Numerical Analysis and Computational Mathematics

Fall Semester 2025 – CSE Section

Prof. Laura Grigori

Assistant: Israa Fakih

Session 14 – December 17, 2025

Solutions – Ordinary differential equations

Solution I (MATLAB)

a) We consider the following MATLAB commands to obtain the result in Fig. 1 (left):

```
C1 = 0.15;    C2 = 0.075;
b1 = 0.002;   b2 = 0;
d1 = 0.0210;  d2 = 0.0325;
t0 = 0;       tf = 600;
y0 = [ 55; 20 ]; % [35; 40];
fun = @( t, y ) [ C1 * y( 1 ) * ( 1 - b1 * y( 1 ) - d2 * y( 2 ) ); ...
                 - C2 * y( 2 ) * ( 1 - b2 * y( 2 ) - d1 * y( 1 ) ) ];
Nh = 5000;
[ tv, uv_forward_euler ] = forward_euler_system( fun, y0, t0, tf, Nh );
plot( tv, uv_forward_euler( 1, : ), '.-b', tv, uv_forward_euler( 2, : ), '.-r' );
grid; axis([-0.1+t0 tf+0.1 -1 80]);
legend('Prey','Predator','Location','SouthEast');
```

We observe that the components of the solution $\mathbf{u}_n \in \mathbb{R}^2$ tend to an equilibrium vector $\mathbf{y}_E \in \mathbb{R}^2$ for t "sufficiently" large.

b) We use the following MATLAB commands to plot the trajectory of the solution in the phase space as in Fig. 1(right):

```
equilibrium_point = [ 47.6190; 27.8388 ];
figure; plot( uv_forward_euler( 1, : ), uv_forward_euler( 2, : ), '-k', ...
      equilibrium_point( 1 ), equilibrium_point( 2 ), '+r' );
xlabel('Prey'); ylabel('Predator');
grid; axis([10 80 10 50]);
```

The trajectory confirms the evolution of the components of the solutions (the number of preys and predators) from the initial condition $\mathbf{y}_0 = (55, 20)^T \in \mathbf{R}^2$ to the equilibrium point $\mathbf{y}_E \in \mathbb{R}^2$ (indicated in red).

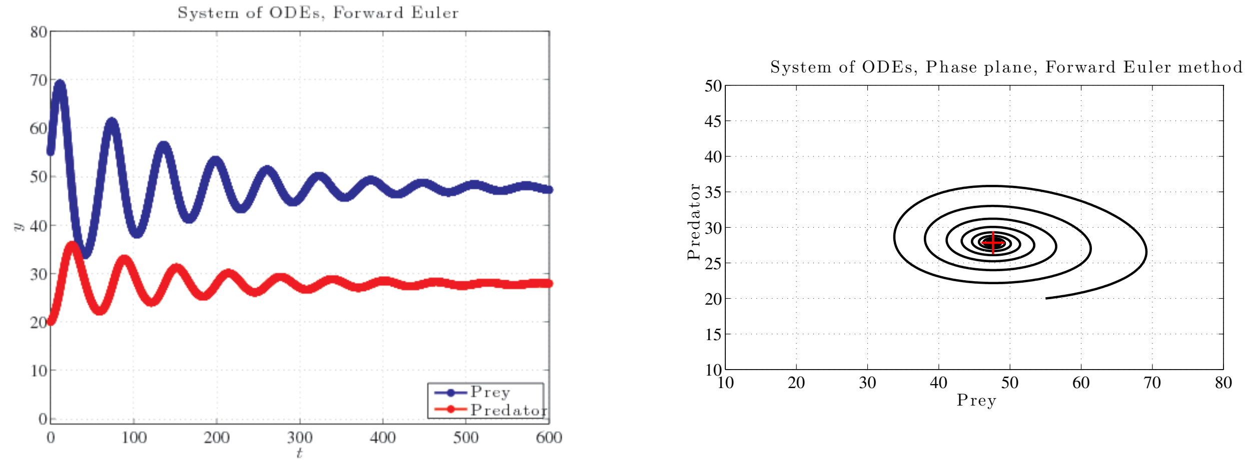


Figure 1: Prey-Predator model with $\mathbf{y}_0 = (55, 20)^T$; approximate solution $\mathbf{u}_n$ vs t (left) and trajectory in the phase space (right).

