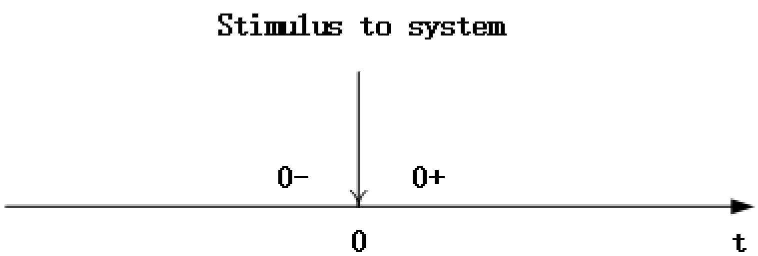
A LTI system described by a linear constant coefficient differential equation:

$$\begin{cases} y''(t) + 3y'(t) + 2y(t) = f'(t) + 3f(t) \\ f(t) = e^{-t}u(t) \\ y(0_-) = 1, y'(0_-) = 2 \end{cases}$$



zero input response: y_{zi}

$$\begin{cases} y''(t) + 3y'(t) + 2y(t) = f'(t) + 3f(t) \\ f(t) = 0 \\ y(0_-) = 1, y'(0_-) = 2 \end{cases} \Longrightarrow \begin{cases} y''(t) + 3y'(t) + 2y(t) = 0 \\ y(0_-) = 1, y'(0_-) = 2 \end{cases}$$

zero state response: $y_{\mathrm{zs}} = y_h + y_p$

$$\begin{cases} y''(t) + 3y'(t) + 2y(t) = f'(t) + 3f(t) \\ f(t) = e^{-t}u(t) \\ y(0_-) = 0, y'(0_-) = 0 \Rightarrow y(0_+) = ?, y'(0_+) = ? \end{cases}$$

About 0_- and 0_+

$$\begin{cases} y''(t) + 3(t) + 2y(t) = e^{-t}\delta(t) + 2e^{-t}u(t) \\ y'(0_-) = y(0_-) = 0 \end{cases}$$

$$\int_{0_-}^{0_+} y''(t)\mathrm{dt} + 3 \cdot \int_{0_-}^{0_+} y'(t)\mathrm{dt} + 2 \cdot \int_{0_-}^{0_+} y(t)\mathrm{dt} = \int_{0_-}^{0_+} e^0\delta(t)\mathrm{dt} + 2 \cdot \int_{0_-}^{0_+} e^{-t}u(t)\mathrm{dt}$$

$$[y'(0_+) - y'(0_-)] + 3[y(0_+) - y(0_-)] + 0 = 1 + 0$$

$$\begin{cases} y(0_+) = y(0_-) = 0 \\ y'(0_+) - y'(0_-) = 1 \end{cases} \Rightarrow \begin{cases} y(0_+) = y(0_-) \\ y'(0_+) = y'(0_-) + 1 = 1 \end{cases}$$

full response: yfull

way 1:

$y_{\mathrm{ful}} = y_{\mathrm{zi}} + y_{\mathrm{zs}}$

way 2:

$$\begin{cases} y''(t)+3y'(t)+2y(t)=f'(t)+3f(t) \\ f(t)=e^{-t}u(t) \\ y(0_-)=1, y'(0_-)=2 \end{cases}$$

Question 1: LTI system definition

$y_{\text{ful}}=y_{\text{zi}}+y_{\text{zs}}$ Dose it match the nature of the LTI system? （当 yzi 存在时，系统在没有输入的情况下已经有响应了，不是不符合线性性吗？）

LTI 系统的定义：0 初始状态下，满足线性时不变性的系统。

在实际应用中，LTI 系统也需要应对非 0 初始状态。两种状态下对系统响应的求解存在差异。

Question 2: Initial relaxation condition (理论课中初始松弛条件的情况):

$$\begin{cases} y''(t)+3y'(t)+2y(t)=f'(t)+3f(t) \\ f(t)=e^{-t}u(t) \\ y(0_-)=0, y'(0_-)=0 \end{cases}$$

