فصل ۸

چند جمله ای ها، برازش منحنی و درونیابی

MATLAB An Introduction With Applications, 6th Edition
Dr. Amos Gilat
The Ohio State University

Slide deck by
Dr. Greg Reese
Miami University

این بخش به این مطالب می پردازد

- **چند جمله ای ها** – توابعی با شکل خاص که در علوم و مهندسی کاربرد دارند
- **برازش منحنی** – یافتن صورت دقیق یک تابع مشخص که با کمترین خطا داده ها را نشان دهد
- **درونیابی** – برآورد مقادیر بین نقاط مشخص داده

چند جمله ای تابعی به این صورت است

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1\, x + a_0$$

که در آن $a_n, a_{n-1}, \dots, a_1, a_0$ اعداد حقیقی هستند و n یک عدد صحیح نامنفی است

- n درجه یا مرتبه چند جمله ای نامیده میشود
- یک عدد ثابت چند جمله ای درجه صفرم است

MATLAB چند جمله ای ها را با یک بردار سطری نشان می دهد

- اولین درایه بردار ضریب x با بیشترین توان است، یعنی a_n
- دومین درایه a_{n-1} است
- درایه nام a_1 است
- درایه n+1، a_0 است

MATLAB یک چند جمله ای درجه n را با برداری به طول $n+1$ نشان می دهد

آن بردار باید شامل تمام ضرایب چند جمله ای باشد، حتی آنهایی که صفر هستند

Polynomial	**MATLAB representation**
$8x + 5$	p = [8 5]
$2x^2 - 4x + 10$	d = [2 –4 10]
$6x^2 - 150$, MATLAB form: $6x^2 + 0x - 150$	h = [6 0 –150]
$5x^5 + 6x^2 - 7x$, MATLAB form: $5x^5 + 0x^4 + 0x^3 + 6x^2 - 7x + 0$	c = [5 0 0 6 –7 0]

تابع polyval در MATLAB مقدار چند جمله ای ها را حساب میکند

```
polyval(p,x)
```

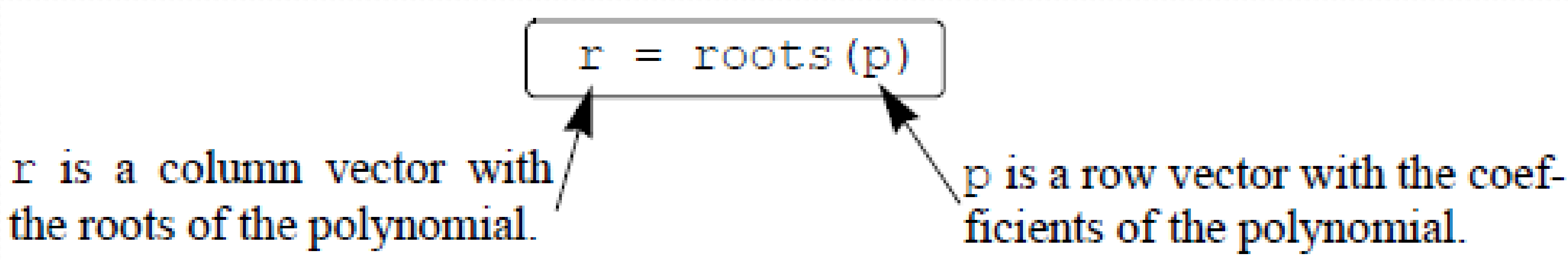
p is a vector with the coefficients of the polynomial.

x is a number, or a variable that has an assigned value, or a computable expression.

- p بردار ضرایب چند جمله ای است
- x یک عدد، بردار، یا ماتریسی از نقاطی است که مقدار چند جمله ای باید در آنها به دست بیاید
  - برای بردارها یا ماتریس ها، polyval محاسبه را درایه به درایه انجام می دهد

ریشه های یک چند جمله ای مقادیری از متغیر مستقل هستند که چند جمله ای را صفر می کنند

- یک چند جمله ای درجه n دقیقاً n ریشه دارد، با این وجود تعدادی از آنها ممکن است تکراری باشند



تابع `roots` در MATLAB تمام ریشه های یک چند جمله ای را پیدا میکند

مثال

هر سه ریشه این چند جمله ای را بیابید

$$f(x) = x^3 - \frac{23}{2}x^2 + \frac{31}{2}x - 5$$

```
>> p = [ 1 -23/2 31/2 -5 ];
>> roots( p )
ans =
   10.0000
    1.0000
    0.5000
```

