)
n= 1	m= 0	... 1	1	– m = 1	... 0	(0, 2-L)	(1, M-1)
n= 2	m= 0	... 2	2	– m = 2	... 0	(0, 3-L)	(2, M-1)
n= M-1	m= 0	...M-1	M-1	– m =	... 0	(0, -2)	(M-1, M-1)
n= M+0	m= 0	...M-1	M+0	– m =	... 1	(0, -1)	(M+0, M-1)
n= M+1	m= 0	...M-1	M+1	– m = L-1	... 2	(0, 0)	(M+1, M-1)
n= M+2	m= 1	..	M+2	– m = L-1	... 3	(0, 1)	(M+2, M-1)
n= M+3	m= 2	..	M+3	– m = L-1	... 4	(0, 2)	(M+3, M-1)
n= M+L-2	m=M-1	...M-1	M+L-2	– m = L-1	... L-1	(0, M-1)	(M+L-2,M-1)
n= M+L-1	m=M	...M-1	M+L-1	– m = L-1	... L	(0, M)	(M+L-1,M-1)
	LB(**m**)	UB(**m**)				max(0, n+1-L)	min(n, M-1)

$$\overset{M+L-1}{y[n]} \mathrel{+}= \overset{M}{h[m]} * \overset{L}{x[n-m]};$$

m = max(0, n-L+1) .. min(n, M-1)

Valid index set example

Case B



M+L-1 M L

y[n] += h[m] * x[n-m];

n=0 m=0
- - - - - - - - - - - - - - - -
n=1 m=0
n=1 m=1
- - - - - - - - - - - - - - - -
n=2 m=0
n=2 m=1
n=2 m=2
- - - - - - - - - - - - - - - -
n=3 m=0
n=3 m=1
n=3 m=2
n=3 m=3
- - - - - - - - - - - - - - - -

n=4 m=0
n=4 m=1
n=4 m=2
n=4 m=3
- - - - - - - - - - - - - - - -
n=5 m=0
n=5 m=1
n=5 m=2
n=5 m=3
- - - - - - - - - - - - - - - -
n=6 m=0
n=6 m=1
n=6 m=2
n=6 m=3
- - - - - - - - - - - - - - - -

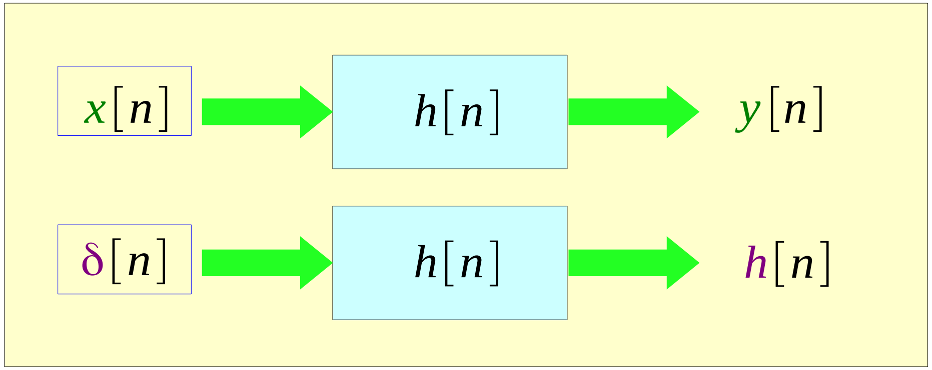
n=7 m=0
n=7 m=1
n=7 m=2
n=7 m=3
- - - - - - - - - - - - - - - -
n=8 m=1
n=8 m=2
n=8 m=3
- - - - - - - - - - - - - - - -
n=9 m=2
n=9 m=3
- - - - - - - - - - - - - - - -
n=10 m=3
- - - - - - - - - - - - - - - -

