

---

```

% This script file computes the optimal investment levels TAOpt and
    TBOpt
% for the two firms (Firm A and Firm B) of the Simplified
    Oligopolistic
% Optimal Influence model which also involves a single consumer C. The
    rest
% of the model parameters (SOpt,Xopt,POpt,Qopt,Fopt) are also
    computed.

% Specifically, the internal model parameters whose optimal values are
    to
% be determined during the grid searching optimization process are the
% following:
% (i):    [TAopt,TBOpt]
% (ii):   [SAopt,SCopt,SBOpt]
% (iii):  [XAopt,XBOpt]
% (iv):   [PAopt,PBOpt] (These are the optimal prices!!!)
% (v):    [QAopt,QBOpt]
% (vi):   [FAopt FBopt FA_rev_opt FB_rev_opt FA_cost_opt FB_rev_opt]

% The grid searching process will be conducted within the 8-
dimensional
% space defined by the Cartesian product of the following external
    model
% parameters:
% (i):    LA (direct influence exerted by consumer C on Firm A).
% (ii):   LB (direct influence exerted by consumer C on Firm B).
% (iii):  PA (initial belief consumer C holds for product A).
% (iv):   PB (initial belief consumer C holds for product B).
% (v):    M (sensitivity coefficient).
% (vi):   K (sensitivity coefficient).
% (vii):  C (marginal cost).
% (viii): G (marginal influence cost or Gamma).

% IMPORTANT NOTE!!!
% Mind that the underlying continuous game may not have an equilibrium
    point
% for any given configuration of the external parameters. Since all
    the
% internal parameters to be determined accept positive values, the
    value of
% (-1) will be utilized in order to indicate the absence of an
    equilibrium
% point for a particular configuration of the external parameters.

% Define ranges and corresponding increment step for each external
    parameter
% of the Simplified Oligopolistic Optimal Influence model.

clc
clear all

```

---

---

```

% Construct a cell array storing the names of model parameters.
ParamsNames = {'LA','LB','PA','PB','M','K','C','G'};

% Set the varying parameter index.
ParamIndex = 6;

% Parameter #1
LA_MIN = 0.25;
LA_MAX = 0.25;
LA_STEP = 0.01;
% Parameter #2
LB_MIN = 0.25;
LB_MAX = 0.25;
LB_STEP = 0.01;
% Parameter #3
PA_MIN = 0.80;
PA_MAX = 0.80;
PA_STEP = 0.01;
% Parameter #4
PB_MIN = 0.80;
PB_MAX = 0.80;
PB_STEP = 0.01;
% Parameter #5
M_MIN = 0.4;
M_MAX = 0.4;
M_STEP = 0.05;
% Parameter #6
K_MIN = 0.1;
K_MAX = 0.9;
K_STEP = 0.05;
% Parameter #7
C_MIN = 0.0001;
C_MAX = 0.0001;
C_STEP = 0.00001;
% Parameter #8 (G or Gamma!!!)
G_MIN = 5.5;
G_MAX = 5.5;
G_STEP = 0.01;

% Set the corresponding ranges for each parameter of the model.
LA_RANGE = [LA_MIN:LA_STEP:LA_MAX];
LB_RANGE = [LB_MIN:LB_STEP:LB_MAX];
PA_RANGE = [PA_MIN:PA_STEP:PA_MAX];
PB_RANGE = [PB_MIN:PB_STEP:PB_MAX];
M_RANGE = [M_MIN:M_STEP:M_MAX];
K_RANGE = [K_MIN:K_STEP:K_MAX];
C_RANGE = [C_MIN:C_STEP:C_MAX];
G_RANGE = [G_MIN:G_STEP:G_MAX];

% Initialize matrices that store fundamental intermediate quantities
% of the
% underlying optimization problem within the simplified oligopolistic
% enviroment as well as the values of the external optimization
% parameters

