```julia
using DifferentialEquations    , Plots
```

# Problem 12-7

```julia
# physical data
begin
    Tᵢₙ = 300. # K
    Tₘₐₓ = 400. # K
    Eₐ = 1e4 # cal/mol
    R = 1.987 # cal/mol/K
    k⁰ = 0.01 # L/mol/s
    v₀ = 2. # L/s
    Cₚ = 30 # cal/mol
    ΔH = -6e3 # cal/mol
    Uₐ = 20 # cal/L/s/K
    ṁ₀ = 50 # g/s
end;
```

## (a)

From mass balance of $A + B \rightarrow C$ in a CSTR,

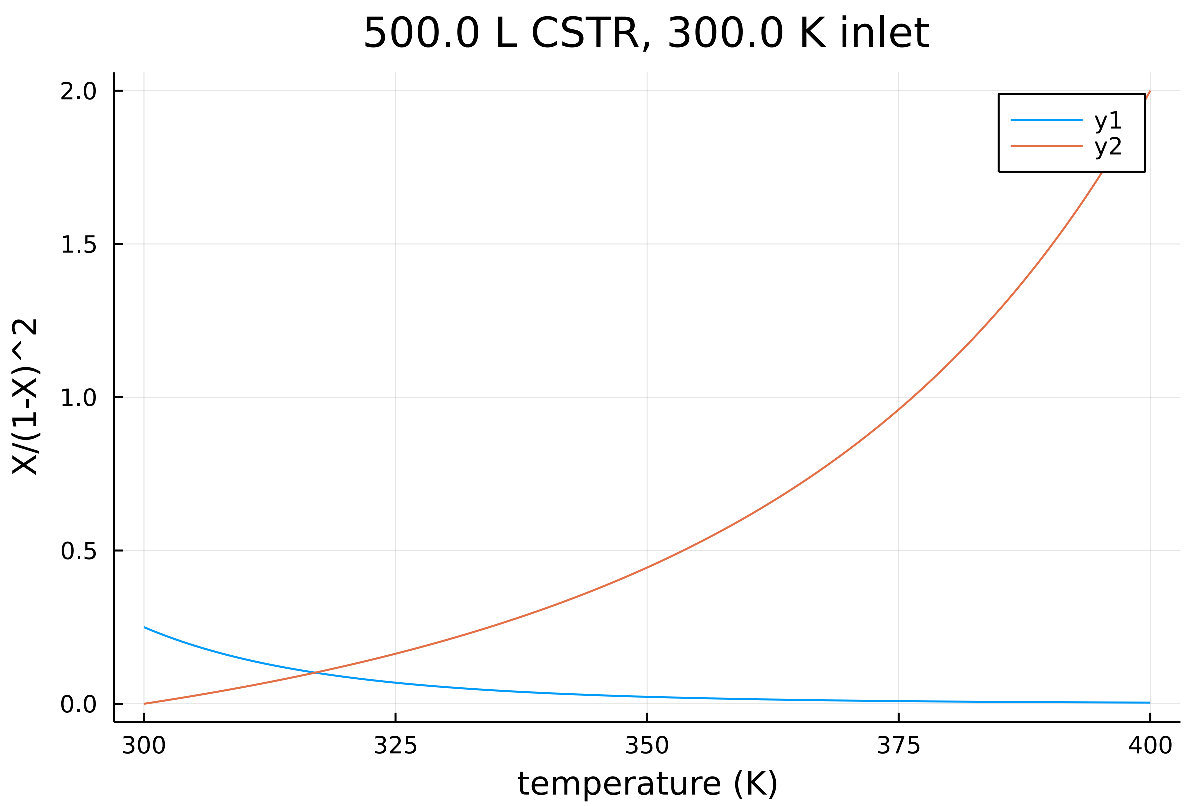
$$\frac{X_A}{(1-X_A)^2} = \frac{VC_{A0}}{v_0} k^0 \exp\left[\frac{-E_a}{R}\left(\frac{1}{T^0} - \frac{1}{T}\right)\right]$$

and from energy balance on an adiabatic system,

$$X_{eb} = \frac{C_p(T - T^0)}{-\Delta H}$$

Letting $y_1 = \frac{X_A}{(1-X_A)^2}$ and $y_2 = \frac{X_{eb}}{(1-X_{eb})^2}$, we can solve for $T$ consistent with both:

```julia
begin
    # conversion from energy balance
    Xeb(T; T⁰) = Cₚ * (T - T⁰) / -ΔH

    # conversion expressions
    y₁(T; V, T⁰, Cₐ₀) = V * Cₐ₀ * k⁰ * exp(-Eₐ * (1/T⁰ - 1/T) / R) / v₀
    y₂(T; T⁰) = Xeb(T; T⁰=T⁰) / (1 - Xeb(T; T⁰=T⁰))^2

    # temperature range
    T = [Tᵢₙ:0.1:Tₘₐₓ...]
end;
```