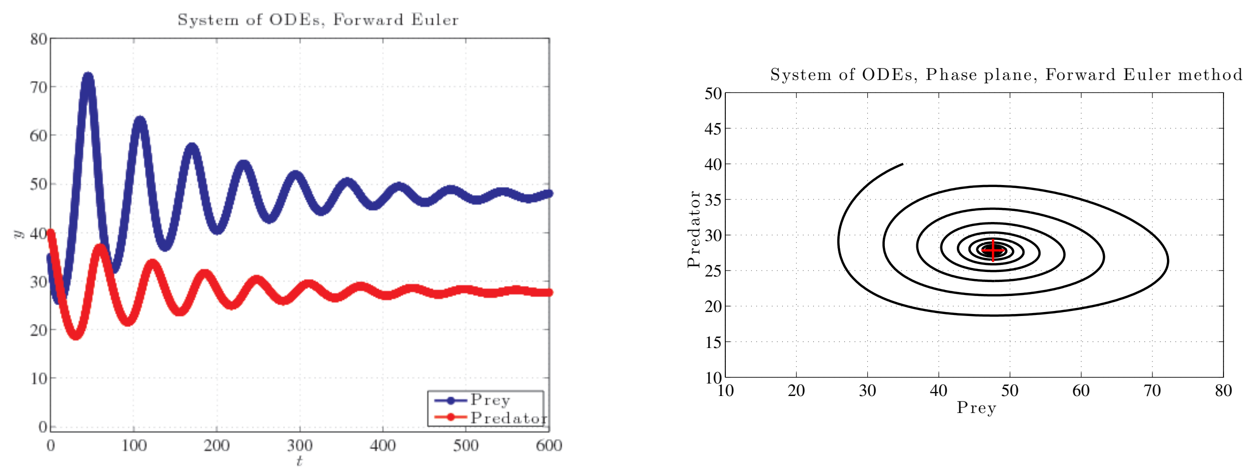


Figure 2: Prey-Predator model with $\mathbf{y}_0 = (35, 40)^T$; approximate solution $\mathbf{u}_n$ vs t (left) and trajectory in the phase space (right).

c) We repeat points a) and b) with a different initial condition $\mathbf{y}_0 = (35, 40)^T$, where the number of predators is larger than the number of preys. We obtain the result of Fig. 2 for which we observe that, even if we start from a different initial condition and the trajectory is different from that obtained at point b), the number of preys and predators still converge to the same equilibrium point $\mathbf{y}_E$. However, we remark that the previous result largely depends on the data chosen for the prey-predator model.

Solution II (MATLAB)

a) We consider the following MATLAB commands to obtain the truss bridge reported in Fig. 3.

```
kbeam = 1e3; mbeam = 2;
alpha = 0.01; beta = alpha;
Nnodes = 29; % number of nodes of the bridge (odd)
m = 2 * Nnodes;
[ K ] = bridge_stiffness_matrix( Nnodes, kbeam );
M = mbeam * speye( m, m );
C = alpha * M + beta * K;
plot_bridge( Nnodes );
```

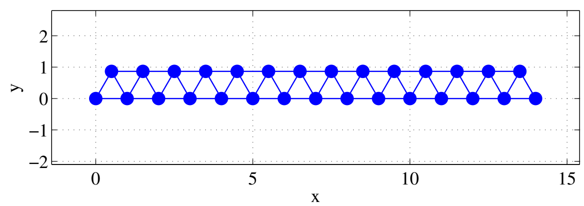


Figure 3: Truss bridge model for $N_{nodes} = 29$.

