```matlab
% Compute 95% confidence interval for smallest delta_t = 1/250

% Define simulation and delta
M = [1000, 2000, 4000, 8000, 16000, 32000, 64000]; % number of simulations
delta = [5/250, 2.5/250, 1/250];



% Create a matrix to store bound of call & put option for each simulation
lowerbnd_call = ones(length(M),1);
upperbnd_call = ones(length(M),1);
lowerbnd_put = ones(length(M),1);
upperbnd_put = ones(length(M),1);


for i = 1:length(M)
  % Call MonteCarlo function to calculate option value and standard deviation
  [Call_Value(i,3), Put_Value(i,3),StdCall(i,3),StdPut(i,3)]...
      = MonteCarlo(M(i),delta(3));
   % Compute the lower & upper bound for 95% CI: mean(Y) ± 1.96 * std(Y)/sqrt(num of simulations)
   lowerbnd_call(i) = Call_Value(i,3) - 1.96 * StdCall(i,3)/sqrt(M(i));
   upperbnd_call(i) = Call_Value(i,3) + 1.96 * StdCall(i,3)/sqrt(M(i));
   lowerbnd_put(i) = Put_Value(i,3) - 1.96 * StdPut(i,3)/sqrt(M(i));
   upperbnd_put(i) = Put_Value(i,3) + 1.96 * StdPut(i,3)/sqrt(M(i));
end

% Rename variable name for table creation purpose
Simulation_M = M';
CallValue = Call_Value(:,3);
Upper_Bound_Call = upperbnd_call;
Lower_Bound_Call = lowerbnd_call;

% Create table for 95% CI of call opti
CI_for_Call = table(Simulation_M, CallValue, Lower_Bound_Call, Upper_Bound_Call);
disp(CI_for_Call)

% Rename variable name for table creation purpose
PutValue = Put_Value(:,3);
Upper_Bound_Put = upperbnd_put;
Lower_Bound_Put = lowerbnd_put;
% Create table for 95% CI of put option
CI_for_Put = table(Simulation_M, PutValue, Lower_Bound_Put, Upper_Bound_Put);
disp(CI_for_Put)
```