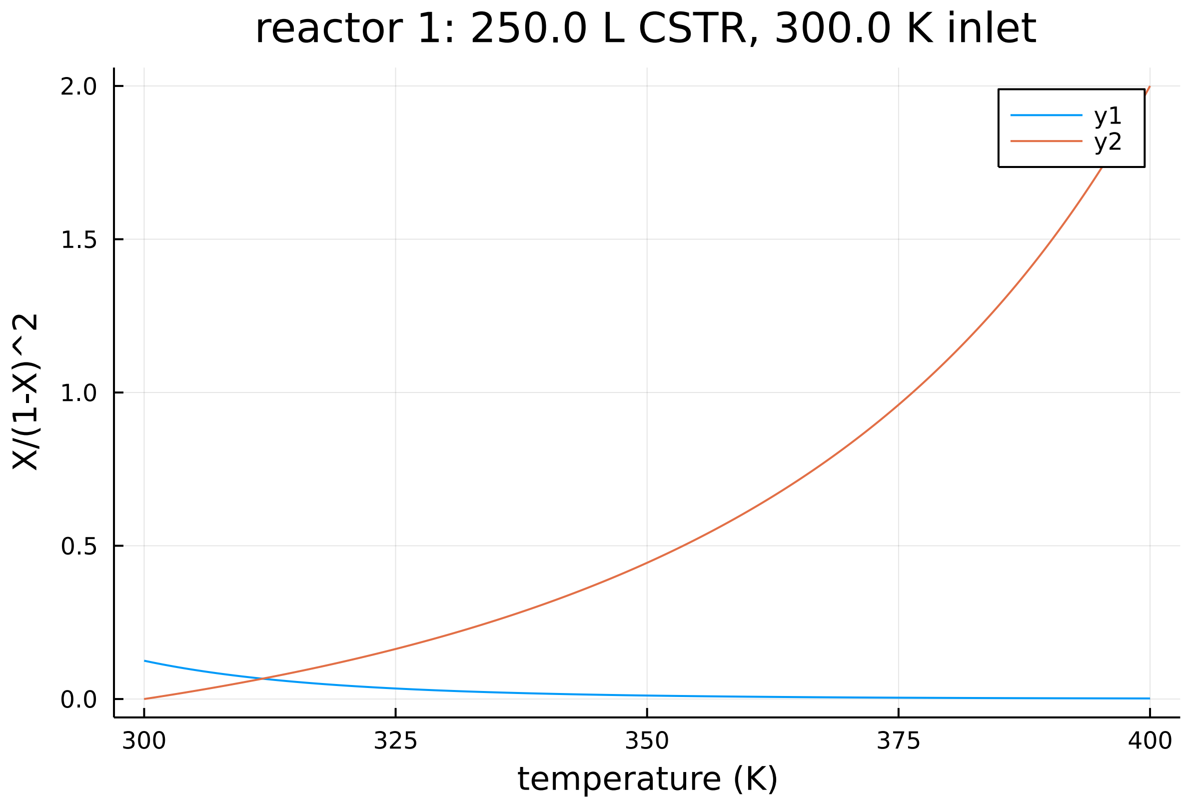


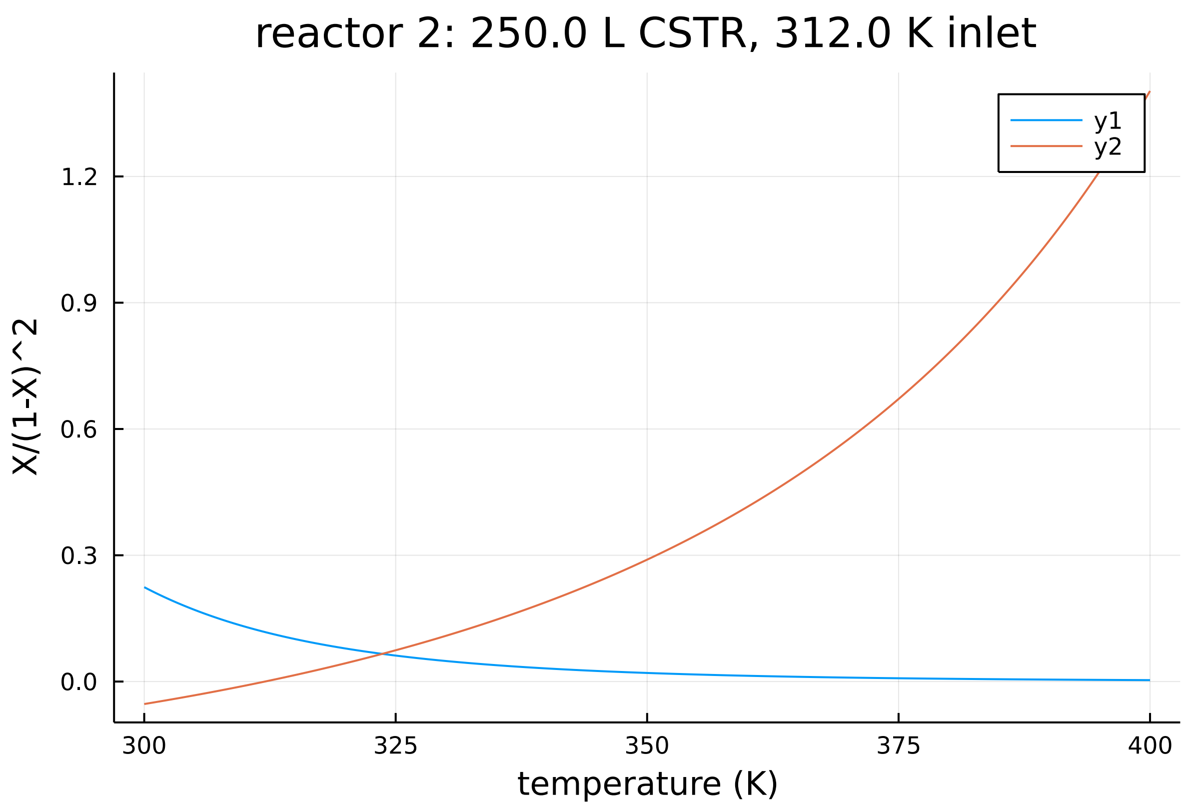

```
# plot curves
begin
    local V = 500. # L
    local T⁰ = 300. # K
    local Cₐ₀ = 0.1 # mol/L

    plot(T, y₁.(T; V=V, T⁰=T⁰, Cₐ₀=Cₐ₀),
        ylabel="X/(1-X)^2",
        xlabel="temperature (K)",
        title="$V L CSTR, $T⁰ K inlet"
    )

    plot!(T, y₂.(T; T⁰=T⁰))
end
```