$\mathbf{m \in [0, M-1]}$
$\mathbf{n-m \in [0, L-1]}$

$\mathbf{n \in [0, L+M-2]}$

Impulse Response

Discrete Time LTI System

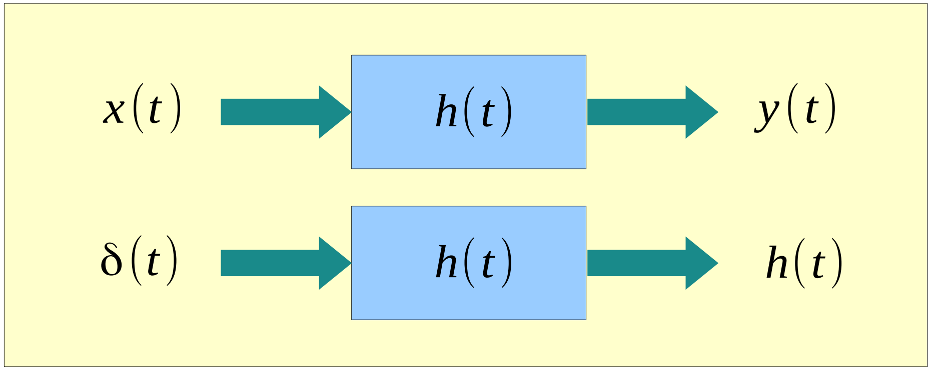


$x[n] \rightarrow \boxed{h[n]} \rightarrow y[n]$

$\delta[n] \rightarrow \boxed{h[n]} \rightarrow h[n]$

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m]$$

Continuous Time LTI System



$x(t) \rightarrow \boxed{h(t)} \rightarrow y(t)$

$\delta(t) \rightarrow \boxed{h(t)} \rightarrow h(t)$

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)\,x(t-\tau)\,d\tau$$

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = \boxed{x[n]}$$

$$a_n h[n] + a_{n-1} h[n-1] + \cdots + a_{n-N} h[n-N] = \boxed{\delta[n]}$$

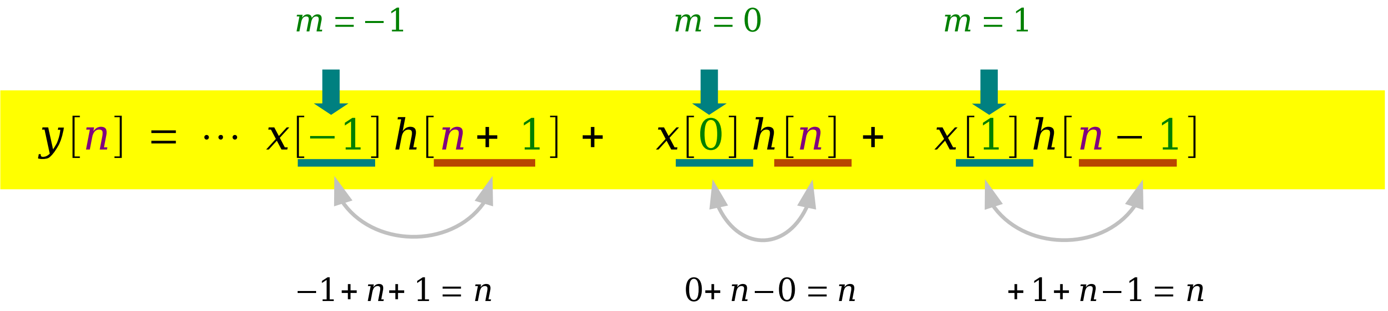
The most general form of
a Discrete Time LTI System

$$h[n] = \frac{1}{a_n}(\delta[n] - a_{n-1} h[n-1] - \cdots - a_{n-N} h[n-N])$$

Convolution Sum

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

$$y[n] = \sum_{\substack{i,j\\ i+j=n}} x[i]\,y[j]$$

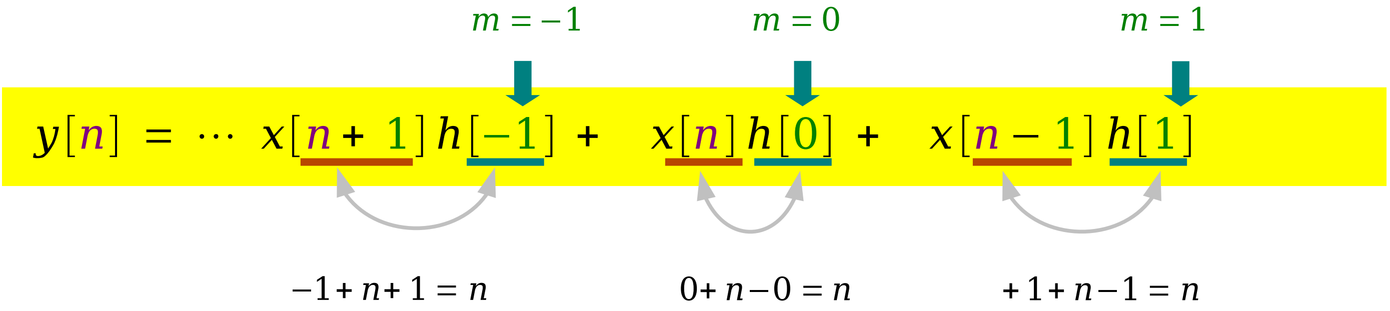


$m = -1$ $m = 0$ $m = 1$

$$y[n] = \cdots\ x[\underline{-1}]\,h[\underline{n+1}] + x[\underline{0}]\,h[\underline{n}] + x[\underline{1}]\,h[\underline{n-1}]$$

