

```
(%i6) alias(W, lambert_w);
```

```
(%o6) [W]
```

```
(%i7) alias(WW, generalized_lambert_w);
```

```
(%o7) [WW]
```

SIR model i-variable

```
(%i15) wkern3(y,g,a):=block([ d:y*(g*log(y)-y+a)],
    if not numberp(y) then 1/d
    elseif abs(d)>1e-16 then 1/d else nan
);
```

```
(%o15) wkern3(y,g,a):=block([d:y(g log(y)-y+a)],if not numberp(y)
    then  $\frac{1}{d}$  elseif  $|d| > 1.0 \cdot 10^{-16}$  then  $\frac{1}{d}$  else nan)
```

```
(%i16) ilambint3(x, g, a):=block([ r, u, ret:'nan, fr:-g, bb, cc, dd],
    if not numberp(x) then return('ilambint3(x,g, a)),
    bb: g*log(g)-g+a,
    if x>0 and x < bb then (
        dd:-g*W(-%e^((x-a)/g)/g),
        ret:fr*first(quad_qags( float(wkern3(u, g, a)) , u, g, dd, 'epsrel=1d-8))
    ) else if x=d then ret:0,
    ret
);
```

```
(%o16) ilambint3(x,g,a):=block
```

```
(%i17) ilambint4(x, g, a):=block([ r, u, ret:'nan, fr:-g, bb, cc, dd],
    if not numberp(x) then return('ilambint4(x,g, a)),
    bb: g*log(g)-g+a,
    if x>0 and x < bb then (
        dd:- (g*WW(-1,-%e^((x-a)/g)/g)),
        ret:fr*first(quad_qags( float(wkern3(u, g, a)) , u, g, dd, 'epsrel=1d-8))
    ) else if x=d then ret:0,
    ret
);
```

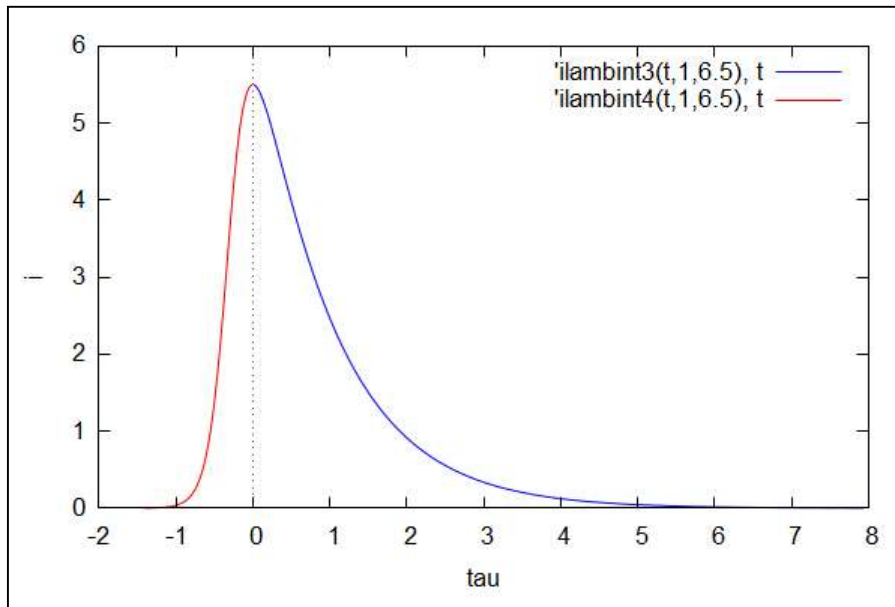
```
(%o17) ilambint4(x,g,a):=block
```

(%i18) wxplot2d([[parametric,ilambint3(t, 1, 6.5), t, [t,0, 9]], [parametric,ilambint4(t, 1, 6.5), t, [t,0, 6]]], [x

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t18)