The intercept is at 317.0 K and 8.5 % conversion.

```
# find the intercept
begin
    local Cₐ₀ = 0.1
    local i = argmin(abs.(y₁.(T; V=500, T⁰=300, Cₐ₀=Cₐ₀) .- y₂.(T; T⁰=300)))
    local t = T[i]
    local x = Xeb(t; T⁰=300) * 100
    md"The intercept is at $t K and $x % conversion."
end
```

Repeating for two 250-L reactors in series:


reactor 1: 250.0 L CSTR, 300.0 K inlet
y1
y2
2.0
1.5
1.0
0.5
0.0
X/(1-X)^2
300
325
350
375
400
temperature (K)


reactor 2: 250.0 L CSTR, 312.0 K inlet
y1
y2
1.2
0.9
0.6
0.3
0.0
X/(1-X)^2
300
325
350
375
400
temperature (K)

The outlet of the first reactor is at 311.8 and 6.0 % conversion, and the second reactor is at 323.5 K and 11.0 % conversion.

```
begin
    local T⁰ = 300.
    local V = 250.
    local Cₐ₀ = 0.1

    # get outlet conditions of first reactor by using above inputs
    local i = argmin(abs.(y₁.(T; V=V, T⁰=T⁰, Cₐ₀=Cₐ₀) .- y₂.(T; T⁰=T⁰)))
    local t1 = T[i]
    local x1 = Xeb(t1; T⁰=T⁰)
    local c1 = round(x1, digits=2) * 100

    # get outlet conditions of second reactor by using rxr#1 output as inputs
    local j = argmin(abs.(y₁.(T; V=V, T⁰=t1, Cₐ₀=(1-x1)*Cₐ₀) .- y₂.(T; T⁰=t1)))
    local t2 = T[j]
    local x2 = (1 - Xeb(t2; T⁰=t1)) * x1 + x1
    local c2 = round(x2, digits=2) * 100

    md"The outlet of the first reactor is at $t1 and $c1 % conversion, and the second reactor is at $t2 K and $c2 % conversion."
end
```

# (b)

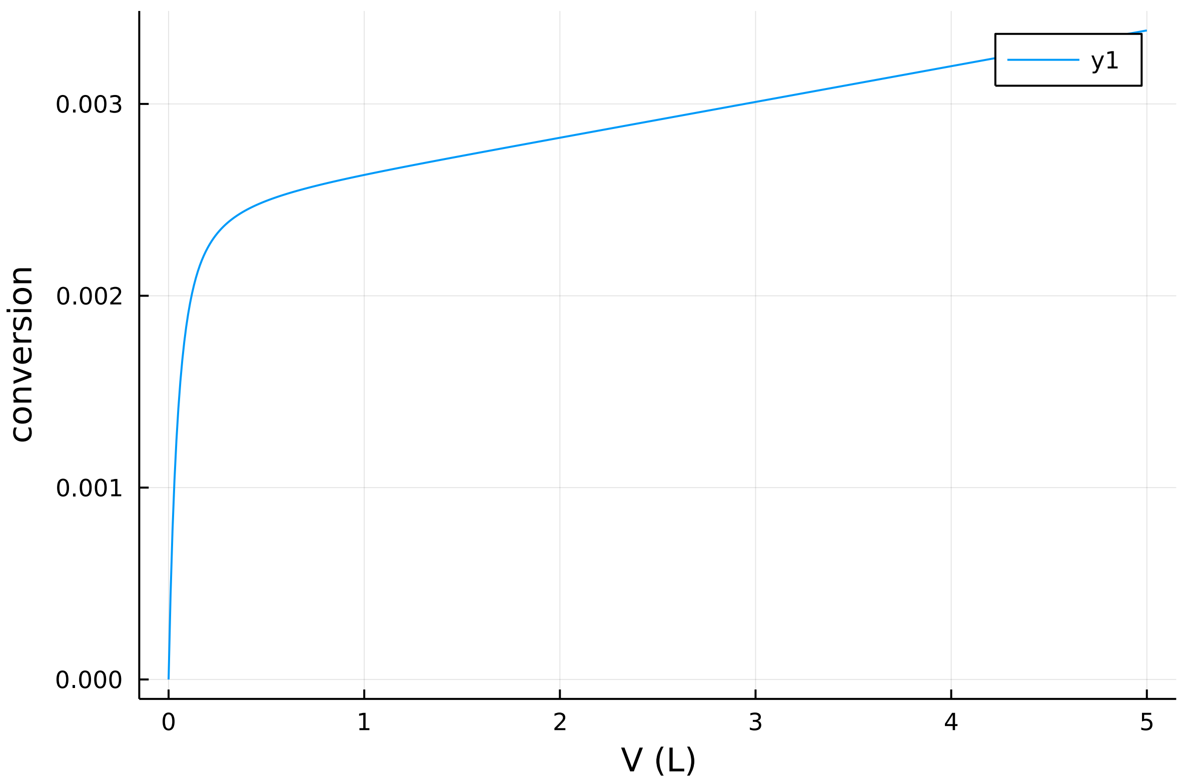
For the same reaction in a PFR with constant-temperature heat exchanger:

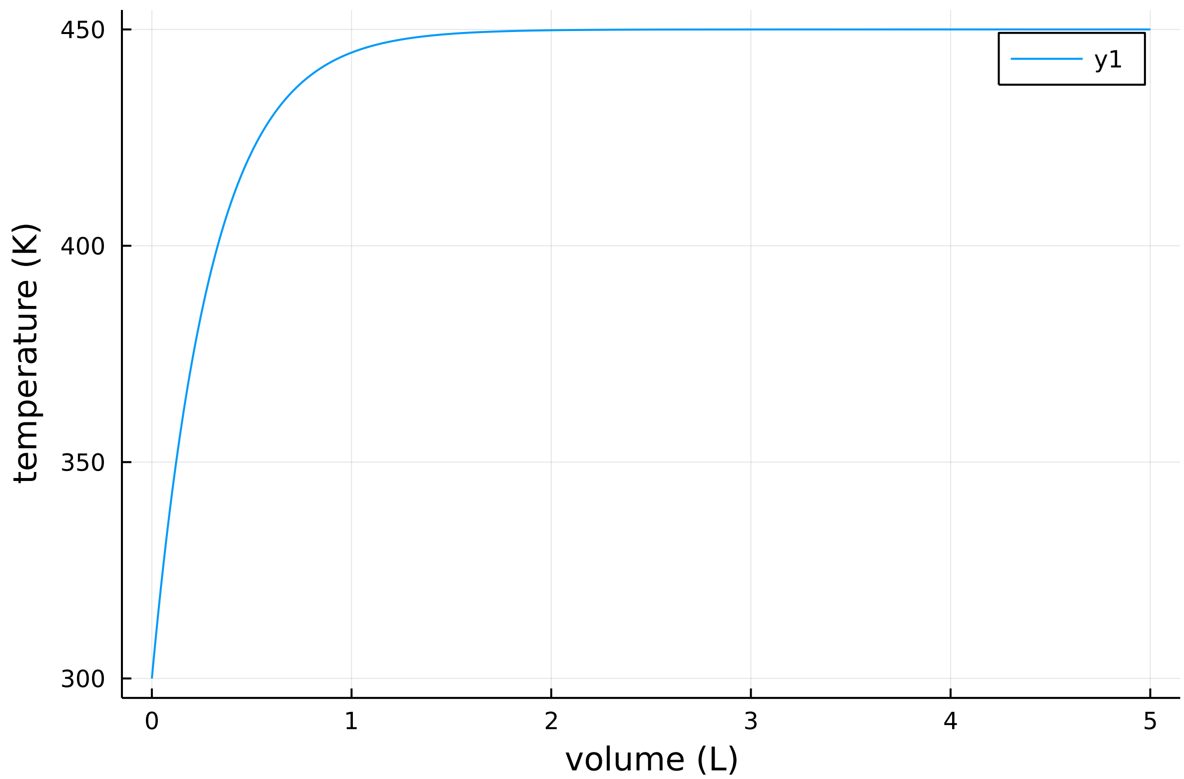
$$\frac{dT}{dV} = \frac{U_a * (T_a - T) + r_A \Delta H}{2C_{A0}(1 - X_A)v_0 C_{p,A}}$$

$$\frac{dX}{dV} = \frac{-r_A}{C_{A0}v_0}$$

$$r_A = -k^0 \exp\left[\frac{-E_a}{R}\left(\frac{1}{T^0} - \frac{1}{T}\right)\right] C_{A0}^2 (1 - X_A)^2$$

```
begin
    function update_f!(f, u, _, t)
        X, T = u

        # dX/dV
        f[1] = -rₐ(T, X) / Cₐ₀ / v₀

        # dT/dV
        f[2] = (Uₐ * (Tₐ - T) + rₐ(T, X) * ΔH) / (Cₐ₀ * (1 - X) * v₀ * Cₚ)
    end

    local X₀ = 0.
    local T₀ = T⁰ = 300.
    local Cₐ₀ = 0.1
    local u₀ = [X₀, T₀]
    local Tₐ = 450.

    rₐ(T, X) = -k⁰ * exp(-Eₐ * (1/T⁰ - 1/T) / R) * Cₐ₀^2 * (1 - X)^2

    local prob = ODEProblem(update_f!, u₀, (0., 5.),
                      saveat=0.01)

    sol_12_7_b = solve(prob)
end;
```