```

---

---

```

% for each step of the multi-dimensional grid searching process.
Topts = []; % Optimal investment levels.
Sopts = []; % Optimal limiting influences.
Xopts = []; % Optimal limiting beliefs.
Popts = []; % Optimal prices.
Qopts = []; % Optimal quantities.
Fopts = []; % Optimal profits.
FilterFlags = []; % Solution filtering flag which may be indicative of
    a non-existing solution.
DigitAccuracies = []; % Length of maximal sequence of identical digits
    within the obtained optimal solutions.
Params = []; % 8-tuple of the varying optimization parameters.

% Initialize internal solver parameters.
% Set the number of initial solution points.
N = 50;
% Set the tolerance value for the minimizer. (Preferable value =
    1e-10)
Tolerance = 1e-15;
% Set the Fvals tolerance value for filtering the obtained solutions.
    (Preferable value = 1e-15)
FvalsTolerance = 1e-15;
% Set the derivative tolerance value for filtering the obtained
    solutions. (Preferable value = 1e-08).
DerivativeTolerance = 1e-10;
% Set the maximum number of iterations to be conducted by the
    optimizer.
MaxIterations = 1000;
% Set the maximum number of function evaluations to be conducted by
    the
% optimizer.
MaxFunctionEvaluations = 10000;

fprintf('Grid Evaluation Process in Progress...\n');
% Perform the actual grid searching.
for LA = LA_RANGE
    for LB = LB_RANGE
        for PA = PA_RANGE
            for PB = PB_RANGE
                for M = M_RANGE
                    for K = K_RANGE
                        for C = C_RANGE
                            for G = G_RANGE
                                % Additional parameters definition.
                                alpha = (K*M - 2) / (M^2 - 4);
                                beta = (2*K - M) / (M^2 - 4);
                                gamma = C / (M - 2);
                                gamma_prime = gamma * (M - 1); %
                                gamma_prime is the gamma' parameter.
                                % Set the current params vector.
                                params = [LA LB PA PB M K C G];
                                Params = [Params;params];

