1)

a)

In [1]:

```
using Pkg
Pkg.build("Mosek")
Pkg.build("MosekTools")
```

```
    Building Mosek → `C:\Users\jungh\.julia\scratchspaces\44cfe95a-1eb2-52ea-b672-e2afdf69b78f\f7c2c7ff998dcc378587bda88dababb5783dda57\build.log`
    Building Mosek → `C:\Users\jungh\.julia\scratchspaces\44cfe95a-1eb2-52ea-b672-e2afdf69b78f\f7c2c7ff998dcc378587bda88dababb5783dda57\build.log`
```

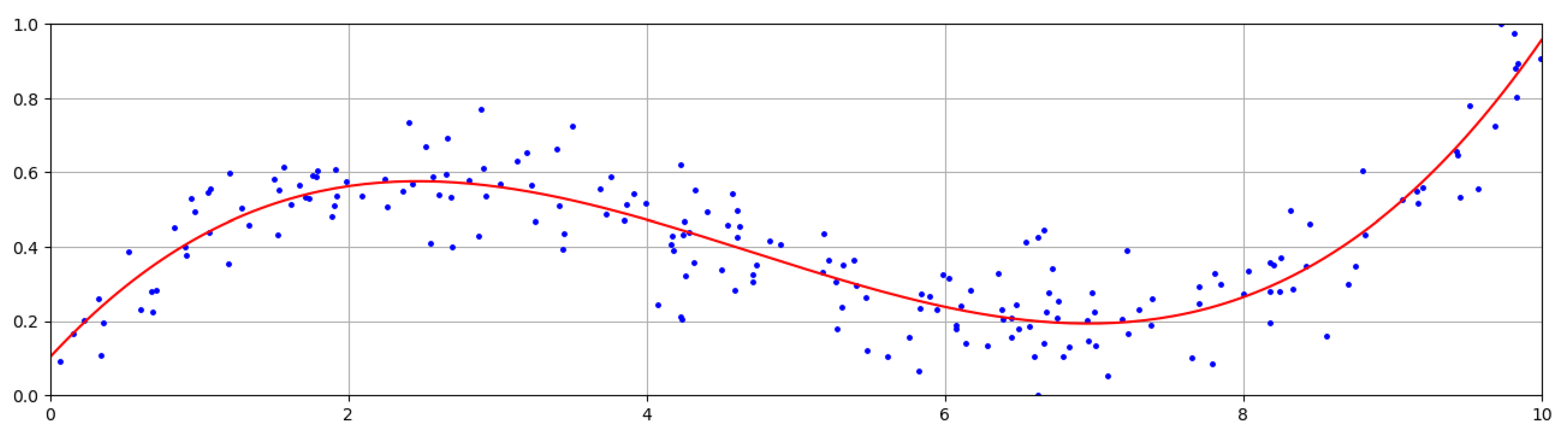
In [50]:

```
using JuMP, Gurobi, MosekTools, Mosek, CSV, DataFrames


dt = CSV.read("xy_data.csv",DataFrame)

x = dt[:,1]
y = dt[:,2]
k = 3
n = length(x)
A = zeros(n,k+1)
for i = 1:n
    for j = 1:k+1
        A[i,j] = x[i]^(k+1-j)
    end
end
m = Model(optimizer_with_attributes(Mosek.Optimizer))
@variable(m, u[1:k+1])
@objective(m, Min, sum( (y - A*u).^2 ))
optimize!(m)
uopt = value.(u)
println(termination_status(m))
println("The objective value is ", objective_value(m))
println("The optimal parameters are ",uopt)
```

```
Problem
  Name                   :
  Objective sense        : min
  Type                   : CONIC (conic optimization problem)
  Constraints            : 6
  Cones                  : 1
  Scalar variables       : 11
  Matrix variables       : 0
  Integer variables      : 0

Optimizer started.
Presolve started.
Eliminator started.
Freed constraints in eliminator : 0
Eliminator terminated.
Eliminator - tries                  : 1                 time                   : 0.00
Lin. dep.  - tries                  : 0                 time                   : 0.00
Lin. dep.  - number                 : 0
Presolve terminated. Time: 0.00
Problem
  Name                   :
  Objective sense        : min
  Type                   : CONIC (conic optimization problem)
  Constraints            : 6
  Cones                  : 1
  Scalar variables       : 11
  Matrix variables       : 0
  Integer variables      : 0

Optimizer  - threads                : 8
Optimizer  - solved problem         : the primal
Optimizer  - Constraints            : 0
Optimizer  - Cones                  : 1
Optimizer  - Scalar variables       : 3                 conic                  : 3
Optimizer  - Semi-definite variables: 0                 scalarized             : 0
Factor     - setup time             : 0.00              dense det. time        : 0.00
Factor     - ML order time          : 0.00              GP order time          : 0.00
Factor     - nonzeros before factor : 0                 after factor           : 0
Factor     - dense dim.             : 0                 flops                  : 0.00e+00
ITE PFEAS    DFEAS    GFEAS    PRSTATUS   POBJ              DOBJ              MU       TIME
0   2.9e-01  8.7e+00  2.4e+00  0.00e+00   4.002384189e+01   3.860962833e+01   1.0e+00  0.00
1   1.6e-02  4.6e-01  4.4e-01  -9.38e-01  2.108968271e+01   3.089876248e+01   5.4e-02  0.00
2   3.2e-03  9.5e-02  8.4e-02  -2.05e-01  6.336187291e+00   1.568712775e+01   1.1e-02  0.00
3   2.4e-04  7.1e-03  1.6e-03  6.03e-01   2.203655278e+00   2.824033239e+00   8.2e-04  0.00
4   1.5e-07  4.4e-06  3.0e-08  9.64e-01   1.671175895e+00   1.671751605e+00   5.1e-07  0.00
5   5.4e-12  1.6e-10  1.1e-12  1.00e+00   1.670930737e+00   1.670930764e+00   1.8e-11  0.00
Optimizer terminated. Time: 0.00

OPTIMAL
The objective value is 1.6709307308625725
The optimal parameters are [0.008441440631036058, -0.11935614636954046, 0.4347550408991947, 0.10340377997714666]
```

