Here, we'll document the steps taken in order to use the Business cycle accounting paper in analyzing the great resession. The steps taken will be as follows:

1. First measure wedges, using data, together with the equilibrium conditions. I.e, solve the model using an LQ approximation.
2. Use the state space representation and the law of motion for states and controls in order to estimate wedges via the Kalman filter.
3. re-estimte the model with the given wedges and shock the model to assess system behavior against the data.

LQ approximation

In order to solve the model and get the relationships of endogenous variables to other variables in the models, we'll use the variant of Vaughan in which we start with the first order conditions that equilibrium must satisfy given the model below:

$$\max_{\{c_t,k_{t+1},h_t\}} \quad E\sum_{t=0}^{\infty}\hat{\beta}^t\left\{\log(c_t)+\phi\log(1-h_t)\right\}$$
$$st: \quad \hat{c}_t+(1+\tau_{xt})(\hat{k}_{t+1}(1+\gamma_n)(1+\gamma_z)-(1-\delta)k_t)=r_t\hat{k}_t+(1-\tau_{ht})w_th_t+\kappa_t$$
$$S_t=PS_{t-1}+Q\epsilon_t, \quad S_t=[\log z_t,\tau_{ht},\tau_{xt},\log g_t]$$
$$c_t,k_{t+1}\geq 0 \text{ in all states}$$

Then, the equations that define the detrendend equilibrium conditions can be written as follows:

$$\hat{y}_t=\hat{k}_t^{\theta}(z_th_t)^{1-\theta}=\hat{c}_t+\hat{k}_{t+1}(1+\gamma_n)(1+\gamma_z)-(1-\delta)\hat{k}_t+\hat{g}_t \tag{1}$$
$$\frac{(1+\tau_{xt})}{c_t}=\hat{\beta}E_t\left[\frac{1}{\hat{c}_{t+1}}\left(\theta\hat{k}_{t+1}^{\theta-1}(h_{t+1}z_{t+1})^{1-\theta}+(1+\tau_{xt+1})(1-\delta)\right)\right] \tag{2}$$
$$\frac{\psi}{1-h_t}=(1-\tau_{ht})(1-\theta)\hat{k_t}^{\theta}h_t^{-\theta}z_t^{1-\theta} \tag{3}$$

We can linearize these equations and find a solution of the following form:

$$X_{t+1}=AX_t+BS_t \tag{4}$$
$$Z_t=CX_t+DS_tlabeleq2 \tag{5}$$
$$S_t=PS_t+Q\epsilon_t \tag{6}$$

with $X_t=\tilde{k}_t, \quad \tilde{Z}_t=[\tilde{k}_{t+1};\tilde{h}t], \quad S_t=[\tilde{z}_t;\tau_{ht};\tau_{xt};\tilde{g}_t]$ where $\tilde{x}_t=\log\frac{x_t}{x^*}$ and x^* is the steady state level of x_t. In order to render equations 1-2 into their log-linear representation, I take the log of the equation and then approximate it by a first order taylor approximation. Then for some two variable function $f(x_t,y_t)=1$, we have:

$$\log f(x_t,y_t)=\log f(x^*,y^*)+\frac{df(x^*,y^*)}{dx_t}\frac{x^*}{f(x^*,y^*)}\underbrace{\frac{(x_t-x^*)}{x^*}}_{=\log\frac{x_t}{x^*}}+\frac{df(x^*,y^*)}{dy_t}\frac{y^*}{f(x^*,y^*)}\underbrace{\frac{(y_t-y^*)}{y^*}}_{=\log\frac{y_t}{y^*}}$$
$$=\frac{df(x^*,y^*)}{dx_t}\frac{x^*}{f(x^*,y^*)}\tilde{x}_t+\frac{df(x^*,y^*)}{dy_t}\frac{y^*}{f(x^*,y^*)}\tilde{y}_t$$

After linearizing equations 1-3, we can use 1 to solve for $\hat{c}_t$ plug this into equations 2 and 3 as follows:

$$0=a_1\tilde{k}_t+a_2\tilde{k}_{t+1}+a_3\tilde{h}_t+a_4\tilde{z}_t+a_5\tilde{\tau}_{ht}+a_6\tilde{g}_t \tag{7}$$
$$0=b_1\tilde{k}_t+b_2\tilde{k}_{t+1}+b_3\tilde{k}_{t+2}+b_4\tilde{h}_t+b_5\tilde{h}_{t+1}+b_6\tilde{z}_t+b_7\tilde{\tau}_{xt}+b_8\tilde{g}_t+b_9\tilde{z}_{t+1}+b_{10}\tilde{\tau}_{xt+1}+b_{11}\tilde{g}_{t+1} \tag{8}$$