$-1 + n + 1 = n$ $0 + n - 0 = n$ $+1 + n - 1 = n$

$$= \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m]$$



$m = -1$ $m = 0$ $m = 1$

$$y[n] = \cdots\ x[\underline{n+1}]\,h[\underline{-1}] + x[\underline{n}]\,h[\underline{0}] + x[\underline{n-1}]\,h[\underline{1}]$$

$-1 + n + 1 = n$ $0 + n - 0 = n$ $+1 + n - 1 = n$

$$= \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

Difference Equation

$$a_n \underline{y[n]} + \underbrace{a_{n-1} \underline{y[n-1]} + \cdots + a_{n-N} \underline{y[n-N]}} = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M]$$

present
output

past outputs
Feedback
recursive

$$a_n y[n] = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M]$$

$$a_i = 0 \quad for\ all\ i$$

Non-recursive
Finite Impulse Response (FIR) filter

$$a_n y[n] = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M] - a_{n-1} y[n-1] - \cdots - a_{n-N} y[n-N]$$

$$a_i \neq 0 \quad for\ some\ i$$

Recursive
Infinite Impulse Response (IIR) filter

Infinite Impulse Response (IIR)

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M]$$

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = b_n x[n]$$ ➡ $b_n h[n]$

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = b_{n-1} x[n-1]$$ ➡ $b_{n-1} h[n-1]$

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = \cdots$$

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = b_{n-M} x[n-M]$$ ➡ $b_{n-M} h[n-M]$

$$h_{all}[n] = b_n h[n] - b_{n-1} h[n-1] + \cdots + b_{n-N} h[n-N]$$

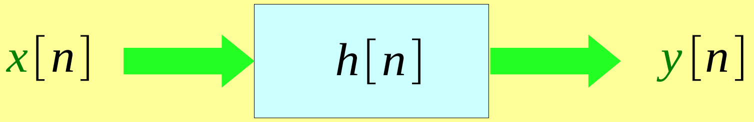
$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = x[n]$$

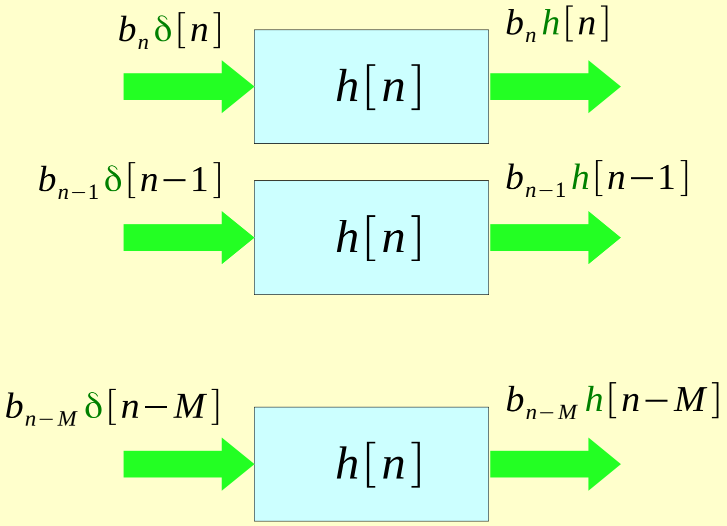
$$h[n] = \frac{1}{a_n}\left(\delta[n] - a_{n-1} h[n-1] - \cdots - a_{n-N} h[n-N]\right)$$

