Fermi – კოსმოსური გამა ტელესკოპი

4 ინტერპოლირება და რეგრესია

პრობლემა: ბირთვული სინთეზი

ექსპერიმენტული მონაცემების დამუშავებას მნიშვნელოვანი როლი აკისრია ახალი მეცნიერული პრინციპების და თეორიის შემუშავებაში. ბირთვული სინთეზის პრობლემა მოითხოვს ღრმა ცოდნას ისეთ სფეროში როგორიცაა ბირთვული მდგრადობა და ბირთვული დაშლა. ბირთვული ენერგიის პრობლემები ასოცირდება გამა გამოსხივებასთან. იმისათვის რომ გაიზომოს კოსმოსიდან მოსული გამა გამოსხივება, შეიქმნა გამა გამოსხივების კოსმოსური ობსერვატორია, რომელიც კოსმოსში გაუშვეს 1991 წელს. იგი ორბიტაზე გაუშვა შატლმა ATLANTIS. იგი შეცვალა უფრო მძლავრმა აპარატურამ, რომელიც გაშვებული იქნა 2008 წლის 11 ივნისს. დღეს იგი ცნობილია როგორც კოსმოსური გამა ტელესკოპი **Fermi** **(http://fermi.gsfc.nasa.gov/)**.

შესავალი

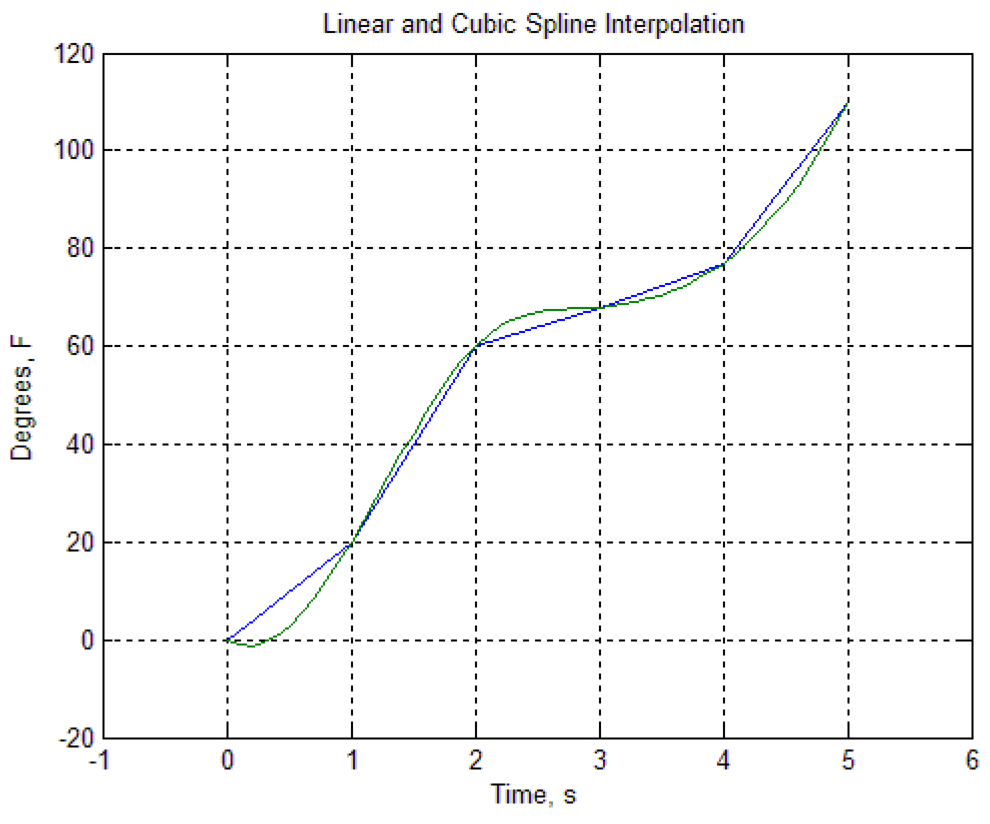
შესავალი

ვთქვათ გვაქვს ექსპერიმენტის ან რაიმე ფიზიკურ პროცესზე დაკვირვების შედეგად მიღებული მონაცემები. ეს მონაცემები სზოგადოდ შეიძლება განვიხილოთ როგორც *y=f(x)* შესაბამისი წერტილიების კოორდინატები. გვინდა გამოვთვალოთ ამ ფუნქციის მნიშვნელობები x ისეთი მნიშვნელობებისათვის, რომელთა შესახებ ინფორმაცია არ გვაქვს. მაგალითად, გვაქვს მონაცემები (a, f(a)) და (c, f(c)). თუ გვსურს შევაფასოთ ფუნქციის მნიშვნელობა b წერტილში f(b), სადაც $a<b<c$, უნდა დავუშვათ, რომ f(a) და f(c) შეერთებულია წრფით და წრფივი ინტერპოლირების საშუალებით გამოვითვლით f(b). თუ დავუშვებთ, რომ ეს წერტილები შეერთებულია კუბური პოლინომის საშუალებით, f(b) გამოსათვლელად გამოვიყენებთ კუბური ინტერპოლირების მეთოდს. ზოგჯერ გვჭირდება მონაცემების მიხედვით

დასმულ წერტილებზე მრუდი ისე გავატაროთ, რომ რაც შეიძლება კარგად მოერგოს დასმულ წერტილებს, რამდენადაც ეს შესაძლებელია, ახლოს გაიაროს ყველა წერტილთან. ამ ამოცანას ემსახურება უმცირეს კვადრატთა მეთოდი, რომელიც ისე შეარჩევს მრუდს, რომ მანძილი მოცემულ წერტილსა და ფუნქციით გამოთვლილ შესაბამის მნიშვნელობას შორის უმცირესი იყოს ყველა წერტილისათვის. ახლა განვიხილავთ ინტერპოლაციისა და მრუდის გამოთვლის (curve fitting) მაგალითებს.

4.1 ინტერპოლირება

განვიხილავთ ინტერპოლირების ორ ტიპს: წრფივი და კუბური (spline) ინტერპოლირება. ორივე შემთხვევაში ვუშვებთ, რომ გვაქვს მონაცემები, რომელიც წარმოდგენილია წერტილის კოორდინატების სახით (x, f(x)). გვინდა შევაფასოთ სიდიდე f(b), რომლის მნიშვნელობა არ გვაქვს საწყის მონაცემებში, მაგრამ სრულდება პირობა $a<b<c$ და ვიცით f(a) და f(c) მნიშვნელობები. ნახ. 9.1 ნაჩვენებია 6 წერტილი, რომლებიც შეერთებულია მონაკვეთებით (სწორი ხაზით) და მესამე რიგის – კუბური პოლინომური წირით. რასაკვირველია ფუნქციის შუალედური მნიშვნელობები დამოკიდებულია იმაზე, თუ ინტერპოლირების რომელ მეთოდს ავირჩევთ მათ განსასაზღვრავად.