In [51]:

```
using PyPlot
npts = 200
xfine = range(0,stop=10,length=npts)
ffine = ones(npts)
for j = 1:k
 ffine = [ffine.*xfine ones(npts)]
end
yfine = ffine * uopt
figure(figsize=(16,4))
plot( x, y, "b.", markersize=5)
plot( xfine, yfine, "r-")
axis([0,10,0,1])
grid()
```



b)

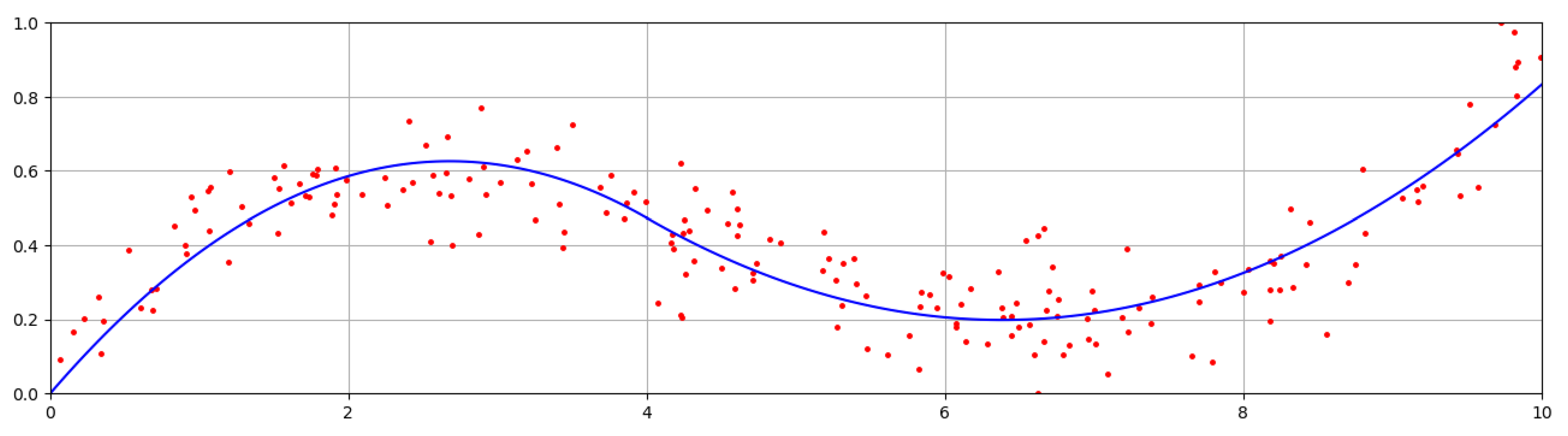
In [52]:

```
lower = findall(x -> x .< 4,x)
upper = findall(x-> x .>= 4,x)
x1 = x[lower]
y1 = y[lower]

x2 = x[upper]
y2 = y[upper]

m1 = Model(with_optimizer(Gurobi.Optimizer))
@variable(m1, p[1:3])
@variable(m1, q[1:3])

@constraint(m1, p[1]*0 .+ p[2]*0 .+ p[3] ==0)
@constraint(m1, p[1]*(4^2) .+ p[2]*4 .+ p[3] == q[1]*4^2 .+ q[2]*4 .+ q[3])

@constraint(m1, p[1]*4*2 .+p[2] == q[1]*4*2 .+ q[2])

@objective(m1, Min, sum((p[1]*x1 .^2 + p[2]*x1 .+ p[3] - y1) .^2) +
    sum((q[1]*x2 .^2 + q[2]*x2 .+ q[3] - y2) .^2))
optimize!(m1)

println(termination_status(m1))
println(objective_value(m1))
```

```
Set parameter Username

Academic license - for non-commercial use only - expires 2022-05-18

Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (win64)

Thread count: 8 physical cores, 16 logical processors, using up to 16 threads

Optimize a model with 3 rows, 6 columns and 11 nonzeros

Model fingerprint: 0x9d7ed823

Model has 12 quadratic objective terms

Coefficient statistics:

  Matrix range     [1e+00, 2e+01]

  Objective range  [7e+01, 5e+03]

  QObjective range [2e+02, 7e+05]

  Bounds range     [0e+00, 0e+00]

  RHS range        [0e+00, 0e+00]

Presolve removed 1 rows and 1 columns

Presolve time: 0.00s

Presolved: 2 rows, 5 columns, 9 nonzeros

Presolved model has 9 quadratic objective terms

Ordering time: 0.00s


Barrier statistics:

 Free vars  : 8

 AA' NZ     : 8.000e+00

 Factor NZ  : 1.500e+01

