Quantitative Macro in Continuous Time

Paolo Martellini
Minneapolis Fed

December 8, 2020

This Class

- Objective: Solving dynamic economic models in continuous time
  - Focus on heterogeneous agents macro-models
- Typical structure:
  - Hamilton-Jacobi-Bellman Equation describes individual decision problem
  - Kolmogorov Forward Equation characterizes the law of motion of the equilibrium distribution
- Why continuous time?
  - FOC approach to policy function (fast maximization)
  - no interaction between decisions and stochastic evolution of the state
  - only local changes of state (sparsity)
  - tight connection between HJB and KF: solve the first, get the second ”for free”

Outline

- Introduction to HJB
- Explicit and Implicit Method
- Adding Income Risk
- Boundary Conditions
- The KFE and the Stationary Distribution
- Extensions and Applications

Discretizing the HJB

- Consider the HJB derivation

$$\rho v(a) = \max_c \; u(c) + v'(a)(ra + y - c) + \dot{v}$$

- Focus on a steady-state, which requires $\dot{v} = 0$
- Discretize the state space: $a_i \in \{\underline{a}, \underline{a} + \Delta_a, .., \bar{a} - \Delta_a, \bar{a}\}$

$$\rho v_i = \max_c \; u(c) + v_i'(ra_i + y - c)$$

- The FOC for consumption is

$$u'(c_i) = v_i'$$

- Approximate v_i' using either the backward or forward derivatives

$$\frac{v_{i+1} - v_i}{\Delta_a} \equiv v'_{i,F}$$
$$\frac{v_i - v_{i-1}}{\Delta_a} \equiv v'_{i,B}$$

How to approximate v': The Upwind Scheme (Barles and Souganidis, 1991)

- Define the forward and backward saving

$$
\begin{aligned}
ra_i + y - c_{i,F} &= ra_i + y - (u')^{-1}(v'_{i,F}) \equiv s_{i,F} \\
ra_i + y - c_{i,B} &= ra_i + y - (u')^{-1}(v'_{i,B}) \equiv s_{i,B}
\end{aligned}
$$

- Since v is concave: $v'_{i,B} > v'_{i,F} \Rightarrow c_{i,B} < c_{i,F} \Rightarrow s_{i,B} > s_{i,F}$. Hence

$$
\mathbb{I}_{\{s_{i,F}>0\}}\mathbb{I}_{\{s_{i,B}<0\}}\mathbb{I}_{\{s_{i,F}<0<s_{i,B}\}} = 0
$$

- Choose the approximation of v'_i and c_i as

$$
\begin{aligned}
v'_i =& v_{i,F}\mathbb{I}_{\{s_{i,F}>0\}} + v_{i,B}\mathbb{I}_{\{s_{i,B}<0\}} + \bar{v}'_i\mathbb{I}_{\{s_{i,F}<0<s_{i,B}\}} \\
c_i =& c_{i,F}\mathbb{I}_{\{s_{i,F}>0\}} + c_{i,B}\mathbb{I}_{\{s_{i,B}<0\}} + (ra_i + y)\mathbb{I}_{\{s_{i,F}<0<s_{i,B}\}}
\end{aligned}
$$

where $\bar{v}'_i = u'(ra_i + y)$, i.e. zero saving is optimal

The Explicit Method

- Let's go back to the HJB, appropriately discretized

$$\rho v_i = u(c_i) + v_i'(ra_i + y - c_i)$$

- Solve by running backward: same logic as VFI in discrete time, $v^{n+1}(a) = u(c^n) + \beta v^n(a')$

$$\rho v_i^n = u(c_i^n) + v_i^{n\prime}(ra_i + y - c_i^n) + \frac{v_i^n - v_i^{n+1}}{\Delta_t}$$

- Algorithm: start with a guess v^0 and update according to

$$v_i^{n+1} = v_i^n - \Delta_t \rho v_i^n + \Delta_t u(c_i^n) + \Delta_t v_i^{n\prime}(ra_i + y - c_i^n)$$

- Pro: simple. Con: i) need small Δ_t ii) slow

The Implicit Method - I

- Under the explicit method

$$\rho v_i^n = u(c_i^n) + v_i^{n\prime}(ra_i + y - c_i^n) + \frac{v_i^n - v_i^{n+1}}{\Delta_t}$$