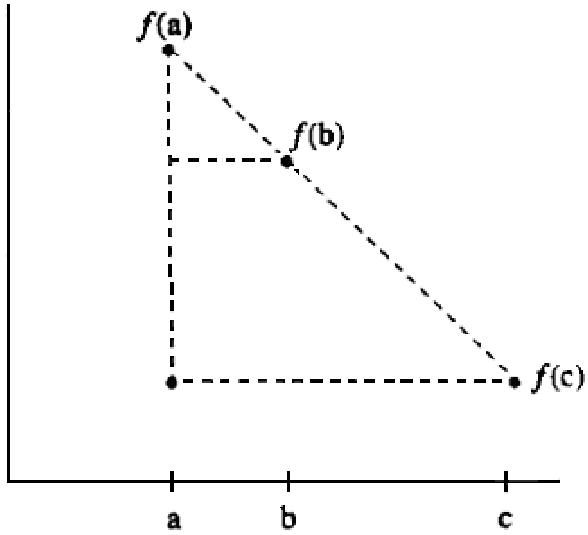
ნახ. 9.1 წრფივი და კუბური ინტერპოლირება

9.1.1 წრფივი ინტერპოლირება

ძირითადი მეთოდი ორ მოცემულ წერტილს შორის მოთავსებული წერტილის კოორდინატის შესაფასებლად წრფივი ინტერპოლირების მეთოდია. ნახ. 9.2–ზე გვაქვს ორი წერტილი. თუ

დავუშვებთ, რომ ფუნქცია ამ ორ წერტილს შორის შეიძლება შეფასდეს როგორც წრფე, რომელიც ამ წერტილებზე გადის, ფუნქციის მნიშვნელობა მათ შორის მოთავსებული ნებისმიერი წერტილისათვის შეგვიძლია გამოვთვალოთ შემდეგი ფორმულით:

$$\frac{f(a)-f(b)}{b-a}=\frac{f(a)-f(c)}{c-a}$$



ნახ. 9.2 მსგავსი სამკუთხედები

$$f(b)=f(a)+\frac{b-a}{c-a}(f(c)-f(a))$$

ვთქვათ ჩვენს ხელთაა ექპერიმენტული მონაცემები. შედარებით მარტივი საქმეა ინტერპოლირების საშუალებით მივიღოთ შუალედური მნიშვნელობები. ინტერპოლირების გამოყენება შესაძლებელია მას შემდეგ, რაც ჩვენს ხელთ არსებულ მონაცემებს შორის ვიპოვით იმ ორ წერტილს, რომელთა შორისაც ჩვენთვის საჭირო წერტილი მდებარეობს. შესაძლებელია ასეთი წერტილები ვერც ვიპოვოთ. MATLAB საშუალებით წრფივი ინტერპოლირება ხორციელდება შემდეგი ფუნქციების საშუალებით. interp1 და interp2.

interp1(x,y,x1)– ფინქცია ახორციელებს ერთგანზომილებიან წრფივ ინტერპოლირებას მონაცემთა ცხრილის საფუძველზე. ვთქვათ გვაქვს მონაცემები x ვექტორის სახით და მისი შესაბამისი მონაცემები y ვექტორის სახით და x1 - ის მნიშვნელობა, რომლისთვისაც გვსურს შესაბამისი y1 სიდიდის ინტერპოლირება. ფუნქცია მოძებნის x ვექტორში იმ მნიშვნელობებს, რომელთა შორისაც x1 სიდიდეა მოთავსებული და წრფივი ინტერპოლირების ფორმულის საშუალებით გამოითვლის y1 სათანადო მნიშვნელობას. შევნიშნავთ, რომ x ვექტორში მონაცემები ზრდის მიხედვით უნდა იყოს დალაგებული, თუ ეს ასე არ არის, MATLAB მიგვითითებს შეცდომაზე. ამასთან ჩვენს მიერ მითითებული სიდიდე უნდა თავსდებოდეს მონაცემთა პირველ და უკანასკნელ მნიშვნელობას შორის, განსხვავებულ შემთხვევაში MATLAB მოგვცემს მნიშვნელობას NAN.

ამ ფუნქციის საილუსტრაციოდ განვიხილოთ ტემპერატურის მონაცემები, რომელიც გაზომილია ცილინდრის ხუფში ავტომობილის ახალი ძრავის გამოცდისას.

დრო, წმ	ტემპერატურა, °F
0.0	0.0

1.0	20.0
2.0	60.0
3.0	68.0
4.0	77.0
5.0	110.0

პირველ რიგში ამ ინფორმაციას ჩავწერთ მატრიცის სახით, სადაც x ვექტორი იქნება დრო, y – ტემპერატურა.

```
x = [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]';
y = [0.0, 20.0, 60.0, 68.0, 77.0, 110.0]';
```

ამის შემდეგ შეგვიძლია ვისარგებლოთ **interp1** ფუნქციით. დროის ისეთი მნიშვნელობებისათვის, რომელიც თავსდება ინტერვალში [0 – 5] წამი. მაგალითად, განვიხილოთ შემდეგი ბრძანებები:

```
y1 = interp1(x,y, 2.6);
y2 = interp1(x,y, 4.9);
```

მივიღებთ y1 = 64.8, y2 = 106.7.

თუ y მატრიცაა და xi ვექტორი, ინტერპოლირება შესრულდება y ყველა სვეტისათვის და მივიღებთ y1 მატრიცას, რომლის სტრიქონების რაოდენობა xi ვექტორის სიგრძის ტოლი იქნება, ხოლო სვეტების რაოდენობა - y ვექტორის სვეტების რაოდენობის. მაგალითად, დავუშვათ ტემპერატურა გავზომეთ ცილინდრის ხუფის სამ სხვადასხვა წერტილში და მივიღეთ შემდეგი ცხრილი:

დრო, წმ	ტემპ. 1, °F	ტემპ. 2, °F	ტემპ. 3, °F
0.0	0.0	0.0	0.0
1.0	20.0	25.0	52.0
2.0	60.0	62.0	90.0
3.0	68.0	67.0	91.0
4.0	77.0	82.0	93.0
5.0	110.0	103.0	96.0

ავაგოთ სამსვეტიანი მატრიცა y ტემპერატურისათვის სამ სხვადასხვა წერტილში:

```
y(: , 1) = [0.0, 20.0, 60.0, 68.0, 77.0, 110.0]';
y(: , 2) = [0.0, 25.0, 62.0, 67.0, 82.0, 103.0]';
y(: , 3) = [0.0, 52.0, 90.0, 91.0, 93.0,  96.0]';
```

თუ გვინდა მივიღოთ ინტერპოლირებული მნიშვნელობა ცილინდრის ხუფის სამივე წერტილში დროის მომენტში 2.6 წამი, MATLAB უნდა მივცეთ ბრძანება:

```
temps = interp1(x,y, 2.6)
```

მივიღებთ:

```
temps =
```

```
    64.8000   65.0000   90.6000
```

interp2 – ახორციელებს წერტილის ინტერპოლაციას ორგანზომილებიან ზედაპირზე z=f(x,y)

9.1.2 კუბური ინტერპოლირება - კუბური სპლაინი

კუბური სპლაინი (წირი) არის გლუვი წირი, რომელიც გაივლის მოცემულ წერტილებზე ისე, რომ წირის მონაკვეთი ყოველ ორ მომდევნო წერტილს შორის მესამე ხარისხის პოლინომს წარმოადგენს. 4.1 ნაჩვენებია კუბური წირი, რომელიც 6 წერტილს აერთებს. იმისათვის, რომ ეს წირი მივიღოთ 5 სხვადასხვა კუბური განტოლება უნდა ამოვხსნათ.

MATLAB-ში კუბური წირი გამოითვლება ფუნქციით **spline**. მისი პირველი ორი არგუმენტი x, y წარმოადგენს ვექტორებს, იმ წერტილების კოორდინატებს, რომლებიც წირმა უნდა შეაერთოს. მესამე არგუმენტი შეიცავს x კოორდინატებს, რომელთა შესაბამისი y უნდა გამოვთვალოთ მესამე რიგის ინტერპოლაციით. x ვექტორის მნიშვნელობები ზრდის მიხედვით უნდა იყოს დალაგებული. **spline** ფუნქციის საილუსტრაციოდ დავუბრუნდეთ ცილინდრის ხუფის ტემპერატურათა მონაცემებს. დროის მნიშვნელობისათვის 2.6 წამი გამოვთვალოთ ტემპერატურა არა წრფივი, არამედ კუბური ინტერპოლირების საშუალებით:

```
x = [0, 1, 2, 3, 4, 5];
y = [0.0, 20.0, 60.0, 68.0, 77.0, 110.0];
temp1 = spline(x,y,2.6)
```

მივიღებთ:

```
temp1 =

   67.3013
```

თუ გვსურს გამოვიყენოთ კუბური ინტერპოლირება ტემპერატურის გამოსათვლელად დროის ორ სხვადასხვა მომენტში, MATLAB ბრძანება ასეთი იქნება:

```
temp2 = spline(x,y,[2.6,4.9])
```

მივიღებთ:

```
temp2 =

    0.6730  1.0520
```

თუ გვინდა მივიღოთ კუბური წირი x მნიშვნელობათა ინტერვალში მთლიანად, უნდა შევქმნათ x ვექტორი საჭირო გარჩევით და ავიღოთ იგი როგორც მესამე პარამეტრი **spline** ფუნქციისათვის.