 Factor Ops : 5.500e+01 (less than 1 second per iteration)

 Threads    : 1


                  Objective                Residual

Iter       Primal          Dual         Primal    Dual     Compl     Time

   0   3.93167351e+01  3.93167351e+01  0.00e+00 1.15e+03  0.00e+00     0s

   1   3.33900550e+01  3.90610078e+01  2.02e-08 1.06e+03  0.00e+00     0s

   2   2.01972815e+01  3.59268095e+01  2.87e-08 8.05e+02  0.00e+00     0s

   3   1.20215358e+01  3.06771419e+01  2.48e-08 5.98e+02  0.00e+00     0s

   4   9.15756523e+00  2.75640401e+01  6.26e-08 5.06e+02  0.00e+00     0s

   5   4.77655376e+00  1.96761937e+01  5.53e-08 3.17e+02  0.00e+00     0s

   6   3.67958142e+00  1.62965491e+01  6.51e-08 2.48e+02  0.00e+00     0s

   7   2.26267536e+00  8.46666464e+00  1.20e-07 1.05e+02  0.00e+00     0s

   8   1.95770246e+00  2.96084814e+00  1.22e-07 1.57e+01  0.00e+00     0s

   9   1.95075320e+00  1.95076104e+00  8.80e-09 1.57e-05  0.00e+00     0s

  10   1.95076002e+00  1.95076002e+00  2.74e-14 1.55e-11  0.00e+00     0s


Barrier solved model in 10 iterations and 0.00 seconds (0.00 work units)

Optimal objective 1.95076002e+00



User-callback calls 68, time in user-callback 0.00 sec

OPTIMAL
1.9507600232199636
```

In [53]:

```
lowerx=value.(p[1:3])
higherx=value.(q[1:3])

using PyPlot, LinearAlgebra
npts = 100
xfine = range(0,stop=4,length=npts)
xfine_higher = range(4,stop=10,length=npts)
ffine = ones(npts)
ffine_h = ones(npts)
for j = 1:2
    ffine = [ffine.*xfine ones(npts)]
end
for j =1:2
    ffine_h = [ffine_h.*xfine_higher ones(npts)]
end
yfine = ffine * lowerx
yfine_high = ffine_h * higherx
figure(figsize=(16,4))
plot( x, y, "r.", markersize=5)
plot( xfine, yfine, "b-")
plot( xfine_higher, yfine_high, "b-")
axis([0,10,0,1])
grid()
```



