Supplementary Material

The codes utilized in this research were crafted using MATLAB 2023. MATLAB, short for MATrix LABoratory, is a widely employed high-level programming language and interactive environment specifically designed for numerical computing, algorithm development, and data analysis.

Part 1: Differential equations

This section defines a system of differential equations modeling a trophic chain of three levels. It computes the rate of change for each species' population based on interactions and dynamics within the ecological system.

```
function ydot = sys5M1(t, x)
%Differential Equations System
global ePCj phiPCj ePCa phiPCa ePAj phiPAj ePAa phiPAa dP dCj GC dAj GA dCa dAa rR
KR phiCjR phiCaR phiAjR phiAaR eCaR eAaR H0PCj H0PCa H0PAj H0PAa H0CjR
H0CaR H0AjR H0AaR;

ydot(1) = (ePCj * phiPCj * x(2) * x(1)) / (1 + x(2) / H0PCj) + (ePCa * phiPCa * x(3) * x(1)) / (1 +
x(3) / H0PCa) + (ePAj * phiPAj * x(4) * x(1)) / (1 + x(4) / H0PAj) + (ePAa * phiPAa * x(5) *
x(1)) / (1 + x(5) / H0PAa) - dP * x(1);
ydot(2) = -(dCj + GC + x(1)) * x(2);
ydot(3) = GC * x(2) - (dCa + (phiPCa * x(1)) / (1 + x(3) / H0PCa)) * x(3) + (eCaR * phiCaR *
x(3) * x(6)) / (1 + x(6) / H0CaR);
ydot(4) = -(dAj + GA + x(1)) * x(4);
ydot(5) = GA * x(4) - (dAa + (phiPAa * x(1)) / (1 + x(5) / H0AaR)) * x(5) + (eAaR * phiAaR *
x(5) * x(6)) / (1 + x(6) / H0AaR);
ydot(6) = rR * x(6) * (1 - x(6) / KR) - ((phiCjR * x(2)) / (1 + x(6) / H0CjR) + (phiCaR * x(3)) / (1
+ x(6) / H0CaR)) * x(6) - ((phiAjR * x(4)) / (1 + x(6) / H0AjR) + (phiAaR * x(5)) / (1 + x(6) /
H0AaR)) * x(6);

ydot = [ydot(1); ydot(2); ydot(3); ydot(4); ydot(5); ydot(6)];
end
```

Part 2: Impulsive effects

Using an impulsive effect approach, this segment computes solutions to the differential equations with sudden alterations in population dynamics over defined time intervals, reflecting the response to impulse-induced changes.

```
function [t, y] = ode15spM1(ti, tf, y10, y20, y30, y40, y50, y60, b, c, pulsestep)

clear y;
clear z;
% The vector y contains the solutions of the formula.
% Vector z stores the components of vector y.
y(1,1) = y10;
y(1,2) = y20;
```

```
y(1,3) = y30;
y(1,4) = y40;
y(1,5) = y50;
y(1,6) = y60;
rep = ceil((tf - ti) / pulsestep);
clear s ytemp;
z = zeros(rep, 6);
z(1,1) = y10;
z(1,2) = y20;
z(1,3) = y30;
z(1,4) = y40;
z(1,5) = y50;
z(1,6) = y60;

% Integrate the system and obtain vector y
fg = @(t, x) sys5M1(t, x);
[s, ytemp] = ode45(fg, [ti, ti + pulsestep], [y10, y20, y30, y40, y50, y60]);
[evalpts, cols] = size(ytemp);
% Store the components of vector Y in a temporary component of Y
for j = 1:evalpts
    t(j) = s(j);
    y(j,1) = ytemp(j,1);
    y(j,2) = ytemp(j,2);
    y(j,3) = ytemp(j,3);
    y(j,4) = ytemp(j,4);
    y(j,5) = ytemp(j,5);
    y(j,6) = ytemp(j,6);
end

% Evaluation occurs at successive intervals.
for k = 1:rep-1
    [sizey, cols] = size(y);
    y1int = y(sizey,1) + 6;
    y2int = y(sizey,2) + b * y(sizey,3) * y(sizey,6);
    y3int = y(sizey,3);
    y4int = y(sizey,4) + c * y(sizey,5) * y(sizey,6);
    y5int = y(sizey,5);
    y6int = y(sizey,6) + 40;
    z(k+1,1) = y1int;
    z(k+1,2) = y2int;
    z(k+1,3) = y3int;
    z(k+1,4) = y4int;
    z(k+1,5) = y5int;
    z(k+1,6) = y6int;
    clear s ytemp;
    % Reintegrate with the new initial conditions
   [s, ytemp] = ode45(fg, [ti + k * pulsestep, ti + (k+1) * pulsestep], [y1int, y2int, y3int, y4int,
y5int, y6int]);
    [evalpts, cols] = size(ytemp);
```

```
    % Store the information to perform integration again with the new initial conditions until the chosen final time.
    for j = 1:evalpts
        t(sizey+j) = s(j);
        y(sizey+j,1) = ytemp(j,1);
        y(sizey+j,2) = ytemp(j,2);
        y(sizey+j,3) = ytemp(j,3);
        y(sizey+j,4) = ytemp(j,4);
        y(sizey+j,5) = ytemp(j,5);
        y(sizey+j,6) = ytemp(j,6);
    end
end
end
```

