Dynamic Macroeconomic Models

Lecture 2

Alessandro Di Nola

University of Konstanz

Notation

- **State variables** (backward-looking):
  - Endogenous: k_t, m_t, b_t
  - Exogenous: A_t, z_t, g_t
- **Control variables** (forward-looking): c_t, n_t
- Static/redundant variables: w_t, r_t, y_t

Recipe to solve DSGE models

- Solving DSGE models with *linear* approximations.
- **Step 1**: Find first-order conditions.
- **Step 2**: list all equilibrium conditions:
  - FOCs
  - Budget constraints
  - Market clearing equations.
- **Step 3**: Find deterministic steady-state
  - Without shocks, economy converges to a steady-state where all endogenous variables are constant:

$$\varepsilon_t = 0, \forall t \implies \lim_{t \to \infty} x_t = \overline{x}.$$

Recipe to solve DSGE models

- **Step 4**: Linear approximation around deterministic steady-state
  - Log-linearized variables $\widetilde{x}_t = \log x_t - \log \overline{x} \simeq (x_t - \overline{x})/\overline{x}$.
- **Step 5**: Solve linearized system of equations obtained in step 4. Several methods in the liter., the most popular are:
  - Blanchard-Kahn based on eigenvalue-eigenvector (Jordan) decomposition. Reference: Blanchard and Kahn (1980).
  - Undetermined coefficients based on matrix quadratic equations. Reference: Uhlig (1999).
- **Step 6**: With solution at hand, compute simulated time series and impulse-responses to shocks.

Roadmap

- In Problem Set 1 you went through steps 1-4 for a protypical RBC model (Hansen 1985).
- Today's lecture: focus on Step 5, i.e. solving linearized system of stochastic difference equations, also known as linearized rational expectation models (LREM).
- Next, we will cover *simulation* and *impulse-response* analysis.

Solving LREM: General Setup

- Denote vector of control variables as u_t and the state vector as s_t.
- Partition state vector as $s_t = \left[s_t^1, s_t^2\right]$, where 1→endogenous, 2→exogenous
- Assume

$$s_{t+1}^2 = \varrho s_t^2 + \varepsilon_{t+1},$$

  where ε_{t+1} is a vector of i.i.d. innovations with mean zero and var-cov matrix Σ.
- Linearized equilibrium equations can be written as

$$\underset{n \times n}{A} \cdot \; E_t x_{t+1} = \underset{n \times n}{B} \cdot x_t$$

  where

$$\underbrace{x_t}_{n \times 1} = \begin{bmatrix} \underset{n_s \times 1}{s_t} \\ \underset{n_c \times 1}{u_t} \end{bmatrix}$$

  and $n = n_s + n_c$.

Solving LREM: An Example

- Stochastic growth model seen in Lecture 1.
- Deterministic steady-state:

$$\bar{k} = \left(\frac{\alpha\beta}{1-\beta\left(1-\delta\right)}\right)^{\frac{1}{1-\alpha}}, \ \bar{c} = k^{\alpha} - \delta k, \ \bar{z} = 1$$

- Log-linearized equations:

$$\sigma\tilde{c}_t - \sigma E_t\tilde{c}_{t+1} + \left[1-\beta\left(1-\delta\right)\right]\rho\tilde{z}_t - \left(1-\alpha\right)\left[1-\beta\left(1-\delta\right)\right]\tilde{k}_{t+1} = 0$$

$$\frac{\bar{c}}{\bar{k}}\tilde{c}_t + \tilde{k}_{t+1} - \frac{1}{\beta}\tilde{k}_t - \frac{1}{\alpha}\left(\frac{1}{\beta} - 1 + \delta\right)\tilde{z}_t = 0$$

$$\tilde{z}_{t+1} = \rho\tilde{z}_t + \varepsilon_{t+1} \implies E_t\tilde{z}_{t+1} = \rho\tilde{z}_t$$