اگر ریشه های یک چند جمله ای را بدانید، ضرایب آن را میتوانید با استفاده از `p = poly( r )` بیابید که در آن r بردار ریشه ها است

```
>> r = [ -1 -1 2 3 ];
>> p = poly( r )
p =
     1     -3     -3      7      6
>> roots( p ) % verify that get roots
ans = 3.0000
      2.0000
     -1.0000
     -1.0000
```

جمع:

برای جمع (منها) کردن دو چند جمله ای، بردارهای ضرایب آنها را جمع (منها) کنید

- اگر یک بردار کوتاه تر باشد، باید آنقدر صفر در ابتدای آن قرار دهید تا هم اندازه با بردار بزرگتر شود

مثال

این دو چند جمله ای را با هم جمع کنید

$$f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$$

و $f_2(x) = 3x^3 - 2x - 6$

```
>> p1 = [ 3 15 0 -10 -3 15 -40 ];
>> p2short = [ 3 0 -2 -6 ];
>> p2 = [ zeros(1,length(p1)-length(p2short)) p2short ]
p2 =
     0     0     0     3     0    -2    -6
>> p1 + p2
ans =
     3    15     0    -7    -3    13   -46
```

ضرب:

دو چند جمله ای را با استفاده از تابع داخلی conv در هم ضرب کنید، به گونه ای که:

```
c = conv( a, b )
```

که در آن a و b دو بردار ضرایب چند جمله ای هستند و c بردار ضرایب حاصلضرب است

- a و b میتوانند درجه های متفاوت داشته باشند

مثال

این دو چند جمله ای را در هم ضرب کنید

$$f_{1(x)} = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$$

و $f_{2(x)} = 3x^3 - 2x - 6$

```
>> p1 = [ 3 15 0 -10 -3 15 -40 ];
>> p2 = [ 3 0 -2 -6 ];
>> conv( p1, p2 )
ans =
9    45    -6    -78    -99    65    -54    -12    -10    240
```

که برابر است با

$$9x^9 + 45x^8 - 6x^7 - 78x^6 - 99x^5 + 65x^4 - 54x^3 - 12x^2 - 10x + 240$$

تقسیم:

دو چند جمله ای را با تابع داخلی deconv بر هم تقسیم کنید، که به این صورت است

```
[q,r] = deconv(u,v)
```

q is a vector with the coefficients of the quotient polynomial.
r is a vector with the coefficients of the remainder polynomial.

u is a vector with the coefficients of the numerator polynomial.
v is a vector with the coefficients of the denominator polynomial.

مثال

این دو چند جمله ای را بر هم تقسیم کنید

$x + 3$ بر $f_1(x) = 2x^3 + 9x^2 + 7x - 6$

```
>> u = [ 2 9 7 -6 ];
>> v = [ 1 3 ];
>> [ a b ] = deconv( u, v )
a =
     2      3      -2
b =
     0      0      0      0
```

خارج قسمت $2x^2 + 3x - 2$ است و باقی مانده ۰

مثال

این دو چند جمله ای را بر هم تقسیم کنید

$f_1(x) = 2x^6 - 13x^5 + 75x^3 + 2x^2 - 60$ بر $x^2 - 5$

```
>> w = [ 2 -13 0 75 2 0 -60 ];
>> z = [ 1 0 -5 ];
>> [ g h ] = deconv( w, z )
g =
     2    -13      10      10      52
h =
     0      0       0       0       0      50     200
```

یا $2x^4 - 13x^3 + 10x^2 + 10x + 52 + \frac{50x+200}{x^2-5}$

میتوانید مشتق یک چند جمله ای، حاصلضرب دو چند جمله ای، یا خارج قسمت دو چند جمله ای را با تابع polyder محاسبه کنید

`k = polyder(p)`	Derivative of a single polynomial. p is a vector with the coefficients of the polynomial. k is a vector with the coefficients of the polynomial that is the derivative.
`k = polyder(a,b)`	Derivative of a product of two polynomials. a and b are vectors with the coefficients of the polynomials that are multiplied. k is a vector with the coefficients of the polynomial that is the derivative of the product.
`[n d] = polyder(u,v)`	Derivative of a quotient of two polynomials. u and v are vectors with the coefficients of the numerator and denominator polynomials. n and d are vectors with the coefficients of the numerator and denominator polynomials in the quotient that is the derivative.