მაგალითად, ნახ. 9.1 ნაჩვენები კუბური წირი მიიღება შემდეგი ბრძანებების შედეგად:

```
x=[0.0, 1.0, 2.0,3.0,4.0,5.0];
y=[0, 20, 60, 68, 77, 110];
newx=0:0.1:5;
newy=spline(x,y,newx);
plot(x,y,newx,newy),...
axis([-1 6 -20 120]),...
title('Linear and Cubic Spline Interpolation'),...
xlabel('Time, s'),...
ylabel('Degrees, F'),...
grid
```

სწორხაზოვანი მონაკვეთები შეესაბამება ბრძანებას **plot**(x,y) და წარმოადგენს წრფივ ინტერპოლირებას, ხოლო გლუვი წირი - **plot**(newx,newy) ბრძანების შედეგია და კუბური ინტერპოლირების შედეგს წარმოადგენს.

სავარჯიშო

დავუშვათ გვაქვს მონაცემთა შემდეგი მწკრივი:

დრო [წმ]	ტემპერატურა, [°F]
0.0	72.5
0.5	78.1
1.0	86.4
1.5	92.3
2.0	110.6
2.5	11.5
3.0	109.3
3.5	110.2
4.0	110.5
4.5	109.9
5.0	110.2

1. შეაერთეთ წერტილები წრფის მონაკვეთებით და კუბური წირით
2. MATLAB საშუალებით გამოითვალეთ ტემპერატურა წრფივი ინტერპოლირებით დროის შემდეგი მნიშვნელობებისათვის

   0.3, 1.25, 2.36, 4.48
3. MATLAB საშუალებით გამოითვალეთ ტემპერატურა კუბური ინტერპოლირებით დროის შემდეგი მნიშვნელობებისათვის

   0.3, 1.25, 2.36, 4.48
4. გამოიყენეთ წრფივი ინტერპოლირების მეთოდი MATLAB –ში და გამოითვალეთ დროის მნიშვნელობები, რომელიც შეესაბამება ტემპერატურათა შემდეგ მნიშვნელობებს:

   81, 96, 100, 106
5. გამოიყენეთ კუბური ინტერპოლირების მეთოდი MATLAB –ში და გამოითვალეთ დროის მნიშვნელობები, რომელიც შეესაბამება ტემპერატურათა შემდეგ მნიშვნელობებს:

   81, 96, 100, 106

პრობლემა: რობოტის მკლავის მანიპულატორი

იმისათვის, რომ რობოტმა რაიმე მოქმედება შეასრულოს, მას უნდა გააჩნდეს კონტროლის სისტემით მართვადი მანიპულატორები, რომელიც შეიმუშავებს მოძრაობის მარშრუტს, რათა იგი გადაადგილდეს ერთი მდებარეობიდან მეორეზე. მარშრუტი გლუვი უნდა იყოს, რათა თავიდან იქნას აცილებული მკვეთრი მოძრაობები, რომლის დროსაც შესაძლოა დაზიანდეს ობიექტი ან თვითონ რობოტის მკლავი. ამიტომ მარშრუტი რობოტის მკლავისათვის განისაზღვრება წერტილების ერთობლიობით, რომელთა გასწვრივაც მკლავმა უნდა იმოძრაოს. იმისათვის, რომ გადაადგილება გლუვი მრუდის გასწვრივ მოხდეს, ინტერპოლირებას მიმართავენ. განვიხილავთ მსგავს პრობლემას, იმ დაშვებით, რომ რობოტის მანიპულატორი მკლავი მოძრაობს სიბრტყეზე, თუმცა საზოგადოდ მოძრაობა სივრცეში ხორციელდება.

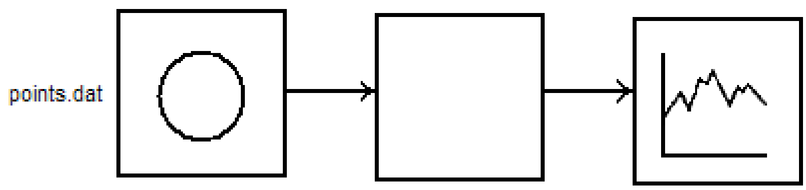
ალგორითმის შემუშავების, ან პრობლემის ამოხსნის მნიშვნელოვანი პირობაა ყურადღებით განვიხილოთ ხომ არ არის რაიმე განსაკუთრებული შემთხვევა, რომელიც მხედველობიდან არ უნდა გამოგვრჩეს. ამ ამოცანაში ვუშვებთ, რომ იმ წერტილების კოორდინატები, რომელთა გასწვრივ რობოტის მკლავმა უნდა იმოძრაოს, მოთავსებულია მონაცემთა ფაილში და ისეთი რიგით არის დალაგებული, რომ მკლავმა უნდა მიაღწიოს გარკვეულ მდებარეობას, აიღოს იქ მდებარე ობიექტი, შემდეგ გადაადგილდეს წერტილში, სადაც დატოვებს აღებულ ობიექტს და კვლავ დაუბრუნდეს საწყის მდებარეობას. ასევე ვუშვებთ, რომ მონაცემებში ჩართულია შუალედური წერტილები, რომელიც საშუალებას აძლევს მას თავი აარიდოს შემხვედრ დაბრკოლებას და ხელი არ შეუშალოს სენსორებს, რომლებიც მონაცემებს აგროვებენ. ყოველ წერტილს გააჩნია სამი კოორდინატი: x და y საწყისი მდებარეების მიმართ და მესამე, რომელიც კოდირებულია შემდეგნაირად:

კოდი	ინტერპრეტაცია
0	საწყისი მდებარეობა
1	შუალედური მდებარეობა
2	ობიექტის აღების მდებარეობა
3	ობიექტის დადების მდებარეობა

ამოცანის დასმა

უნდა გავატაროთ კუბური მრუდი მოცემულ წერტილებზე, რომელიც უნდა გაიაროს რობოტის მკლავის მანიპულატორმა თავის გზაზე, საწყისი მდებარეობიდან ობიქტის აღების, მისი დაბინავების გავლით კვლავ საწყის მდებარეობამდე.