Solving LREM: An Example

- Let $x_t = \left[\tilde{k}_t, \tilde{z}_t, \tilde{c}_t\right]'$, with $s_t = \left[\tilde{k}_t, \tilde{z}_t\right]'$ and $u_t = \tilde{c}_t$.
- Linearized equations can be written in matrix form:

$$\begin{bmatrix} \left(\frac{1}{\beta} - 1 + \delta\right)(1-\alpha) & 0 & \frac{1}{\beta}\sigma \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{k}_{t+1} \\ E_t \tilde{z}_{t+1} \\ E_t \tilde{c}_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{1}{\beta} - 1 + \delta\right)\rho & \frac{1}{\beta}\sigma \\ \frac{1}{\beta} & \frac{1}{\alpha}\left(\frac{1}{\beta} - 1 + \delta\right) & -\frac{\bar{c}}{\bar{k}} \\ 0 & \rho & 0 \end{bmatrix} \begin{bmatrix} \tilde{k}_t \\ \tilde{z}_t \\ \tilde{c}_t \end{bmatrix}$$

- Note: $E_t s^1_{t+1} = s^1_{t+1}$, since endogenous states are *predetermined*. So here $E_t k_{t+1} = k_{t+1}$.

Solving LREM

- We need to solve

$$\underset{n\times n}{A} \cdot \; E_t x_{t+1} = \underset{n\times n}{B} \cdot x_t$$

- *Provided that A is invertible*, can write

$$E_t x_{t+1} = \underbrace{A^{-1}B}_{M} \cdot x_t \tag{1}$$

- Remark: the system above is not a solution yet!
- M only tells us how the variables x_t will evolve *given* initial starting point.
- We only have initial conditions for the state variables.
- Where do we “start” the control variables?

Solving LREM

- Decouple the system using the Jordan decomposition of M

$$MD = D\Lambda \rightarrow M = D\Lambda D^{-1}$$

  where Λ is diagonal matrix with **eigenvalues** on the main diagonal. D is matrix with corresponding **eigenvectors**.

  Math review

- Arrange eigenvalues from smallest to largest in absolute value.
- Reorder eigenvectors accordingly:
  - the i-th column of D corresponds with the i-th eigenvalue of position (i, i) in Λ.

Solving LREM

- Λ can be written as

$$\begin{bmatrix} \underset{es\times es}{\Lambda_1} & 0 \\ 0 & \underset{eu\times eu}{\Lambda_2} \end{bmatrix}$$

  where *es*: # of stable eigenvalues (i.e. smaller than one in absolute value) and *eu*: # of unstable eigenvalues.

- Write (1) as

$$E_t x_{t+1} = D\Lambda D^{-1} \cdot x_t$$

$$\underbrace{D^{-1}E_t x_{t+1}}_{E_t \widehat{x}_{t+1}} = \Lambda \underbrace{D^{-1} \cdot x_t}_{\widehat{x}_t}$$

- Hence

$$\begin{bmatrix} E_t \widehat{s}_{t+1} \\ E_t \widehat{u}_{t+1} \end{bmatrix} = \begin{bmatrix} \underset{es\times es}{\Lambda_1} & 0 \\ 0 & \underset{eu\times eu}{\Lambda_2} \end{bmatrix} \begin{bmatrix} \widehat{s}_t \\ \widehat{u}_t \end{bmatrix}$$

Solving LREM

- Further, define

$$D^{-1} = \begin{bmatrix} \underset{es \times ns}{d_{11}} & \underset{es \times nu}{d_{11}} \\ \underset{eu \times ns}{d_{21}} & \underset{eu \times nc}{d_{22}} \end{bmatrix}$$

  so that the transformed variables are related to the original variables as follows:

$$\begin{bmatrix} \widehat{s}_t \\ \widehat{u}_t \end{bmatrix} = \begin{bmatrix} \underset{es \times ns}{d_{11}} & \underset{es \times nu}{d_{11}} \\ \underset{eu \times ns}{d_{21}} & \underset{eu \times nc}{d_{22}} \end{bmatrix} \begin{bmatrix} s_t \\ u_t \end{bmatrix} \tag{2}$$

- After decoupling, our task is much easier! We now have an independent system of two equations:

$$\begin{aligned} E_t \widehat{s}_{t+1} &= \Lambda_1 \widehat{s}_t \\ E_t \widehat{u}_{t+1} &= \Lambda_2 \widehat{u}_t \end{aligned}$$

Solving LREM

- Iterating forward yields

$$\begin{aligned} E_t \widehat{s}_{t+T} &= \Lambda_1^T \widehat{s}_t \\ E_t \widehat{u}_{t+T} &= \Lambda_2^T \widehat{u}_t \end{aligned}$$