研究系统性质时采用初始松弛条件，解决实际问题时需要将系统的初始状态及输入都考虑在内，即系统的全解应包含零输入部分和零状态部分。

Zero-Input Response（Symbolic method）

$y''(t)+3y'(t)+2y(t)=f(t),\ f(t)=e^{-t}u(t),\ y(0_-)=1, y'(0_-)=2$

Find the zero_input response:

1. Represent differentiation by using the **diff** function
2. Specify a differential equation by using ==
3. use simplify to **simplify** the uncertainty model

```
clear
syms y(t)
D2y = diff(y,t,2);
Dy = diff(y,t);
eqn = D2y+3*Dy+2*y==0;
conds = [y(0)==1, Dy(0)==2];
ysol = dsolve(eqn, conds);
yzi = simplify(ysol)
```

Zero-State Response （Symbolic method）

$y''(t)+3y'(t)+2y(t)=f(t),\ f(t)=e^{-t}u(t),\ y(0_-)=1, y'(0_-)=2$

Find the zero_state response:

1. Represent differentiation by using the **diff** function
2. Specify a differential equation by using ==
3. $y(0_-) = 0,\ y'(0_-) = 0$, so $y(0_+) = 0,\ y'(0_+) = 0$

```
clear
syms y(t)
D2y = diff(y,t,2);
Dy = diff(y,t);
eqn = D2y+3*Dy+2*y==exp(-t)*heaviside(t);
conds = [y(0)==0, Dy(0)==0];
yzs = dsolve(eqn, conds);
simplify(yzs)
fplot(yzs,[0 5])
```

it is the same as the result : $y_{st} = \left(e^{-2t} - \mathrm{e}^{-t} + te^{-t}\right)u(t)$

```
% 教程例题，课堂讲解时请删除
clear;clf;
syms y(t)
D2y = diff(y,t,2);
Dy = diff(y,t);
eqn = D2y-5*Dy+6*y==exp(-2*t)*heaviside(t);
conds = [y(0)==0, Dy(0)==0];
ysol = dsolve(eqn, conds);
yzs = simplify(ysol)
```

Some tips

$y''(t) + 2y'(t) + 3y(t) = f'(t) + 2f(t),\ f(t) = 5\sin 2\pi t$, find the zero-state response

1. When there are multiple equations, enclose them in parentheses, just like multiple conditions
2. When there are multiple equations, you will get multiple results, use dot to refer to the results you need.

```
clear; clf;
syms y(t1) f(t1)
D2y = diff(y,t1,2);
Dy = diff(y,t1);
Df = diff(f,t1);
eqn1 = D2y+2*Dy+3*y==Df+2*f;
eqn2 = f==5*sin(2*pi*t1);
eqns = [eqn1 eqn2];
conds = [y(0)==0, Dy(0)==0];    % the right side of the equation is continuous, so the conditio
ysol = dsolve(eqns, conds)
yzs = simplify(ysol.y);  % when have several eqn, the result will have several output. pick the
fplot(yzs,[0 5]);xlabel('t');ylabel('y(t)');title('Zero-State Response'),grid on;
```

Zero-State Response （Numerical method）

$$\begin{cases} y''(t) + 3y'(t) + 2y(t) = f(t) & f(t) = e^{-t}u(t) \\ y(0_-) = 0; y'(0_-) = 0 & y(0_+) = ?;\ y'(0_+) = ? \end{cases}$$

- When trying to find out the zero-state response, we need to know the state of 0+.
- As we have already mentioned, 0+ comes from 0- and the differential equation.
- In face, for zero-state response, 0- is always 0.
- Therefore, 0+ is actually only related to the differential equation which describes the system.
- So it can be considered that 0+ state only depends on the characteristics of the system.
- Thus the zero-state response depends on the characteristics of the system and the input signal.

