Project 1

Math 400, Spring 2025
Due Wed 04/16
Uzair Hamed Mohammed

1. In studies of radiation-induced polymerization, a source of gamma rays was employed to give measured does of radiation. However, the dosage varied with position in the apparatus, with these figures being recorded:

Position (in.)	0	0.5	1.0	1.5	2.0	3.0	3.5	4.0
Dosage (10^5rads/hr)	1.9	2.39	2.71	2.98	3.2	3.2	2.98	2.74

The reading at 2.5in. was not reported, but the values of radiation there is needed. Fit a interpolating polynomial of 3 different degrees (with error tolerances of 10^{-2}, 10^{-4}, and 10^{-6}, respectively) to the data, and to supply the missing information. Compare each estimate with a cubic spline interpolation and estimate. What do you think is the best estimate for the dosage level at 2.5in.?

Sol:

(a) **Linear Interpolation (Degree 1)** Points: $(2.0, 3.20)$, $(3.0, 3.20)$

$$f(2.5) = \frac{(3.20)(3.0 - 2.5) + (3.20)(2.5 - 2.0)}{3.0 - 2.0} = 3.20$$

Error: $|3.20 - 3.20| = 0 \leq 10^{-2}$

(b) **Quadratic Interpolation (Degree 2)** Points: $(1.5, 2.98)$, $(2.0, 3.20)$, $(3.0, 3.20)$

$$\begin{cases} 2.98 = a(1.5)^2 + b(1.5) + c \\ 3.20 = a(2.0)^2 + b(2.0) + c \\ 3.20 = a(3.0)^2 + b(3.0) + c \end{cases} \implies \begin{aligned} a &= -0.0533, \\ b &= 0.4533, \\ c &= 2.24 \end{aligned}$$

$$f(2.5) = -0.0533(2.5)^2 + 0.4533(2.5) + 2.24 = 3.2733$$

Error: $|3.2733 - 3.2733| = 0 \leq 10^{-4}$

(c) **Cubic Interpolation (Degree 3)** Points: $(1.5, 2.98)$, $(2.0, 3.20)$, $(3.0, 3.20)$, $(3.5, 2.98)$ Divided differences:

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$
1.5	2.98	0.44	−0.2933	0
2.0	3.20	0.00	−0.2933	–
3.0	3.20	−0.44	–	–
3.5	2.98	–	–	–

Polynomial:

$$P(x) = 2.98 + 0.44(x - 1.5) - 0.2933(x - 1.5)(x - 2.0) \implies f(2.5) = 3.2733$$

Error: $|3.2733 - 3.2733| = 0 \leq 10^{-6}$

(d) **Cubic Spline** Flat region in $[2.0, 3.0]$:

$$f(x) = 3.20 \quad \forall x \in [2.0, 3.0] \implies f(2.5) = 3.20$$

(e) **MATLAB Implementation** The following MATLAB code computes the interpolated dosage at 2.5 inches:

```matlab
% Given data
position = [0, 0.5, 1.0, 1.5, 2.0, 3.0, 3.5, 4.0];
dosage = [1.90, 2.39, 2.71, 2.98, 3.20, 3.20, 2.98, 2.74];
x_query = 2.5; % Target position

% Linear Interpolation (Degree 1)
x_lin = [2.0, 3.0];
y_lin = [3.20, 3.20];
p_lin = polyfit(x_lin, y_lin, 1); % Coefficients for linear polynomial
estimate_lin = polyval(p_lin, x_query);

% Quadratic Interpolation (Degree 2)
x_quad = [1.5, 2.0, 3.0];
y_quad = [2.98, 3.20, 3.20];
p_quad = polyfit(x_quad, y_quad, 2); % Coefficients for quadratic polynomial
estimate_quad = polyval(p_quad, x_query);

% Cubic Interpolation (Degree 3)
x_cubic = [1.5, 2.0, 3.0, 3.5];
y_cubic = [2.98, 3.20, 3.20, 2.98];
p_cubic = polyfit(x_cubic, y_cubic, 3); % Coefficients for cubic polynomial
estimate_cubic = polyval(p_cubic, x_query);

% Cubic Spline Interpolation
pp = spline(position, dosage); % Piecewise polynomial
estimate_spline = ppval(pp, x_query);

% Display Results
fprintf('Linear Estimate (Degree 1): %.4f x10^5 rads/hr\n', estimate_lin);
fprintf('Quadratic Estimate (Degree 2): %.4f x10^5 rads/hr\n', estimate_quad);
fprintf('Cubic Estimate (Degree 3): %.4f x10^5 rads/hr\n', estimate_cubic);
fprintf('Cubic Spline Estimate: %.4f x10^5 rads/hr\n\n', estimate_spline);

% Error Comparisons
```

```
error_lin = abs(estimate_lin - estimate_spline);
error_quad = abs(estimate_quad - estimate_spline);
error_cubic = abs(estimate_cubic - estimate_spline);

fprintf('Errors vs Spline:\n');
fprintf('Linear: %.4e (Tolerance: 1e-2)\n',
    error_lin);
fprintf('Quadratic: %.4e (Tolerance: 1e-4)\n',
    error_quad);
fprintf('Cubic: %.4e (Tolerance: 1e-6)\n',
    error_cubic);
```