- The implicit method leverages the linearity of the HJB (conditional on the choice of c)

$$\rho v_i^{n+1} = u(c_i^n) + v_i^{n+1\prime}(ra_i + y - c_i^n) + \frac{v_i^n - v_i^{n+1}}{\Delta_t}$$

- In compact form,

$$\rho v^{n+1} = u^n + A^n v^{n+1} + \frac{v^n - v^{n+1}}{\Delta_t}$$

The Implicit Method - II

$$A^n =$$

$$\begin{pmatrix} -\frac{(s_{1,F}^n)^+}{\Delta_a} + \frac{(s_{1,B}^n)^-}{\Delta_a} & \frac{(s_{1,F}^n)^+}{\Delta_a} & 0 & \dots & 0 & 0 & 0 \\ -\frac{(s_{2,B}^n)^-}{\Delta_a} & -\frac{(s_{2,F}^n)^+}{\Delta_a} + \frac{(s_{2,B}^n)^-}{\Delta_a} & \frac{(s_{2,F}^n)^+}{\Delta_a} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -\frac{(s_{I-1,B}^n)^-}{\Delta_a} & -\frac{(s_{I-1,F}^n)^+}{\Delta_a} + \frac{(s_{I-1,B}^n)^-}{\Delta_a} & \frac{(s_{I-1,F}^n)^+}{\Delta_a} \\ 0 & 0 & 0 & \dots & 0 & -\frac{(s_{I,B}^n)^-}{\Delta_a} & -\frac{(s_{I,F}^n)^+}{\Delta_a} + \frac{(s_{I,B}^n)^-}{\Delta_a} \end{pmatrix}$$

- For example, $A^n_{(2,\cdot)} v = -\frac{(s_{2,B}^n)^-}{\Delta_a} v_1 + \left(-\frac{(s_{2,F}^n)^+}{\Delta_a} + \frac{(s_{2,B}^n)^-}{\Delta_a}\right) v_2 + \frac{(s_{2,F}^n)^+}{\Delta_a} v_3$
- The matrix A^n
  - has dimension *IxI*
  - is very sparse (computationally efficient)
  - each row sums to zero, terms in diagonal/off-diagonal are negative/positive (”intensity matrix”)

The Implicit Method - III

- Algorithm: start with a guess v^0 and update it according to

$$B^n v^{n+1} = b^n, \qquad B^n = \left(\frac{1}{\Delta_t} + \rho\right) I - A^n, \qquad b^n = u^n + \frac{1}{\Delta_t} v^n$$

- Comments:
  - Pro: i) fast ii) works with any Δ_t
  - Con: i) slightly harder to code ii) requires solving a (potentially large) linear system

Introducing Idiosyncratic Income Shocks - I

- Consider the stochastic income process $y_j \in \{y_L, y_H\}$ with Poisson transition rates $\lambda_{LH}, \lambda_{HL}$
- The individual HJB becomes derivation

$$\rho v_j(a) = \max_c u(c) + v_j'(a)(ra + y_j - c) + \lambda_{jj'}(v_{j'}(a) - v_j(a))$$

- Discretizing, and using the implicit method,

$$\rho v_{i,j}^{n+1} = u(c_i^n) + v_{i,j}^{n+1\prime}(ra_i + y_j - c_i^n) + \lambda_{jj'}(v_{i,j'}^{n+1} - v_{i,j}^{n+1}) + \frac{v_{i,j}^n - v_{i,j}^{n+1}}{\Delta_t}$$

- In compact form, stacking $v = [v_{1,1}, v_{2,1}, \dots, v_{I,1}, v_{1,2}, v_{2,2}, \dots, v_{I,2}]$

$$\rho v^{n+1} = u^n + A^n v^{n+1} + \frac{v^n - v^{n+1}}{\Delta_t}$$

Introducing Idiosyncratic Income Shocks - II

$$A^n = \left[\begin{array}{c|c} A^n_{a,L} & 0 \\ \hline 0 & A^n_{a,H} \end{array}\right] + A_y$$

- The matrix A^n
  - has dimension $(Ix2)x(Ix2)$
  - is very sparse, but less than before (larger bandwidth)
  - each row sums to zero, terms in diagonal/off-diagonal are negative/positive (”intensity matrix”)

