Take the growth model with taxes,
\[
v(k, K) = \max_{k', h, c} \{ \log (c) + \theta \log (1 - h) + \beta v(k', K') \}
\]
subject to the budget constraint
\[
c + k' \leq \left( 1 + (1 - \tau) (r(K, N) - \delta) \right) k + w(K, N) h + T(K, N),
\]
the law of motion for aggregate capital
\[
K' = H_K(K),
\]
the aggregate labor input function
\[
N = H_N(K),
\]
and the government budget constraint
\[
T + G = \tau (r(K, N) - \delta) K.
\]
To solve this model, we will iterate "in the time domain". Roughly speaking, suppose we know the equilibrium at some date, or we just guess it, meaning we have a guess for \(v^0 (k, K)\) and \(H^0_K(K)\). It turns out we won’t need to guess \(H^0_N(K)\). We solve the problem
\[
V(k, K; N) = \max_{k', h, c} \{ \log (c) + \theta \log (1 - h) + \beta v^0 (k', H^0_K(K)) \}
\]
subject to
\[
c + k' \leq \left( 1 + (1 - \tau) (r(K, N) - \delta) \right) k + w(K, N) h + r(K, N) - \delta) K - G.
\]
We are treating \(N\) like a state here, although it is not because it depends on \(K\), but in a way we don’t know yet; and also be clear that \(v \neq V\). The decision rules are
\[
c = g_c(k, K; N)
\]
\[
h = g_h(k, K; N)
\]
\[
k' = g_k(k, K; N).
\]
The first step is to solve the equation
\[
N = g_h(K, K; N)
\]
for \(N = H_N(K)\), which imposes the labor market clearing condition. Given that, we have
\[
H^1_K(K) = g_k(K, K; H_N(K)),
\]
the implied law of motion for aggregate capital, and
\[
v^1(k, K) = \log (c(k, K; H_N(K))) + \theta \log (1 - g_h(k, K; H_N(K))) + \beta v^0 (k', H^1_K(K))
\]
as the updated value function. We continue until this process converges.

This algorithm is particularly easy to implement in an LQ setting, where we won’t even need to guess \(H^0_K(K)\). But it is still easy to implement in non-LQ settings, although for this problem it will turn out that just solving the functional equations for the RCE directly, without doing dynamic programming, will often be more efficient.