b) We use the following commands to define the matrices $\widetilde{M}$, $\widetilde{C}$, and $\widetilde{K}$, with $\widetilde{m} = 2N_{nodes} - 3$.

```
i_t = [ 3 : m - 1 ];
m_t = m - 3;
M_t = M( i_t, i_t ); C_t = C( i_t, i_t ); K_t = K( i_t, i_t );
```

c) In order to rewrite the system, we set $\widetilde{\mathbf{y}}(t) = (\widetilde{\mathbf{w}}(t)^T, \widetilde{\mathbf{d}}(t)^T)^T \in \mathbb{R}^{2\widetilde{m}}$ with $\widetilde{\mathbf{w}}(t) = \widetilde{\mathbf{d}}'(t) \in \mathbb{R}^{\widetilde{m}}$ an auxiliary vector representing the velocity of the nodes. The initial condition reads $\widetilde{\mathbf{y}}_0 = (\widetilde{\mathbf{v}}_0^T, \widetilde{\mathbf{d}}_0^T)^T$. By introducing the block invertible matrix $\widetilde{P} = \begin{bmatrix} \widetilde{M} & 0 \\ 0 & \widetilde{I} \end{bmatrix}$ with $\widetilde{I} \in \mathbb{R}^{\widetilde{m} \times \widetilde{m}}$ the identity matrix, we obtain that $\widetilde{A} = \widetilde{P}^{-1} \begin{bmatrix} -\widetilde{C} & -\widetilde{K} \\ \widetilde{I} & 0 \end{bmatrix}$ and $\widetilde{\mathbf{g}}(t) = \widetilde{P}^{-1} \left(\widetilde{\mathbf{b}}(t)^T, \mathbf{0} \right)^T$.

```
I_t = speye( m_t, m_t );
P_t = [ M_t, sparse( m_t, m_t ); sparse( m_t, m_t ), I_t ];
A_t = P_t \ [ - C_t, - K_t; I_t, sparse( m_t, m_t ) ];
```

d) We consider the following MATLAB commands for the external force $\mathbf{f}_{15}^{ext}(t) = (0, -q_1(t))^T$:

```
node_force = 15; % node in which the force is applied
b = zeros( m, 1 ); b( 2 * node_force, 1 ) = -1;
b_t = b( i_t, 1 );
y0_t = zeros( 2 * m_t, 1 );
t0 = 0; tf = 250;
```

```
t_ref = 25;
g_t = @( t ) P_t \ [ b_t * ( ( t / t_ref ) * ( t ≤ t_ref ) + ...
                             1 * ( t > t_ref ) ); zeros( m_t, 1 ) ];
Nh = 2500;
h = ( tf - t0 ) / Nh;
[ tv, uv ] = backward_euler_system_nhcc( A_t, g_t, y0_t, t0, tf, Nh );
d_t = uv( m_t + 1 : end, : );
d = zeros( m, Nh + 1 );
d( i_t, : ) = d_t( :, : ); % displacement vector (including the constraints)
inode = 15; % node in which we are interested to evaluate the displacement
u_inode_x = d( 2 * inode - 1, : ); u_inode_y = d( 2 * inode, : );
plot( tv, u_inode_x, '-b', tv, u_inode_y, '-r' );
grid on; xlabel('t'); ylabel('disp');
legend('disp x','disp y' );
```

We obtain the result displayed in Fig. 4(a). We observe that both the horizontal and vertical displacements converge to constant values after some oscillations for $t > t_{ref}$.

Fig. 4(b) is obtained by setting $\mathbf{f}_{15}^{ext}(t) = (0, -q_2(t))^T$ and redefining $\widetilde{\mathbf{g}}(t)$ as:

```
g_t = @( t ) P_t \ [ b_t * ( ( t / t_ref ) * ( t ≤ t_ref ) ); zeros( m_t, 1 ) ];
```

The results of Fig. 4(c–f) are obtained by setting $\mathbf{f}_{15}^{ext}(t) = (0, -q_k(t))^T$ with $k = 3, 4, 5$. For example, for the case $k = 3$ we use the MATLAB command:

```
omega = 0.25;
g_t = @( t ) P_t \ [ b_t * sin( omega * t ); zeros( m_t, 1 ) ];
```

We observe that in Fig. 4(c–d) the amplitude of the oscillations induced on the displacement by the external force $\mathbf{f}_{15}^{ext}(t)$ remains relatively small for $\omega_3 = 0.25$ and $\omega_5 = 0.65$.