which can be placed in matrix form as:

$$0=\underbrace{\begin{bmatrix}1&0&0\\0&0&0\\0&b_3&b_5\end{bmatrix}}_{A_1}\begin{bmatrix}\tilde{k}_{t+1}\\\tilde{k}_{t+2}\\\tilde{h}_{t+1}\end{bmatrix}+\underbrace{\begin{bmatrix}0&-1&0\\a_1&a_2&a_3\\b_1&b_2&a_4\end{bmatrix}}_{A_2}\begin{bmatrix}\tilde{k}_t\\\tilde{k}_{t+1}\\\tilde{h}_t\end{bmatrix}+\text{stochastic terms}$$

Then we solve for A by getting the generalized eigenvalues and eigenvectors of A_2 and $-A_1$ since A_1 is not invertible and we can't just take the eigenvalues of $-A_1^{-1}A_2$. Then we should have the same number of eigen values in the unit circle (positive and smaller than 1) as the number of endogenous states in our model - k_t only in this case. Hence if V is the eigenvector, we can reorder so that for the eigenvalue diagonal matrix Λ, the eigen value in the unit circle is in $\Lambda(1,1)$ and $V(1,1)$ is the column vector associated with $\Lambda(1,1)$ of length equal to number of states, so that we have the follwing solutions for A and C

$$A = V_{11}\Lambda(1,1)V_{11}^{-1}$$
$$C = V_{21}V_{11}^{-1}$$

and V_{21} is the rest of the column vector associated with $\Lambda(1,1)$. We can then substitute equations 15 - 11 into the first order condiions - equations 7 and 8. This will form a linear system in B and D. We can then solve for the parameters embedded in B and D that define how control and states respond to exogenous shocks. So after solving for this, we have the the system of equations in 15 - 11 define the evolution of states and controls with respect to shocks. Now, we have to estimate the parameters that define the evolution of these exogenous stochastc variables. We will use the Kalman filter for this.

kalman filter

Now, we'll set up how we'll estimate the parameters of the model with the Kalman filter. These parameters are stacked up in the vector Θ. Then we can write the likelihood functionas follows:

$$L(\Theta) = \sum_{t=0}^{T-1} \left\{ \log|\Omega_t| + trace\left(\Omega_t^{-1}u_t u_t'\right) \right\}$$

In order to build the likelihood function, suppose that X_t is our (full) state vector and we have data on Y_t. Then, we'll start with the following state space representation:

$$X_{t+1} = AX_t + BS_t \tag{9}$$
$$Y_t = CX_t + \omega_t \tag{10}$$
$$\omega_t = D\omega_t + \eta_t \tag{11}$$

We need to estimate A, B, C, D and R where $E\eta_t\eta_t' = R$. Then we can make the following redefinition $\bar{Y}_t = Y_{t+1} - DY_t$ and rewrite the system as:

$$X_{t+1} = AX_t + B\epsilon_{t+1} \tag{12}$$
$$\bar{Y}_t = \bar{C}X_t + CB\epsilon_{t+1} + \eta_{t+1} \tag{13}$$

However, the system that we'll use in the computer will be in the following form:

$$\hat{X}_{t+1} = A\hat{X}_t + K_t u_t$$
$$\bar{Y}_t = \bar{C}\hat{X}_t + u_t$$

where K_t is the kalman gain. The procedure will be to start with some initial X_0 and Σ_0 and in each iteration, update these two matrices and add them to the lekilihood function. u_t measures how far our estimate, measured by $\bar{C}X$ are from what we observe in the data, i.e, what we see in Y_t. The Kalman gain is given by:

$$K_t = \underbrace{(BB'C' + A\Sigma_t\bar{C}')}_{=\Sigma_{X_{t+1},u_t}}\Omega_t^{-1}$$

where Ω_t and Σ_t are time-dependent covariance matrices for u_t and $X_t - \hat{X}_t$, respectively and are given by:

$$\Omega_t = \bar{C}\Sigma_t\bar{C}' + R + CBB'C' = \Sigma_{u_t,u_t}$$
$$\Sigma_t = A\Sigma_t A' + BB' - (BB'C' + A\Sigma_t\bar{C}')\Omega_t^{-1}(\bar{C}\Sigma_t A' + CBB')$$