Part 3: Solution visualization

Integrating the previous functions, this part generates figures depicting population dynamics for different scenarios (various values of MR) in the trophic chain model. It presents the abundance of species over time under distinct environmental conditions, aiding visualization and interpretation of model behavior.

```
% With this code, the graphs of the solutions are obtained
clear all
close all
clc

global ePCj phiPCj ePCa phiPCa ePAj phiPAj ePAa phiPAa dP dCj GC dAj GA dCa dAa rR KR phiCjR phiCaR phiAjR phiAaR eCaR eAaR H0PCj H0PCa H0PAj H0PAa H0CjR H0CaR H0AjR H0AaR;

% Proposal 2
MP = 100;
MCj = 1;
MCa = 2.5;
MAj = 0.8;
MAa = 1.5;
% MR = 0.5;

% Energetic parameters
alpha = 0.75;
beta = 0.5;

% Intrinsic reproductive rates
r0R = 2.7;

% Carrying capacity
K0R = 4 * 3.8;

% Mortality rate of predator
```

```
dP0 = 0.74;
dCj0 = 0.74;
dAj0 = 0.74;
dCa0 = 0.74;
dAa0 = 0.74;

dCj = (dCj0) * ((MCj) ^ (alpha - 1));
dAj = (dAj0) * ((MAj) ^ (alpha - 1));
dCa = (dCa0) * ((MCa) ^ (alpha - 1));
dAa = (dAa0) * ((MAa) ^ (alpha - 1));

GC = 53 * (MCa ^ 0.27);
GA = 53 * (MAa ^ 0.27);

ePCj0 = 0.2;
ePCa0 = 0.7;
ePAj0 = 0.5;
ePAa0 = 0.4;
eCaR0 = 0.5;
eCjR0 = 0.3;
eAaR0 = 0.5;
eAjR0 = 0.2;

ePCj = ePCj0 * (MCj / MP);
ePCa = ePCa0 * (MCa / MP);
ePAj = ePAj0 * (MAj / MP);
ePAa = ePAa0 * (MAa / MP);

F = @(x, y) (1 + (x / y) ^ (-beta));
Phi = @(x, y) (1 - exp(-(x / y) ^ 2));

fPCj = 0.3;
fPCa = 0.1;
fPAj = 0.5;
fPAa = 0.2;
fCjR = 0.4;
fCaR = 0.4;
fAjR = 0.4;
fAaR = 0.4;

phiPCj = fPCj * F(MP, MCj) * Phi(MP, MCj);
phiPCa = fPCa * F(MP, MCa) * Phi(MP, MCa);
phiPAj = fPAj * F(MP, MAj) * Phi(MP, MAj);
phiPAa = fPAa * F(MP, MAa) * Phi(MP, MAa);

H = @(x, y) (x .^ (alpha)) / (y);

h0PCj = 0.2;
h0PCa = 0.3;
```

```
h0PAj = 0.2;
h0PAa = 0.3;
h0CjR = 0.2;
h0CaR = 0.4;
h0AjR = 0.2;
h0AaR = 0.4;

H0PCj = ((h0PCj) / (ePCj0 * phiPCj)) * H(MP, MCj);
H0PCa = ((h0PCa) / (ePCa0 * phiPCj)) * H(MP, MCa);
H0PAj = ((h0PAj) / (ePAj0 * phiPAj)) * H(MP, MAj);
H0PAa = ((h0PAa) / (ePAa0 * phiPAj)) * H(MP, MAa);

% Image 1: b = 3.2 and c = 0.7
% Image 2: b = 4 and c = 2.5

% Previous code up to the point where the ode15spM1 function is defined

% Specific values of MP

% Proposals for different values of MP
MR_values = [0.04, 0.5, 0.7, 15];

% Create a figure to organize subplots
figure;

for i = 1:length(MR_values)
    MR = MR_values(i);

    % Rest of the code with equations dependent on MR
    rR = r0R * ((MR) ^ (alpha - 1));
    KR = K0R * ((MR) ^ (-alpha));
    dP = (dP0) * ((MP) ^ (alpha - 1));
    eCaR = eCaR0 * (MR / MCa);
    eAaR = eAaR0 * (MR / MAa);
    phiCjR = fCjR * F(MCj, MR) * Phi(MCj, MR);
    phiCaR = fCaR * F(MCa, MR) * Phi(MCa, MR);
    phiAjR = fAjR * F(MAj, MR) * Phi(MAj, MR);
    phiAaR = fAaR * F(MAa, MR) * Phi(MAa, MR);
    H0CjR = ((h0CjR) / (eCjR0 * phiCjR)) * H(MCj, MR);
    H0CaR = ((h0CaR) / (eCaR0 * phiCaR)) * H(MCa, MR);
    H0AjR = ((h0AjR) / (eAjR0 * phiAjR)) * H(MAj, MR);
    H0AaR = ((h0AaR) / (eAaR0 * phiAaR)) * H(MAa, MR);

    % Obtain solutions for the current value of MP
    [t, y] = ode15spM1(0, 30, 18, 160, 140, 160, 240, MR, 4, 2.5, 1);

    % Organize graphs in subplots
    subplot(2, 2, i); % 2 rows, 2 columns, current position
    plot(t, y(:, 1), 'r-', 'LineWidth', 4);
```

```
    hold on;
    plot(t, y(:, 2), 'b-', 'LineWidth', 4);
    plot(t, y(:, 3), 'b--', 'LineWidth', 4);
    plot(t, y(:, 4), 'm-', 'LineWidth', 4);
    plot(t, y(:, 5), 'm--', 'LineWidth', 4);
    plot(t, y(:, 6), 'g-', 'LineWidth', 4);

    legend('Predator','Native juvenil','Native adult','Invasive juvenil','Invasive adult','Basal
resource');
    xlabel('Time (years)', 'FontWeight', 'bold');
    ylabel('Abundance (n)', 'FontWeight', 'bold');
    title(['Abundance for MR = ' num2str(MR)]);
    set(gca, 'FontSize', 25);
    set(gca, 'LineWidth', 2);
end
```