For example, if $f_1(x) = 3x^2 - 2x + 4$, and $f_2(x) = x^2 + 5$, the derivatives of $3x^2 - 2x + 4$, $(3x^2 - 2x + 4)(x^2 + 5)$, and $\frac{3x^2-2x+4}{x^2+5}$ can be determined by:

```
>> f1= 3 -2 4];
>> f2=[1 0 5];
```

Creating the vectors of coefficients of f_1 and f_2.

```
>> k=polyder(f1)
k =
     6    -2
```

The derivative of f_1 is: $6x - 2$.

```
>> d=polyder(f1,f2)
d =
    12    -6    38   -10
```

The derivative of $f_1 * f_2$ is: $12x^3 - 6x^2 + 38x - 10$.

```
>> [n d]=polyder(f1,f2)
n =
     2    22   -10
d =
     1     0    10     0    25
```

The derivative of $\frac{3x^2-2x+4}{x^2+5}$ is: $\frac{2x^2+22x-10}{x^4+10x^2+25}$.

برازش منحنی فرآیندی است که در آن یک تابع ریاضی به گونه ای تنظیم میشود که تا حد ممکن بر گروهی از داده ها منطبق باشد

- سپس میتوان از آن تابع به عنوان مدل ریاضی داده ها استفاده کرد

در این قسمت روشهای ساده برازش منحنی و ابزارهای مرتبط با آنها در MATLAB بررسی خواهند شد

دو راه کلی برای برازش یک چند جمله ای بر نقاط داده

1. چند جمله ای باید از هر نقطه بگذرد
2. لازم نیست چند جمله ای از هر نقطه بگذرد

چند جمله ای هایی که از تمام نقاط داده میگذرند:

اگر n نقطه داده (x_i, y_i) داشته باشیم، میتوان چند جمله ای از درجه n-1 ساخت که از همه n نقطه بگذرد. به عنوان مثال

- با داشتن دو نقطه، میتوان یک معادله خط (چند جمله درجه اول) $y = mx + b$ نوشت که از دو نقطه عبور کند
- با داشتن سه نقطه، میتوان یک معادله درجه دو (چند جمله ای درجه دو) $y = ax^2 + bx + c$ نوشت که از هر سه نقطه بگذرد

چند جمله ای هایی که الزاماً از تمام نقاط نمی گذرند:

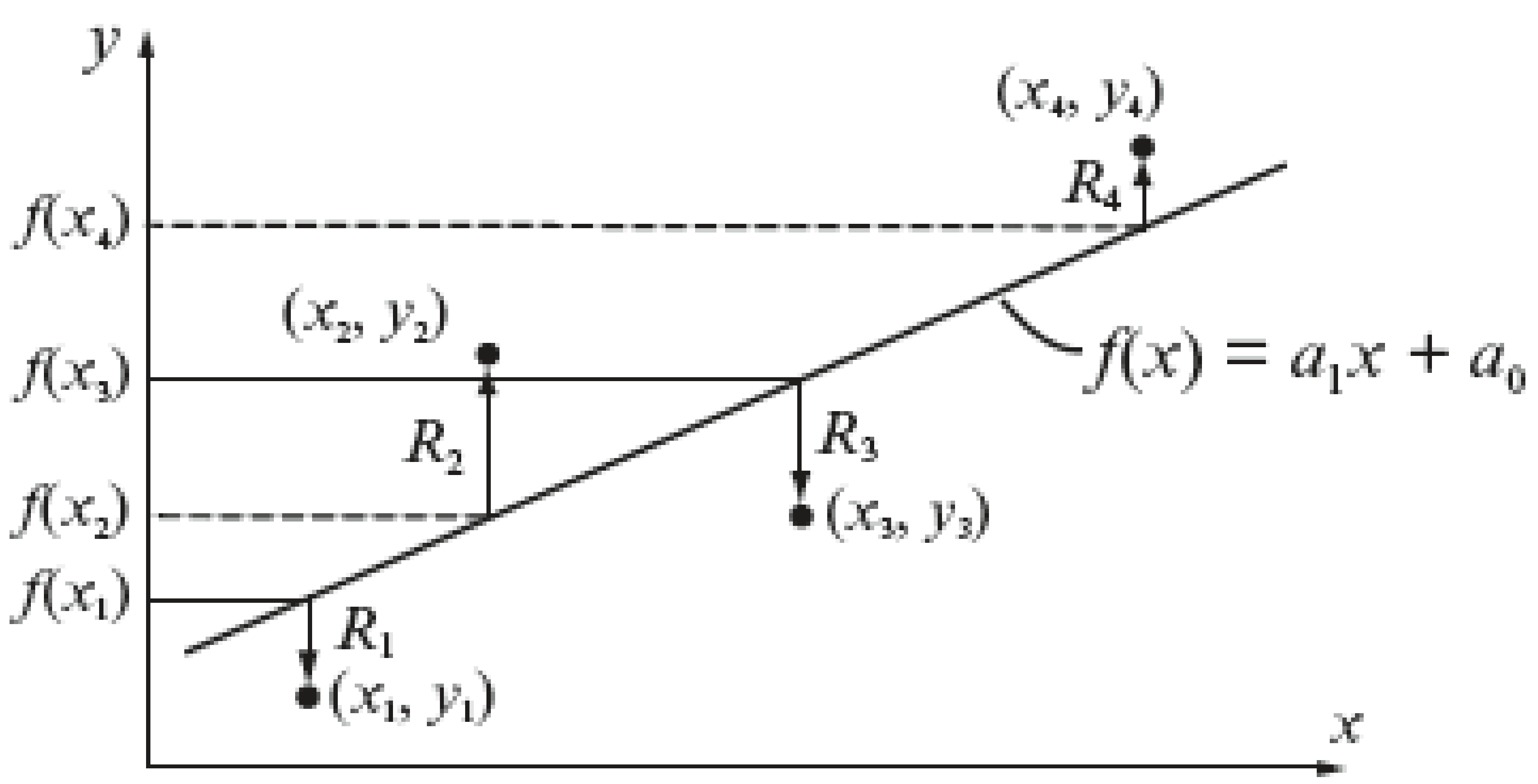
اگر n نقطه داده (x_i, y_i) داشته باشیم، اغلب میتوان یک چند جمله ای از درجه کمتر از $n-1$ ساخت که از تمام نقاط نگذرد ولی باز هم کل مجموعه را تخمین بزند