IIR and a Superposition of Inputs

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = \boxed{x[n]}$$

$$h[n] = \frac{1}{a_n}(\delta[n] - a_{n-1} h[n-1] - \cdots - a_{n-N} h[n-N])$$





$$a_n h[n] + a_{n-1} h[n-1] + \cdots + a_{n-N} h[n-N] = \boxed{b_n x[n]}$$

$$a_n h[n] + a_{n-1} h[n-1] + \cdots + a_{n-N} h[n-N] = \boxed{b_{n-1} x[n-1]}$$

$$a_n h[n] + a_{n-1} h[n-1] + \cdots + a_{n-N} h[n-N] = \boxed{b_{n-M} x[n-M]}$$

IIR as an Sum of All Impulse Responses

$$a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = \boxed{b_n x[n]} + \boxed{b_{n-1} x[n-1]} + \cdots + \boxed{b_{n-M} x[n-M]}$$

$$x_{all}[n] \longrightarrow \boxed{h_{all}[n]} \longrightarrow y[n]$$

$$x_{all}[n] = \boxed{b_n x[n]} + \boxed{b_{n-1} x[n-1]} + \cdots + \boxed{b_{n-M} x[n-M]}$$

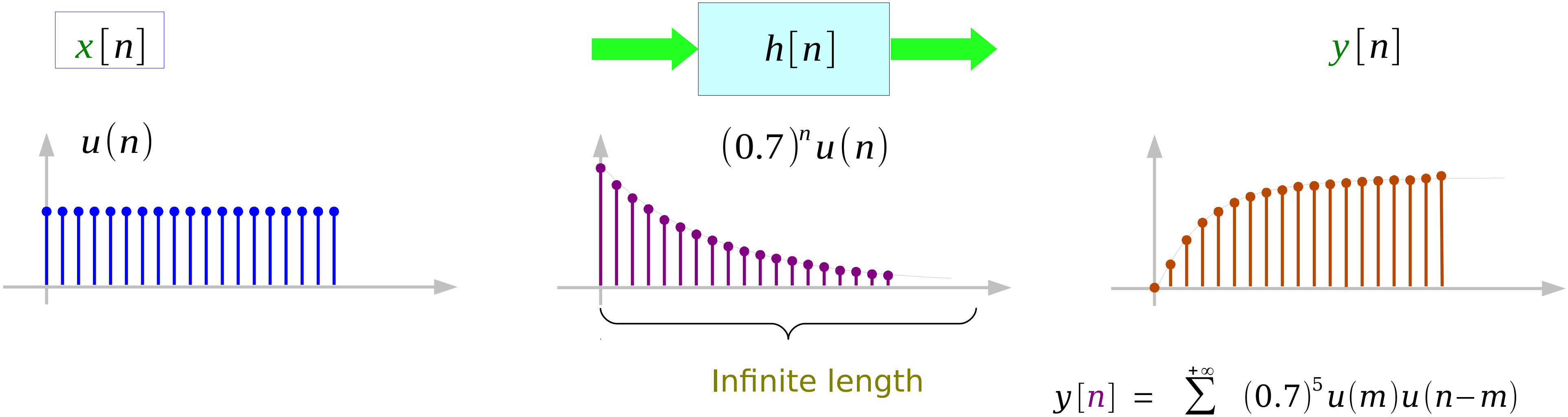
$$h_{all}[n] = b_n h[n] - b_{n-1} h[n-1] + \cdots + b_{n-N} h[n-N]$$

$$\left.\begin{aligned} & a_n y[n] + a_{n-1} y[n-1] + \cdots + a_{n-N} y[n-N] = \boxed{x[n]} \\ & h[n] = \frac{1}{a_n}(\delta[n] - a_{n-1} h[n-1] - \cdots - a_{n-N} h[n-N]) \end{aligned}\right\}$$

IIR Example

Discrete Time LTI System

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\, h[n-m]$$



$$y[n] = \sum_{m=-\infty}^{+\infty} (0.7)^5 u(m) u(n-m)$$

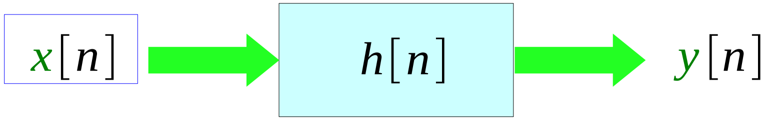
$$= \sum_{m=0}^{n} (0.7)^5 = \frac{1-(0.7)^{n+1}}{1-0.7}$$