- Eigenvalues in Λ_1 are all less than 1 in abs. value $\Longrightarrow$ $E_t \widehat{s}_{t+T} \to 0$ as $T \to \infty$.
- Eigenvalues in Λ_2 are all larger than 1 in abs. value $\Longrightarrow$ $E_t \widehat{u}_{t+T} \to \pm\infty$ as $T \to \infty$, **unless we impose** $\widehat{u}_t = 0$.
- (In a deterministic model) This restriction basically pins down the **saddle-path** equations.
- Recalling (2), condition $\widehat{u}_t = 0$ can be stated as

$$0 = \underset{eu \times ns}{d_{21}} s_t + \underset{eu \times nc}{d_{22}} u_t \tag{3}$$

Solving LREM: Solution for Controls

- *Provided that d_{22} is invertible,*

$$(3)\ \underset{eu\times ns}{d_{21}}\, s_t + \underset{eu\times nc}{d_{22}}\, u_t = 0 \implies u_t = \underbrace{-d_{22}^{-1} d_{21}}_{U} \cdot s_t \qquad (4)$$

- A necessary condition for d_{22} being invertible is that $eu = nc$.
- This means **# unstable eigenvalues = # control/forward-looking variables.** (BK conditions).
- Case $eu > nc$: system (3) has no solution
  - Intuition: we have too many restrictions on the initial values of the control variables.
- Case $eu < nc$: system (3) has infinitely many solutions
  - Intuition: there are too few restrictions on the initial values of the control variables → infinitely many solutions may satisfy those restrictions.

Blanchard-Kahn Conditions: A Summary

Proposition

If the number of eigenvalues of M larger than one in absolute value is equal to the number of forward-looking variables, then there is a unique solution to the system.

Proposition

If the number of eigenvalues of M larger than one in absolute value is greater than the number of forward-looking variables, then there is no solution to the system.

Proposition

If the number of eigenvalues of M larger than one in absolute value is less than the number of forward-looking variables, then there is an infinity of solutions.[1]

[1] You will also find this case referred to as *"indeterminate"* in the literature.

Solving LREM: Solution for States

- If BK conditions are satisfied, we can uniquely determine the "policy function" for controls: equation (4), i.e. $u_t = U \cdot s_t$.
- But we also need a transition equation for the state vector s_t.
- Go back to $E_t x_{t+1} = M \cdot x_t$, with $x_t = [s_t, u_t]'$.
- Decompose M into blocks:

$$\begin{bmatrix} E_t s_{t+1} \\ E_t u_{t+1} \end{bmatrix} = \begin{bmatrix} \underset{ns \times ns}{m_{11}} & \underset{ns \times nc}{m_{12}} \\ \underset{nc \times ns}{m_{21}} & \underset{nc \times nc}{m_{22}} \end{bmatrix} \begin{bmatrix} s_t \\ u_t \end{bmatrix}$$

- Given partitioning, the first row of this system is

$$E_t s_{t+1} = m_{11} s_t + m_{12} u_t.$$

- Replacing (4) into the above equation yields:

$$E_t s_{t+1} = \underbrace{\left(m_{11} - m_{12} d_{22}^{-1} d_{21} \right)}_{\Pi} s_t$$

Solving LREM: Solution for States

- Note:

$$s_{t+1} = \begin{bmatrix} s^1_{t+1} \\ E_t s^2_{t+1} + \varepsilon_{t+1} \end{bmatrix}$$

and

$$E_t s^2_{t+1} = \varrho s^2_t.$$

- Hence:

$$s_{t+1} = \begin{bmatrix} s^1_{t+1} \\ E_t s^2_{t+1} + \varepsilon_{t+1} \end{bmatrix} = \underset{ns \times ns}{\Pi} \begin{bmatrix} s^1_t \\ s^2_t \end{bmatrix} + \underbrace{\begin{bmatrix} \underset{n_{s1} \times n_{s2}}{0} \\ \underset{n_{s2} \times n_{s2}}{I} \end{bmatrix}}_{W} \varepsilon_{t+1}$$

Solving LREM: Solution

- Summing up, we can write the solution in a recursive form as:

$$\begin{aligned} s_{t+1} &= \underset{n_s \times n_s}{\Pi} s_t + \underset{n_s \times n_{s2}}{W} \varepsilon_{t+1} && (5) \\ u_t &= \underset{n_c \times n_s}{U} s_t && (6) \end{aligned}$$

- Given initial conditions for the states, s_0, compute the equilibrium sequence $\{s_{t+1}\}_{t=0}^{\infty}$ iterating on (5) for a given sequence of innovations $\{\varepsilon_{t+1}\}_{t=0}^{\infty}$.
- Finally, use (6) to determine $\{u_t\}_{t=0}^{\infty}$.
- Terminology: in the language of state-space systems, (5) is called **transition** or **state equation**, whereas (6) is called **observation** or **measurement equation**.