رایج ترین روش انجام این کار روش برازش حداقل مربعات است

برای انجام برازش حداقل مربعات چند جمله ای $p(x)$ از مرتبه n بر روی نقاط داده (x_i,y_i)

1. تفاضل $p(x_i)-y_i$ را در هر نقطه حساب کنید
   - این تفاضل *باقی مانده* یا *خطا* نامیده میشود
2. هر مقدار تفاضل را به توان دو برسانید
3. مربعات را با هم جمع کنید
4. مقدار $n+1$ ضریب $p(x)$ که این حاصلجمع را حداقل میکند بیابید

مثال



تابع polyfit در MATLAB برازش حداقل مربعات یک چند جمله ای بر نقاط داده را به دست می دهد

```
p = polyfit(x,y,n)
```

p is the vector of the coefficients of the polynomial that fits the data.

x is a vector with the horizontal coordinate of the data points (independent variable).
y is a vector with the vertical coordinate of the data points (dependent variable).
n is the degree of the polynomial.

زمانی که از `polyfit` روی یک مجموعه از m نقطه داده استفاده می کنید

- میتوانید از هر چند جمله ای درجه n به صورتی که $n \le m\text{-}1$ استفاده کنید
  - اگر $n = m\text{-}1$ چند جمله ای از همه نقاط خواهد گذشت

یک چند جمله ای درجه m-1 یا با درجه بالا الزاماً بهترین برازش بر تمام نقاط را ارائه نمی دهد زیرا ممکن است به طور قابل ملاحظه ای بین نقاط منحرف شود

اغلب لازم است توابعی را برازش کنیم که چند جمله ای نیستند. چهار تابع زیر زیاد استفاده میشوند و میشود آنها را با تکنیکهای ریاضی به چند جمله ای تبدیل کرد (در واقع چند جمله ای خطی). سپس میتوان از `polyfit` برای برازش آنها استفاده کرد

$y = bx^m$	(power function)
$y = be^{mx}$ or $y = b10^{mx}$	(exponential function)
$y = m\ln(x) + b$ or $y = m\log(x) + b$	(logarithmic function)
$y = \dfrac{1}{mx + b}$	(reciprocal function)