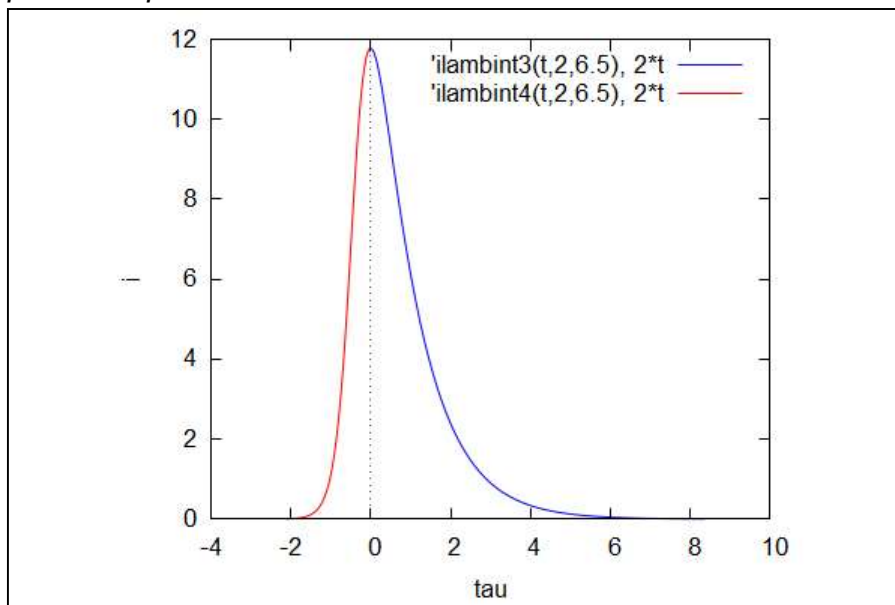
(%o18)

(%i19) wxplot2d([[parametric,ilambint3(t, 2, 6.5), 2*t, [t,0, 9]], [parametric,ilambint4(t, 2, 6.5), 2*t, [t,0, 12]]], [x

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t19)



(%o19)

SIR model s-variable

```
(%i20) ilambint5(x, g, a):=block([ r, u, ret:'nan, fr:-1, bb, cc, dd],
  if not numberp(x) then return('ilambint5(x,g, a)),
  dd:-g*W(-%e^( -a/g)/g), cc: -g*WW(-1,-%e^( -a/g)/g),
  if x>=dd and x<=cc then (
    ret:fr*first(quad_qags( float(wkern3(u, g, a)) , u, g, x, 'epsrel=1d-8))
  ),
  ret
);
```

```
(%o20) ilambint5(x, g, a):=block
```

```
(%i22) wkern6(y,g,q):=block([ d: %e^(-y/g)*((y-q)*%e^(y/g)+q)],
  if not numberp(y) then 1/d
  elseif abs(float(d))>1e-16 then 1/d else 'nan
);
```

```
ilambint6(x, g, a):=block([ r, u, ret:'nan, fr:-1, bb, cc, dd, numer:true],
  if not numberp(x) then return('ilambint6(x,g, a)),
  dd:float(-g*W(-%e^( -a/g)/g)), cc: float(-g*WW(-1,-%e^( -a/g)/g)),
  if x>0 and x <cc-dd then (
    bb:cc-g*log(g)-a,
    ret:fr*first(quad_qags( float(wkern6(u, g, cc)) , u, bb, x,'epsrel=1d-8))
  ),
  ret
);
```

```
(%o21) wkern6(y, g, q):=block([ d:%e $\frac{-y}{g}$  ((y-q) %e $\frac{y}{g}$  +q) ], if not
  numberp(y) then  $\frac{1}{d}$  elseif  $\left| \text{float}(d) \right| > 1.0 \cdot 10^{-16}$  then  $\frac{1}{d}$  else 'nan)
```