$$= \frac{10}{3}\left(1-(0.7)^{n+1}\right)$$

Discrete Time Exponential γⁿ

Discrete Time LTI System

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\, h[n-m]$$



$x[n]$

$h[n]$

$y[n]$

$$e^{\lambda t} = \gamma^{t}$$

$$\lambda = \ln \gamma$$

$$\gamma = e^{\lambda}$$

$$\lambda = -0.3$$

$$\gamma = e^{-0.3} = 0.7408$$

$$e^{-0.3t} = 0.7408^{t}$$

$$\gamma = e^{\lambda}$$

$$\lambda = \ln \gamma$$

$$\gamma = 4$$

$$\lambda = \ln 4 = 1.386$$

$$e^{1.386t} = 4^{t}$$

$$e^{\lambda n} = \gamma^{n}$$

Finite Impulse Response (FIR)

$$a_n y[n] = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M]$$

Tapped Delay Line

Transversal Filter

$$a_n h[n] = b_n \delta[n] + b_{n-1} \delta[n-1] + \cdots + b_{n-M} \delta[n-M]$$

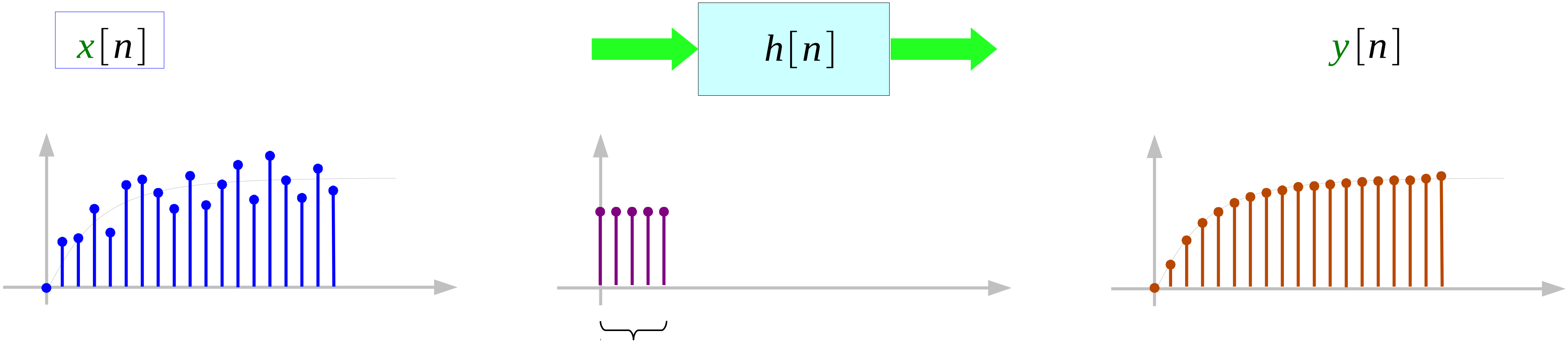
$$h[k] = \begin{cases} 0 & (k \le 0) \\ b_k / a_n & (0 \le k \le M) \\ 0 & (k > M) \end{cases}$$