In MATLAB, **lsim** is the function to find out the zero-state response in this way.

$$y = \mathrm{lsim}(\mathrm{sys}, f, t)$$
$$\mathrm{sys} = \mathrm{tf}(b, a)$$

$$a_3 y'''(t) + a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_3 f'''(t) + b_2 f''(t) + b_1 f'(t) + b_0 f(t)$$
$$a = [a_3, a_2, a_1, a_0]$$
$$b = [b_3, b_2, b_1, b_0]$$

Note: if the Nth derivative is missing in the differential equation, the corresponding element in the vector should be set to zero.

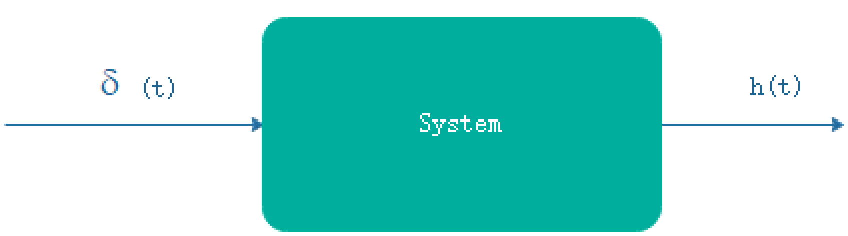
$y''(t) + 3y'(t) + 2y(t) = f(t),\ f(t) = e^{-t}u(t),\ y(0_-) = 1, y'(0_-) = 2$, find the zero-state response

```
clear;clf;
t = 0:0.01:10;
sys = tf(1,[1 3 2]);
f = exp(-t).*heaviside(t);
y=lsim(sys,f,t);
plot(t,y);xlabel('t');ylabel('y(t)');title('Numeric method with lsim'),grid on;
```

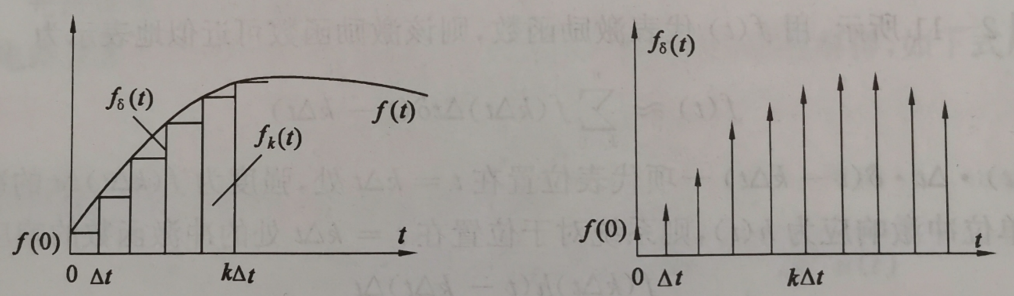
Convolution with Impulse Response (Numerical Method)

We can also find out zero-state response by convolving the impulse response and the input signal.

$$h = \mathrm{impulse}(\mathrm{sys}, t) \text{ or } [h\ \ t] = \mathrm{impulse}(\mathrm{sys})$$
$$g = \mathrm{step}(\mathrm{sys}, t) \text{ or } [g\ t] = \mathrm{step}(\mathrm{sys})$$



$$f_{k+1}(t) \approx f(k\Delta t)\Delta t\delta(t-k\Delta t)$$



(a) 用脉冲信号近似表示　　(b) 用冲激信号近似表示

$$f(t) \approx f_\delta(t) \approx f(0)\Delta t\delta(t) + \cdots + f(k\Delta t)\Delta t\delta(t-k\Delta t) + \cdots$$
$$= \sum_{k=0}^{\infty} f(k\Delta t)\Delta t\delta(t-k\Delta t) \qquad (2\text{—}66)$$