All of these functions can easily be fitted to given data with the `polyfit` function. This is done by rewriting the functions in a form that can be fitted with a linear polynomial ($n = 1$), which is

$$y = mx + b$$

The logarithmic function is already in this form, and the power, exponential, and reciprocal equations can be rewritten as:

$$\ln(y) = m\ln(x) + \ln b \qquad \text{(power function)}$$

$$\ln(y) = mx + \ln(b) \quad \text{or} \quad \log(y) = mx + \log(b) \qquad \text{(exponential function)}$$

$$\frac{1}{y} = mx + b \qquad \text{(reciprocal function)}$$

از `polyfit` برای این توابع به این صورت استفاده کنید

Function		**`polyfit` function form**
power	$y = bx^m$	`p=polyfit(log(x),log(y),1)`
exponential	$y = be^{mx}$ or	`p=polyfit(x,log(y),1)` or
	$y = b10^{mx}$	`p=polyfit(x,log10(y),1)`
logarithmic	$y = m\ln(x) + b$ or	`p=polyfit(log(x),y,1)` or
	$y = m\log(x) + b$	`p=polyfit(log10(x),y,1)`
reciprocal	$y = \dfrac{1}{mx + b}$	`p=polyfit(x,1./y,1)`

خروجی p دو عنصر دارد: p(1) ضریب m و p(2)، b است

یک راه مناسب برای فهمیدن اینکه هر کدام از توابع برازش مناسبی دارد یا نه رسم آنها با محورهای مشخص شده است. اگر داده ها خطی به نظر برسند، آن تابع را در `polyfit` استفاده کنید

x axis	*y* axis	Function
linear	linear	linear $y = mx + b$
logarithmic	logarithmic	power $y = bx^m$
linear	logarithmic	exponential $y = be^{mx}$ or $y = b10^{mx}$
logarithmic	linear	logarithmic $y = m\ln(x) + b$ or $y = m\log(x) + b$
linear	linear (plot 1/*y*)	reciprocal $y = \dfrac{1}{mx + b}$

اینها توابعی برای رسم در حالتهای مختلف محورها هستند

- plot – x خطی، y خطی
- semilogx– x لگاریتمی، y خطی
- semilogy– x خطی، y لگاریتمی
- loglog– x لگاریتمی، y لگاریتمی

نکات دیگر برای اتخاب توابع:

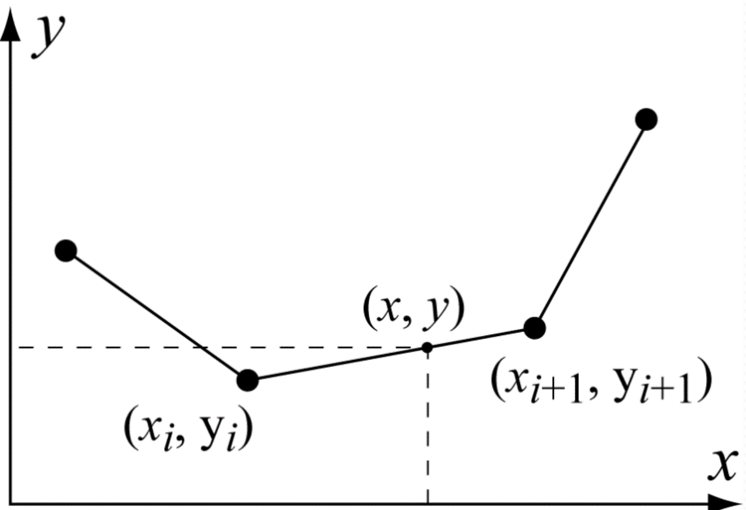
- توابع توانی
  - میتوانند از مبدأ بگذرند
  - تنها میتوانند بر داده هایی برازش شوند که همگی مثبت یا منفی باشند
- توابع لگاریتمی نمیتوانند نقاط با $x \leq 0$ را مدل کنند
- تابع توانی زمانی که $x = 0$ باشد ۰ است
- تابع نسبت نمیتواند $y = 0$ را مدل کند

درونیابی تخمین مقادیر بین دو نقطه داده است. MATLAB توانایی دورنیابی با استفاده از چند جمله ای ها یا تبدیل فوریه را دارد

- در مورد درونیابی با استفاده از تبدیل فوریه در کتاب بحث نشده است

درونیابی یک بعدی:

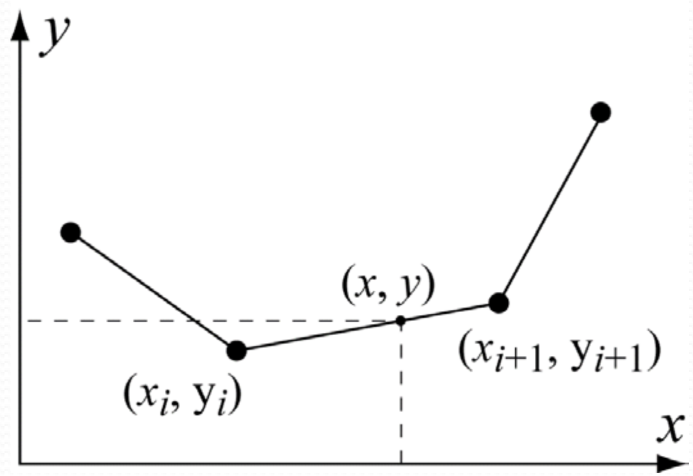
درونیابی خطی تخمین مقادیر بین دو نقطه داده به وسیله وصل کردن نقاط با یک خط مستقیم و سپس استفاده از مقادیر روی خط به عنوان مقدار تخمین زده شده است



