Supplement QM
Multilayer Quantum Well-and-Barrier Analysis Software Program

CONTENTS

I. INTRODUCTION

Supplement QM. This supplement contains the listings of all the Matlab .m files used in determining the reflectance and transmittance characteristics of multilayer quantum well-and-barrier structures. In addition, there are files for displaying graphical results. A number of Matlab script .m files contain examples of various multilayer quantum well-and-barrier structures.

II. MAIN PROGRAM

The main program, called AQM.m, is the first program in the listing.

AQM.m

```
% MAIN PROGRAM NANE: AQM
% Step 1: Specify Initial Quantities
clear; clf; clc; Prelim; Structure;
% Step 2: Draw Quantum Well and Barrier Structure
[L] = DrawMQW(PotentP,PotentQ,PotentF,ThickP,ThickQ,ThickF,DS,CG);
% Step 3: Vary the Vacuum Energy
for E = EnergyMin:EnergyDel:EnergyMax
E = E + Eo;
 % Step 4: Convert EM Quantities to QM Quantities
Convert;
% Step 5: Compute TE and TM Incident Wavefunctions
[PsiI_TE,PsiI_TM] = PsiI(ThetaI,epP,muP);
% Step 6: Region P, Q, and F Calculations
RegionP; RegionQ; RegionF;
% Step 7: Compute Thin-Film Phases and Matrix Q
Phases; MatrixQ;
% Step 8: Compute the Matrices MR AND MT
[MR,MT] = MatrixMRMT(Q);
% Step 9: Compute Reflectance and Transmittance
Wavefunction; Probability;
end
% Step 10: Plot Reflectance and Transmittance
[FIN] = PlotRef(RTE,RTM,EnergyMin,EnergyMax,EnergyDel,ThetaI,C,CG);
[RPB] = PlotTra(TTE,TTM,EnergyMin,EnergyMax,EnergyDel,ThetaI,C,CG);
```

III. MATLAB .M FUNCTION FILES

All the Matlab .m Function files are next listed in alphabetical order.

Angle.m

```
% FUNCTION: Angle
function [Theta] = Angle(ThetaI,IndexP,Index)
Theta = asind((IndexP/Index)*sind(ThetaI));
end
```

Color.m

```
% FUNCTION: Color
function [v] = Color(n,CG)
a = real(n); fff = 1*10^-4;
if CG == 1
if a >= -10.0, v = [1 0 0]; end
if a >= -0.10, v = [1 1 1]; end
if a >=   fff , v = [1 0 1]; end
if a >= +0.10, v = [0 0 1]; end
if a >= +1.00, v = [0 1 0]; end
if a >= 10.00, v = [1 1 0]; end
if a >= 40.00, v = [0 1 1]; end
if a >= 70.00, v = [0 0 0]; end
else
if a >= -10.0, v = [1 1 1]*0.20; end
if a >= -0.10, v = [1 1 1]*0.30; end
if a >= +0.00, v = [1 1 1]*0.80; end
if a >= +0.10, v = [1 1 1]*0.90; end
if a >= +1.00, v = [1 1 1]*0.95; end
if a >= 10.00, v = [1 1 1]*1.00; end
if a >= 40.00, v = [1 1 1]*1.00; end
if a >= 70.00, v = [1 1 1]*1.00; end
end;
end
```