激函数之和对于原来函数 $f(t)$ 近似的程度，也完全取决于时间间隔 Δt 划分

$$\begin{aligned}
\delta(t) &\rightarrow h(t)\\
\delta(t-k\Delta t) &\rightarrow h(t-k\Delta t)\\
f(k\Delta t)\Delta t\cdot\delta(t-k\Delta t) &\rightarrow f(k\Delta t)\Delta t\cdot h(t-k\Delta t)\\
\sum_{k=0}^{n} f(k\Delta t)\Delta t\cdot\delta(t-k\Delta t) &\rightarrow \sum_{k=0}^{n} f(k\Delta t)\Delta t\cdot h(t-k\Delta t)
\end{aligned}$$

$$\sum_{k=0}^{n} f(k\Delta t)\Delta t\cdot h(t-k\Delta t) \xrightarrow{\Delta t\to 0} \int_0^t f(\tau)h(t-\tau)d\tau$$

$y''(t) + 3y'(t) + 2y(t) = f(t),\ f(t) = e^{-t}u(t),\ y(0_-) = 1, y'(0_-) = 2$

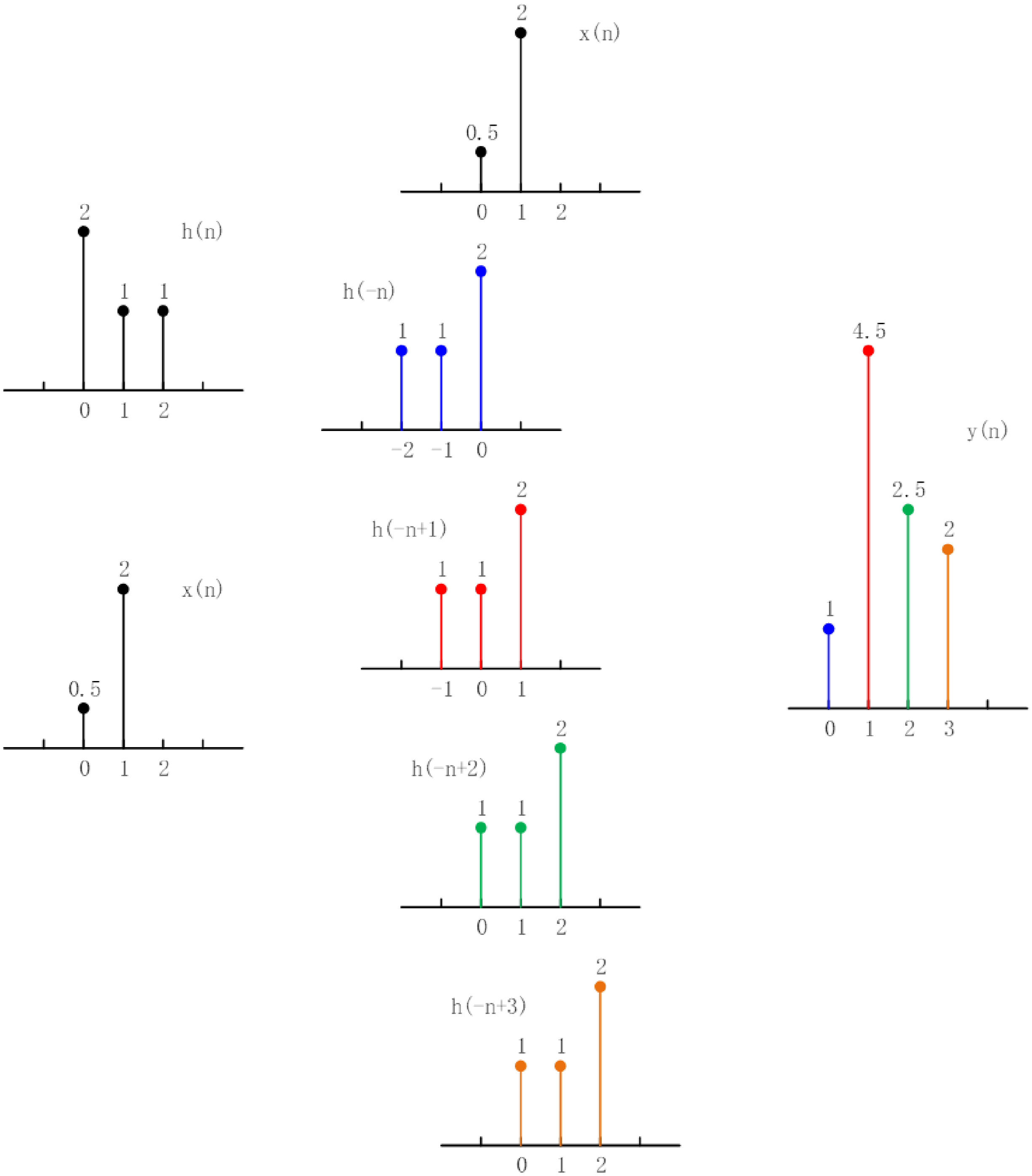
```
clear;clf;
dt = 0.01;
t = 0:0.01:5;
sys = tf(1,[1,3,2]);
h = impulse(sys,t);
f = exp(-t).*heaviside(t);
y = conv(h,f)*dt;   % for continuous method, a dt is required
n = length(y);
tt = (0:n-1)*dt; % the length of the result is changed
```