Consumption Choice at the Borrowing Constraint

- At $\underline{a} = a_1$, we need $s_j(\underline{a}) = r\underline{a} + y_j - c_j(\underline{a}) \geq 0$. This implies the state constraint boundary condition

$$v'_{1,j} = u'(c_{1,j}) \geq u'(ra_1 + y_j)$$

- In practice,
  - compute optimal consumption $c_{1,j,F}$ using the forward derivative $\left(\frac{v_{2,j} - v_{1,j}}{\Delta_a}\right)$
  - compute $s_{1,j,F} = ra_1 + y_j - c_{1,j,F}$
  - choose consumption as follows

$$c_{1,j} = \begin{cases} c_{1,j,F} & \text{if } s_{1,j,F} > 0 \\ ra_1 + y_j & \text{o/w} \end{cases}$$

  - in the intensity matrix A, $s_{1,j,B} = 0$
- Use symmetric argument at $\bar{a}$ (not necessary if $\bar{a}$ is large enough)
- This approach delivers a *viscosity solution*: weak solution to HJB, in the presence of kinks

Solving for the Stationary Distribution: The Kolmogorov Forward Equation - I

- The law of motion of the distribution satisfies derivation

$$0 = \dot{g}_j = -[s_j(a)g_j(a)]' - \lambda_{jj'}g_j(a) + \lambda_{j'j}g_{j'}(a)$$

- Discretizing, and appropriately choosing the backward/forward derivatives,

$$0 = -\frac{(s^n_{i,j,F})^+ g_{i,j} - (s^n_{i-1,j,F})^+ g_{i-1,j}}{\Delta_a} - \frac{(s^n_{i+1,j,B})^- g_{i+1,j} - (s^n_{i,j,B})^- g_{i,j}}{\Delta_a} - \lambda_{jj'}g_{i,j} + \lambda_{j'j}g_{i,j'}$$

- (Make sure you understand why the expression above is a correct approximation)
- In compact form,

$$0 = A^T g$$

Solving for the Stationary Distribution: The Kolmogorov Forward Equation - II

- Intuition: the mass at a_i is the sum of the inflows from a_{i-1} and a_{i+1}, and the (negative) outflows from a_i details
- Algorithm:
  - once the HJB has converged, set A equal to the 'last' A^n (steady-state transitions)
  - use solver that is suited for non full-rank matrix
  - g is equal to the solution $\tilde{g}$ of the linear system, divided by $\sum_i \sum_j \tilde{g}_{ij} \, \Delta_a$ (normalization)
- GE in Aiyagari models: compute aggregate savings and update r until capital market clears

Extensions and Applications

- non-concave value function (e.g. lumpy housing w/ downpayment constraint) details
- life-cycle model with stochastic aging, $h \in \{1, 2, ..., H\}$ details

$$\rho v(a, h) = \max_c \; u(c) + v'(ra + y_h - c) + \phi[v(a, h+1) - v(a, h)]$$

- search and matching model details

$$\text{joint value of match } z\text{: } \rho v(z) = z + \lambda_1 \beta \int_{\hat{z}} max\{v(\hat{z}) - v(z), 0\} + \delta(u - v(z))$$

$$\text{value of unemployment: } \rho u = \ell + \lambda_0 \beta \int_{\hat{z}} max\{v(\hat{z}) - u, 0\}$$

- diffusion processes, portfolio choice, aggregate uncertainty, learning (stopping time)

References

These notes rely **heavily** on

- Achdou, Han, Lasry, Lions, Moll. 2020. ”Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach”. Forthcoming at Review of Economic Studies. Online Appendices and codes on Moll’s website

Cool paper, coming soon:

- Thomas Phelan (Cleveland Fed) and Keyvan Eslami (Ryerson) on Markov Chain Approximation methods

Derivation of HJB

back

- Consider the standard discrete-time Bellman Equation with time length Δ_t

$$v_t(a_t) = \max_{c_t} \Delta_t u(c_t) + exp(-\rho\Delta_t) v_{t+\Delta_t}(a_{t+\Delta_t})$$

$$st. \quad a_{t+\Delta_t} = a_t + \Delta_t(r_t a_t + y_t - c_t), \qquad a_{t+\Delta_t} \geq \underline{a}$$