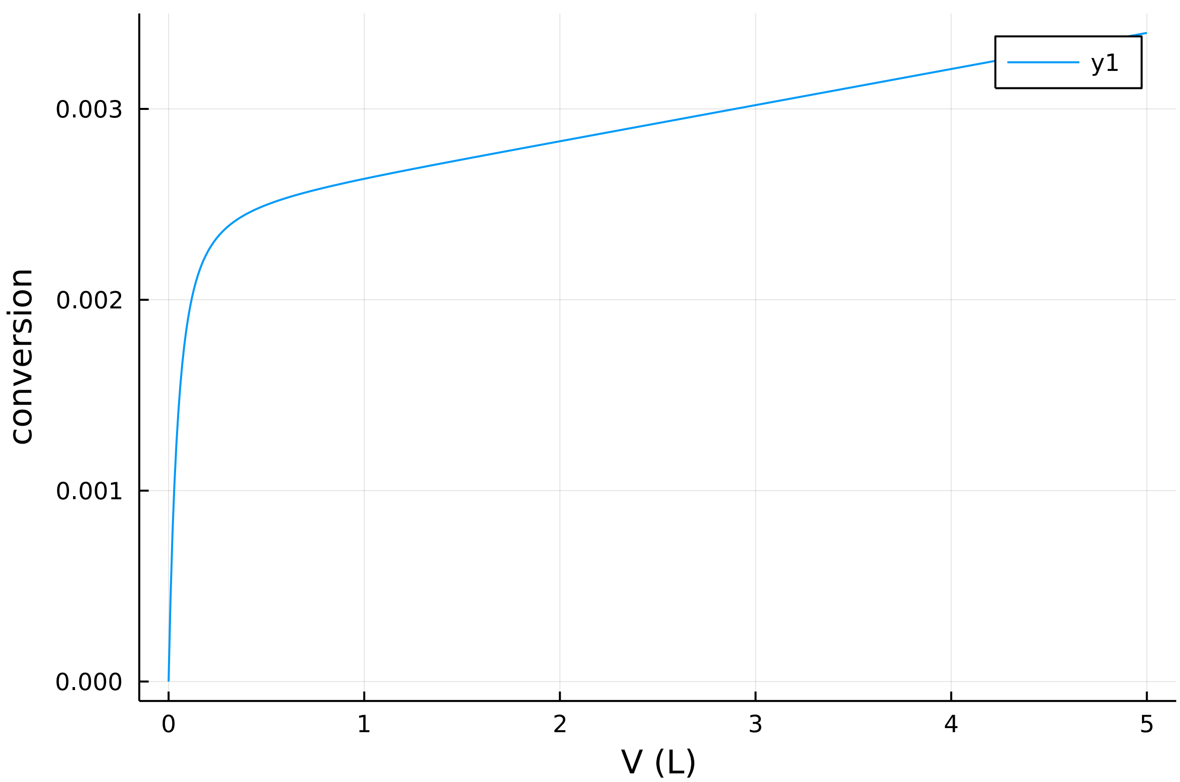


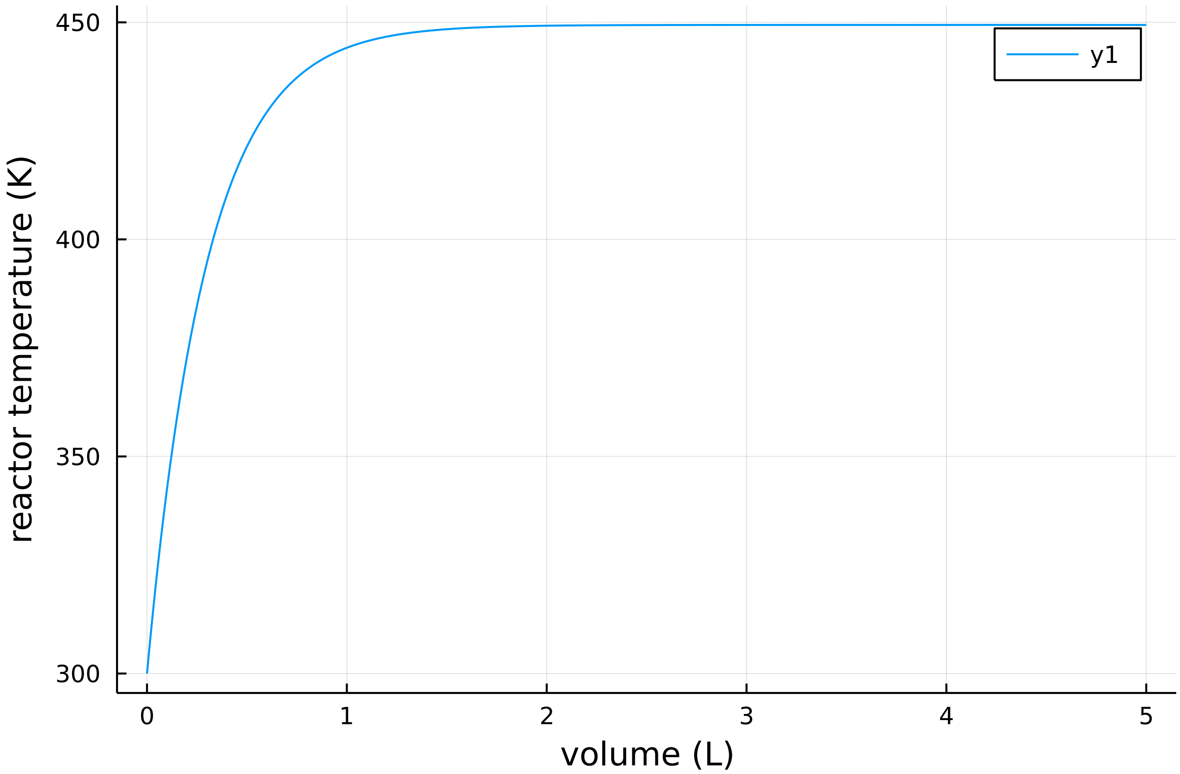

## (c)

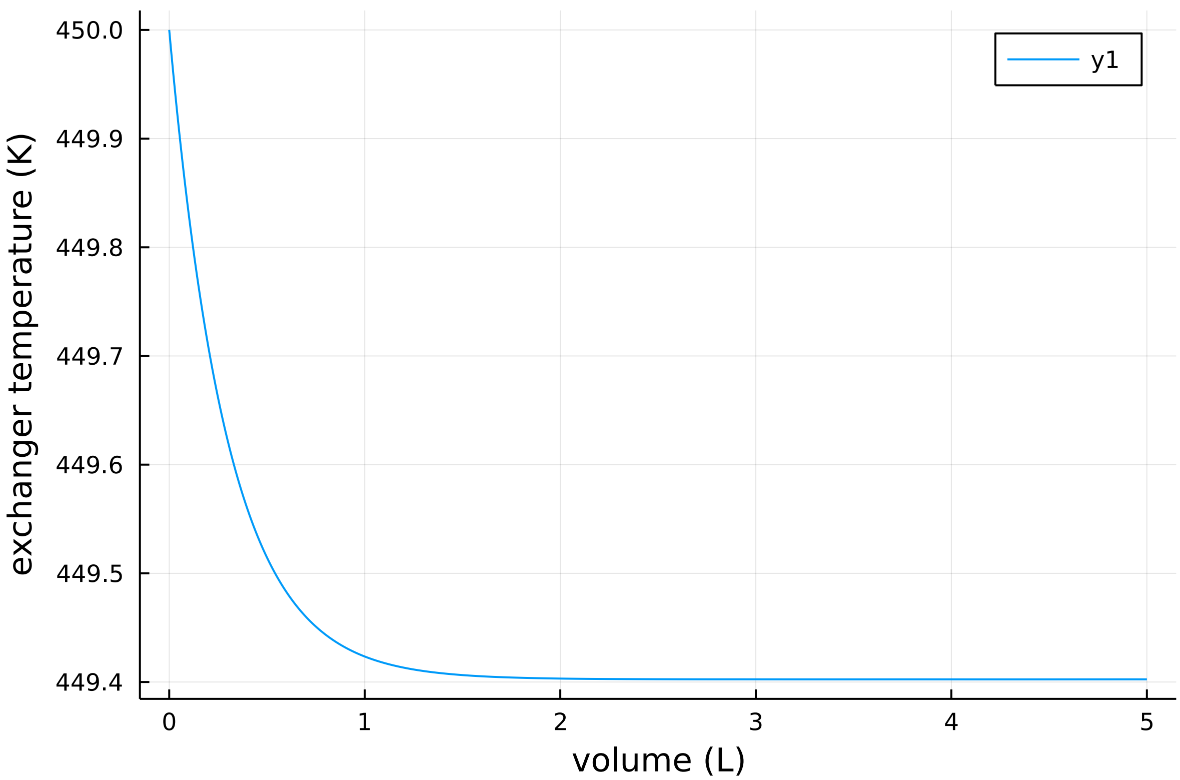
Add the heat exchanger expression:

$$\dot{m}_0 c_{p,0} \frac{dT_a}{dV} = U_A (T - T_0)$$

```
begin
    function update_f2!(f, u, _, t)
        X, T, Tₐ = u

        # dX/dV
        f[1] = -rₐ(T, X) / Cₐ₀ / v₀

        # dT/dV
        f[2] = (Uₐ * (Tₐ - T) + rₐ(T, X) * ΔH) / (Cₐ₀ * (1 - X) * v₀ * Cₚ)

        # dTₐ/dV
        f[3] = Uₐ * (T - Tₐ) / (ṁ₀ * Cₚ)
    end

    local X₀ = 0.
    local T₀ = local T⁰ = 300.
    local Cₐ₀ = 0.1
    local Tₐ₀ = 450.
    local u₀ = [X₀, T₀, Tₐ₀]

    sol_12_7_c = solve(ODEProblem(update_f2!, u₀, (0., 5.),
                        saveat=0.01))
end;
```




450
400
350
300
reactor temperature (K)
y1
0
1
2
3
4
5
volume (L)

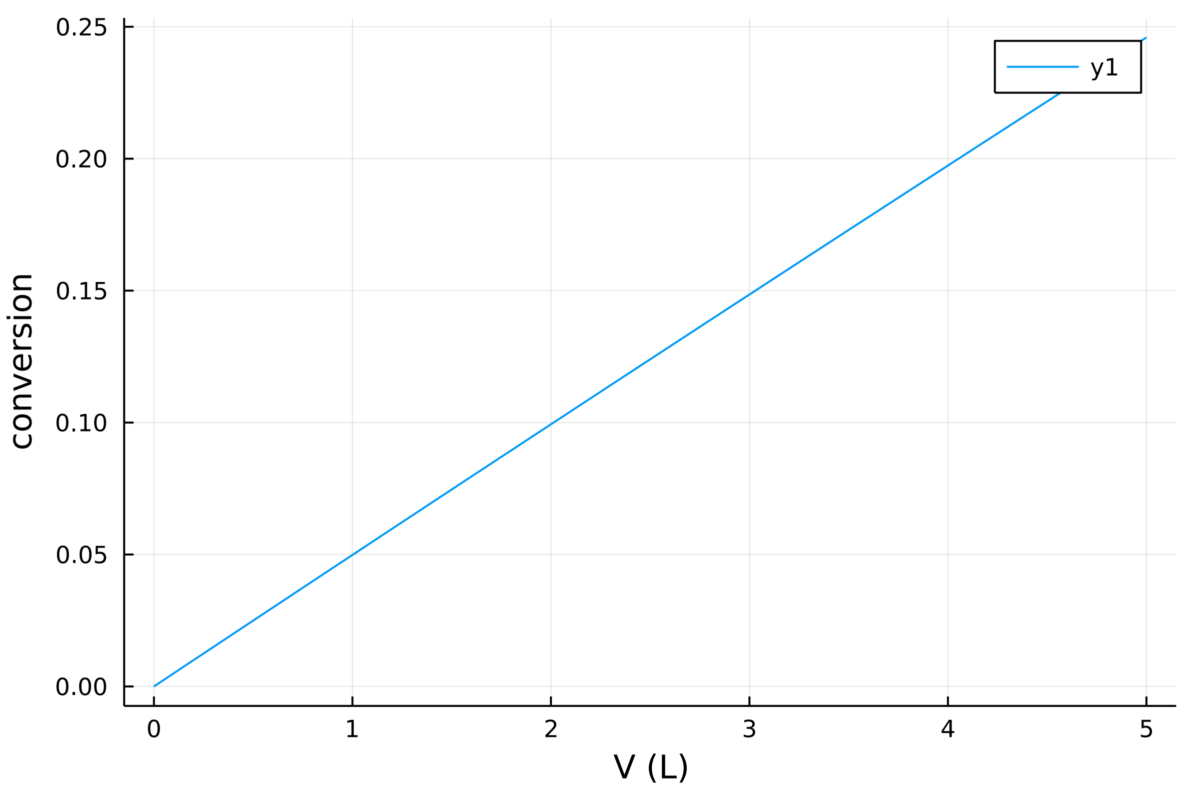

450.0
449.9
449.8
449.7
449.6
449.5
449.4
exchanger temperature (K)
y1
0
1
2
3
4
5
volume (L)