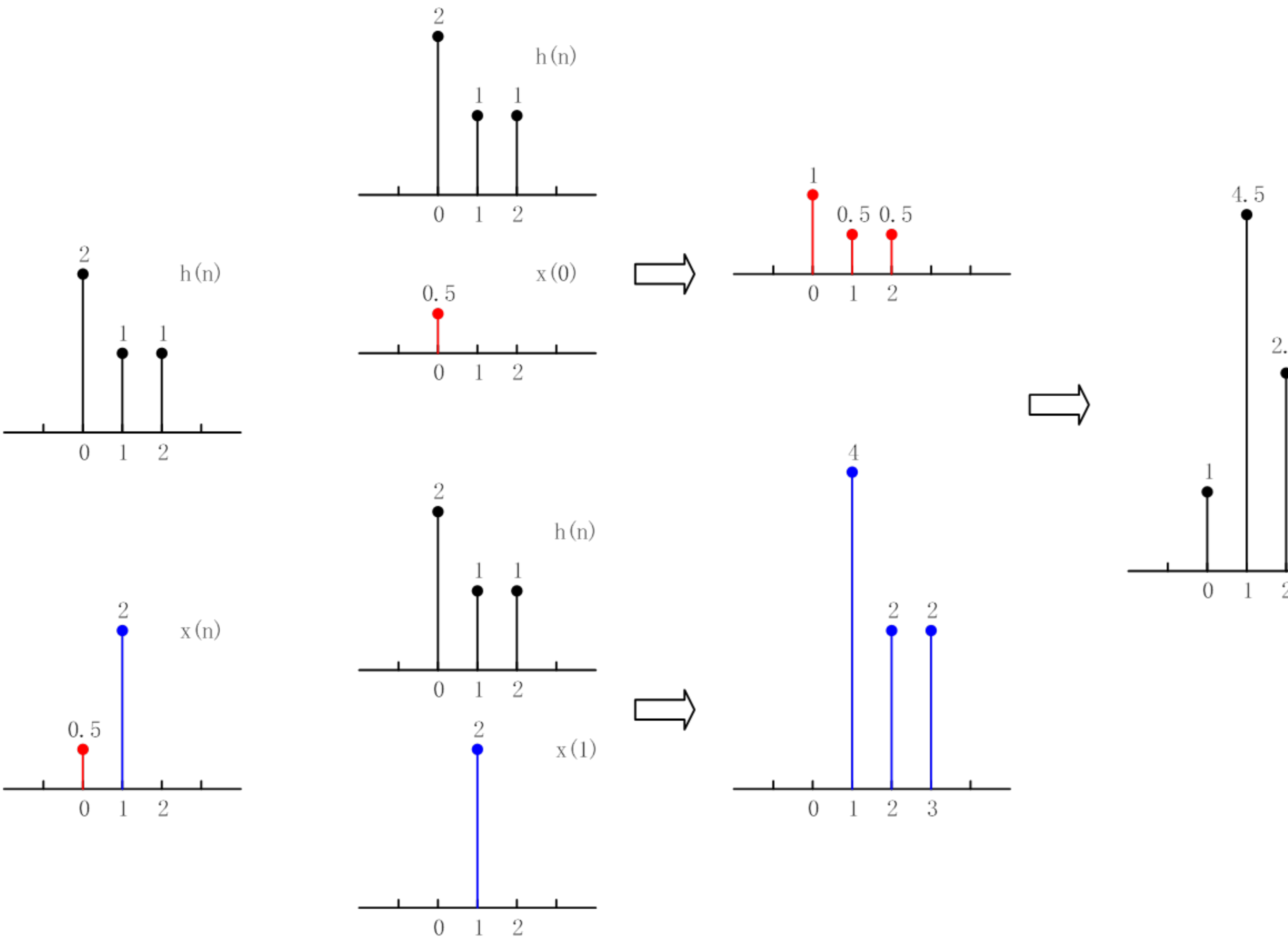
```
subplot(1,2,1);plot(tt,y);xlabel('t');ylabel('y(t)');title('Convolution method'),grid on;

syms y(t1)
D2y = diff(y,t1,2);
Dy = diff(y,t1);
eqn = D2y+3*Dy+2*y==exp(-t1)*heaviside(t1);
conds = [y(0)==0, Dy(0)==0];
ysol = dsolve(eqn, conds);
yzs = subs(ysol,'t1',tt);
subplot(1,2,2);plot(tt,yzs);xlabel('t');ylabel('y(t)');title('Symbolic method'),grid on;
```