- Subtract $exp(-\rho\Delta_t) v_{t+\Delta_t}(a_t)$ from both sides, divide by Δ_t and take the limit $\Delta_t \to 0$

$$\frac{v_t(a_t) - exp(-\rho\Delta_t) v_{t+\Delta_t}(a_t)}{\Delta_t} = \max_{c_t} u(c_t) + exp(-\rho\Delta_t) \frac{v_{t+\Delta_t}(a_{t+\Delta_t}) - v_{t+\Delta_t}(a_t)}{\Delta_t} \frac{\Delta_a}{\Delta_a}$$

$$\rho v_t(a_t) - \dot{v}_t = \max_{c_t} u(c_t) + v_t'(a_t)(r_t a_t + y_t - c_t)$$

- Note: the LHS is the derivative of $-exp(-\rho x) v_{t+x}(a_t)$ in $x = 0$

Derivation of HJB w/ Income Shocks

back

- Consider the standard discrete-time Bellman Equation with time length Δ_t

$$v_{j,t}(a_t) = \max_{c_t} \Delta_t u(c_t) + exp(-\rho\Delta_t)\left[(1-\Delta_t\lambda_{jj'})v_{j,t+\Delta_t}(a_{t+\Delta_t}) + \Delta_t\lambda_{jj'}v_{j',t+\Delta_t}(a_{t+\Delta_t})\right]$$

$$st. \quad a_{t+\Delta_t} = a_t + \Delta_t(r_t a_t + y_{j,t} - c_t), \qquad a_{t+\Delta_t} \geq \underline{a}$$

- Subtract $exp(-\rho\Delta_t)v_{j,t+\Delta_t}(a_t)$ from both sides, divide by Δ_t and take the limit $\Delta_t \to 0$

$$\frac{v_{j,t}(a_t) - exp(-\rho\Delta_t)v_{j,t+\Delta_t}(a_t)}{\Delta_t} = \max_{c_t} u(c_t) + exp(-\rho\Delta_t)\frac{v_{j,t+\Delta_t}(a_{t+\Delta_t}) - v_{j,t+\Delta_t}(a_t)}{\Delta_t}$$
$$+\lambda_{jj'}(v_{j',t+\Delta_t}(a_{t+\Delta_t}) - v_{j,t+\Delta_t}(a_{t+\Delta_t}))$$

$$\rho v_{j,t}(a_t) - \dot{v}_{j,t} = \max_{c_t} u(c_t) + v'_{j,t}(a_t)(r_t a_t + y_{j,t} - c_t) + \lambda_{jj'}(v_{j',t}(a_t) - v_{j,t}(a_t))$$

Intensity Matrix w/ Income Shocks

back

- The matrix below has dimension (IxI)

$$A^n_{a,j} =$$

$$\begin{pmatrix}
-\frac{(s^n_{1,j,F})^+}{\Delta_a} + \frac{(s^n_{1,j,B})^-}{\Delta_a} & \frac{(s^n_{1,j,F})^+}{\Delta_a} & 0 & \dots & 0 & 0 & 0 \\
-\frac{(s^n_{2,j,B})^-}{\Delta_a} & -\frac{(s^n_{2,j,F})^+}{\Delta_a} + \frac{(s^n_{2,j,B})^-}{\Delta_a} & \frac{(s^n_{2,j,F})^+}{\Delta_a} & \dots & 0 & 0 & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & \dots & -\frac{(s^n_{I-1,j,B})^-}{\Delta_a} & -\frac{(s^n_{I-1,j,F})^+}{\Delta_a} + \frac{(s^n_{I-1,j,B})^-}{\Delta_a} & \frac{(s^n_{I-1,j,F})^+}{\Delta_a} \\
0 & 0 & 0 & \dots & 0 & -\frac{(s^n_{I,j,B})^-}{\Delta_a} & -\frac{(s^n_{I,j,F})^+}{\Delta_a} + \frac{(s^n_{I,j,B})^-}{\Delta_a}
\end{pmatrix}$$

Intensity Matrix w/ Income Shocks

back

- Each block has dimension (IxI)

$$A_y = \left[\begin{array}{ccc|ccc} -\lambda_{LH} & & & \lambda_{LH} & & \\ & -\lambda_{LH} & & & \lambda_{LH} & \\ & & -\lambda_{LH} & & & \lambda_{LH} \\ \hline \lambda_{HL} & & & -\lambda_{HL} & & \\ & \lambda_{HL} & & & -\lambda_{HL} & \\ & & \lambda_{HL} & & & -\lambda_{HL} \end{array}\right]$$