فرض کنید نقاط داده (x_i, y_i) و (x_{i+1}, y_{i+1}) داریم و بخواهیم مقدار y را در x تخمین بزنیم، با فرض $x_i < x < x_{i+1}$. شکل (x, y) را روی یک خط راست بین دو نقطه داده نشان میدهد، یعنی نقطه درونیابی شده به صورت خطی. معادله y به این صورت است

$$y = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} x + \frac{y_i x_{i+1} - y_{i+1} x_i}{x_{i+1} - x_i}$$

در درونیابی خطی



- منحنی بین دو نقطه داده شیب ثابت دارد
- در حالت کلی، شیب منحنی در هر نقطه داده تغییر می کند

میتوان درونیابی هموارتری با استفاده از *spline*های درجه دو یا سه، که چند جمله ای هایی هستند که ضرایب آنها تنها بر اساس نقاط داده نزدیک نقطه درونیابی شده به دست می آید، انجام داد

تابع `interp1()` درونیابی یک بعدی انجام می دهد

```
yi = interp1(x,y,xi,'method')
```

yi is the interpolated value.

x is a vector with the horizontal coordinates of the input data points (independent variable).
y is a vector with the vertical coordinates of the input data points (dependent variable).
xi is the horizontal coordinate of the interpolation point (independent variable).

Method of interpolation, typed as a string (optional).

- The vector x must be monotonic (with elements in ascending or descending order).
- xi can be a scalar (interpolation of one point) or a vector (interpolation of many points). yi is a scalar or a vector with the corresponding interpolated values.

- MATLAB can do the interpolation using one of several methods that can be specified. These methods include:

`'nearest'`	returns the value of the data point that is nearest to the interpolated point.
`'linear'`	uses linear spline interpolation.
`'spline'`	uses cubic spline interpolation.
`'pchip'`	uses piecewise cubic Hermite interpolation, also called `'cubic'`

- When the `'nearest'` and the `'linear'` methods are used, the value(s) of `xi` must be within the domain of `x`. If the `'spline'` or the `'pchip'` methods are used, `xi` can have values outside the domain of `x` and the function `interp1` performs extrapolation.
- The `'spline'` method can give large errors if the input data points are nonuniform such that some points are much closer together than others.
- Specification of the method is optional. If no method is specified, the default is `'linear'`.

MATLAB ابزاری برای انجام درونیابی به صورت تعاملی دارد. به وسیله آن شما میتوانید

- برازش منحنی با استفاده از چند جمله ای های تا درجه ۱۰ انجام دهید
- برازش با استفاده از چند جمله ای های مختلف را با رسم آنها روی یک نمودار مقایسه کنید
- خطاهای باقی مانده را رسم و مقایسه کنید
- مقادیر درونیابی شده را محاسبه کنید

برای فعال کردن این ابزار

1. نقاط داده را رسم کنید
2. Tools|Basic Fitting را انتخاب کنید
3. دکمه فلش سمت راست را دو بار کلیک کنید تا پنجره به شکل تصویر ۳-۸ در آید