```

---

---

```

[TAopt,TBopt,FilterFlag,DigitsAccuracy] =
SimplifiedOligopolisticOptimalInfluences(N,Tolerance,FvalsTolerance,DerivativeTol
    T = [TAopt TBopt];
    if(FilterFlag==0)
        [S,X,P,Q,F] =
RetrieveOptimalModelParameters(T,C,G,LA,LB,PA,PB,alpha,beta,gamma,gamma_prime);
    else
        S = -1*ones(1,3);
        X = -1*ones(1,2);
        P = -1*ones(1,2);
        Q = -1*ones(1,2);
        F = -1*ones(1,6);
    end;
    % Print current solution.
    param_value_string =
strcat([ParamsNames{ParamIndex} ' = ' num2str(params(ParamIndex))]);
    fprintf('%s TAopt =
%f TBopt = %f FilterFlag = %d DigitsAccuracy = %d
\n',param_value_string,TAopt,TBopt,FilterFlag,DigitsAccuracy);
    Topts = [Topts;T];
    FilterFlags =
[FilterFlags;FilterFlag];
    DigitsAccuracies =
[DigitAccuracies;DigitsAccuracy];
    Sopts = [Sopts;S];
    Xopts = [Xopts;X];
    Popts = [Popts;P];
    Qopts = [Qopts;Q];
    Fopts = [Fopts;F];
end
end
end
end
end
end
end

% Set parameters' indices and corresponding values for the parameters
that
% remain constant.
ConstIndices = setdiff(1:length(params),ParamIndex);
ConstValues = params(ConstIndices);