1. INPUT/OUTPUT დიაგრამა



ნახ. 9.3 I/O დიაგრამა

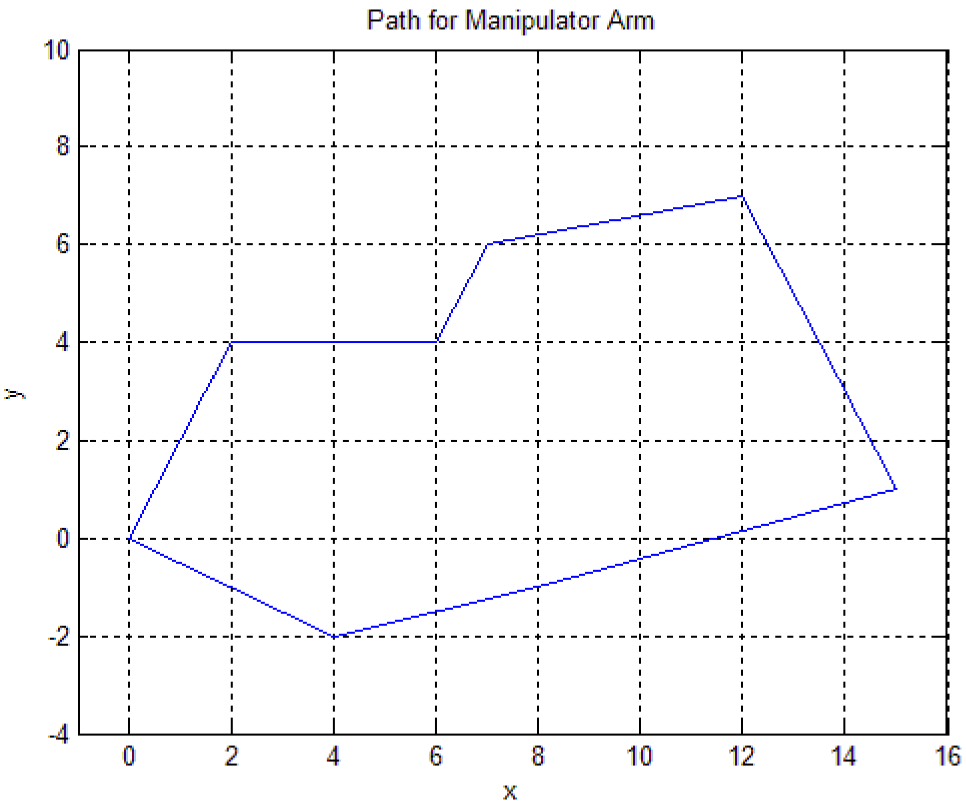
2. სახელდახელო ამოხსნა

განვსაზღვროთ, ხომ არ არის რაიმე განსაკუთრებული შემთხვევა, რომელიც მხედველობაში უნდა მივიღოთ ალგორითმის შედგენისას. განვიხილოთ მონაცემთა ფაილი, რომელიც შეიცავს წერტილებს შემდეგი მონაცემებით:

x	y	კოდი	კოდის ინტერპრეტაცია
0	0	0	საწყისი მდებარეობა
2	4	1	შუალედური მდებარეობა
6	4	1	შუალედური მდებარეობა
7	6	2	ობიექტის აღების მდებარეობა
12	7	1	შუალედური მდებარეობა
15	1	3	ობიექტის დაბინავების მდებარეობა

8	-1	1	შუალედური მდებარეობა
4	-2	1	შუალედური მდებარეობა
0	0	0	საწყისი მდებარეობა

სწორი ხაზებით შეერთებული ეს წერტილები ნაჩვენებია ნახ. 9.4 – ზე.
დავყოთ მარშრუტი სამ ნაწილად: საწყისი მდებარეობიდან ობიექტის აღების მდებარეობამდე, ობიექტის აღების მდებარეობიდან მისი დაბინავების მდებარეობამდე და აქედან უკან საწყის მდებარეობამდე. ასე ორი მიზეზის გამო მოვიქეცით: პირველი, მანიპულატორი უნდა შეჩერდეს ყოველი მათგანის დსრულებისას, ასე, რომ ეს მონაკვეთები რეალურად გამოყოფილი მონაკვეთებია. მეორე – ფუნქცია **spline** მოითხოვს, რომ x კოორდინატები ზრდის მიხედვით იყოს დალაგებული. ამგვარად გზის ყოველ მონაკვეთში კოორდინატები ზრდის მიხედვით უნდა იყოს დალაგებული. დავუშვებთ, რომ ეს მომენტი გათვალიწინებულია მონაცემების შედგენისას



ნახ. 9.4 სწორი ხაზებით შეერთებული წერტილები მონაცემთა ფაილიდან რობოტის მანიპულატორისათვის

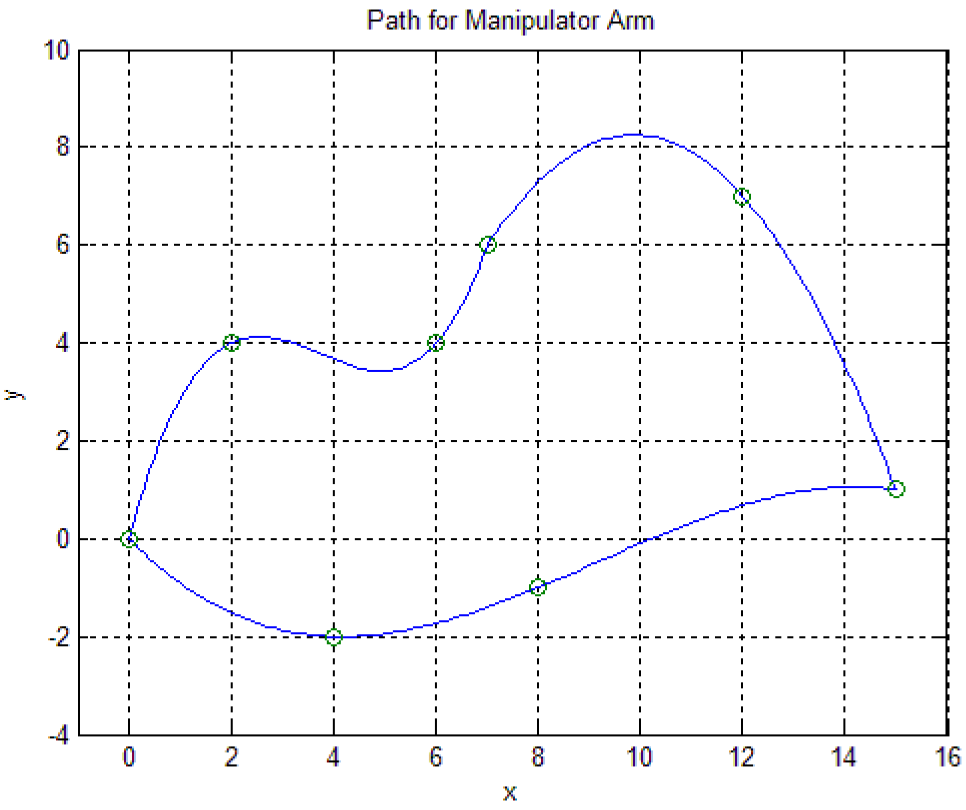
3. MATLAB ამოხსნა

მინაცემთა სამ ნაწილად გაყოფა მნიშვნელოვანი საფეხურია ალგორითმის შედგენაში. ნაკლებად მნიშვნელოვანია ის თუ რამდენ წერტილს შევარჩევთ კუბური წირისათვის. რადგან კოორდინატი შეიძლება იყოს ძალიან მცირე და ძალიან დიდი სიდიდე, გადავწყვიტეთ

ვიპოვოთ უმცირესი მანძილი წერტილებს შორის და მისი მეათედი გამოვიყენოთ როგორც ნაზრდი ინტერპოლირებისათვის საჭირო წერტილების x კოორდინატების შერჩევისათვის. ამ მოსაზრებით ყოველ ორ მოცემულ წერტილს შორის მინიმუმ 10 ინტერპოლირებული წერტილი მაინც გვექნება.

```
%     This program reads the data file contaning the
%     points for a path for a manipulator arm to go to
%     a location to grasp an object, then move to
%     another location to release the object, and
%     then move back to the start posision

load points.dat;
x=points(:,1);
y=points(:,2);
code=points(:,3);
%
%     Generate the three separate paths
%
grasp=find(code==2);
release=find(code==3);
lenx=length(x);
x1=x(1:grasp);            y1=y(1:grasp);
x2=x(grasp:release);      y2=y(grasp:release);
x3=x(release:lenx);       y3=y(release:lenx);
%
%     Compute time sequences
%
incr=min(abs(x(2:lenx)-x(1:lenx-1)))/10;
t1=x(1):incr*sign(x(grasp)-x(1)):x(grasp);
t2=x(grasp):incr*sign(x(release)-x(grasp)):x(release);
t3=x(release):incr*sign(x(lenx)-x(release)):x(lenx);
%
%     Compute spline path
%
s1=spline(x1,y1,t1);
s2=spline(x2,y2,t2);
s3=spline(x3,y3,t3);
%
%     Plot spline path
%
axis([-1, 16, -4, 10])
plot([t1 t2 t3],[s1 s2 s3], …
[x1' x2' x3'],[y1' y2' y3'],'o'),...
title('Path for Manipulator Arm'),...
xlabel('x'), ylabel('y'), grid
```