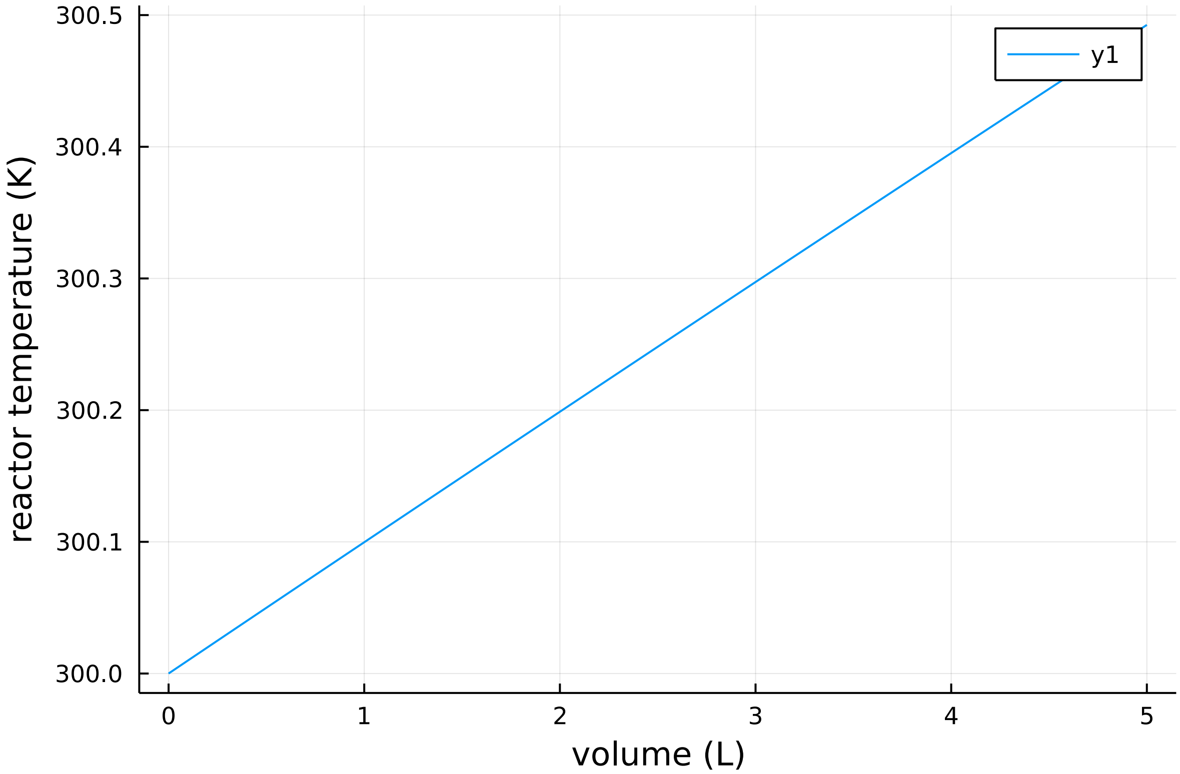
## (e)

Change the energy balance expression:

$$\frac{dT}{dV} = \frac{r_A \Delta H}{2C_{A0}(1 - X_A)v_0 C_{p,A}}$$

```julia
begin
    function update_f3!(f, u, _, t)
        X, T = u

        # dX/dV
        f[1] = -rₐ(T, X) / Cₐ₀ / v₀

        # dT/dV
        f[2] = rₐ(T, X) * ΔH / (Cₐ₀ * (1 - X) * v₀ * Cₚ)
    end

    local X₀ = 0.
    local T₀ = local T⁰ = 300.
    local Cₐ₀ = 0.1
    local Tₐ₀ = 450.
    local u₀ = [X₀, T₀]

    sol_12_7_e = solve(ODEProblem(update_f3!, u₀, (0., 5.),
                       saveat=0.01))
end;
```





300.5
300.4
300.3
300.2
300.1
300.0
reactor temperature (K)
0
1
2
3
4
5
volume (L)
y1

```julia
using DifferentialEquations , Plots
```

# Problem 12-15

```julia
# physical data
begin
    Eₐ = 4e4 # cal/mol
    R = 1.987 # cal/mol/K
    k⁰ = 6.6e-3 # 1/min
    Fₐ₀ = 80. # mol/min
    cₚ = 50 # cal/mol
    ΔH = -7500. # cal/mol
    Uₐ = 8e3 # cal/min/K
    Tₐ = 300 # K
    τ = 100 # min
end;
```

## (a)

Heat generation:

$$G = r_A \Delta H$$

Heat removal:

$$R = U_A(T_a - T)$$

Rate law:

$$-r_A = kC_{A0}(1 - X_A)$$

$$k = k^0 \exp\left[\frac{-E_a}{R}\left(\frac{1}{T^0} - \frac{1}{T}\right)\right]$$

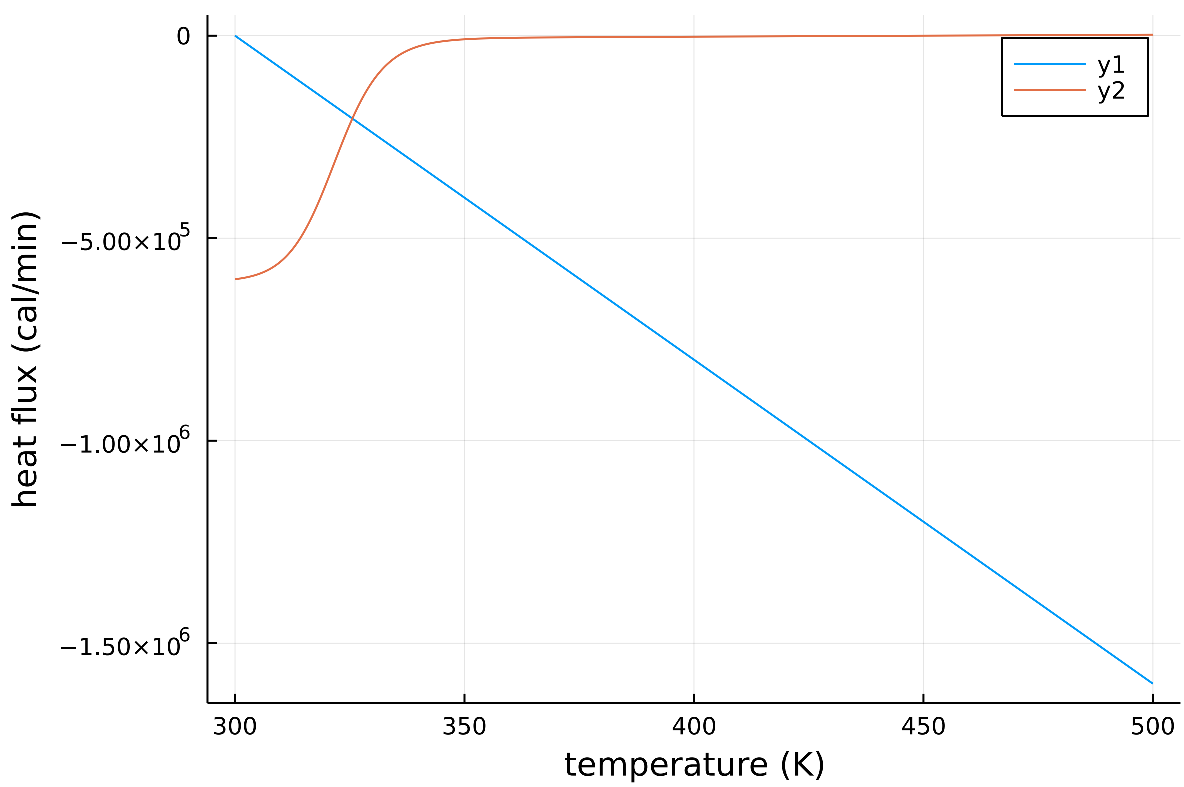
Performance equation:

$$k\tau = \frac{X_A}{1 - X_A}$$

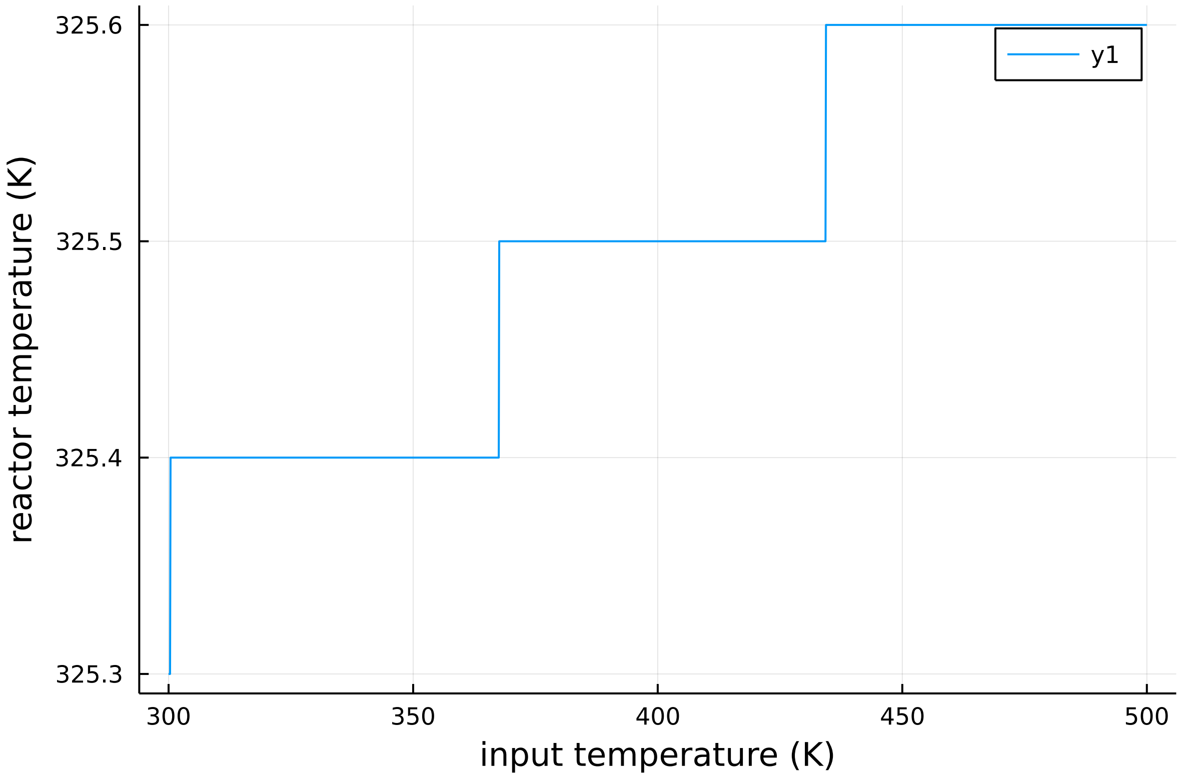
```julia
begin
    T = [300:0.1:500...]

    rₐ(T) = k(T) * Fₐ₀ * (1 - X(T))
    X(T) = k(T) / (1 + k(T))
    k(T) = k⁰ * exp(-Eₐ * (1/350 - 1/T) / R)

    heat_inp(T; T₀) = cₚ * (T - T₀)
    heat_rem(T) = Uₐ * (Tₐ - T)
    heat_gen(T) = rₐ(T) * ΔH
end;
```

The curves intersect at 326 K, corresponding to 33 % conversion.

```julia
begin
    Rs = heat_rem.(T)
    Gs = heat_gen.(T) + heat_inp.(T, T₀=450.)
    local i = argmin(abs.(Rs .- Gs))
    local t = Int(round(T[i]))
    local c = Int(round(X(T[i]) * 100))
    md"The curves intersect at $t K, corresponding to $c % conversion."
end
```




**(b)**




```
begin
    function find_T(T₀)
        Rs = heat_rem.(T)
        Gs = heat_gen.(T) + heat_inp.(T, T₀=T₀)
        i = argmin(abs.(Rs .- Gs))
        return T[i]
    end

    plot(T, find_T.(T), xlabel="input temperature (K)", ylabel="reactor temperature
(K)")
end
```