DrawMQW.m

```
% FUNCTION: DrawMQW
function [L] = DrawMQW(NP,NQ,NF,TP,TQ,TF,DS,CG)
Scale = 70;
figure(1); clf
if CG == 1
BC = [1.0 0.9 0.7]; else
BC = [1 1 1]; end
set(1,'Color',BC)
HA = axes('position',[0.05 0.10 0.35 0.80]);
H = sum(TF) + 100;
HP = plot([0 0 1 1],[0 H H 0],'-k');
set(HP,'Color',BC)
hold on
MI =  0; MA = Scale;
HF = fill([0.3 0.3 0.7 0.7],[MI MA MA MI],Color(NP,CG));
HT = text(0.85,(MI+MA)/2,[sprintf('%4.3f',NP) ]);
set(HT,'FontSize',[15],'FontWeight','Bold')
if NP == 0
HT = text(0.1,(MI+MA)/2,'Vacuum'); else
HT = text(0.1,(MI+MA)/2,'Barrier'); end
set(HT,'FontSize',[15],'FontWeight','Bold')
HT = text(0.42,(MI+MA)/2,'Entrance');
set(HT,'FontSize',[15],'FontWeight','Bold')
wsize = size(TF,2);
MI = MA;
for ws = 1:wsize
MA = MI + Scale*TF(ws);
HF = fill([0.3 0.3 0.7 0.7],[MI MA MA MI],Color(NF(ws),CG));
HT = text(0.75,(MI+MA)/2,[sprintf('%4.3f',real(NF(ws)))]);
set(HT,'FontSize',[15],'FontWeight','Bold')
if imag(NF(ws)) > 0
HT = text(0.83,(MI+MA)/2,['+␣' sprintf('%4.3f',imag(NF(ws))) 'i']);
set(HT,'FontSize',[15],'FontWeight','Bold')
end
HT = text(0.2,(MI+MA)/2,[sprintf('%4.2f',TF(ws))]);
set(HT,'FontSize',[15],'FontWeight','Bold')
MI = MA;
end
MA = MI + Scale;
HF = fill([0.3 0.3 0.7 0.7],[MI MA MA MI],Color(NQ,CG));
HT = text(0.85,(MI+MA)/2,[sprintf('%4.3f',NQ) ]);
set(HT,'FontSize',[15],'FontWeight','Bold')
if NQ == 0
HT = text(0.1,(MI+MA)/2,'Vacuum'); else
HT = text(0.1,(MI+MA)/2,'Barrier'); end
set(HT,'FontSize',[15],'FontWeight','Bold')
HT = text(0.47,(MI+MA)/2,'Exit');
set(HT,'FontSize',[15],'FontWeight','Bold')
set(HA,'Color',BC,'XColor',BC,'YColor',BC)
HT = title(DS);
set(HT,'FontSize',[15],'FontWeight','Bold')
HX = xlabel('Thickness␣(nm)␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣P.E.␣(eV)');
set(HX,'FontSize',[15],'FontWeight','Bold','Color',[0 0 0])
L = 1;
end
```

MatrixA.m

```
% FUNCTION: MatrixA
function [A] = MatrixA(ax,ay,az)
A = zeros(6,6);
A(2,6) = -ax; A(3,5) = +ax; A(5,3) = -ax; A(6,2) = +ax;
A(1,6) = +ay; A(3,4) = -ay; A(4,3) = +ay; A(6,1) = -ay;
A(1,5) = -az; A(2,4) = +az; A(4,2) = -az; A(5,1) = +az;
end
```

MatrixC.m

```
% FUNCTION: MatrixC
function [C] = MatrixC(ep,mu)
C = eye(6,6);
C(3,3) = ep; C(6,6) = mu;
end
```

MatrixM.m

```
% FUNCTION: MatrixM
function [M] = MatrixM(NR,NL,C)
M11 = C;      M12 = C*NL;
M21 = C*NR;   M22 = C;
M = [M11 M12; M21 M22];
end
```

MatrixMRMT.m

```
% FUNCTION: MatrixMRMT
function [MR,MT] = MatrixMRMT(Q)
Q11 = Q(1:6,1:6);     Q12 = Q(1:6,7:12);
Q21 = Q(7:12,1:6);    Q22 = Q(7:12,7:12);
MR = Q21*inv(Q11); MT = inv(Q11);
end
```

MatrixN.m

```
% FUNCTION: MatrixN
function [N] = MatrixN(A,W)
N = inv(W)*A;
end
```

MatrixP.m

```
% FUNCTION: MatrixP
function [P] = MatrixP(w,n,theta,hc)
phi = 2*pi*n*(w/hc)*cosd(theta);
P = zeros(12,12);
P(1:6,1:6)    = eye(6,6)*exp(+i*phi);
P(7:12,7:12) = eye(6,6)*exp(-i*phi);
end
```

MatrixW.m

```
% FUNCTION: MatrixW
function [W] = MatrixW(ep,mu,n)
W = zeros(6,6);
W(1,1) = -ep/n; W(2,2) = -ep/n; W(3,3) = -ep/n;
W(4,4) = +mu/n; W(5,5) = +mu/n; W(6,6) = +mu/n;
end
```