FIR Example

Discrete Time LTI System

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m]$$

moving average filter



finite length

Convolution Sums in FIR Systems

$$a_n y[n] = b_n x[n] + b_{n-1} x[n-1] + \cdots + b_{n-M} x[n-M]$$

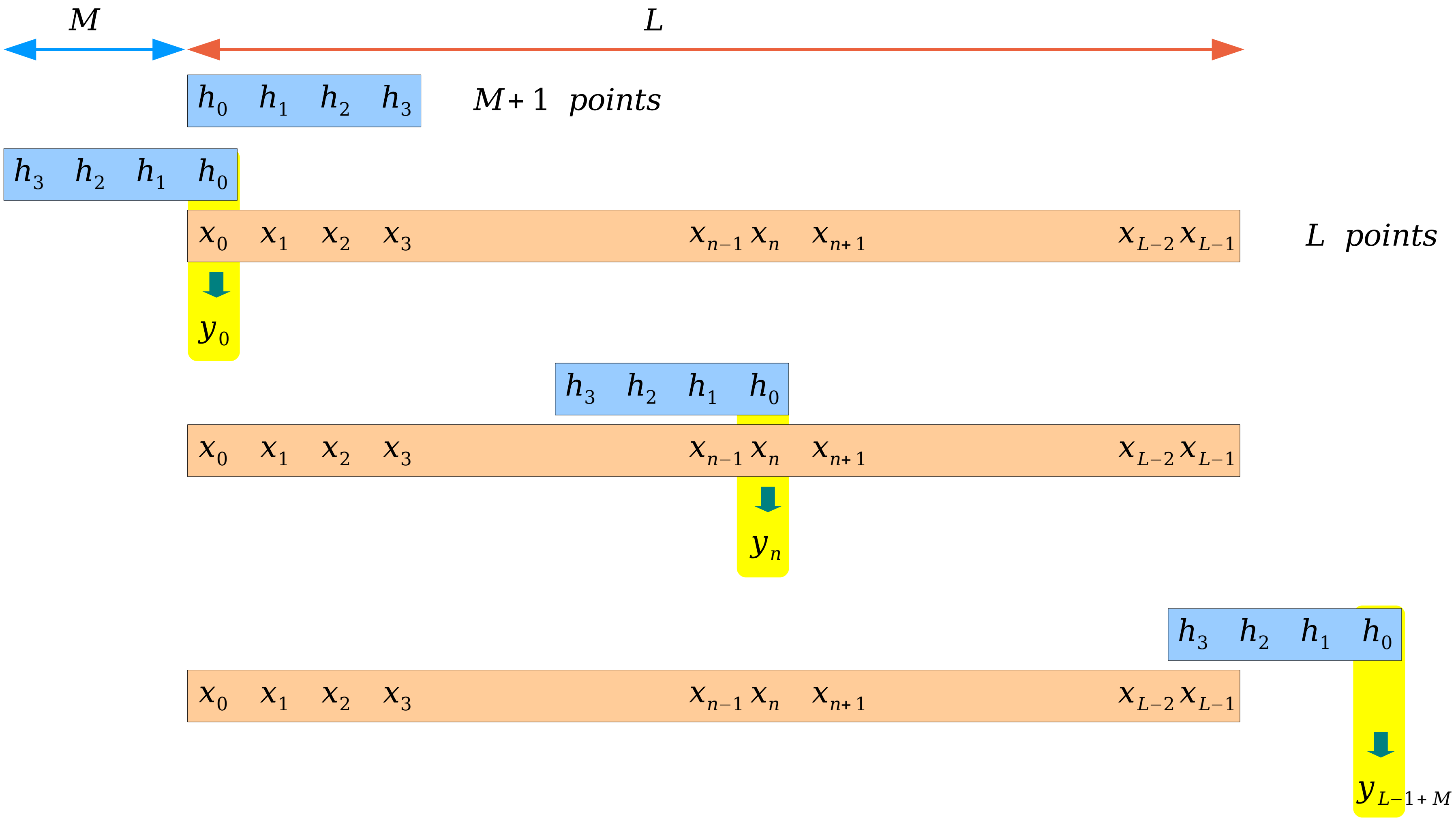
Computing Convolution Sums of an FIR

- Flip and Slide Form
- LTI Form
- Convolution Table
- Direct Form

Flip and Slide Form