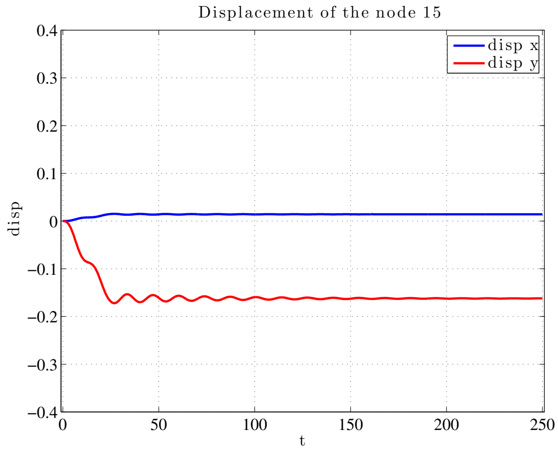
For $\omega_4 = 0.4688$, the amplitude of the oscillations of the displacements is very large compared to the previous cases. Indeed, the dynamical system is in condition of resonance for $\omega = \omega_4 = 0.4688$ since this angular frequency is close to one of the natural frequencies of the structural model. We can compute the values of these natural angular frequencies by solving a generalized eigenvalue problem as originally done in Series 11. Specifically, we consider the following MATLAB command to compute the 5 smallest natural angular frequencies:

```
omega_natural = sort( sqrt( eigs( K_t, M_t, 5, 'SM' )' ) )
% omega_natural =
%      0.4688    1.6652    2.3342    3.6036    5.5365
```

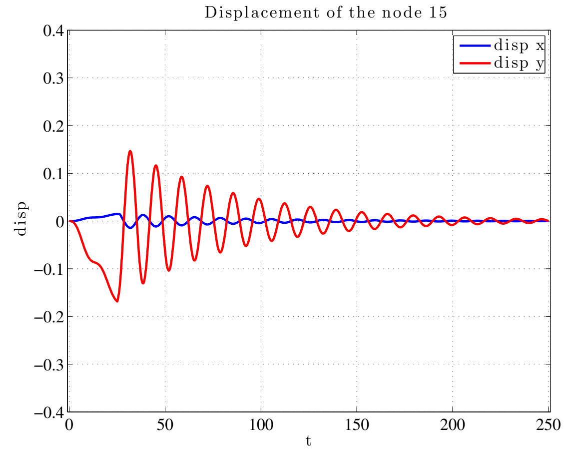
In Fig. 4(f) we report the result obtained for the same external force but without damping ($C = 0$). We observe that the amplitude of the oscillations is larger than in the previous (damped) case.

The following MATLAB commands can be used to visualize with an animation the dynamics of the truss bridge under the action of the external forces.

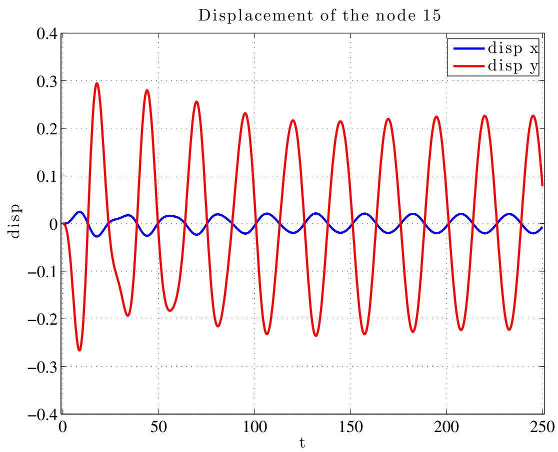
```
for i = 1 : 10 : Nh + 1
    plot_bridge( Nnodes, d( :, i ) );
    pause( 0.1 )
```