ნახ. 9.5 კუბური ინტერპოლირების შედეგად მიღებული წირით შეერთებული წერტილები

4. **შემოწმება**

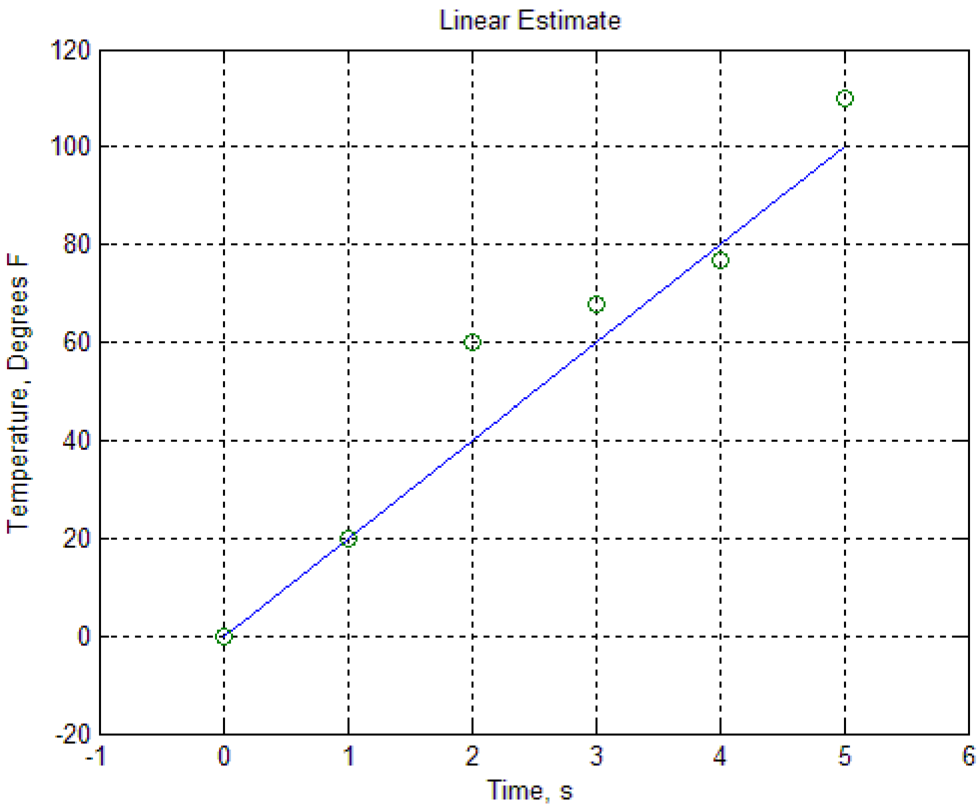
შევამოწმოთ პროგრამა მონაცემთა ზემოთგანხილული ფაილის მაგალითზე. მივიღებთ ნახ. 9.5.

4.2 უმცირეს კვადრატთა მეთოდი

დავუშვათ გვაქვს ექსპერიმენტული მონაცემები, ავაგეთ გრაფიკი და ცხადად ჩანს, რომ ისინი ძირითადად წრფის გასწვრივ დალაგდნენ. თუმცა თუ გავავლებთ წრფეს, მხოლოდ რამდენიმე მათგანი გადაიკვეთება. უმცირეს კვადრატთა მეთოდი გამოიყენება ისეთი წრფის მოსაძებნად, რომელიც ყველა წერტილს ყველაზე ახლოს ჩაუვლის წერტილიდან წრფემდე მანძილის კვადრატის მინიმალურის მოძებნის საფუძველზე. თუმცა ეს წრფე განიხილება, როგორც ოპტიმალური 'მიახლოება' (best fitting), შეიძლება ისეც მოხდეს, რომ არცერთი წერტილი არ მოხვდეს უშუალოდ მასზე. (უნდა აღინიშნოს, რომ ეს მეთოდი ბევრად განსხვავდება ინტერპოლირებისაგან, რადგან წირი, რომლითაც ვსარგებლობთ წრფივი და კუბური ინტერპოლირებისას, გაივლის ყველა წერტილზე.) ამ თავში განვიხილავთ ჯერ წრფის, შემდეგ კი პოლინომური წირის გავლებას ექსპერიმენტული მონაცემების შესაბამის წერტილებზე უმცირე კვადრატთა მეთოდის გამოყენებით.

9.1.3 წრფივი რეგრესია

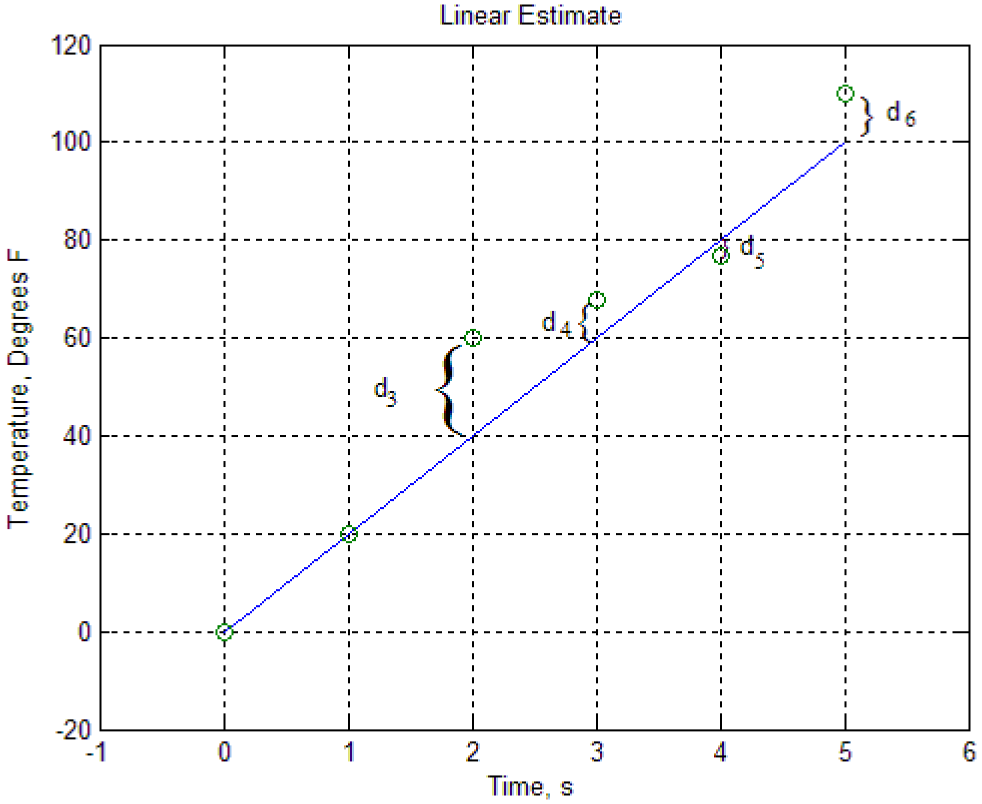
წრფივი რეგრესია ისეთ პროცესს ეწოდება, რომელიც განსაზღვრავს ისეთ წრფივ განტოლებას, რომელიც ოპტიმალურად გაივლის არსებული მონაცემების შესაბამის წერტილებზე, წრფემდე მანძილების კვადრატების ჯამის მინიმიზაციის საფუძველზე. კარგად რომ გავიგოთ ეს პროცესი დავუბრუნდეთ ტემპერატურის მონაცემებს, რომლის ანათვალიც აღებულია ახალი ძრავის ცილინდრის ხუფში. თუ ავაგებთ ამ მონაცემებს, ვნახავთ, რომ ისინი წრფის გასწვრივ განლაგდებიან. წრფე y=20x ერთი შეხედვით კარგად ასახავს წერტილთა მდებარეობას. ნახ. 9.6 ნაჩვენებია ეს წერტილები და გავლებული წრფე. შემდეგი ბრძანებები იძლევა ნახ. 9.7.