PlotRef.m

```
% FUNCTION: PlotRef
function [FIN] = PlotRef(RTE,RTM,LMIN,LMAX,LDEL,THE,C,CG)
x = LMIN:LDEL:LMAX;
HA = axes('position',[0.50 0.50 0.45 0.30]);
if CG == 1
HH = plot(x,100*real(RTE),'-r',x,100*real(RTM),'-b');
else
HH = plot(x,100*real(RTE),'ok',x,100*real(RTM),'xk');
end
set(HH,'LineWidth',[5]); set(HH,'MarkerSize',[1])
Hx = xlabel('K.E.␣=␣E-Eo␣(eV)');
set(Hx,'FontSize',[15],'FontWeight','Bold')
Hy = ylabel('Reflectance␣(␣%␣)');
set(Hy,'FontSize',[15],'FontWeight','Bold')
axis([min(x) max(x) -5 105])
LINC = (LMAX-LMIN)/10;
set(HA,'xtick',[LMIN:LINC:LMAX])
set(HA,'ytick',[0 20 40 60 80 100])
set(HA,'FontSize',[15],'FontWeight','Bold')
set(HA,'LineWidth',[2])
if CG == 1; set(HA,'Color',[0 1 1]); else
set(HA,'Color',[1 1 1]*0.95); end
HT = text(min(x),110,'Spin␣States:␣');
set(HT,'FontSize',[15],'FontWeight','Bold')
if CG == 1
HT = text(min(x),110,'␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣U␣');
set(HT,'FontSize',[15],'FontWeight','Bold','Color',[1 0 0])
HT = text(min(x),110,'␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣L␣');
set(HT,'FontSize',[15],'FontWeight','Bold','Color',[0 0 1])
else
HT = text(min(x),110,'␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣U␣o');
set(HT,'FontSize',[15],'FontWeight','Bold','Color',[0 0 0])
HT = text(min(x),110,'␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣L␣x␣');
set(HT,'FontSize',[15],'FontWeight','Bold','Color',[0 0 0])
end
HT = text(min(x),130.0,['Incid␣=␣␣' sprintf('%4.2f',THE) '␣Deg']);
set(HT,'FontSize',[15],'FontWeight','Bold')
HT = text(min(x),150.0,'Richard␣P.␣Bocker,␣Ph.D.');
set(HT,'FontSize',[15],'FontWeight','Bold')
HT = text(min(x),140.0,'Carlsbad,␣California,␣USA');
set(HT,'FontSize',[15],'FontWeight','Bold')
HT = text(min(x),160.0,'Multiple␣Quantum␣Well-Barrier␣Analysis');
set(HT,'FontSize',[15],'FontWeight','Bold'); clc
grid on
FIN = 'Finished';
end
```

PlotTra.m

```
% FUNCTION: PlotTra
function [RPB] = PlotTra(TTE,TTM,LMIN,LMAX,LDEL,THE,C,CG)
x = LMIN:LDEL:LMAX;
HA = axes('position',[0.50 0.10 0.45 0.30]);
if CG == 1
HH = plot(x,100*real(TTE),'-r',x,100*real(TTM),'-b');
else
HH = plot(x,100*real(TTE),'ok',x,100*real(TTM),'xk');
end
set(HH,'LineWidth',[5]); set(HH,'MarkerSize',[1])
Hx = xlabel('K.E.␣=␣E-Eo␣(eV)');
set(Hx,'FontSize',[15],'FontWeight','Bold')
Hy = ylabel('Transmittance␣(␣%␣)');
set(Hy,'FontSize',[15],'FontWeight','Bold')
axis([min(x) max(x) -5 105])
LINC = (LMAX-LMIN)/10;
set(HA,'xtick',[LMIN:LINC:LMAX])
set(HA,'ytick',[0 20 40 60 80 100])
set(HA,'FontSize',[15],'FontWeight','Bold')
set(HA,'LineWidth',[2])
if CG == 1; set(HA,'Color',[0 1 1]); else
set(HA,'Color',[1 1 1]*0.95); end
grid on
RPB = 'Thats␣all␣Folks'; clc
end
```

PsiI.m

```
% FUNCTION: PsiI
function [PsiI_TE,PsiI_TM] = PsiI(ThetaI,epP,muP)
R = sqrt(epP/muP); S = sqrt(muP/epP);
Vec_TE = [0 1 0 +R*cosd(ThetaI) 0 -R*sind(ThetaI)].';
Vec_TM = [-S*cosd(ThetaI) 0 +S*sind(ThetaI) 0 1 0].';
PsiI_TE = Vec_TE/sqrt(Vec_TE'*Vec_TE);
PsiI_TM = Vec_TM/sqrt(Vec_TM'*Vec_TM);
end
```

IV. MATLAB .M SCRIPT FILES

Matlab .m Script files are next listed in alphabetical order.

