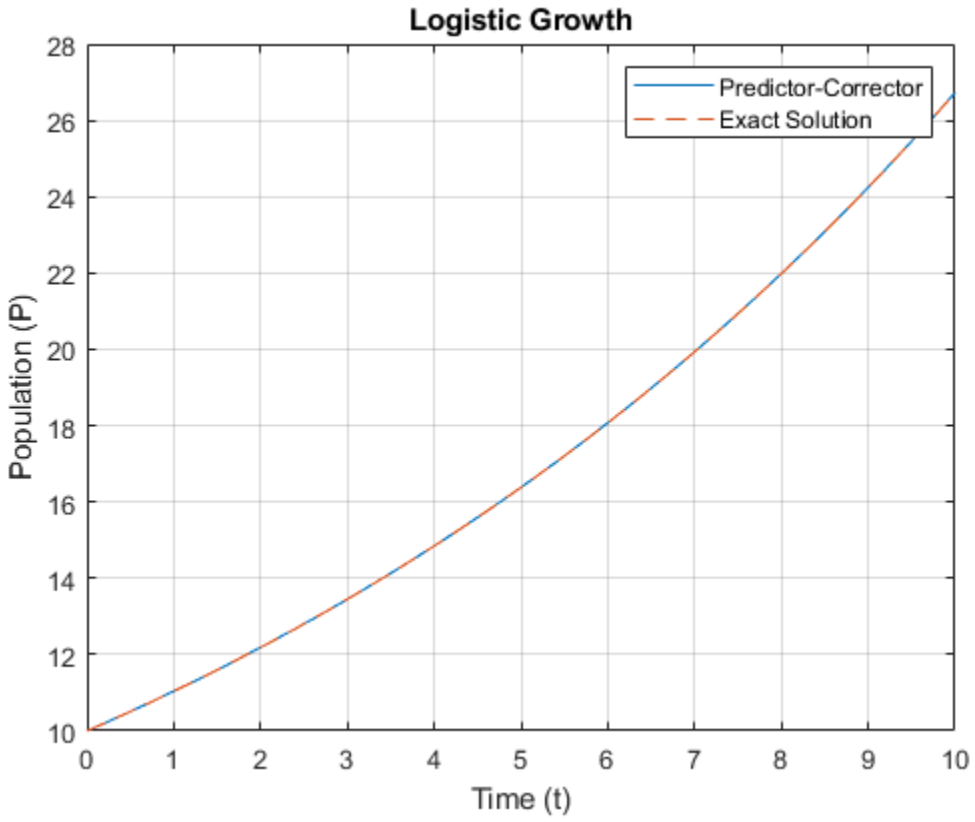
```
function logistic_growth_plot()

    % Parameters
    t0 = 0.0;
    P0 = 10.0;
    h = 0.1;
    N = 100;
    r = 0.1;
    K = 1000;

    [t_vals, P_vals] = predictor_corrector(P0, t0, h, N, r, K);

    % Visualization
    figure;
    plot(t_vals, P_vals, 'DisplayName', 'Predictor-Corrector');
    hold on;
    plot(t_vals, arrayfun(@(t) exact_solution(t, P0, r, K),
 t_vals), '--', 'DisplayName', 'Exact Solution');
    xlabel('Time (t)');
    ylabel('Population (P)');
    legend();
    title("Logistic Growth");
    grid on;

    function dpdt = logistic_growth(t, P, r, K)
        dpdt = r * P * (1 - P / K);
    end

    function P = exact_solution(t, P0, r, K)
        P = (K * P0 * exp(r * t)) / (K + P0 * (exp(r * t) - 1));
    end

    function [t, P] = predictor_corrector(y0, t0, h, N, r, K)
        t = t0;
        P = y0;

        % Bootstrap using 4th order Runge-Kutta
        for i = 1
            k1 = h * logistic_growth(t(end), P(end), r, K);
            k2 = h * logistic_growth(t(end) + 0.5 * h, P(end) + 0.5 * k1, r,
 K);
            k3 = h * logistic_growth(t(end) + 0.5 * h, P(end) + 0.5 * k2, r,
 K);
            k4 = h * logistic_growth(t(end) + h, P(end) + k3, r, K);

            P(end+1) = P(end) + (k1 + 2 * k2 + 2 * k3 + k4) / 6;
            t(end+1) = t(end) + h;
        end

        for i = 2:N
            % Predictor
```

```
            P_predict = P(end) + h * (1.5 * logistic_growth(t(end), P(end), r,
 K) - 0.5 * logistic_growth(t(end-1), P(end-1), r, K));
            t_predict = t(end) + h;

            % Corrector
            P_correct = P(end) + h/2 * (logistic_growth(t(end), P(end), r, K)
 + logistic_growth(t_predict, P_predict, r, K));

            P(end+1) = P_correct;
            t(end+1) = t_predict;
        end
    end

end
```



Published with MATLAB® R2023a