ნახ. 9.6 წრფივი შეფასება

```
x = [0, 1, 2, 3, 4, 5 ];
y = [0, 20, 60, 68, 77, 110 ];
y1 = 20*x;
plot(x,y1,x,y,'o'), title('Linear Estimate'),...
xlabel('Time, s'), ylabel('Temperature, Degrees F'), grid
axis([-1,6,-20,120])
```

ვნახოთ რამდენად კარგად ასახავს ეს წრფივი შეფასება ჩვენს მონაცემებს. პირველ რიგში გავზომოთ მანძილები წერტილებიდან წრფემდე. ეს მანძილები ნაჩვენებია ნახ. 9.7-ზე



ნახ. 9.7 მანძილი წერტილსა და წრფივ შეფასებას შორის

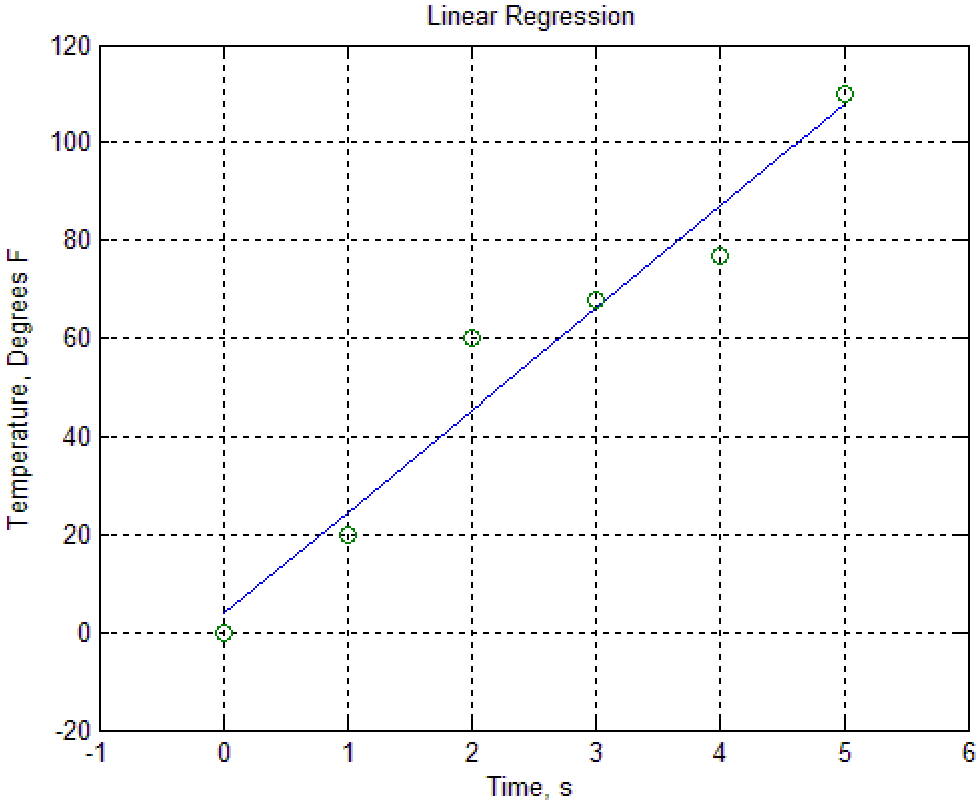
პირველი ორი წერტილი პირდაპირ წრფეზე მოთავსდა, ასე, რომ d_1 და d_2 0-ის ტოლია. d_3 ტოლია 60-40, ანუ 20. ასევე შეიძლება შევაფასოთ დანარჩენი მანძილებიც. თუ გამოვითვლით მანძილების ჯამს, დადებითმა და უარყოფითმა მანძილებმა შეიძლება ერთმანეთი გააბათილონ და მივიღებთ უფერო მცირე მნიშვნელობას, ვიდრე მოსალოდნელია, ამის თავიდან ასაცილებლად უნდა შევკრიბოთ აბსოლუტური სიდიდეები ან კვადრატში აყვანილი მნიშვნელობები. წრფივი შეფასების ხარისხის ზომად ვირჩევთ წერტილებსა და წრფივ შეფასებას შორის მანძილების კვადრატების ჯამს. ეს ჯამი გამოითვლება MATLAB საშუალებით შემდეგი ბრძანებებით:

```
sum_sq = sum((y-y1).^2)
```

მივიღებთ: sum_sq =573.

თუ გავავლებთ სხვა წრფეს, შეგვიძლია ასეთივე ჯამი მისთვისაც გამოვითვალოთ. ამ ორ წრფეს შორის უკეთესი შეფასება ის იქნება, რომლისთვისაც ჯამის მნიშვნელობა უფრო მცირეა. იმისათვის, რომ ვიპოვოთ საუკეთესო წრფივი შეფასება, რომლისთვისაც კვადრატული ჯამის მნიშვნელობა უმცირესია, უნდა დავწეროთ განტოლება, რომელიც გამოითვლის კვადრატული ჯამის მნიშვნელობებს წრფის ზოგად ფორმულაზე დაყრდნობით: $y = mx + b$. დავწერთ განტოლებას, რომელიც წარმოადგენს მანძილების კვადრატების ჯამს. ამ განტოლების ცვლადებია m და b. გამოვითვლით ამ განტოლების წარმოებულს m-ით და b –ით და გავუტოლებთ მას 0. ამგვარად გამოთვლილი m და b განსაზღვრავს იმ წრფეს, რომელსაც ექნება მანძილების კვადრატების ჯამის უმცირესი მნიშვნელობა მოცემული წერტილების მიმართ. ეს სიდიდეები შეიძლება გამოთვლილი იქნას ფუნქციით **polyfit**. ეს ფუნქცია წარმოდგენილი იქნება მოგვიანებით, პოლინომური რეგრესიის განხილვის შემდეგ. ნახ. 9.8 წარმოადგენს წრფეს უმცირესი კვადრატული გადახრით, რომელიც ოპტიმალურად

ასახავს ჩვენს მონაცემებს. იგი შეესაბამება ოპტიმალურ წრფივ შეფასებას. გადახრის კვადრატების ჯამი ტოლია 356.82