Convert.m

```
% SCRIPT: Convert
% Conversion of EM Quantities to QM Quantities
% Introduce Relativistic Lorentz Factor
UP = PotentP; UQ = PotentQ; UF = PotentF;
gP = (E - UP)/Eo; gQ = (E - UQ)/Eo; gF = (E - UF)/Eo;
epP = (gP + 1)*Eo; muP = (gP - 1)*Eo;
epQ = (gQ + 1)*Eo; muQ = (gQ - 1)*Eo;
epF = (gF + 1)*Eo; muF = (gF - 1)*Eo;
IndexP = sqrt(epP.*muP);
IndexQ = sqrt(epQ.*muQ);
IndexF = sqrt(epF.*muF);
pc = sqrt(gF + 1).*sqrt(gF - 1)*Eo;
```

MatrixQ.m

```
% SCRIPT: MatrixQ
Q = inv(MIR);
for n = 1:N
Q = Q*Mn(:,:,n)*inv(Pn(:,:,n))*inv(Mn(:,:,n));
end
Q = Q*MTT;
```

Phases.m

```
% SCRIPT: Phases
for n = 1:N
[Pn(:,:,n)] = MatrixP(ThickF(n),pc(n),ThetaF(n),hc);
end
```

Prelim.m

```
% SCRIPT: Prelim
disp('␣Matlab:␣Multiple␣Quantum␣Well-and-Barrier␣Analysis␣Program')
disp('␣Author:␣Richard␣P.␣Bocker,␣Ph.D.')
disp('␣'); disp('␣Begin␣Calculations'); disp('␣')
c      = 3.00*10^8;          % Speed of Light in Vacuum (m/s)
h      = 4.14*10^-15;        % Planck Constant (eV*s)
Me     = 9.11*10^-31;        % Electron Rest Mass (kg)
Eo     = 5.12*10^5;          % Electron Rest Mass Energy (eV)
hc     = 1.241*10^3;         % (eV*nm)
Mecc = 0.511*10^6;           % (eV)
ThetaI = 0.00;               % (degrees)
CG = input('␣Type␣1␣for␣Color␣or␣2␣for␣Gray␣Plots:␣');
```

Probability.m

```
% SCRIPT: Probability
AA = PsiR_TE; BB = PsiI_TE; CC = PsiR_TM; DD = PsiI_TM;
EE = PsiT_TE; FF = PsiT_TM;
NUM = sqrt(AA(1:3)'*AA(1:3))*sqrt(AA(4:6)'*AA(4:6));
DEN = sqrt(BB(1:3)'*BB(1:3))*sqrt(BB(4:6)'*BB(4:6));
RTE(L) = (NUM/DEN);
NUM = sqrt(CC(1:3)'*CC(1:3))*sqrt(CC(4:6)'*CC(4:6));
DEN = sqrt(DD(1:3)'*DD(1:3))*sqrt(DD(4:6)'*DD(4:6));
RTM(L) = (NUM/DEN);
NUM = sqrt(EE(1:3)'*EE(1:3))*sqrt(EE(4:6)'*EE(4:6));
DEN = sqrt(BB(1:3)'*BB(1:3))*sqrt(BB(4:6)'*BB(4:6));
TTE(L) = (NUM/DEN)*Ratio;
NUM = sqrt(FF(1:3)'*FF(1:3))*sqrt(FF(4:6)'*FF(4:6));
DEN = sqrt(DD(1:3)'*DD(1:3))*sqrt(DD(4:6)'*DD(4:6));
TTM(L) = (NUM/DEN)*Ratio;
clear AA BB CC DD EE FF
L = L + 1;
```

RegionF.m

```
% SCRIPT: RegionF
for n = 1:N
[ThetaF(n)] = Angle(ThetaI,IndexP,IndexF(n));
[CF]   = MatrixC(1,1);
[WF]   = MatrixW(epF(n),muF(n),IndexF(n));
[AFR]  = MatrixA(cosd(90-ThetaF(n)),0,cosd(ThetaF(n)));
[AFL]  = MatrixA(cosd(90-ThetaF(n)),0,cosd(180-ThetaF(n)));
[NFR]  = MatrixN(AFR,WF);
[NFL]  = MatrixN(AFL,WF);
[Mn(:,:,n)] = MatrixM(NFR,NFL,CF);
end
```