- For example, $A_{y,(2,\cdot)} v = \lambda_{LH}(v_{2,H} - v_{2,L})$

Derivation of KFE

back

- The fraction of individuals with wealth below a satisfies

$$Pr(\tilde{a}_{t+\Delta_t} \le a, \tilde{y}_{t+\Delta_t} = y_j) = (1-\Delta_t\lambda_{jj'})Pr(\tilde{a}_t \le a-\Delta_t s_j(a), \tilde{y}_t = y_j)+\Delta_t\lambda_{j'j}Pr(\tilde{a}_t \le a-\Delta_t s_j(a), \tilde{y}_t = y_{j'})$$

- Using the definition of the cdf G_j,

$$G_j(a, t + \Delta_t) = (1 - \Delta_t\lambda_{jj'})G_j(a - \Delta_t s_j(a), t) + \Delta_t\lambda_{j'j}G_{j'}(a - \Delta_t s_{j'}(a), t)$$

- Substract $G_j(a, t)$ from both sides, divide by Δ_t and compute the limit $\Delta_t \to 0$

$$\frac{G_j(a, t + \Delta_t) - G_j(a, t)}{\Delta_t} = \frac{G_j(a - \Delta_t s_j(a), t) - G_j(a, t)}{\Delta_t} - \lambda_{jj'}G_j(a - \Delta_t s_j(a), t) + \lambda_{j'j}G_{j'}(a - \Delta_t s_{j'}(a), t)$$

$$\dot{G}(a, t) = - g_j(a, t)s_j(a) - \lambda_{jj'}G_j(a, t) + \lambda_{j'j}G_{j'}(a, t)$$

Connecting Individual Decisions to Aggregate Transitions

back

- For simplicity, consider the case without income shocks, i.e. $\lambda_{LH} = \lambda_{HL} = 0$
- The $i-th$ column of A reads

$$A_{(\cdot,i)} = \begin{pmatrix} 0 \\ \dots \\ \frac{(s^n_{i-1,F})^+}{\Delta_a} \\ -\frac{(s^n_{i,F})^+}{\Delta_a} + \frac{(s^n_{i,B})^-}{\Delta_a} \\ -\frac{(s^n_{i+1,B})^-}{\Delta_a} \\ \dots \\ 0 \end{pmatrix}$$

Connecting Individual Decisions to Aggregate Transitions

back

- For simplicity, consider the case without income shocks, i.e. $\lambda_{LH} = \lambda_{HL} = 0$
- The $i - th$ row of A^T reads

$$A^T_{(i,\cdot)} = \begin{pmatrix} 0 & \dots & \frac{(s^n_{i-1,F})^+}{\Delta_a} & -\frac{(s^n_{i,F})^+}{\Delta_a} + \frac{(s^n_{i,B})^-}{\Delta_a} & -\frac{(s^n_{i+1,B})^-}{\Delta_a} & \dots 0 \end{pmatrix}$$

Connecting Individual Decisions to Aggregate Transitions

back

- For simplicity, consider the case without income shocks, i.e. $\lambda_{LH} = \lambda_{HL} = 0$
- The $i - th$ row of A^T reads

$$A^T_{(i,\cdot)} = \begin{pmatrix} 0 & \dots & \frac{(s^n_{i-1,F})^+}{\Delta_a} & -\frac{(s^n_{i,F})^+}{\Delta_a} + \frac{(s^n_{i,B})^-}{\Delta_a} & -\frac{(s^n_{i+1,B})^-}{\Delta_a} & \dots 0 \end{pmatrix}$$

- Then, the $i - th$ row of $A^T g$ reads

$$0 = \left(\frac{(s^n_{i-1,F})^+}{\Delta_a} \right) g_{i-1} + \left(-\frac{(s^n_{i,F})^+}{\Delta_a} + \frac{(s^n_{i,B})^-}{\Delta_a} \right) g_i + \left(-\frac{(s^n_{i+1,B})^-}{\Delta_a} \right) g_{i+1}$$