ნახ. 9.8 წრფივი რეგრესია უმცირესი კვადრატული გადახრით

9.1.4 პოლინომური რეგრესია

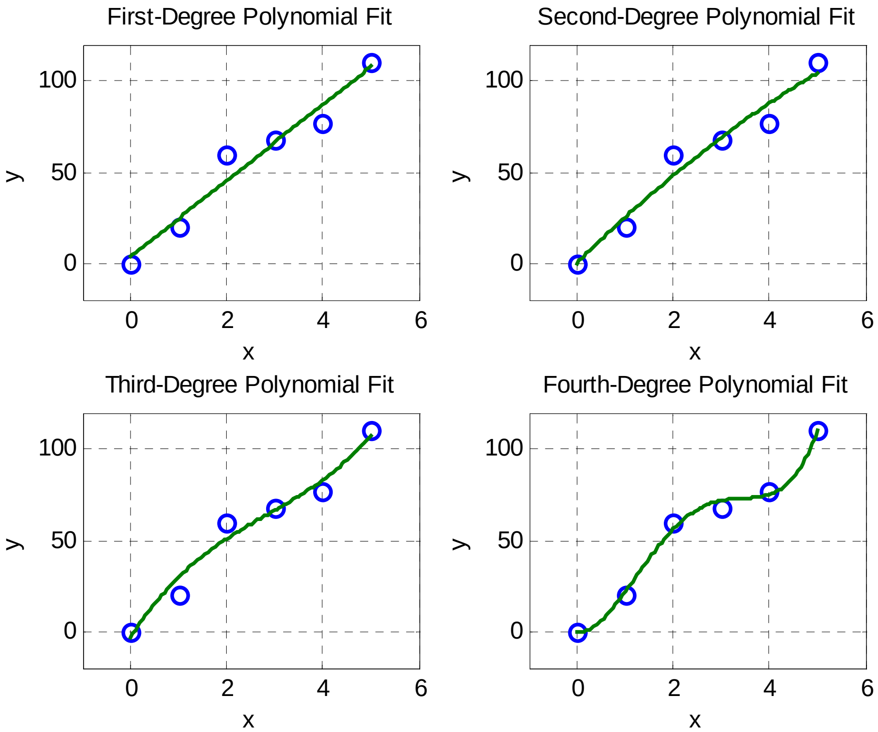
უკვე წარმოგიდგინეთ უმცირეს კვადრატთა მეთოდი, რომელიც გამოითვლის ექსპერიმენტული მონაცემების ოპტიმალურად ამსახველ წრფეს. იგივე შედეგს მივაღწევთ პოლინომური რეგრესიის მეთოდის გამოყენებით, წრფე ხომ იგივე პირველი რიგის პოლინომია. შეგახსენებთ, რომ ერთცვლადიანი პოლინომის ზოგადი ფორმულაა:

$$f(x) = a_0 x^N + a_1 x^{N-1} + a_2 x^{N-2} + \cdots + a_{N-1} x + a_N$$

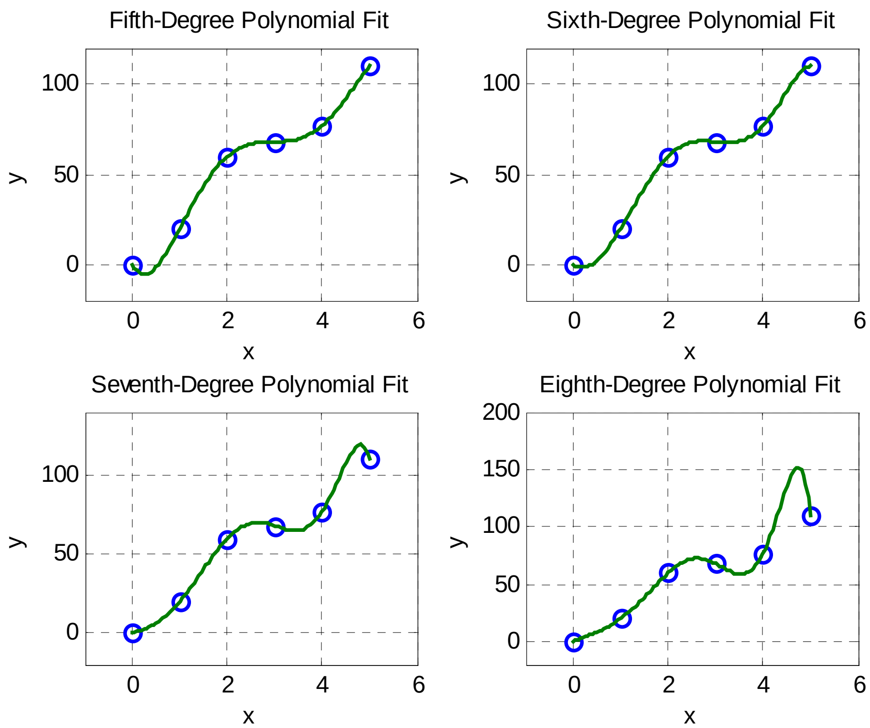
პოლინომის ხარისხი მისი ცვლადის ხარისხის უმაღლესი მაჩვენებლის ტოლია. ამრიგად კუბური, მესამე ხარისხის პოლინომის ზოგადი ფორმულა ასეთია:

$$g(x) = a_0 x^3 + a_1 x^2 + a_2 x + a_3$$

ნახ. 9.9 და ნახ. 9.10 ნაჩვენებია ახალი ძრავის ცილინდრის ხუფში ათვლილი ტემპერეტურის მონაცემების სხვადასხვა რიგის პოლინომური რეგრესიის შედეგად მიღებული მრუდები.



ნახ. 9.9 პოლინომური რეგრესიის შედეგად მიღებული მრუდები



ნახ. 9.10 პოლინომური რეგრესიის შედეგად მიღებული მრუდები

ყურადღება მიაქციეთ, რაც უფრო იზრდება პოლინომის რიგი, მით უფრო მეტია წერტილების რაოდენობა, რომელზეც წირი უშუალოდ გაივლის. თუ მოცემული გვაქვს N წერტილი და გამოვიყენებთ N რიგის პოლინომურ რეგრასიას, მიღებული წირი ყველა წერტილზე გაივლის.

მაღალი რიგის პოლინომური შეფასებისას ყურადღება უნდა მივაქციოთ ჩვენს ხელთ არსებულ მონაცემთა თავისებურებებს. მაღალი რიგის პოლინომს წერტილებს შორის შესაძლოა ცვალებადობის ფართო ინტერვალი ახასიათებდეს, რამაც შეიძლება მცდარ გზაზე დაგვაყენოს მონაცემების შეფაებისას გარკვეულ ინტერვალში.

9.1.5 polyfit და polyval ფუნქცია

MATLAB –ში პოლინომური რეგრასია ხორციელდება ფუნქციით **polyfi**t. ფუნქციას აქვს სამი არგუმენტი. მონაცემთა x - y კოორდინატები და პოლინომის რიგი N. ფუნქცია გვაძლევს N რიგის პოლინომის კოეფიციენტებს პოლინომის x ცვლადის ხარისხის კლების მიმართულებით. (ყურადღება მიაქციეთ, N რიგის პოლინომისათვის ვღებულობთ N+1 კოეფიციენტს.)

ნახ. 9.8 გამოსახული მონაცემების საუკეთესო წრფივი შეფასების გადხრის კვადრატების ჯამი = 356.82. მისი გამოთვლა და მრუდის აგება შესაძლებელიია შემდეგი ბრძანებებით:

```
x = [0, 1, 2, 3, 4, 5];
y = [0 20 60 68 77 110];
coef = polyfit(x,y,1);
m=coef(1);
b=coef(2);
ybest=m*x+b;
sum_sq = sum((y-ybest).^2);
plot(x,ybest,x,y,'o'),...
xlabel('Time, s'), ylabel('Temperature, Degrees F'),...
title('Linear Regression'),...
grid
axis([-1,6,-20, 120])
```

ფუნქცია **polyval** გამითვლის პოლინომს, რომელსაც ახასიათებს უმცირესი კვადრატული გადახრა მოცემული წერტილებიდან. მისი პირველი არგუმენტია პოლინომის კოეფიციენტები, მეორე – x ვექტორი, რომლის არგუმენტებია x ცვლადის ის მნიშვნელობები, რომელთათვისაც გვინდა გამოვითვალოთ პოლინომის მნიშვნელობები. წინა მაგალითში წრფივი რეგრესიის წერტილები გამივითვალეთ მიღებული კოეფიციენტების საშუალებით – ybest=m*x+b, იგივე შედეგს მოგვცემს ფუნქცია **polyva**l:

```
ybest = polyval(coef,x);
```