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

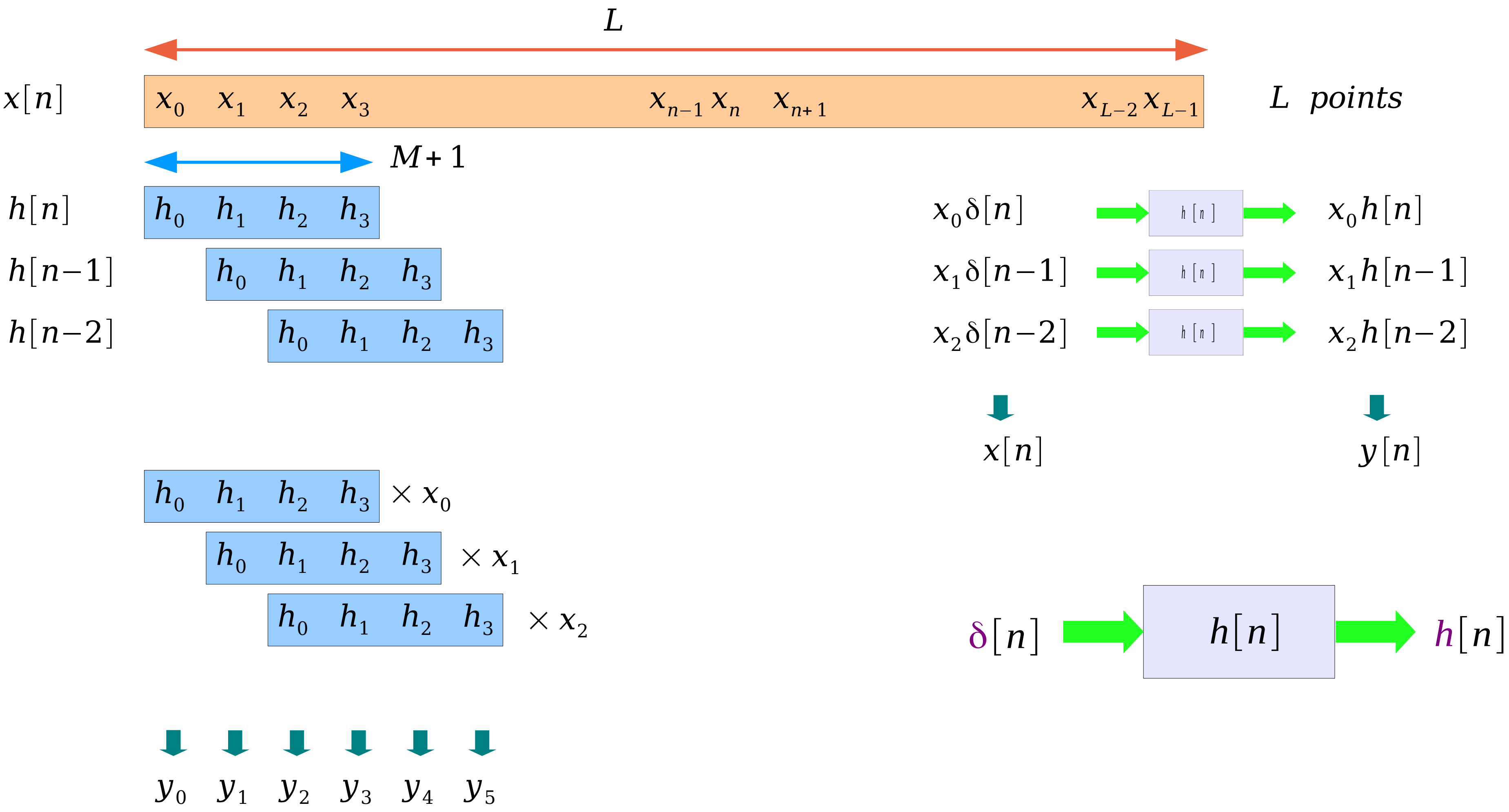
$$y[n] = \sum_{\substack{i,j \\ i+j=n}} x[i]\,y[j]$$



LTI Form

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

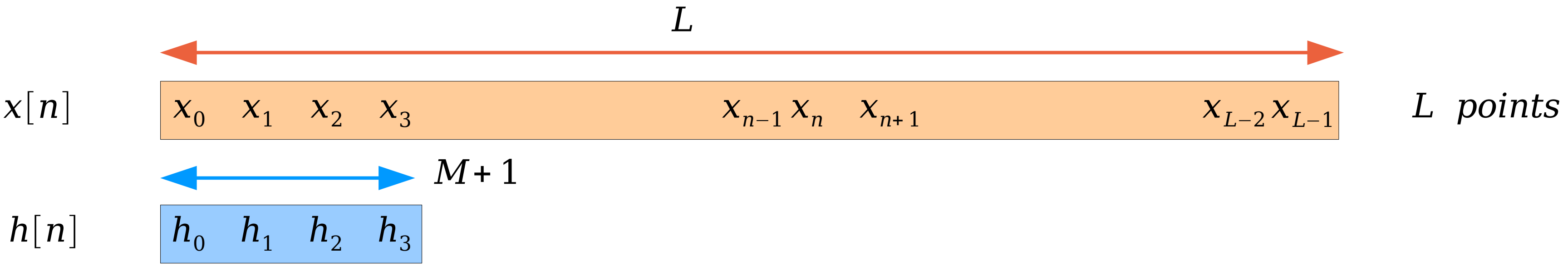
$$y[n] = \sum_{\substack{i,j\\ i+j=n}} x[i]\,y[j]$$



Convolution Table

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] \qquad = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