Then we can finally update our X_{t+1}

$$\hat{X}_{t+1} = A\hat{X}_t + K_t u_t \tag{14}$$

Now in order to update Σ_{t+1}, we just need to rewrite $X_{t+1} - \hat{X}_{t+1}$ and take expectations so that:

$$\begin{aligned}
&E\left(X_{t+1} - \hat{X}_{t+1}\right)\left(X_{t+1} - \hat{X}_{t+1}\right)' = \\
&(A - K_t\bar{C})(X_t - \hat{X}_t)(X_t - \hat{X}_t)'(A - K_t\bar{C})' + BB' - BB'C'K_t' - K_tCBB'K_tCBBC'K_t' + K_tRK_t' \Rightarrow \\
&\Sigma_{t+1} = (A - K_t\bar{C})\Sigma_t(A - K_t\bar{C})' + BB' - BB'C'K_t' - K_tCBB'K_tCBBC'K_t' + K_tRK_t'
\end{aligned}$$

Finite Element method

Consider the following problem:

$$\begin{aligned}
\max_{\{c_t, k_{t+1}\}} \quad & \sum_{t=0}^{\infty} \beta^t \log(c_t) \\
st: \quad & c_t + k_{t+1} - (1-\delta)k_t = Ak_t^{\alpha} \\
& c_t, k_{t+1} \geq 0 \text{ in all states}
\end{aligned}$$

We want to solve for the consumption function that solves this problem with weighted residual methods. In fact, imagine we have the residual equation below:

$$R(x;\theta) = F(d^n(x;\theta))$$

where this equation we can think of as the first order conditions of a functional equation and we can think of $d^n(x;\theta)$ as the policy functions associated with the functional equation. These weighted residual methods, get the residuals close to zero in the integral sense, i.e.

$$\int_{\Omega} \phi_i(x) R(x;\theta)dx = 0, \quad i = 0, 1, ..., n$$

In this class of models, there are three sets of weight functions that will help us to pin down the $\theta's$. These are:

1. Least Squares: $\phi_i = dR(x;\theta)/d\theta_i$
2. Collocation: $\phi_i = \delta(x - x_i)$ where δ is the Dirac delta function. This set of weights implies that the residuals is set to zero at n points $x_1, ..., x_n$ called collocation points.
3. Galerkin: $\phi_i = \psi_i(x)$, which implies that the set of weight functions is the same as the basis functions used to represent d

In this vein and returning to the original problem, we can write the euler equation as follows:

$$\begin{aligned}
\frac{1}{c_t} &= \frac{\beta}{c_{t+1}}(1 - \delta + \alpha A k_{t+1}^{\alpha-1}) \\
\frac{1}{c(k)} &= \frac{\beta}{c(Ak^{\alpha} + (1-\delta)k - c(k))}(1 - \delta + \alpha A\left(Ak^{\alpha-1} + (1-\delta)k_t - c(k)\right)^{\alpha-1}) \\
F(c)(k) &= \frac{c(k)\beta\left(1 - \delta + \alpha A\left(Ak^{\alpha} + (1-\delta)k_t - c(k)\right)^{\alpha-1}\right)}{c(Ak^{\alpha-1} + (1-\delta)k - c(k))} - 1
\end{aligned} \tag{15}$$

We want to approximate $c(k)$ by finding an approximation $c^n(k;\theta)$ that sets $F(c)$ approximately equal to 0 for all k. Hence we can approximate our consumption function by the follwing function of k

$$c^n(k;\theta) = \theta_1\psi_1(k) + \theta_2\psi_2(k) + ... + \theta_n\psi_n(k) = \sum_{i=1}^{n} \theta_i\psi_i(k)$$

Finite element method

In the finite element method, instead of using polynomial over the entire domain of k, we will use basis functions that are non-zero on only small regions of the demain of k. Hence we can break up the domain into nodes - points on the domain of k - and then we define the elements as the intervals generated by the set of nodes chosen. For example, for the nodes $(0, 2, 3)$, we have two elements (intervals), i.e. $e_1 = [0, 2], \quad e_2 = [2, 3]$. Hence, for each element i, $c_i^n(k;\theta)$ can be written as:

$$c_i^n(k;\theta) = \theta_i \psi_i(k) + \theta_{i+1}\psi_{i+1}(k) \tag{16}$$

where each $\psi_i(k)$ is given by:

$$\psi_i(k) = \begin{cases} \frac{k-k_{i-1}}{k_i-k_{i-1}} & \text{if } k \in [k_{i-1}, k_i] \\ \frac{k_{i+1}-k}{k_{i+1}-k_i} & \text{if } k \in [k_i, k_{i+1}] \\ 0 & \text{elsewhere} \end{cases}$$