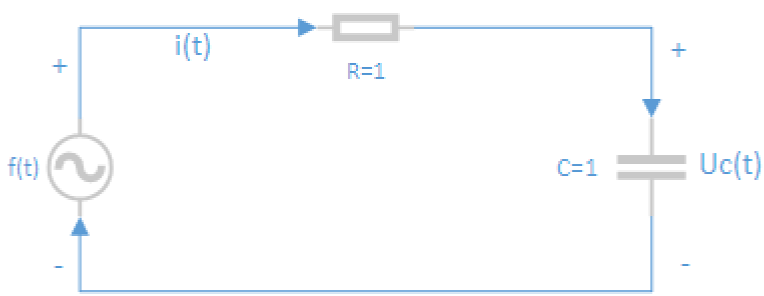
Subs

$y''(t) + 3y'(t) + 2y(t) = f(t),\ f(t) = e^{-t}u(t),\ y(0_-) = 1, y'(0_-) = 2$, find the zero-state response

```
clear;clf;
t = 0:0.01:10;
syms y(t1)
D2y = diff(y,t1,2);
Dy = diff(y,t1);
eqn = D2y+3*Dy+2*y==exp(-t1)*heaviside(t1);
conds = [y(0)==0, Dy(0)==0];
ysol = dsolve(eqn, conds);
yzs = double(subs(ysol,'t1',t));
plot(t,yzs);xlabel('t');ylabel('y(t)');title('Symbolic method'),grid on;
```


x(n)
2
0.5
0 1 2
h(n)
2
1 1
0 1 2
h(−n)
2
1 1
−2 −1 0
h(−n+1)
2
1 1
−1 0 1
x(n)
2
0.5
0 1 2
h(−n+2)
2
1 1
0 1 2
h(−n+3)
2
1 1
0 1 2
y(n)
4.5
2.5
2
1
0 1 2 3





分压原理

$$i = c\frac{du_c(t)}{\mathrm{dt}}$$

$$\mathrm{Rc}\frac{du_c(t)}{\mathrm{dt}} + u_c(t) = f(t)$$