Solving LREM: Example

- Go back to our example with stochastic growth model.
- Recall: state variables are $s_t = [k_t, z_t]'$ and controls are $u_t = c_t$.
- Hence solution in state-space form is:

$$
\begin{aligned}
\begin{bmatrix} k_{t+1} \\ z_{t+1} \end{bmatrix} &= \begin{bmatrix} \pi_{kk} & \pi_{kz} \\ 0 & \rho_z \end{bmatrix} \begin{bmatrix} k_t \\ z_t \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \varepsilon_{t+1} \\
c_t &= \begin{bmatrix} u_{ck} & u_{cz} \end{bmatrix} \begin{bmatrix} k_t \\ z_t \end{bmatrix}
\end{aligned}
$$

- In this example, there are two state variables and one control/forward-looking variable, i.e. $n_s = 2$, $n_c = 1$,
- Among the state variables, one is endogenous (capital) and the other one (technology) is exogenous, i.e. $n_{s^1} = 1, n_{s^2} = 1$.

Blanchard-Kahn: Matlab implementation

- All you have to do is writing the system of stochastic linear difference equations in the setup $A \cdot E_t x_{t+1} = B \cdot x_t$.
- Then call matlab function [outputs] = fun_blanchard_kahn1(inputs).
- This function takes as inputs the matrices A and B and returns the state-space matrices Π, U and W.
- Also returns an error flag, `err`. `err` is negative if Blanchard-Kahn condition for saddle-path stability is violated.

```
[ PI,U,W,err ] = fun_blanchard_kahn1(A,B);
```

Blanchard-Kahn: Results

```
Steady-state values for k,c and z:
2.2931

1.1077

1

Num. of stable eig: 2
Num. of unstable eig: 1
System is saddle-path stable
```

Blanchard-Kahn: Results, cont'd

```
Matrix PI is:
0.9061    0.2078
     0    0.9500

Matrix W is:
0
1

Matrix U is:
0.3033    0.4726
```

Simulation: Matlab implementation

- Produce simulated time series iterating on the **state-space form**.
- Pseudo-code to generate a simulated series of lenght T periods, given sequence of random shocks $\{\varepsilon_{t+1}\}_{t=0}^{\infty}$.

```
Define ns, ns_endo, ns_exo, nc, P, W, U
s0   = zeros(ns,1); %initial conditions states
epsi = sigmae*randn(ns_exo,T); %random numbers
s    = zeros(ns,T); s(:,1) = s0;
u    = zeros(nc,T);
for t=2:T % start for-loop
 s(:,t) = P*s(:,t-1) + W*epsi(t);
 u(:,t) = U*s(:,t);
end % close for-loop
```

Impulse Response Functions: Matlab implementation

- Produce impulse responses iterating on the **state-space form**.
- Pseudo-code to generate impulse responses, of lenght T periods, to a one stdev shock to TFP, i.e. $\varepsilon_1 = \sigma_e$ and $\varepsilon_t = 0, \forall t \geq 2$.

```
Define ns,ns_endo,ns_exo,nc, P, W, U
s0   = zeros(ns,1); % initial conditions states
epsi = zeros(ns_exo,T); epsi(1,2) = sigmae;
s    = zeros(ns,T); s(:,1) = s0;
u    = zeros(nc,T);
for t=2:T % start for-loop
s(:,t) = P*s(:,t-1) + W*epsi(t);
u(:,t) = U*s(:,t);
end % close for-loop
```

Results I

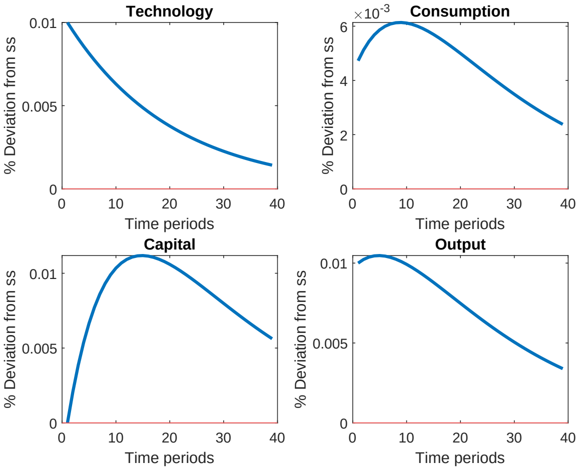


Figure: IRF of z, c, k, y to a one st. dev. positive shock to technology.