$$y[n] = \sum_{\substack{i,j \\ i+j=n}} x[i]\,y[j]$$



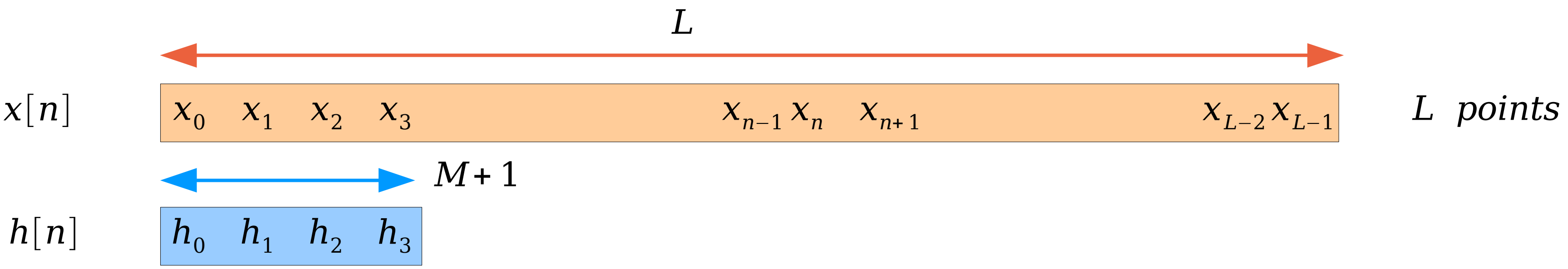
	x_0	x_1	x_3	x_4	x_5	x_6
h_0	$h_0 x_0$	$h_0 x_1$	$h_0 x_3$	$h_0 x_4$	$h_0 x_5$	$h_0 x_6$
h_1	$h_1 x_0$	$h_1 x_1$	$h_1 x_3$	$h_1 x_4$	$h_1 x_5$	$h_1 x_6$
h_2	$h_2 x_0$	$h_2 x_1$	$h_2 x_3$	$h_2 x_4$	$h_2 x_5$	$h_2 x_6$
h_3	$h_3 x_0$	$h_3 x_1$	$h_3 x_3$	$h_3 x_4$	$h_3 x_5$	$h_3 x_6$

$$y[n] = \sum_{\substack{i,j \\ i+j=n}} x[i]\,y[j]$$

Direct Form

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] \qquad = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m]$$

$$y[n] = \sum_{\substack{i,j \\ i+j=n}} x[i]\,y[j]$$



$$y[n] = \sum_{m=-\infty}^{+\infty} x[n-m]\,h[m] \qquad\qquad y[n] = \sum_{m=max[n-(L-1),0]}^{min[n,M]} x[n-m]\,h[m]$$

$$0 \le m \le M \qquad\qquad 0 \le m \le M$$

$$0 \le n-m \le L-1 \qquad\qquad -(L-1) \le m-n \le 0$$

$$m \le n \le L-1+m \qquad\qquad n-(L-1) \le m \le n$$

$$\Big\downarrow \qquad\qquad\qquad \Big\downarrow$$

$$0 \le n \le L-1+M \qquad\qquad max[n-(L-1),0] \le m \le min[n,M]$$

Convolution Property

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{+\infty} x[m]\, h[n-m]$$

$$y[n] = \delta[n] * h[n] = \sum_{m=-\infty}^{+\infty} \delta[m]\, h[n-m]$$

$$h[n] = \delta[n] * h[n]$$

$$x[n] * A\delta[n-n_0] = A\, x[n-n_0]$$

Frequency Response

$$y(t) = h(t) * x(t) \qquad = \int_{-\infty}^{+\infty} h(\tau)\, x(t-\tau)\, d\tau$$

$$x(t) = A\, e^{j\Phi}\, e^{j\omega t} \longrightarrow \boxed{h(t)} \longrightarrow y(t) = H(jw) \cdot A\, e^{j\Phi}\, e^{j\omega t}$$

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)\, A\, e^{j\Phi}\, e^{jw(t-\tau)}\, d\tau$$

$$= \int_{-\infty}^{+\infty} h(\tau)\, A\, e^{j\Phi}\, e^{jwt}\, e^{-j\omega\tau}\, d\tau$$

$$= \underline{A\, e^{j\Phi}\, e^{jwt} \cdot} \; \underline{\int_{-\infty}^{+\infty} h(\tau)\, e^{-j\omega\tau}\, d\tau}$$

$$= \underline{x(t)} \quad \cdot \quad \underline{H(jw)}$$

Direct Form
Convolution Table
LTI Form
Matrix Form
Flip-and-side form
Overlap-Add Block Convolution

Block Processing Method
Sample Processing Method

Orfanidis intro to signal processing

References

[1] http://en.wikipedia.org/
[2] J.H. McClellan, et al., Signal Processing First, Pearson Prentice Hall, 2003
[3] M. J. Roberts, Fundamentals of Signals and Systems
[4] S. J. Orfanidis, Introduction to Signal Processing
[5] B. P. Lathi, Signals and Systems