RegionP.m

```
% SCRIPT: RegionP
[CP]   = MatrixC(1,1);
[WP]   = MatrixW(epP,muP,IndexP);
[APR]  = MatrixA(cosd(90-ThetaI),0,cosd(ThetaI));
[APL]  = MatrixA(cosd(90-ThetaI),0,cosd(180-ThetaI));
[NPR]  = MatrixN(APR,WP);
[NPL]  = MatrixN(APL,WP);
[MIR]  = MatrixM(NPR,NPL,CP);
```

RegionQ.m

```
% SCRIPT: RegionQ
[ThetaT] = Angle(ThetaI,IndexP,IndexQ);
[CQ] = MatrixC(1,1);
[WQ] = MatrixW(epQ,muQ,IndexQ);
[AQR] = MatrixA(cosd(90-ThetaT),0,cosd(ThetaT));
[AQL] = MatrixA(cosd(90-ThetaT),0,cosd(180-ThetaT));
[NQR] = MatrixN(AQR,WQ);
[NQL] = MatrixN(AQL,WQ);
[MTT] = MatrixM(NQR,NQL,CQ);
Ratio = cosd(ThetaT)/cosd(ThetaI);
```

Structure.m

```
% SCRIPT: Structure
disp('␣Following␣Examples␣Illustrate␣Software␣Capability:')
disp('␣Code␣01:␣MQW01:␣Layers␣01:␣Finite␣Qunatum␣Barrier')
disp('␣Code␣02:␣MQW02:␣Layers␣01:␣Finite␣Quantum␣Well')
disp('␣Code␣03:␣MQW03:␣Layers␣03:␣Quantum␣Barrier␣Tunneling')
disp('␣Code␣04:␣MQW04:␣Layers␣03:␣Finite␣Quantum␣Well')
disp('␣Code␣05:␣MQW05:␣Layers␣07:␣Well-and-Barrier␣Structure')
disp('␣Code␣06:␣MQW06:␣Layers␣11:␣Scattering-Periodic␣Potential')
C = input('␣Type␣in␣Two-Digit␣SUB␣Structure␣Code␣Number:␣␣');
if C == 01; MQW01; end; if C == 02, MQW02; end
if C == 03, MQW03; end; if C == 04, MQW04; end
if C == 05, MQW05; end; if C == 06, MQW06; end
```

Wavefunction.m

```
% SCRIPT: Wavefunction
PsiR_TE = MR*PsiI_TE;
PsiR_TM = MR*PsiI_TM;
PsiT_TE = MT*PsiI_TE;
PsiT_TM = MT*PsiI_TM;
```

V. MATLAB .M SCRIPT FILES - EXAMPLES

In addition, there are six Matlab .m Script files depicting examples of different multilayer quantum well-and-barrier structures. One can easily add more to this list by examining the content of these files as well as the Structure.m script file.

MQW01.m

```
% SCRIPT: MQW01
DS = 'MQW01:␣Finite␣Quantum␣Barrier␣(␣1␣Layer␣)';
N = 1;                      % Number of Thin-Film Layers
EnergyMin =   0.00;         % Minimum Particle Energy (eV)
EnergyMax =  80.00;         % Maximum Particle Energy (eV)
EnergyDel =   0.10;         % Energy Increment (eV)
ThickP  = 0.20;             % Region P Gap Width (nm)
PotentP = 0.00;             % Region P Potential Energy (eV)
ThickQ  = 0.20;             % Region Q Gap Width (nm)
PotentQ = 0.00;             % Region Q Potential Energy (eV)
ThickF  = [0.20];           % Thin-Film Thickness (nm)
PotentF = [40.0];           % Thin-Film Potential Energy (eV)
```

MQW02.m

```
% SCRIPT: MQW02
DS = 'MQW02:␣Finite␣Quantum␣Well␣(␣1␣Layer␣)';
N = 1;                      % Number of Thin-Film Layers
EnergyMin =   0.00;         % Minimum Particle Energy (eV)
EnergyMax =   8.00;         % Maximum Particle Energy (eV)
EnergyDel =   0.01;         % Energy Increment (eV)
ThickP  = 1.00;             % Region P Gap Width (nm)
PotentP = 0.00;             % Region P Potential Energy (eV)
ThickQ  = 1.00;             % Region Q Gap Width (nm)
PotentQ = 0.00;             % Region Q Potential Energy (eV)
ThickF  = [1.50];           % Thin-Film Thickness (nm)
PotentF = [-3.00];          % Thin-Film Potential Energy (eV)
```