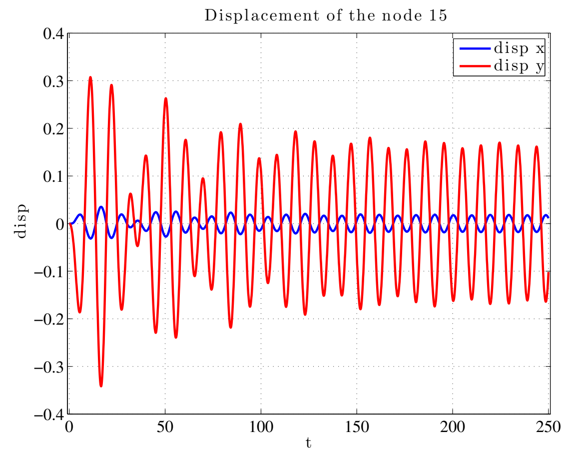
(a) $q_1(t)$



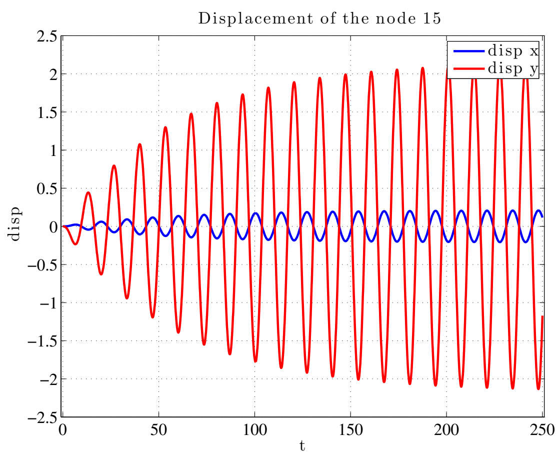
(b) $q_2(t)$



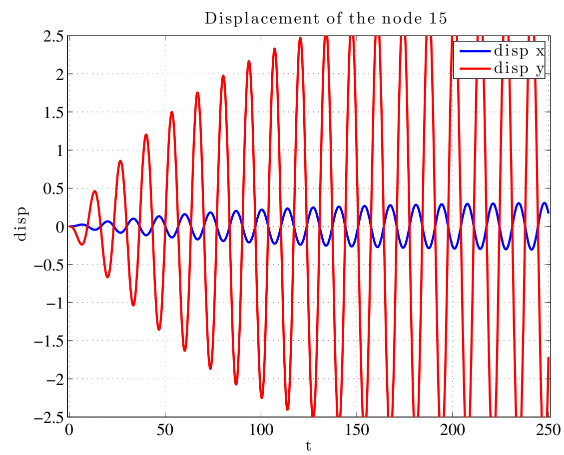
(c) $q_3(t)$



(d) $q_5(t)$



(e) $q_4(t)$, $\alpha = \beta = 0.01$



(f) $q_4(t)$, $\alpha = \beta = 0$

Figure 4: Numerical approximations of the displacements $\mathbf{u}_{15,x}(t)$ (blue) and $\mathbf{u}_{15,y}(t)$ (red) vs. time t for $\mathbf{f}_{15}^{ext}(t) = (0, -q_k(t))^T$ with $k = 1, \ldots, 5$.

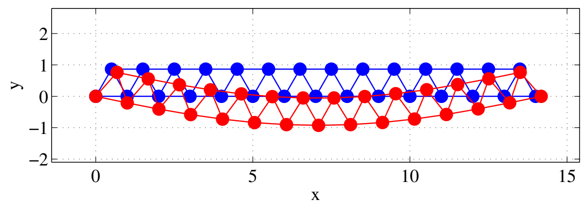


Figure 5: Deformed configuration of the truss bridge model at time $t = 62$ obtained for $\mathbf{f}_{15}^{ext}(t) = (0, -q_4(t))^T$.

```
end
```

The deformed configuration of the bridge obtained for $\mathbf{f}_{15}^{ext}(t) = (0, -q_4(t))^T$ at time $t = 62$ is reported for reference in Fig. 5.