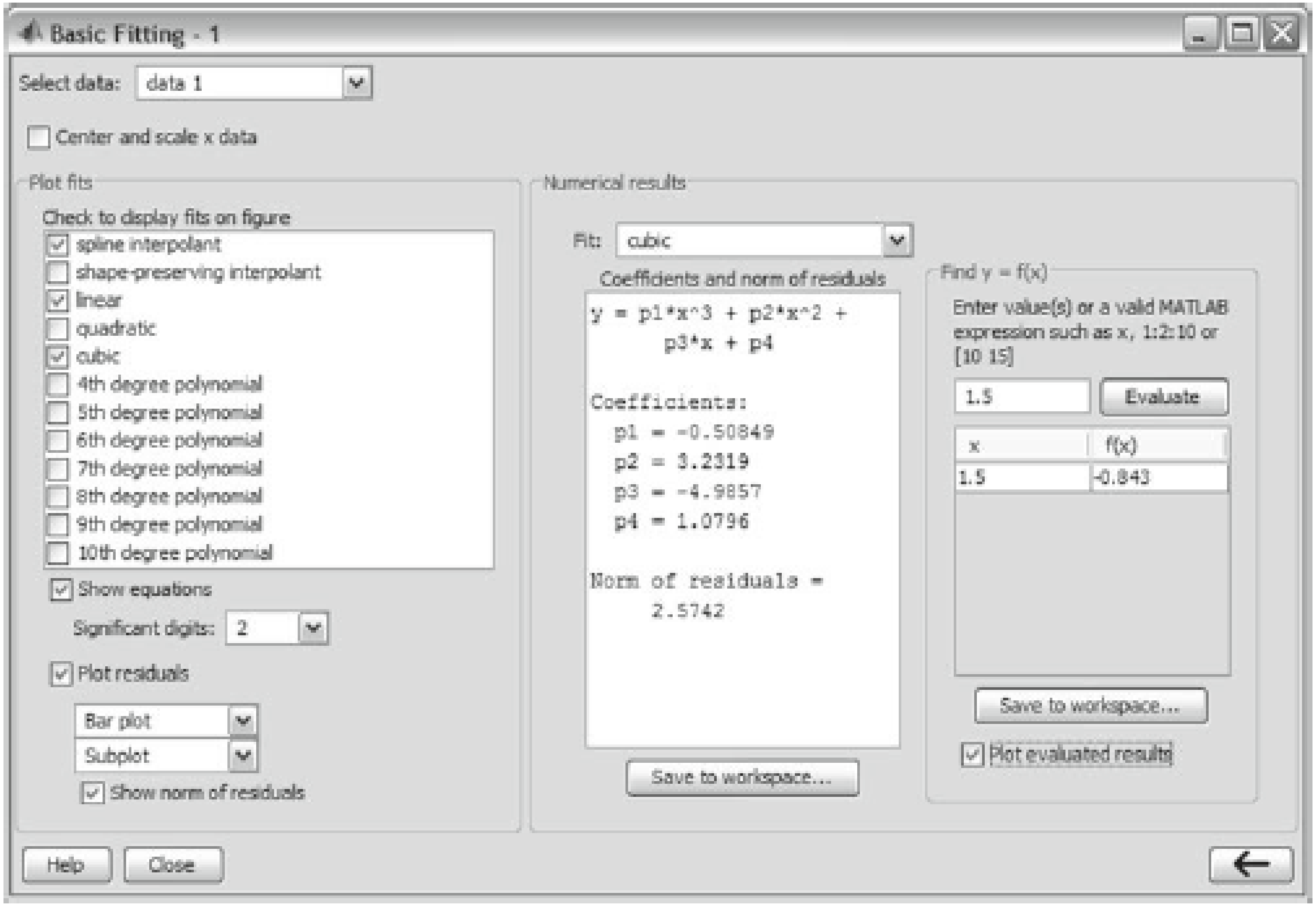


Figure 8-3: The Basic Fitting Window.

قسمتهای مختلف پنجره Basic Fitting

- **Select data**
  - اگر بیش از یک سری داده رسم شده باشد، امکان انتخاب این که روی کدامیک از آنها کار کنید را می دهد
    - در یک زمان تنها روی یک سری داده میتوانید کار کنید، ولی میتوانید چند برازش به صورت همزمان روی آن داده ها انجام دهید
- **Center and scale x data**
  - داده ها را طوری تنظیم میکند که میانگین آنها صفر و انحراف معیارشان یک باشد
- **Check to display fits on figure**
  - انتخاب نوع برازشی که میخواهید انجام شده و نمایش داده شود

- **Show equations**
  - اگر انتخاب شود، معادله برازشهای انتخاب شده در جعبه بالای این قسمت نمایش داده می شود
- **Plot residuals**
  - ترسیم شدن یا نشدن خطای باقیمانده درونیابی و نوع ترسیم آن را مشخص میکند
- **Show norm of residuals**
  - مشخص میکند که نُرم خطاها نمایش داده شود یا نه. نُرم یعنی
    - معیاری برای کیفیت برازش
    - نُرم کوچکتر یعنی برازش بهتر

- **Fit**
  - انتخاب اینکه جزئیات کدام برازش بررسی شود
- **Coefficients and norm of residuals**
  - مقادیر عددی ضرایب معادله برازش و مقدار نُرم را نمایش میدهد
- **Find y = *f*(*x*)**
  - این امکان را می دهد که مقادیر برازش شده برای مقادیر وارد شده متغیر مستقل بررسی شود