Results II

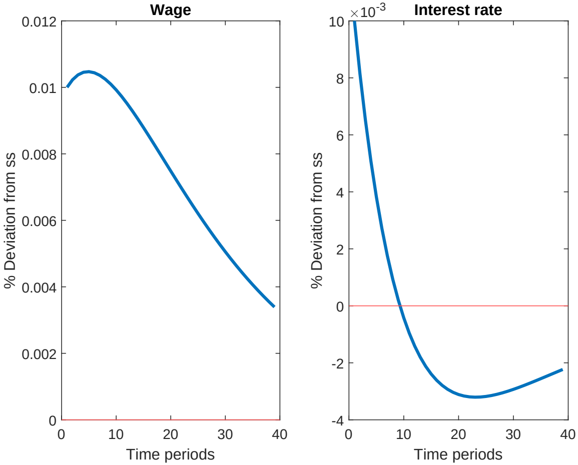


Figure: IRF of w, r to a one st. dev. positive shock to technology.

Results III

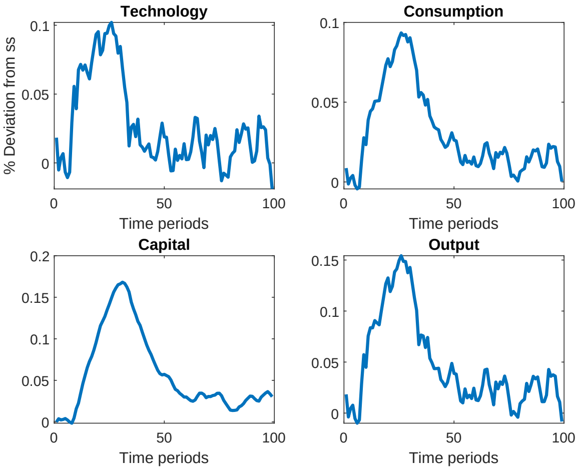


Figure: Simulated time series, given random sequence $\{\varepsilon_{t+1}\}_{t=0}^{\infty}$.

Final remarks

- Matlab codes to replicate material on these notes can be found in ILIAS:
  - main file: `main_stochastic_growth_model.m`;
  - auxiliary file: `fun_blanchard_kahn1.m`.

Final remarks

- Matlab codes to replicate material on these notes can be found in ILIAS:
  - main file: `main_stochastic_growth_model.m`;
  - auxiliary file: `fun_blanchard_kahn1.m`.
- We solved the model for $(\tilde{k}_t, \tilde{z}_t, \tilde{c}_t)_{t=0}^{\infty}$. What if we are interested in other variables such as output, wage rate, etc.?
  - Either add them in the system $A \cdot E_t x_{t+1} = B \cdot x_t$ or compute the simulated series $(\tilde{y}_t, \tilde{w}_t, \tilde{r}_t)_{t=0}^{\infty}$ *after* solving the system.

Final remarks

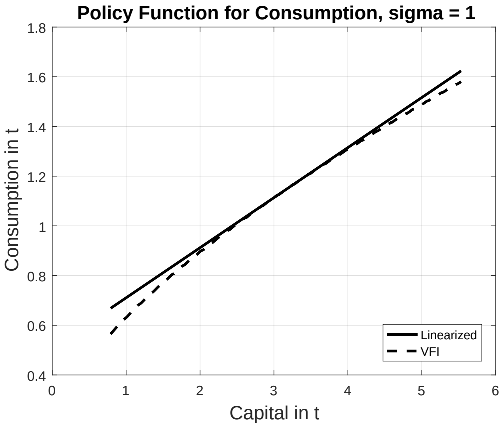
- Matlab codes to replicate material on these notes can be found in ILIAS:
  - main file: `main_stochastic_growth_model.m`;
  - auxiliary file: `fun_blanchard_kahn1.m`.
- We solved the model for $(\tilde{k}_t, \tilde{z}_t, \tilde{c}_t)_{t=0}^{\infty}$. What if we are interested in other variables such as output, wage rate, etc.?
  - Either add them in the system $A \cdot E_t x_{t+1} = B \cdot x_t$ or compute the simulated series $(\tilde{y}_t, \tilde{w}_t, \tilde{r}_t)_{t=0}^{\infty}$ *after* solving the system.
- Did we obtain the "true" solution?
  - We computed a local approximation around a particular point (i.e. deterministic steady state).
  - Whether such local approximation is accurate enough depends on the model and on the research question at hand.