% Perform plotting operations.
plot_parameters_tuples(Topts,Sopts,Xopts,Popts,Qopts,Fopts,Params,ParamIndex,Const

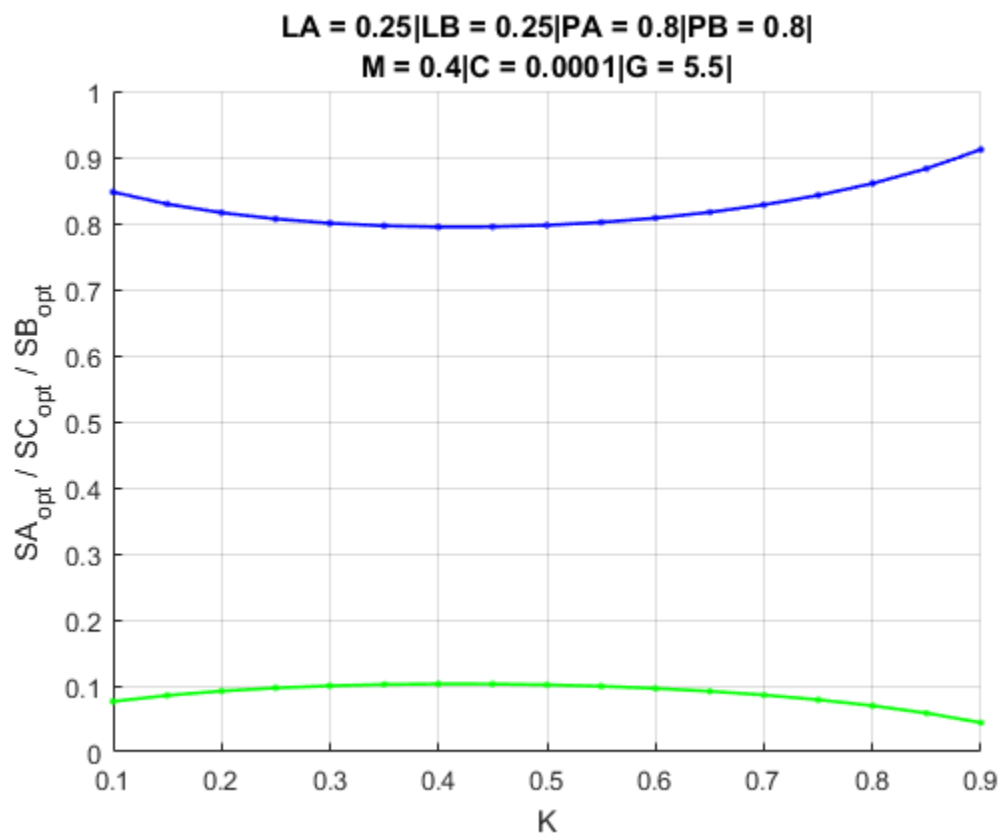
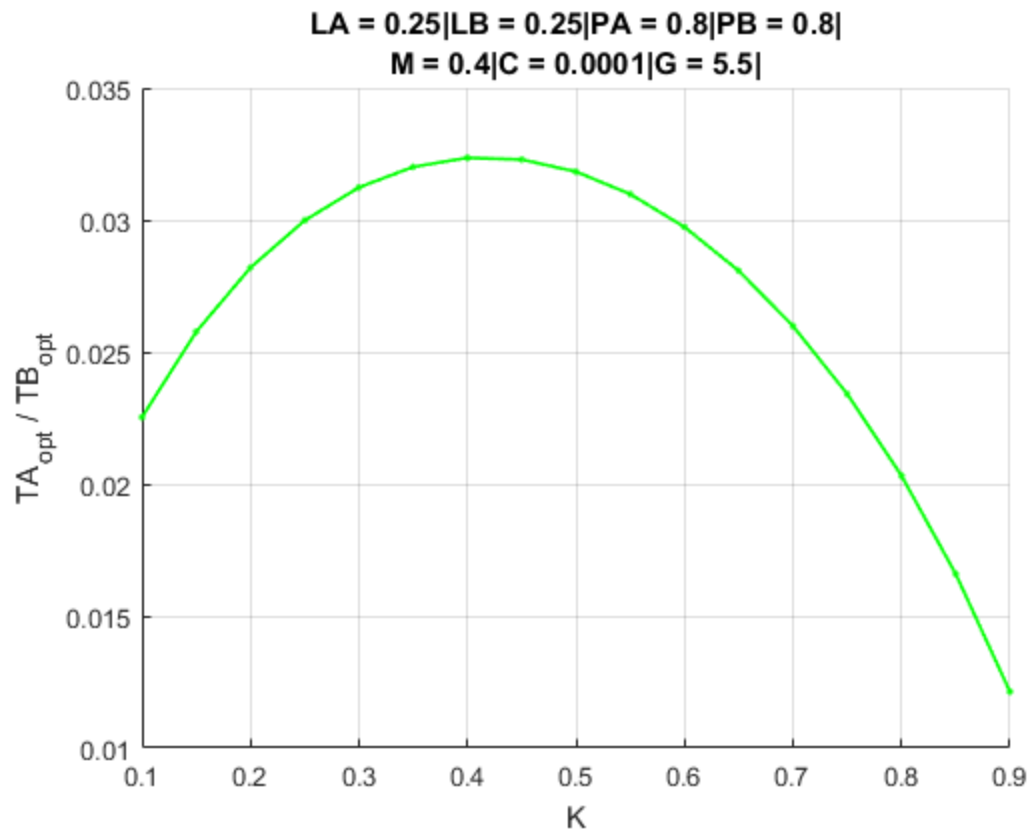
Grid Evaluation Process in Progress...
K = 0.1 TAopt = 0.022528 TBopt = 0.022528 FilterFlag = 0
    DigitsAccuracy = 16
K = 0.15 TAopt = 0.025768 TBopt = 0.025768 FilterFlag = 0
    DigitsAccuracy = 15
K = 0.2 TAopt = 0.028196 TBopt = 0.028196 FilterFlag = 0
    DigitsAccuracy = 17

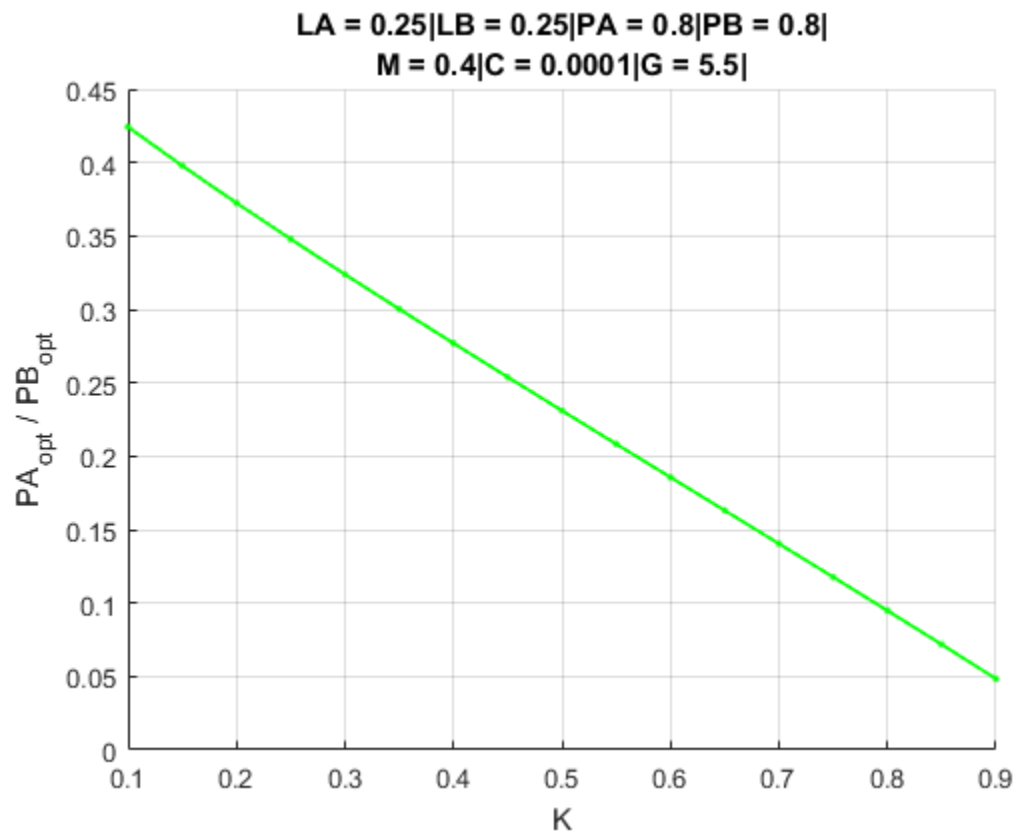
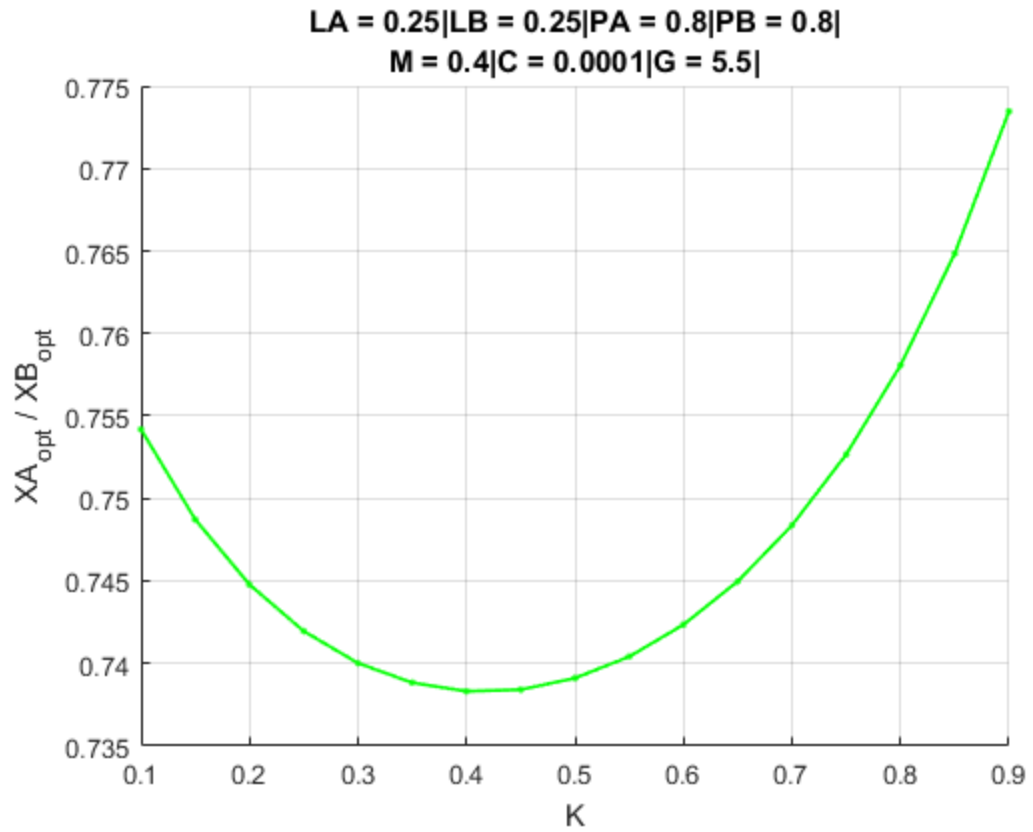
```