شکل ۴-۸ مثالی از پنجره ترسیمات همراه با تغییرات اعمال شده در رابط Basic Fitting است

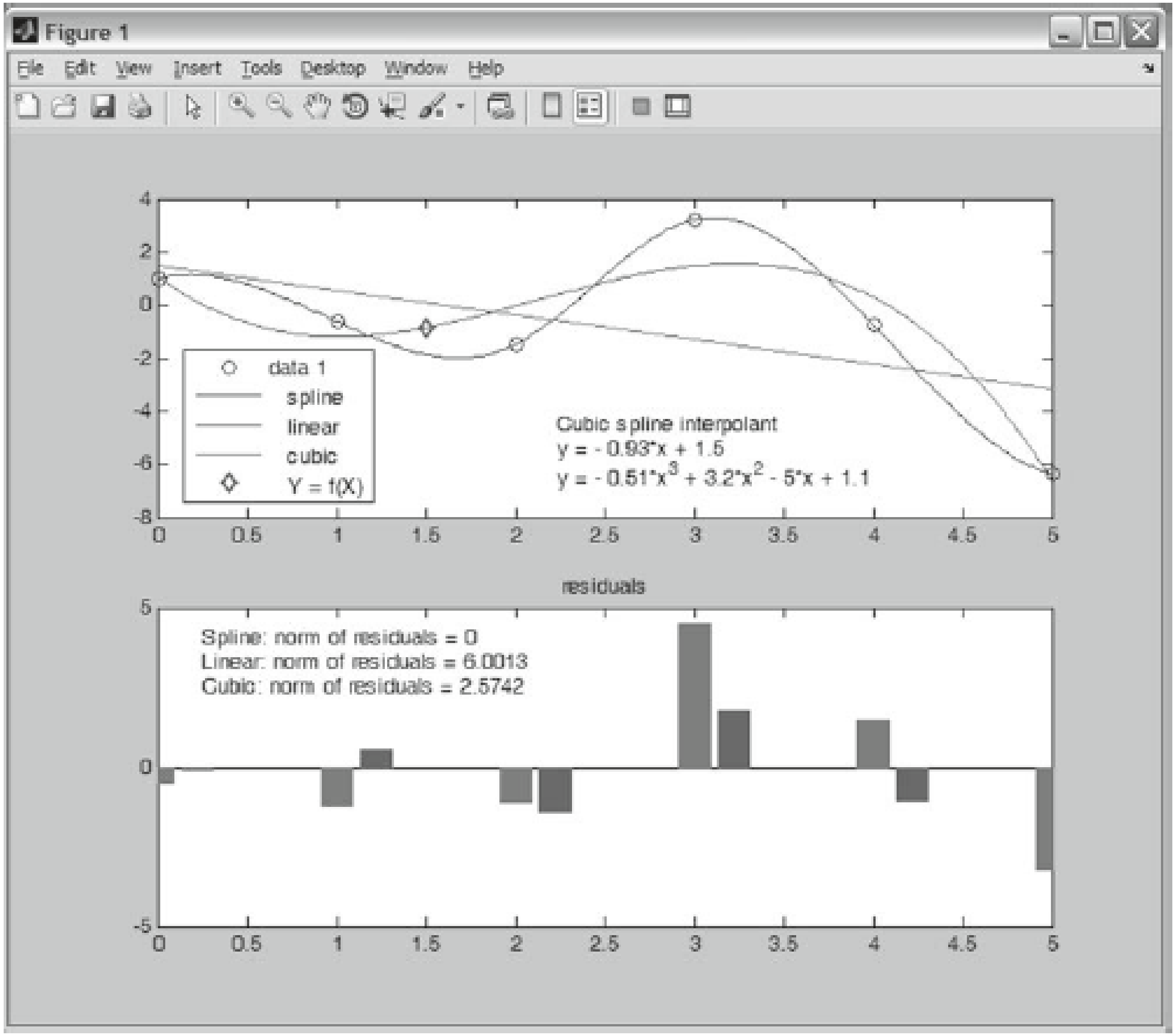


Figure 8-4: A Figure Window modified by the Basic Fitting Interface.

شماره تمرین های منتخب

- ۴
- ۶
- ۸
- ۱۲
- ۱۵
- ۱۷
- ۲۳

- ۲۵
- ۲۶
- ۲۷
- ۳۰
- ۳۱
- ۳۲
- ۳۳