ავიღოთ მონაცემები, რომლითაც ვსარგებლობდით ზემოხსენებულ მაგალითში და გამოვიყენოთ პოლინომური რეგრესიის მეთოდი, პოლინომის რიგი შევცვალოთ 2-დან 9-მდე. როგორც უკვე აღვნიშნეთ, მოსალოდნელია, რომ დაბალი რიგის პოლინომის შემთხვევაში წირი ყველა მოცემულ წერტილზე არ გაივლის, მაგრამ მეექვსე რიგის პოლინომი უკვე ყველა წერტილს უნდა მოიცავდეს. როგორც ნახზიდან კარგად ჩანს, მეექვსედან მეცხრე რიგის ჩათვლით

ყველა პოლინომი ჩვენს მიერ განხილულ ყველა წერტილს მოიცავს და მათი შესაბამისი უმცირესი კვადრატული ჯამი 0-ის ტოლია, თუმცა დაბალი რიგის პოლინომის შესაბამისი მრუდები უკეთ ასახავს ჩვენი მონაცემების საერთო ტენდენციას. ზემითაღწერილი პოლინომური მრუდები აიგება შემდეგი ბრძანებებით:

```
x =  [ 0, 1, 2, 3, 4, 5 ];
y = [ 0, 20, 60, 68, 77, 110 ];
newx = 0:0.5:5;
for n=1:9;
     f (:,n) = polyval(polyfit(x,y,n), newx) ';
end
```

გრაფიკის ასაგებად ამ ბრძანებებს დაემატება:

```
plot(newx, f(:,2), x,y, 'o')
```

საჭიროა კიდევ დამატებითი ბრძანებები გრაფიკის გასაფორმებლად: სათაურის დაწერა, შესაბამისი წარწერა ღერძებზე, ღერძების საზღვრების განსაზღვრა და subplot ბრძანების გამოყენება.

ამ თავში გავეცანით განსხვავებას ინტერპოლირებასა და უმცირეს კვადრატთა მეთოდს შორის. წარმოგიდგინეთ ინტერპოლირების ორი სხვადასხვა ტიპი: წრფივი და კუბური ინტერპოლირება. განვიხილეთ MATLAB ბრძანებები ინტერპოლირებისათვის. შემდეგ გაგაცანით უმცირეს კვადრატთა მეთოდის ძირითადი პრინციპი და სხვადასხვა რიგის პოლინომური რეგრესია. გიჩვენეთ, თუ როგორ შევარჩიოთ მონაცემებისათვის საუკეთესო წირი უმცირეს კვადრატთა მეთოდის საშუალებით სხვადასხვა რიგის პოლინომის გამოყენებით.

ბრძანებები და ფუნქციები

polyfit	გამითვლის პოლინომს უნცირეს კვადრატთა მეთოდით
polyval	გამოითვლის პოლინომის მნიშვნელობებს
spline	კუბური ინტერპოლირება
interp1	ერთგანზომილებიანი წრფივი ინტერპოლირება
interp2	ორგანზომილებიანი წრფივი ინტერპოლირება

ამოცანები

1 – 7 პრობლემა დაკავშირებულია ამ თავში განხილულ ამოცანებთან, ხოლო 8 – 11 – ახალ საინჟინრო ამოცანებს უკავშირდება

რობოტის მკლავის მანიპულატორი. პრობლემები უკავშირდება ამ თავში განხილულ ამოცანას რობოტის მკლავის მანიპულატორის შესახებ. მონაცემები ჩაწერილია ფაილში points.dat.

1. დაწერეთ პროგრამა იმისათვის, რომ წინასწარ შეამოწმოთ მონაცემები ფაილში points.dat. არის თუ არა დალაგებული ზრდის მოხედვით x კოორდინატის მნიშვნელობები მარშრუტის სამ სხვადასხვა მონაკვეთზე?

2. დავუშვათ ფაილი points.dat მიიცავს სამზე მეტ მონაკვეთს, მაგალითად ეს შეძლება იყოს რამდენიმე ობიექტის გადატანა ახალ ადგილზე. დაწერეთ პროგრამა, რომელიც დაითვლის ინდივიდუალური მარშრუტების რაოდენობას, რომელიც მთავრდება ან ობიექტის აღებით, ან მისი დადებით ახალ ადგილზე, ან საწყის მდებარეობაში დაბრუნებით.
3. შეცვალე კუბური ინტერპოლაციის პროგრამა ისე, რომ მან დაბეჭდოს მთელი მარშრუტის ინტერპოლირებული მონაცემები და ჩაწეროს იგი ფაილში path.dat. წაშალეთ მონაცემები, რომლებიც მეორდება.
4. დაწერეთ პროგრამა, რომელიც წაიკითხავს ფაილს path.dat, რომელიც მე-3 ამოცანაშია აღწერილი, ააგებს საწყის მარშრუტს და დასვამს წრეებს იმ წერტილებზე, სადაც რობოტის მკლავი შეჩერდა. (რობოტის მკლავი ჩერდება რომ აიღოს ობიექტი, დადოს იგი ან იმ შემთხვევაში თუ დაუბრუნდა საწყის მდებარეობას.)
5. შეცვალე მე-4ამოცანაში აღწერილი პროგრამა ისე, რომ მარშრუტი შეიცავდეს სამზე მეტ მონაკვეთს, როგორც მე-2 ამოცანაშია აღწერილი.

ნავთობის ჭაბურღილი.(oil well production) დავუშვათ გვინდა გავიგოთ ნავთობის მოქმედი ჭაბურღილის პროდუქცია როგორაა დამოკიდებული ტემპერატურაზე. გვაქვს ექსპერიმენტული მონაცემები, რომელიც გვიჩვენებს საშუალოდ დღის განმავლობაში წარმოებული ნავთობის რაოდენობას ბარელებში და შესაბამის საშუალოდ ტემპერატურას დღის განმავლობაში. მონაცემები ჩაწერილია ASCII ფაილში oil.dat.

6. რადგანაც მონაცემები არცერთი პარამეტრის მიხედვით არ არის დალაგებული, პირველ რიგში საჭიროა მათი გადალაგება. დაწერეთ პროგრამა, რომელიც კითხულობს ფაილს oil.dat და ქმნის ორ ახალ ფაილს: ფაილი oiltmp.dat უნდა შეიცავდეს მონაცემებს დალაგებულს ნავთობის რაოდენობის ზრდის მიხედვით ტემპერატურას შესაბამისი მონაცემებით, ხოლო ფაილი tmpoil.dat – ტემპერატურის ზრდის მიხედვით დალაგებულ მონაცემებს ნავთობის რაოდენობის შესაბამისი მნიშვნელობებითურთ. თუ გვაქვს წერტილები x კოორდინატის ერთნაირი მნიშვნელობებით, ახალ ფაილში უნდა დარჩეს მხოლოდ ერთი მინაცემი, ხოლო შესაბამისი y უნდა იყოს ერთნაირი x კოორდინატის მქონე y კოორდინატების საშუალო მნიშვნელობა. .
7. დაწერეთ პროგრამა, რომელიც ააგებს tempoil.dat ფაილის მონაცემების გრაფიკს წერტილების მეორე და მესამე რიგის პოლინომური აპროქსიმაციის საფუძველზე. დაბეჭდეთ მიღებული პოლინომის გამოსახულება და უმცირესი კვადრატული ცთომილება.
8. დაწერეთ პროგრამა, რომელიც ააგებს oiltmp.dat ფაილის მონაცემების გრაფიკს წერტილების მეორე და მესამე რიგის პოლინომური აპროქსიმაციის საფუძველზე. დაბეჭდეთ მიღებული პოლინომის გამოსახულება და უმცირესი კვადრატული ცთომილება.
9. დავუშვათ ტემპერატურის ზრდის მიხედვით დალაგებული მონაცემები ნავთობის შესაბამისი რაოდენობით გვინდა შევაფასოთ (აღვწეროთ) მესამე რიგის პოლინომით. დაწერეთ პროგრამა, რომელიც საშუალებას მოგვცემს შევიყვანოთ ტემპერატურის მნიშვნელობა და მივიღოთ ნავთობის რაოდენობის შესაბამისი მნიშვნელობა.