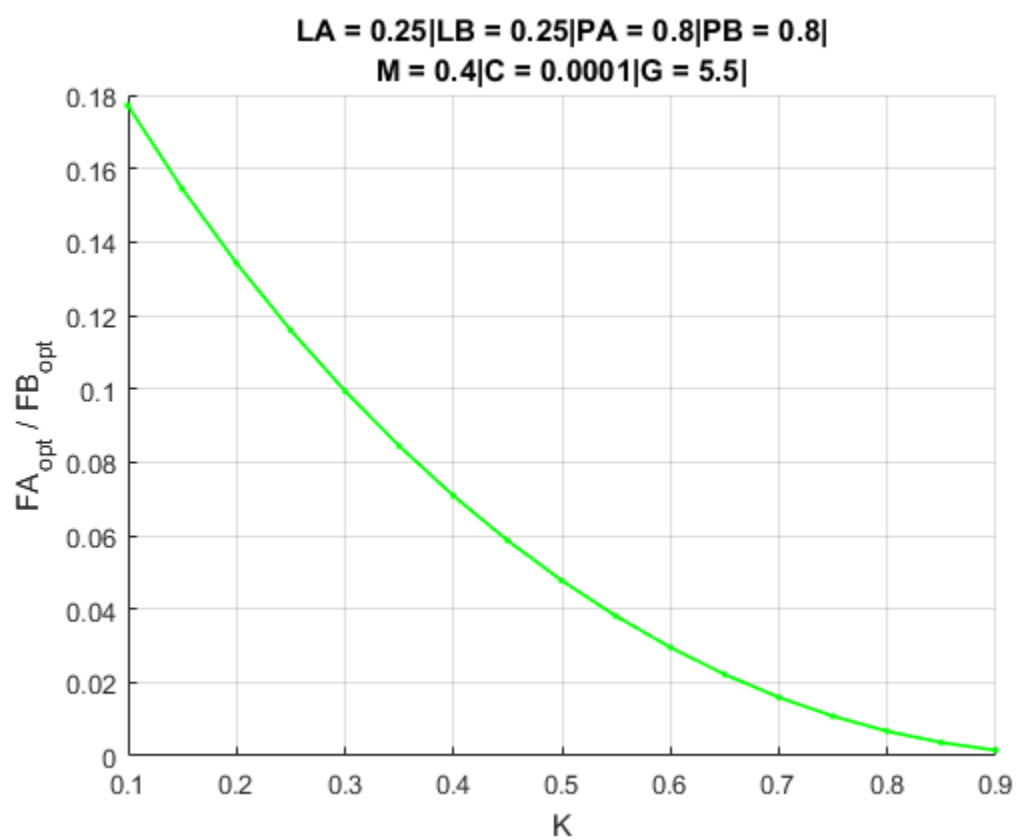
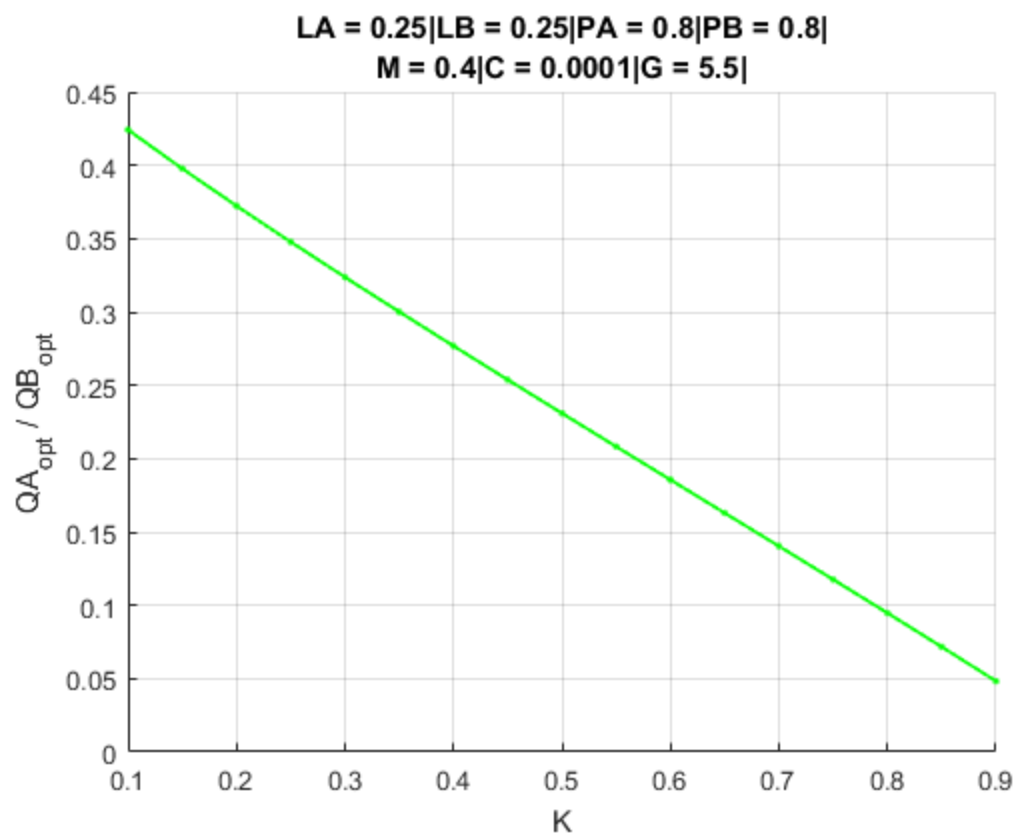
---