Non-concave Value Function

back

- If v is concave, $v'_{i,j,F} < v'_{i,j,B} \Rightarrow c_{i,j,F} > c_{i,j,B} \Rightarrow s_{i,j,F} < s_{i,j,B}$
- What if v has a (convex) kink? Then, $v'_{i,j,F} > v'_{i,j,B}$ and $s_{i,j,F} > s_{i,j,B} \Rightarrow$ upwind scheme fails
- Solution: define

$$\mathbb{I}^{both}_{i,j} := \mathbb{I}_{\{s_{i,j,F} > 0 > s_{i,j,B}\}}$$

$$\mathbb{I}^{unq}_{i,j} := \mathbb{I}_{\{s_{i,j,F} < 0 \text{ and } s_{i,j,B} < 0\}} + \mathbb{I}_{\{s_{i,j,F} > 0 \text{ and } s_{i,j,B} > 0\}}$$

and $H_{i,j,F} := u(c_{i,j,F}) + v'_{i,j,F} s_{i,j,F}$ and $H_{i,j,B} := u(c_{i,j,B}) + v'_{i,j,B} s_{i,j,B}$

- Use the upwind scheme

$$v'_{i,j} = v'_{i,j,F}\left[\mathbb{I}_{\{s_{i,j,F}>0\}}\mathbb{I}^{unq}_{i,j} + \mathbb{I}_{\{H_{i,j,F}>H_{i,j,B}\}}\mathbb{I}^{both}_{i,j}\right] + v'_{i,j,B}\left[\mathbb{I}_{\{s_{i,j,B}<0\}}\mathbb{I}^{unq}_{i,j} + \mathbb{I}_{\{H_{i,j,F}<H_{i,j,B}\}}\mathbb{I}^{both}_{i,j}\right] + \bar{v}'_{i,j}\mathbb{I}_{\{s_{i,j,F}<0<s_{i,j,B}\}}$$

Life-Cycle Model with Stochastic Aging

back

- Solve the problem at $h = H$: equivalent to standard HJB w/ discount rate $\rho + \phi$
- Backward induction: at $h < H$, the function $v(a, h+1)$ is just a constant

$$\rho v^{n+1}(a, h) = u(c^n) + v^{n+1\prime}(ra + y_h - c^n) + \phi[v(a, h+1) - v^{n+1}(a, h)],$$

so that $b^n = u^n + \frac{1}{\Delta_t} v^n + \phi v(\cdot, h+1)$

- A is not an intensity matrix: need to add newborns at $h = 1$

$$0 = A^T g + \phi g_0,$$

where the vector $g_0 = [g_0(a_1, 1), g_0(a_2, 1), .., g(a_I, 1), \underbrace{0}_{(H-1)I}]$ is a parameter of the model

Search and Matching Model - I

back

- Define $w = [v(z_1), v(z_2), ..., v(z_I), u]$, $y = [z_1, ..., z_I, \ell]$
- Stack the HJBs for v_i and u together and obtain

$$\rho w = y + \lambda\beta \sum_{\hat{z}} f(\hat{z}) max\{w(\hat{z}) - w, 0\} + \delta(u - w)$$

- Under the implicit scheme,

$$\rho w^{n+1} = y + \lambda\beta \sum_{\hat{z}} f(\hat{z}) \mathbb{I}^n_{\{w(\hat{z})>w\}} (w^{n+1}(\hat{z}) - w^{n+1}) + \delta(u^{n+1} - w^{n+1}) + \frac{w^n - w^{n+1}}{\Delta_t}$$

- In compact form,

$$B^n w^{n+1} = b^n, \qquad B^n = \left(\frac{1}{\Delta_t} + \rho\right) I - A^n, \qquad b^n = y + \frac{1}{\Delta_t} w^n$$

Search and Matching Model - II

back

- It is easy to show that $v(z)$ is strictly increasing
- Why do we need an iterative procedure? Because we do not know R st $u = v(R)$
- Consider the case in which there are $I = 4$ possible values of z

$$A^n = \begin{pmatrix} x_1 & \lambda_1 f(z_2) & \lambda_1 f(z_3) & \lambda_1 f(z_4) & \delta \\ & x_2 & \lambda_1 f(z_3) & \lambda_1 f(z_4 & \delta \\ & & x_3 & \lambda_1 f(z_4) & \delta \\ & & & x_4 & \delta \\ \lambda_0 f(z_1)\mathbb{I}_1^n & \lambda_0 f(z_2)\mathbb{I}_2^n & \lambda_0 f(z_3)\mathbb{I}_3^n & \lambda_0 f(z_4)\mathbb{I}_4^n & x_u \end{pmatrix},$$

where $i)$ $x_i \leq 0$ is the negative of the sum of the off-diagonal elements $ii)$ $\mathbb{I}_i^n = \{v^n(z_i) \geq u^n\}$