Hence, the residual equation takes as many values as the number of elements considered. And at each element i, $c_i^n(k;\theta)$ it is only a function of k, θ_i, θ_{i+1}, and parameters. And finally (using Galerkin), after setting $\phi_i(k) = \psi_i(k)$ and integrating over the domain of k, the following function

$$\int_{k_i}^{k_{i+1}} \psi_i(k) R(k;\theta) dk = 0, \quad i = 0, 1, ..., n$$

We will derive the derivative of this residual equation analytically. Here is how we write this residual equation

$$F(c)(k) = \frac{c^n(k;\theta)\beta\left(1 - \delta + \alpha A\left(Ak^\alpha + (1-\delta)k_t - c^n(k;\theta)\right)^{\alpha-1}\right)}{c^n(Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta);\theta)} - 1$$

The more complicated derivative is that of the denominator of this differential equation. Let's take a closer look at it:

$$\begin{aligned}\frac{\partial c^n(Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta);\theta)}{\partial\theta_i} =& \psi_i(Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta)) + \theta_i \frac{\partial\psi_i(Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta))}{\partial c^n(k;\theta)} \frac{\partial c^n(k;\theta)}{\partial\theta_i} \\ &+ \theta_{i+1} \frac{\partial\psi_{i+1}(Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta))}{\partial c^n(k;\theta)} \frac{\partial c^n(k;\theta)}{\partial\theta_i}\end{aligned}$$

To illustrate this point further, let $Ak^{\alpha-1} + (1-\delta)k - c^n(k;\theta) = k'(k;\theta)$. This implies that for if we want to get $\frac{\partial c^n(k'(k;\theta);\theta)}{\partial\theta_i}$, we can do so according to:

$$\frac{\partial c^n(k'(k;\theta);\theta)}{\partial\theta_i} = \psi_i(k'(c^n(k;\theta))) + \frac{\partial c^n(k;\theta)}{\partial\theta_i} \sum_j \theta_j * \frac{\partial\psi_j(k'(c^n(k;\theta)))}{\partial c^n(k;\theta)}$$

To see that this works, note consider $k'(k;\theta)$ such that $c^n(k'(k;\theta_1,\theta_2);\theta_2,\theta_3)$. Then

$$c^n(k'(k;\theta_1,\theta_2);\theta_2,\theta_3) = \theta_2\left(\frac{k_3 - k'(k;\theta_1,\theta_2)}{k_3 - k_2}\right) + \theta_3\left(\frac{k'(k;\theta_1,\theta_2) - k_2}{k_3 - k_2}\right)$$

$$\frac{\partial c^n(k'(k;\theta);\theta)}{\partial\theta} = \begin{bmatrix} \underbrace{\psi_1(k, c^n(k;\theta_1,\theta_2))}_{=0} + \theta_1 \underbrace{\frac{\partial\psi_1(k'(k, c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)}}_{=0} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_1} + \cdots + \theta_N \frac{\partial\psi_N(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_1} \\ \psi_2(k, c^n(k;\theta_1,\theta_2)) + \theta_1 \frac{\partial\psi_1(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_2} + \cdots + \theta_N \frac{\partial\psi_N(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_2} \\ \vdots \\ \psi_N(k, c^n(k;\theta_1,\theta_2)) + \theta_1 \frac{\partial\psi(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_N} + \cdots + \theta_N \frac{\partial\psi(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial\theta_N} \end{bmatrix}$$

$$\frac{\partial c^n(k'(k;\theta);\theta)}{\partial \theta} = \begin{bmatrix} \theta_2 \frac{\partial \psi_2(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_1} + \theta_3 \frac{\partial \psi_3(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_1} \\ \psi_2(k'(k,c^n(k;\theta_1,\theta_2))) + \theta_2 \frac{\partial \psi_2(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_2} + \theta_3 \frac{\partial \psi_3(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_2} \\ \psi_3(k'(k,c^n(k;\theta_1,\theta_2))) + \theta_2 \frac{\partial \psi_2(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \underbrace{\frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_3}}_{=0} + \theta_3 \frac{\partial \psi_3(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \underbrace{\frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_3}}_{=0} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \theta_2 \frac{\partial \psi_2(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_1} + \theta_3 \frac{\partial \psi_3(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_1} \\ \psi_2(k'(k,c^n(k;\theta_1,\theta_2))) + \theta_2 \frac{\partial \psi_2(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_2} + \theta_3 \frac{\partial \psi_3(k'(k,c^n(k;\theta_1,\theta_2)))}{\partial c^n(k;\theta_1,\theta_2)} * \frac{\partial c^n(k;\theta_1,\theta_2)}{\partial \theta_2} \\ \psi_3(k'(k,c^n(k;\theta_1,\theta_2))) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$