2

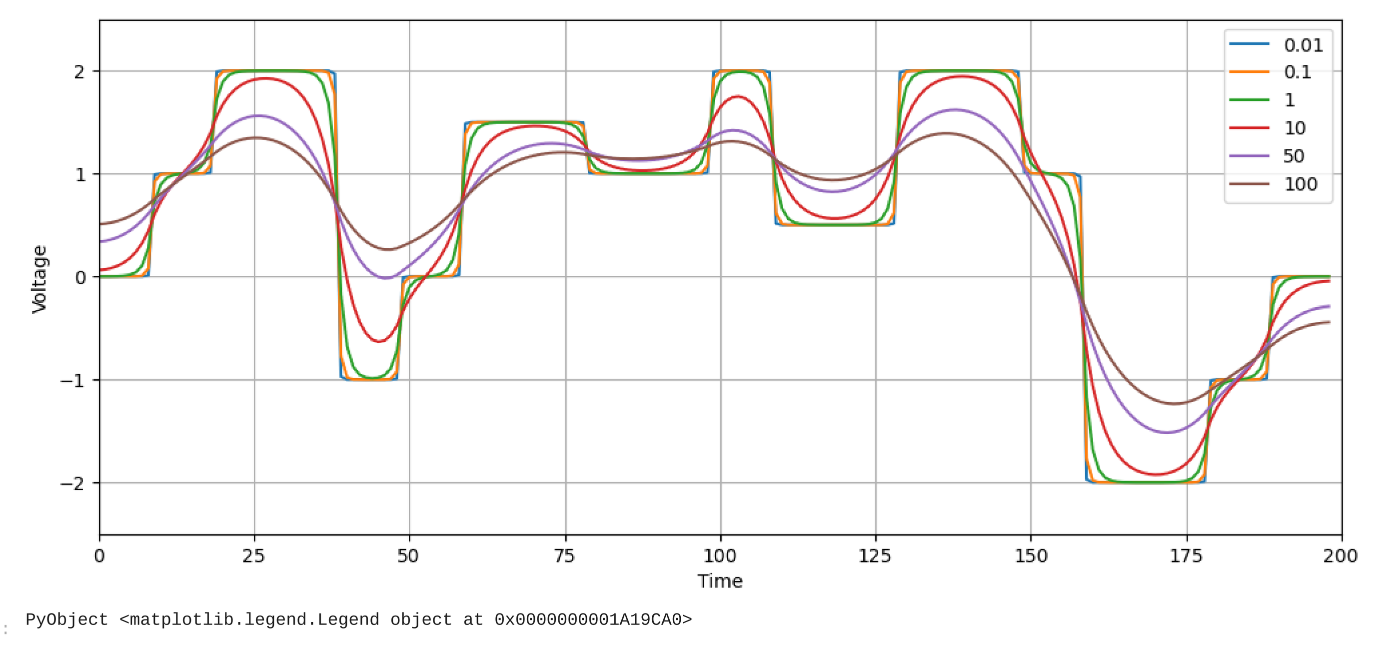
In [59]:

```
vt=CSV.read("voltages.csv",DataFrame)
using JuMP, Mosek

T =199
λ = [0.01,0.1,1,10,50,100]
m3 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m3, u[1:T])
@expression(m3, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m3, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m3,Min,sum(A)+λ[1]*sum(b))
optimize!(m3)
uval1=value.(u)
println("λ=",λ[1]," : ",objective_value(m3))

m4 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m4, u[1:T])
@expression(m4, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m4, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m4,Min,sum(A)+λ[2]*sum(b))
optimize!(m4)
uval2=value.(u)
println("λ=",λ[2],": ",objective_value(m4))

m5 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m5, u[1:T])
@expression(m5, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m5, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m5,Min,sum(A)+λ[3]*sum(b))
optimize!(m5)
uval3=value.(u)
println("λ=",λ[3],": ",objective_value(m5))

m6 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m6, u[1:T])
@expression(m6, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m6, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m6,Min,sum(A)+λ[4]*sum(b))
optimize!(m6)
uval4=value.(u)
println("λ=",λ[4],": ",objective_value(m6))

m7 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m7, u[1:T])
@expression(m7, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m7, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m7,Min,sum(A)+λ[5]*sum(b))
optimize!(m7)
uval5=value.(u)
println("λ=",λ[5],": ",objective_value(m7))

m8 = Model(with_optimizer(Mosek.Optimizer,LOG=0))
@variable(m8, u[1:T])
@expression(m8, A, sum((vt[i,1] .- u[i])^2 for i = 1:n))
@expression(m8, b, sum((u[i+1] .- u[i])^2 for i = 1:n-1))
@objective(m8,Min,sum(A)+λ[6]*sum(b))
optimize!(m8)
uval6=value.(u)

println("λ=",λ[6],": ",objective_value(m8))
```

```
λ=0.01 : 0.3137858162216389
λ=0.1: 2.7044936151372667
λ=1.0: 14.310953235785519
λ=10.0: 50.14338969336234
λ=50.0: 108.3882168626505
λ=100.0: 140.85580642068211
```

In [55]:

```
using PyPlot
figure(figsize=(12,5))
plot(0:T-1, uval1,label="0.01");
plot(0:T-1, uval2,label="0.1");
plot(0:T-1, uval3,label="1");
plot(0:T-1, uval4,label="10");
plot(0:T-1, uval5,label="50");
plot(0:T-1, uval6,label="100");
xlabel("Time"); ylabel("Voltage"); grid()

axis([0,200,-2.5,2.5])
legend(loc ="best")
```



Out[55]: PyObject <matplotlib.legend.Legend object at 0x0000000001A19CA0>