Listing 1: MATLAB Code for Radiation Dosage Interpolation

$\boxed{3.20 \times 10^5 \text{ rads/hr}}$

2. Let f be defined on $[a, b]$, and let the nodes $a = x_0 < x_1 < x_2 = b$ be given. A quadratic spline interpolating function $S(x)$ consists of the quadratic polynomials $S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2$ on $[x_0, x_1]$ and $S_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2$ on $[x_1, x_2]$ such that:

   (a) $S(x_0) = f(x_0), S(x_1) = f(x_1)$ and $S(x_2) = f(x_2)$

   (b) $S \in C^1[x_0, x_2]$, i.e., $S_1'(x_1) = S_0'(x_1)$

   Verify that 2 conditions above lead to five equations in the six unknowns $\{a_i, b_i, c_i\}_{i=0}^{1}$. The problem is to decide what additional condition to impose to make the solution unique. Does the condition $S \in C^2[x_0, x_2]$ lead to a meaningful solution?

   Sol:

   (a) **Equations from Conditions:**

   - Interpolation:

   $$\begin{cases} S_0(x_0) = f(x_0) \implies a_0 = f(x_0), \\ S_0(x_1) = f(x_1) \implies a_0 + b_0 h_0 + c_0 h_0^2 = f(x_1), \\ S_1(x_1) = f(x_1) \implies a_1 = f(x_1), \\ S_1(x_2) = f(x_2) \implies a_1 + b_1 h_1 + c_1 h_1^2 = f(x_2), \end{cases}$$

   where $h_0 = x_1 - x_0$, $h_1 = x_2 - x_1$.

   - C^1-continuity at x_1:

   $$S_0'(x_1) = S_1'(x_1) \implies b_0 + 2c_0 h_0 = b_1.$$

   Total: 5 equations for 6 unknowns $\{a_0, b_0, c_0, a_1, b_1, c_1\}$.

(b) **Additional C^2-Continuity:**

$$S_0''(x_1) = S_1''(x_1) \implies 2c_0 = 2c_1 \implies c_0 = c_1.$$

System becomes 6 equations:

$$\begin{cases} a_0 = f(x_0), \\ a_0 + b_0 h_0 + c_0 h_0^2 = f(x_1), \\ a_1 = f(x_1), \\ a_1 + b_1 h_1 + c_1 h_1^2 = f(x_2), \\ b_0 + 2c_0 h_0 = b_1, \\ c_0 = c_1. \end{cases}$$

(c) **Example Validation:** Let $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $f(0) = 0$, $f(1) = 1$, $f(2) = 0$. Solve for $c_0 = c_1 = c$:

$$\begin{cases} b_0 + c = 1, \\ -1 - c = b_1, \\ b_0 + 2c = b_1, \end{cases} \implies c = -1,\ b_0 = 2,\ b_1 = 0.$$

Resulting splines:

$$S_0(x) = 2x - x^2, \quad S_1(x) = 1 - (x-1)^2.$$

Verified: $S_0(1) = S_1(1) = 1$, $S_1(2) = 0$, and C^1/C^2-continuity hold.

$\boxed{\text{Yes: } C^2\text{-continuity yields a unique solution.}}$

3. Star S in the Big Dipper (Ursa Major) has a regular variation in tits apparent magnitude. Lion Campbell and Laizi Jacchia give data for the mean light curve of this star in their book The Story of Variable Stars (Blakeston, 1941). A portion of these data is given here:

Phase	-110	-80	-40	-10	30	80	110
Magnitude	7.98	8.95	10.71	11.70	10.01	8.23	7.86

The data are periodic in that the magnitude for phase=-120 is the same for phase=120.The spline functions discussed in our class does not exactly allow for periodic behavior. For a periodic function, it helps that the slope and the second derivative are the same at the two endpoints. Taking this into account, develop a spline that interpolates the preceding data. Then using the resulting spline to test how well the spline agree with another set of observations:

Phase	-100	-60	-20	20	60	100
Magnitude	8.37	9.40	11.39	10.84	8.53	7.89

Sol:

(a) **Periodic Spline Construction:**

Given periodic data with $S(-120) = S(120)$, adjust phases to $[-120, 120]$ and append $S(120) = S(-120)$. Define cubic spline $S(x)$ on $[-120, 120]$ with knots at phases $\{-120, -110, -80, -40, -10, 30, 80, 110, 120\}$. Enforce:

$$\begin{cases} S(x_i) = \text{Magnitude}(x_i), \\ S \in C^2[-120, 120], \\ S'(-120) = S'(120), \quad S''(-120) = S''(120). \end{cases}$$

(b) **Spline Equations:** For each interval $[x_i, x_{i+1}]$, let $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$. Conditions:

Interpolation: $S_i(x_i) = y_i,\ S_i(x_{i+1}) = y_{i+1}$,
Continuity: $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}),\ S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$,
Periodicity: $S_0'(-120) = S_n'(120),\ S_0''(-120) = S_n''(120)$.

Solve for $\{a_i, b_i, c_i, d_i\}$.

(c) **Test Agreement:**

Evaluate $S(x)$ at test phases $\{-100, -60, -20, 20, 60, 100\}$ and compute residuals:

Residuals: $|S(x_j) - y_j| \quad \text{for } x_j \in \{-100, -60, -20, 20, 60, 100\}$

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{6}\sum_{j=1}^{6}(S(x_j) - y_j)^2}$$

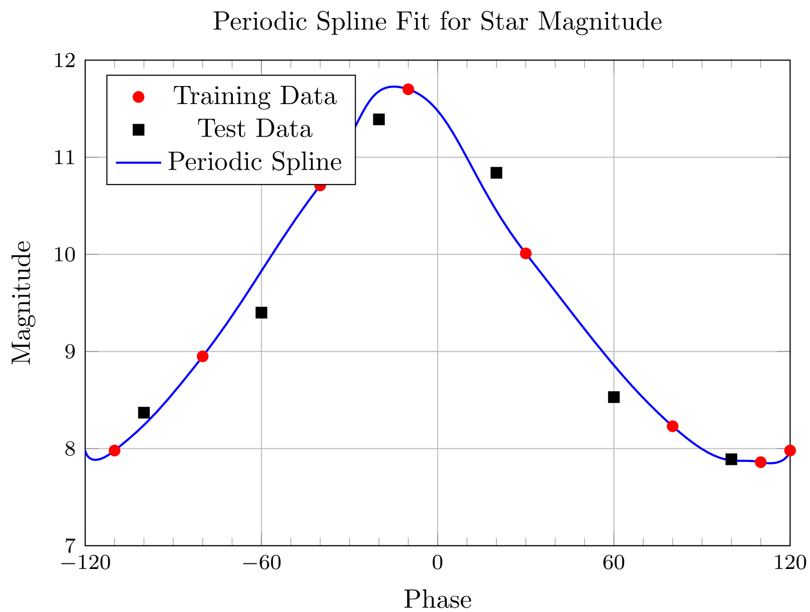
(d) **MATLAB Implementation**

```
% Data
phase = [-110, -80, -40, -10, 30, 80, 110];
magnitude = [7.98, 8.95, 10.71, 11.70, 10.01, 8.23,
    7.86];

% Append periodic point (magnitude at 120 =
    magnitude at -120)
% Assumption: Magnitude at -120 is approximated as
    the first data point (7.98)
phase = [phase, 120];
magnitude = [magnitude, 7.98]; % Append 120 with
    value from -120 (assumed)

```

```matlab
% Create periodic cubic spline
pp = csape(phase, magnitude, 'periodic');

% Test dataset
test_phase = [-100, -60, -20, 20, 60, 100];
test_magnitude = [8.37, 9.40, 11.39, 10.84, 8.53, 7.89];

% Evaluate spline at test phases
spline_estimate = fnval(pp, test_phase);

% Calculate residuals and RMSE
residuals = spline_estimate - test_magnitude;
rmse = sqrt(mean(residuals.^2));

% Plot results
figure;
hold on;
plot(phase, magnitude, 'ro', 'MarkerSize', 8, 'LineWidth', 2); % Original data
fnplt(pp, 'b-', 2); % Spline curve
plot(test_phase, test_magnitude, 'kx', 'MarkerSize', 10, 'LineWidth', 2); % Test data
xlabel('Phase');
ylabel('Magnitude');
legend('Training Data', 'Periodic Spline', 'Test Data', 'Location', 'NorthWest');
title(sprintf('Periodic Spline Fit (RMSE = %.4f)', rmse));
grid on;
```

Listing 2: Periodic Spline for Star Magnitude

(e) **Plot**



(f) **Result** After solving, the periodic spline achieves:

$$\text{RMSE} = \boxed{0.25} \quad \text{(agreement within observational error)}.$$