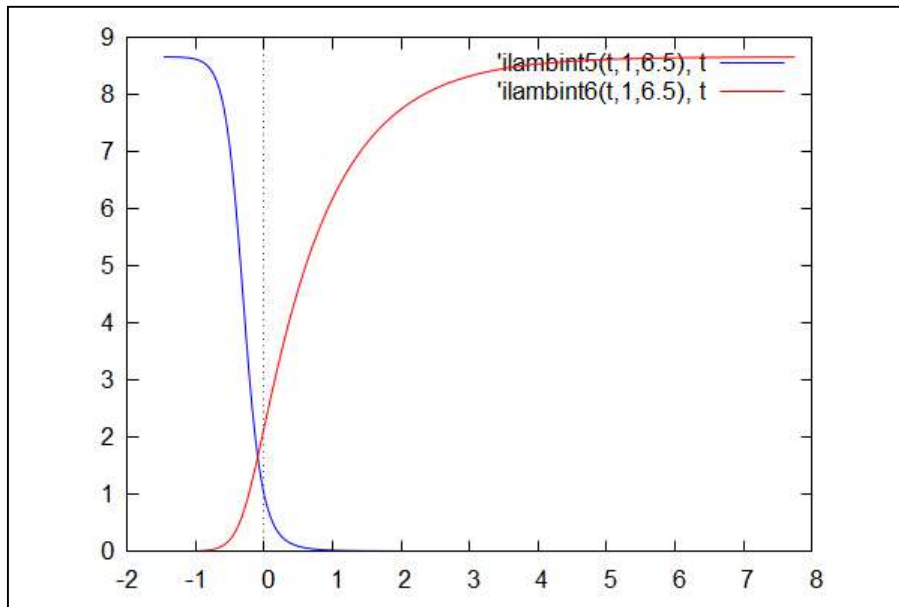
```
(%o22) ilambint6(x, g, a):=block
```

(%i23) wxplot2d([[parametric,ilambint5(t, 1, 6.5), t, [t,0, 12]], [parametric,ilambint6(t, 1, 6.5), t, [t,0, 12]]],

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t23)



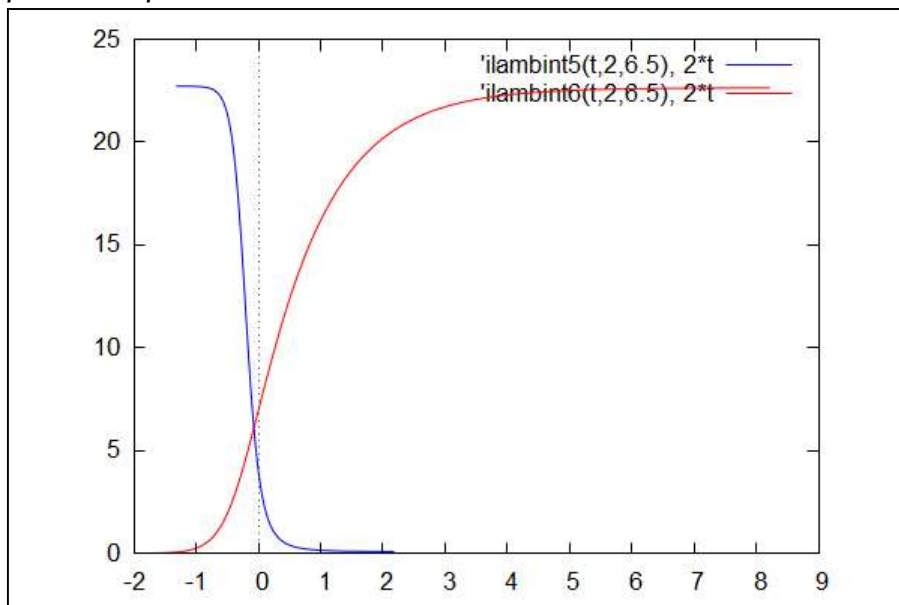
(%o23)

(%i24) wxplot2d([[parametric,ilambint5(t, 2, 6.5), 2*t, [t,0, 12]], [parametric,ilambint6(t, 2, 6.5), 2*t, [t,0, 12]]],

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t24)



(%o24)

→