Computational Approach to solving Krusell-Smith:

1. Initialize the algorithm: Set parameters, grid bounds, number of grid points. States of the economy are $S = (k, \epsilon, K, z)$. Conjecture a law of motion for how average capital evolves. Guess values for a_0, a_1, b_0, b_1. Initialize capital policy function and value function. Draw a panel of shocks $\mathcal{E}$ – this is used to simulate your model.

   – Note: It is important that you draw a panel of shock up-front. Why? When you simulate, you want to make sure your economy is facing the "same shocks" so you can converge on values of a_0, a_1, b_0, b_1 (the objects we are interested in solving for). If you don't, the code won't converge.

2. Value Function Iteration: Take regression parameters a_0, a_1, b_0, b_1 and the law of motion for mean capital as given. Solve household's decision problem. Once done, you'll have a capital policy function and value function: $\{V(k, \epsilon, K, z), k'(k, \epsilon, K, z))\}$

   – Note: To solve the HH's problem, we're going to need to rely on interpolation and function minimization techniques. K' — which we need to know to calculate the continuation value $(E_{\epsilon,z}[V(k', \epsilon, K', z)])$ — likely does not fall on one of our pre-specified average capital (K) grid points. So, we need to interpolate what the continuation value would be.

3. Simulate Model: Take value and capital policy function solved for in (2) and panel of shocks $\mathcal{E}$ that was obtained when you initialized the program. Simulate a panel of data on capital choice and calculate a time series of average capital $\mathcal{K}$. Given $\mathcal{K}$, re-estimate regression coefficients $\hat{a}_0, \hat{a}_1, \hat{b}_0, \hat{b}_1$.

   – Note: Don't forget that you'll need to sort your data based on which aggregate state of the world the economy is in, since you have two separate regression equations to estimate.

If $|\hat{a}_0 - a_0| + |\hat{a}_1 - a_1| + |\hat{b}_0 - b_0| + |\hat{b}_1 - b_1| < \tilde{\varepsilon}$, you are done. Otherwise, update regression coefficients and repeat steps (2) - (3) until convergence.

Pseudo Code to Solve Krusell-Smith:

Algorithm 1 Krusell-Smith

procedure MAIN CODE
    call DRAWSHOCKS( )
    return $\mathcal{E}$

    $a_0 = a_0^{init}, a_1 = a_1^{init}, b_0 = b_0^{init}, b_1 = b_1^{init}$
    convergence flag $= 0$
    while convergence flag $= 0$ **do**
        call VFI( )
        return $\{V(k, \epsilon, K, z), k'(k, \epsilon, K, z)\}$

        call SIMULATECAPITALPATH( )
        return $\mathcal{K}$

        call ESTIMATEREGRESION( )
        return $\{\hat{a}_0, \hat{a}_1, \hat{b}_0, \hat{b}_1\}$

        if $|\hat{a}_0 - a_0| + |\hat{a}_1 - a_1| + |\hat{b}_0 - b_0| + |\hat{b}_1 - b_1| > \tilde{\varepsilon}$ **then**   ▷ Try $\lambda = 0.5$
            $a_0 \leftarrow \lambda \hat{a}_0 + (1 - \lambda) a_0$
            $a_1 \leftarrow \lambda \hat{a}_1 + (1 - \lambda) a_1$
            $b_0 \leftarrow \lambda \hat{b}_0 + (1 - \lambda) b_0$
            $b_1 \leftarrow \lambda \hat{b}_1 + (1 - \lambda) b_1$
        else if $|\hat{a}_0 - a_0| + |\hat{a}_1 - a_1| + |\hat{b}_0 - b_0| + |\hat{b}_1 - b_1| < \tilde{\varepsilon}$ **then**
            convergence flag $= 1$
        end if

    end while
end procedure

function DrawShocks( )
  See `HelpfulFunctuons` file
  return $\mathcal{E}$
end function

function VFI( )
  See `HelpfulFunctuons` file
  return $\{V(k,\epsilon,K,z), k'(k,\epsilon,K,z)\}$
end function

function SimulateCapitalPath( )

person/time	z_1	z_2	...
$n=1$	$k_2^1 = g^i(K^{ss}, \varepsilon_1^1, K^{ss}, z_1)$	$k_3^1 = g^i(k_2^1, \varepsilon_2^1, \overline{K}_2, z_2)$	...
$n=2$	$k_2^2 = g^i(K^{ss}, \varepsilon_1^2, K^{ss}, z_1)$	$k_3^2 = g^i(k_2^2, \varepsilon_2^2, \overline{K}_2, z_2)$	...
...	...	...	...
$n=N$	$k_2^N = g^i(K^{ss}, \varepsilon_1^N, K^{ss}, z_1)$	$k_3^N = g^i(k_2^N, \varepsilon_2^N, \overline{K}_2, z_2)$	...
Agg. Capital	$\overline{K}_2 = \frac{1}{N}\sum_{n=1}^N k_2^n$	$\overline{K}_3 = \frac{1}{N}\sum_{n=1}^N k_3^n$	

  $\mathcal{K} = [\bar{K}_2, \bar{K}_3, \ldots \bar{K}_T]$
  return $\mathcal{K}$
end function

function EstimateRegresion( )
  Sort $\mathcal{K}$ based on aggregate state z
  Re-estimte regression
  return $\{\hat{a}_0, \hat{a}_1, \hat{b}_0, \hat{b}_1\}$
end function