```
dX_t:=piecewise([#t < `&tau;`,#`&sigma;`*dW_t],[#t <= `&tau;`,#`&thetav;`*dt+`&sigma;`*dW_t])
Error: Unexpected 'identifier'. [line 1, col 32]

`&sigma;`:=1
 1

`&sigma;`
 1

a:=`&sigma;`
 1

reset
 reset

a
 1

reset()

a
 a

sigma
 sigma

`&sigma;`=1
 σ = 1

x:=`&sigma;`+5
 σ + 5

reset()

dX_t:=piecewise([#t < `&tau;`,#`&sigma;`dW_t],[#t <= `&tau;`,#`&thetav;`*dt+`&sigma;`*dW_t])
Error: Unexpected 'identifier'. [line 1, col 32]

reset()

x=`&sigma;`+5;
 x = σ + 5

x
 x

x:=

x:=`&sigma;`+1
 σ + 1

x
 σ + 1

fprint(Unquoted, Text, "xmu.m", generate::MATLAB(S)):

x
 σ + 1

S
 S

fprint(Unquoted, Text, "xmu.m", generate::MATLAB(x)):
Warning: Rewriting symbol '`&sigma;`' to 'sigma'.

reset()


dX_t:=piecewise([#t < `&tau;`,#sigma*dW_t],[#t <= tau,#thetav*dt+sigma*dW_t])
```

$$\begin{cases} \#\text{sigma } dW_t & \text{if } \#t < \tau \\ dW_t \text{ sigma} + \#\text{thetav } dt & \text{if } \#t \le \text{tau} \end{cases}$$

```
dX_t:=piecewise([#t < `&tau;`,#sigma*dW_t],[#t <= tau,#thetav*d_t+sigma*dW_t])
```

$$\begin{cases} \#\text{sigma}\ \text{dW}_\text{t} & \text{if } \#\text{t} < \tau \\ \text{dW}_\text{t}\ \text{sigma} + \#\text{thetav}\ d_t & \text{if } \#\text{t} \le \text{tau} \end{cases}$$

```
dX_t:=piecewise([t < tau,#sigma*dW_t],[t <= tau,thetav*d_t+sigma*dW_t])
```

$$\begin{cases} \#\text{sigma}\ \text{dW}_\text{t} & \text{if } t < \text{tau} \\ \text{dW}_\text{t}\ \text{sigma} + d_t\ \text{thetav} & \text{if } t \le \text{tau} \end{cases}$$

```
fprint(Unquoted, Text, "dX_t.m", generate::MATLAB(dX_t)):
reset()
P_theta_gt_t:=(1-p)*exp(-lambda*t)
```

$-\mathrm{e}^{-\text{lambda}\ t}\,(p-1)$

```
fprint(Unquoted, Text, "P_theta_gt_t.m", generate::MATLAB(P_theta_gt_t)):
reset()
R_T:=abs(12)
```

12

```
theta:=9: r_T:=abs(T)-abs(theta)
Error: The identifier 'theta' is protected. [_assign]
reset()

abs(`&tau;`)
```

$|\tau|$

```
R_T:=Est * abs(T-`&tau;`)
```

$\text{Est}\ |\tau - T|$

```
fprint(Unquoted, Text, "R_T.m", generate::MATLAB(R_T)):
Warning: Rewriting symbol '`&tau;`' to 'tau'.
reset()
R_T:=E(`&tau;`)+2*E*int((`&Pi;`-0.5), x=0..T)
```

$\mathrm{e}(\tau) + 2\ T\ \mathrm{e}\ \#(\Pi - 0.5)$

```
int(#f, #x=#a..#b)
```

$-\#\text{f}\ (\#\text{a} - \#\text{b})$

```
reset()
standardScalarProduct := linalg::scalarProduct:
unprotect(linalg):
linalg::scalarProduct := proc(u, v)
    local F, f, t;
begin
    // (0)
    f := expr(u[1] * v[1]);

     // (1)
    t := indets(f);
    if t = {} then t := genident("t") else t := op(t, 1) end_if;

     // (2)
    F := int(f, t = 0..1);

     // (3)
    u::dom::coeffRing::coerce(F)
end:
```