MQW03.m

```
% SCRIPT: MQW03
DS = 'MQW03:␣Quantum␣Barrier␣Tunneling␣(␣3␣Layers␣)';
TFT = [0.25  0.50  0.25];
TFE = [10.0  0.00  10.0];
N = 3;                       % Number of Thin-Film Layers
EnergyMin =    0.00;         % Minimum Particle Energy (eV)
EnergyMax =   16.00;         % Maximum Particle Energy (eV)
EnergyDel =    0.004;        % Energy Increment (eV)
ThickP  = 0.50;              % Region P Gap Width (nm)
PotentP = 0.00;              % Region P Potential Energy (eV)
ThickQ  = 0.50;              % Region Q Gap Width (nm)
PotentQ = 0.00;              % Region Q Potential Energy (eV)
ThickF  = TFT;               % Thin-Film Thicknesses (nm)
PotentF = TFE;               % Thin-Film Potential Energies (eV)
clear TFT TFE
```

MQW04.m

```
% SCRIPT: MQW04
DS = 'MQW04:␣Quantum␣Well␣Energy␣Levels␣(␣3␣Layers␣)';
vvv = 1.00;
aaa = sqrt((49*hc*hc)/(128*Mecc*vvv));
TFT = [1  2  1];
TFE = [1  0  1];
N = 3;                       % Number of Thin-Film Layers
EnergyMin =    0.00;         % Minimum Particle Energy (eV)
EnergyMax =    2.00;         % Maximum Particle Energy (eV)
EnergyDel =    0.001;        % Energy Increment (eV)
ThickP  = 5.00;              % Region P Gap Width (nm)
PotentP = 0.00;              % Region P Potential Energy (eV)
ThickQ  = 5.00;              % Region Q Gap Width (nm)
PotentQ = 0.00;              % Region Q Potential Energy (eV)
ThickF  = TFT*aaa;           % Thin-Film Thicknesses (nm)
PotentF = TFE*vvv;           % Thin-Film Potential Energies (eV)
clear vvv aaa TFT TFE
```

MQW05.m

```
% SCRIPT: MQW05
DS = 'MQW05:␣Well-and-Barrier␣Structure␣(␣7␣Layers␣)';
TFT = [2.5  1  1.5  1  1.5  1  2.5]*0.3;
TFE = [.5  0  1  0  1  0  .5]*2.0;;
N = 7;                       % Number of Thin-Film Layers
EnergyMin =    0.00;         % Minimum Particle Energy (eV)
EnergyMax =    3.00;         % Maximum Particle Energy (eV)
EnergyDel =    0.003;        % Energy Increment (eV)
ThickP  = 1.00;              % Region P Gap Width (nm)
PotentP = 0.00;              % Region P Potential Energy (eV)
ThickQ  = 1.00;              % Region Q Gap Width (nm)
PotentQ = 0.00;              % Region Q Potential Energy (eV)
ThickF  = TFT;               % Thin-Film Thicknesses (nm)
PotentF = TFE;               % Thin-Film Potential Energies (eV)
clear TFT TFE
```

MQW06.m

```
% SCRIPT: MQW06
DS = 'MQW06:␣Scattering-Periodic␣Potential␣(␣11␣Layers␣)';
TFT = [1  1  1  1  1  1  1  1  1  1  1]*0.3;
TFE = [1  0  1  0  1  0  1  0  1  0  1]*2.0;
N = 11;                        % Number of Thin-Film Layers
EnergyMin =     0.00;          % Minimum Particle Energy (eV)
EnergyMax =     4.00;          % Maximum Particle Energy (eV)
EnergyDel =     0.004;         % Energy Increment (eV)
ThickP   = 1.00;               % Region P Gap Width (nm)
PotentP = 0.00;                % Region P Potential Energy (eV)
ThickQ   = 1.00;               % Region Q Gap Width (nm)
PotentQ = 0.00;                % Region Q Potential Energy (eV)
ThickF   = TFT;                % Thin-Film Thicknesses (nm)
PotentF = TFE;                 % Thin-Film Potential Energies (eV)
clear TFT TFE
```