---

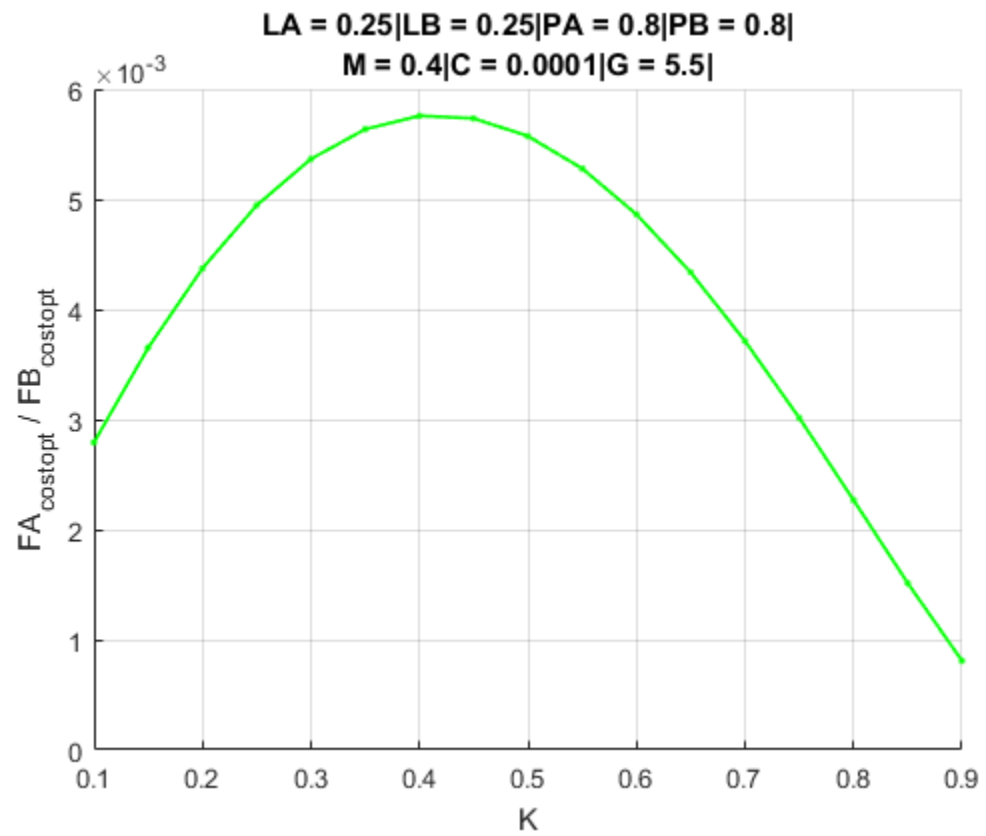
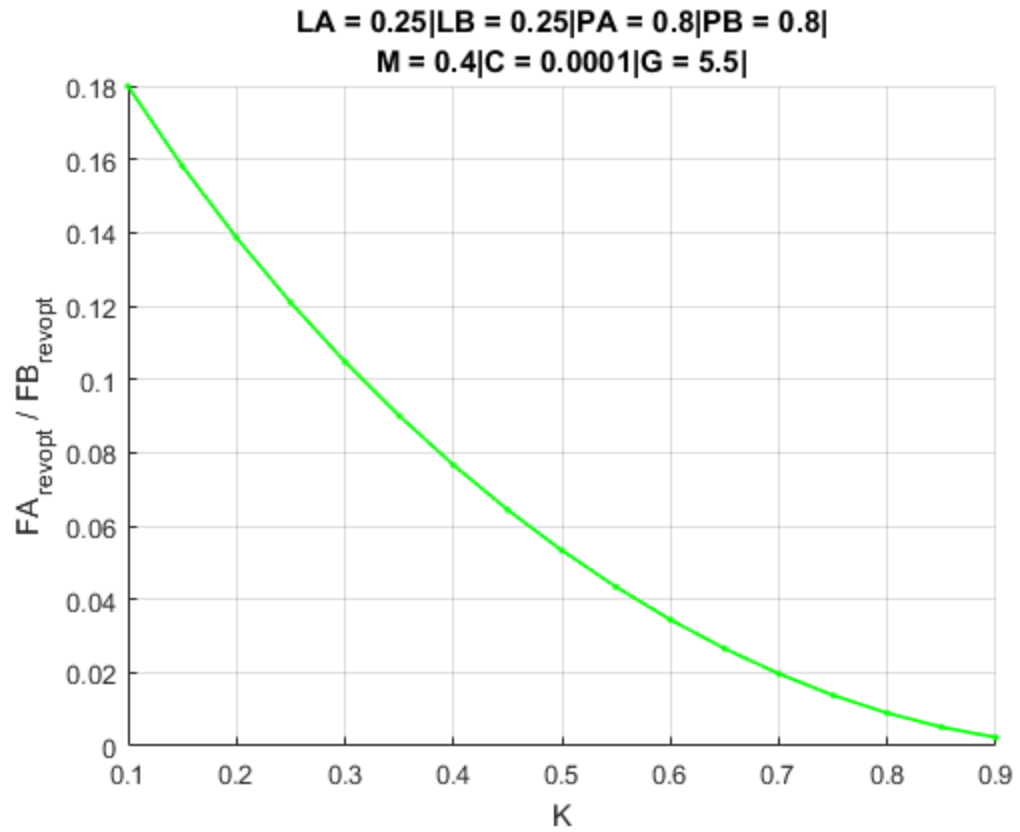
*K = 0.25 TAOpt = 0.029985 TBopt = 0.029985 FilterFlag = 0  
DigitsAccuracy = 16  
K = 0.3 TAOpt = 0.031235 TBopt = 0.031235 FilterFlag = 0  
DigitsAccuracy = 18  
K = 0.35 TAOpt = 0.032012 TBopt = 0.032012 FilterFlag = 0  
DigitsAccuracy = 13  
K = 0.4 TAOpt = 0.032355 TBopt = 0.032355 FilterFlag = 0  
DigitsAccuracy = 16  
K = 0.45 TAOpt = 0.032290 TBopt = 0.032290 FilterFlag = 0  
DigitsAccuracy = 12  
K = 0.5 TAOpt = 0.031830 TBopt = 0.031830 FilterFlag = 0  
DigitsAccuracy = 15  
K = 0.55 TAOpt = 0.030980 TBopt = 0.030980 FilterFlag = 0  
DigitsAccuracy = 14  
K = 0.6 TAOpt = 0.029733 TBopt = 0.029733 FilterFlag = 0  
DigitsAccuracy = 13  
K = 0.65 TAOpt = 0.028076 TBopt = 0.028076 FilterFlag = 0  
DigitsAccuracy = 12  
K = 0.7 TAOpt = 0.025982 TBopt = 0.025982 FilterFlag = 0  
DigitsAccuracy = 12  
K = 0.75 TAOpt = 0.023412 TBopt = 0.023412 FilterFlag = 0  
DigitsAccuracy = 11  
K = 0.8 TAOpt = 0.020311 TBopt = 0.020311 FilterFlag = 0  
DigitsAccuracy = 12  
K = 0.85 TAOpt = 0.016592 TBopt = 0.016592 FilterFlag = 0  
DigitsAccuracy = 12  
K = 0.9 TAOpt = 0.012128 TBopt = 0.012128 FilterFlag = 0  
DigitsAccuracy = 13*











---

*Published with MATLAB® R2017a*