Linearization vs VFI

- Compare linearized solution we have seen so far with a "global" solution method such as VFI.
- VFI delivers a global solution, i.e. a policy function for consumption and next-period capital *specified over the entire state space*.
- Linearization, instead, provides a solution that is accurate enough only in a (small) neighborhood around the approximation point.

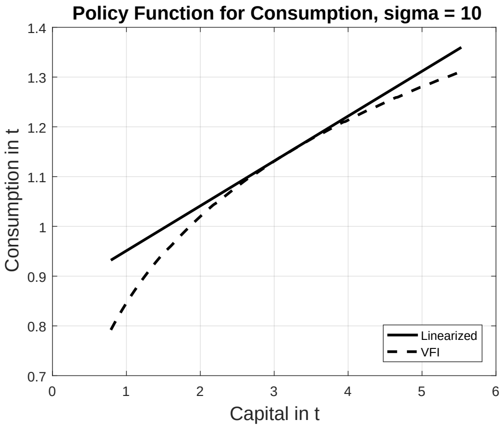
Accuracy I

- The fit is good, especially near the steady state.



Accuracy II

- The approximation gets worse as CRRA σ gets bigger (the policy function becomes more concave).



Appendix

Eigenvalues and Eigenvectors

Definition

Let A be a square matrix and let λ be a scalar. If $A\mathbf{v} = \lambda\mathbf{v}$ for some nonzero vector $\mathbf{v}$, then we say that λ is an eigenvalue of A and $\mathbf{v}$ is the corresponding eigenvector.

Intuition: In general $A\mathbf{v}$ is not proportional to $\mathbf{v}$. But if it is, then the proportionality factor, λ, is called eigenvalue and $\mathbf{v}$ is an eigenvector of A corresponding to λ.

Eigenvalues and Eigenvectors: Properties

Theorem

Let A be a square matrix and let λ be a scalar. The following statements are equivalent:

(a) *λ is an eigenvalue of A and $\mathbf{v}$ is the corresponding eigenvector.*

(b) *$A - \lambda I$ is a singular matrix.*

(c) *$det(A - \lambda I) = 0$.*

Remarks: (a) is the definition of eigenvalue. (c) implies that λ is an eigenvalue of A if it is a zero of the characteristic polynomial of A. Eigenvectors are not unique: if $\mathbf{v}$ is an eigenvector, then also any multiple $\alpha\mathbf{v}$, with $\alpha \neq 0$, is an eigenvector. Most softwares standardize an eigenvector to have unit length, i.e. $\|\mathbf{v}\| = 1$.

Jordan decomposition

For simplicity, let's assume that the $n \times n$ matrix A has n distinct real eigenvalues. Then we can state the following theorem.

Theorem

Form the matrix D

$$D = [v_1 \ v_2 \ldots v_n]$$

whose columns are the n eigenvectors corresponding to the n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$. Then $D^{-1}AD = \Lambda$ where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix}.$$

Back

Example

Let matrix A be

$$A = \begin{bmatrix} -1 & 3 \\ 2 & 0 \end{bmatrix}$$

The characteristic polynomial is $\lambda^2 + \lambda - 6 = 0$ whose roots are $\lambda_1 = -3$ and $\lambda_2 = 2$. The eigenvector(s) for $\lambda_1 = -3$ solve $Av = -3v$ hence

$$\mathbf{v}_1 = (1,\ -2/3)^\top$$

for example is an eigenvector. The eigenvector(s) for $\lambda_2 = 2$ solve $Av = 2v$ hence

$$\mathbf{v}_2 = (1,\ 1)^\top$$

is an eigenvector. Ordering eigenvalues from smallest to largest (in absolute value) we get

$$\Lambda = \begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix}, D = \begin{bmatrix} 1 & 1 \\ 1 & -2/3 \end{bmatrix}, D^{-1} = \begin{bmatrix} \frac{2}{5} & \frac{3}{5} \\ \frac{3}{5} & -\frac{3}{5} \end{bmatrix}.$$

Verify that $D^{-1}AD = \Lambda$.