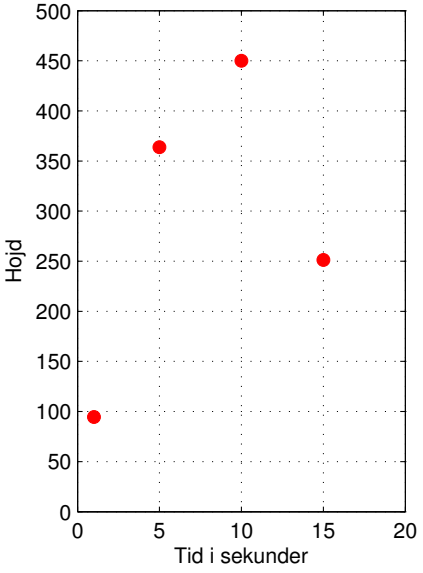
# Interpolation och polynomanpassning

Delar av GNM kap 4

## Motiverande exempel - Interpolation

Vi vet från grundkurser i fysik att en boll som kastas rakt upp följer en parabola $v_0 t - \frac{mg}{2}t^2$ om man inte beaktar luftmotståndet. Antag att vi nu skickar upp ett objekt med hög fart så att luftmotståndet inte är försumbart. Med mätinstrument finner vi att höjden $y(t)$ vid vissa tidpunkter ges av

| $t$ | 1 | 5 | 10 | 15 |
|---|---|---|---|---|
| $y(t)$ | 94.59 | 363.75 | 450 | 251.2 |




Problem: Var befinner sig objektet vid tidpunkten $t = 12$?

Genom att titta på grafen kan man kanske grovt hitta en approximation för var den bör befinna sig. I den här kursen lär vi oss hur man gör det systematiskt, och hur man använder ytterligare information om funktionen för att approximera funktionsvärden som inte finns i tabellen. Tänk dig t.ex. att en ingenjör med kunskap om fysiken säger att höjden kan skrivas som

$$y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

Vi förväntar oss att $a_0 \approx 0$, men annars har vi ingen information om konstanterna $a_0, a_1, a_2, a_3$. Hur kan vi använda denna information för att bättre bestämma $y(12)$?

## Problemformulering - Interpolation

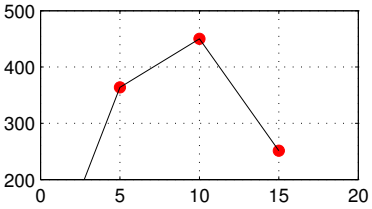
Givet $n$ datapunkter

| $x$ | $x_1$ | $\cdots$ | $x_n$ |
|---|---|---|---|
| $y$ | $y_1$ | $\cdots$ | $y_n$ |

och antag att en funktion $f(x)$ går genom samtliga punkter, d.v.s. $y_1 = f(x_1), \ldots, y_n = f(x_n)$. Bestäm $f(x)$ för ett värde som inte är givet i tabellen $x \neq x_1, \ldots, x \neq x_n$. Tänk: "vi läser mellan raderna"

Funktionen $f(x)$ är typiskt känd men innehåller några okända konstanter, till exempel ett polynom där koefficienterna är okända.

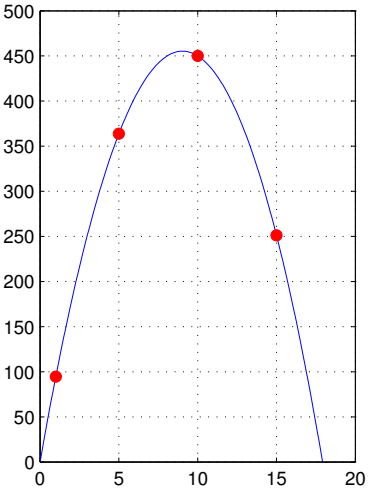
## Urval av metoder och tekniker




- (GNM sid 136) Styckvis linjär interpolation: Dra räta linjer mellan datapunkterna. Alt. 1 beräkna $k$ och $m$ om $y(x) = kx + m$. Alt. 2: Den räta linjen som går genom två punkter $(x_1, y_1)$ och $(x_2, y_2)$ är

$$\ell(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1).$$

- (GNM 4.1E-1) Polynominterpolation (viktig):
  - Ställ upp en ekvation för varje datapunkt $\Rightarrow$ Ett linjärt ekvationssystem. Till exempel för luftmotståndsexemplet ovan får vi




$$\begin{pmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 5 & 5^2 & 5^3 \\ 1 & 10 & 10^2 & 10^3 \\ 1 & 15 & 15^2 & 15^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 363.75 \\ 450 \\ 251.2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \approx \begin{pmatrix} 0.0036 \\ 100 \\ -5.4 \\ -0.01 \end{pmatrix}$$

    Nu hittar vi lätt
    $y(12) = 0 + 100 \cdot 12 - 5.4 \cdot 12^2 - 0.01 \cdot 12^3 = 405.12.$

    Obs: Antalet obekanta är lika med antalet ekvationer när man gör polynominterpolation. Om vi har fler ekvationer än obekanta finns det i regel ingen funktion som går genom alla punkter. Istället kan man använda andra metoder, framförallt minstakvadratmetoden som vi behandlar senare i kursen.

- (GNM sida 137) Felet i polynominterpolation är

$$\frac{f^n(\xi)}{n!}(x - x_1)(x - x_2) \cdots (x - x_n)$$

  för något värde $\xi$ i interpolationsintervallet.

- (GNM sida 139) Runges fenomen: Interpolation med polynom av hög grad kan leda till ofysikaliska och dåliga approximationer. Typiskt stora oscillationer i kanten med ekvidistanta data.

- (GNM sida 135) Newtons interpolationspolynom: Ansätt istället

$$y(x) = c_1 + c_2(x - x_1) + c_3(x - x_1)(x - x_2) + \cdots + c_n(x - x_1)(x - x_2) \cdots (x - x_n)$$

  Leder också till ett linjärt ekvationssystem men med "snällare" tal.

- (GNM Exempel 4.13) Hermiteinterpolation. Om derivatorna i varje datapunkt också är givna kan vi för varje delintervall göra en anpassning av ett kubiskt polynom.

## Relaterade MATLAB-koncept och kommandon

Matriserna som man behöver för interpolation kan ofta ställas upp enkelt i med hjälp av kolumnvektorer. Till exempel

```
tv=[1;5;10;15];
yv=[94.59;363.75;450;251.2];
```

Obs: `tv` och `yv` är kolumnvektorer inte radvektorer.
Kom ihåg att `.` betyder att matlab ska beräkna elementvis och `.^` är elementvis upphöjt till. Matrisen i exemplet ovan ges av

```
A=[ones(4,1), tv.^1, tv.^2, tv.^3]
```

$\Rightarrow$

$$\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 25 & 125 \\ 1 & 10 & 100 & 1000 \\ 1 & 15 & 225 & 3375 \end{matrix}$$

Ett linjärt ekvationssystem kan man i matlab lösa med kommandot `\`

```
c=A\yv;
```

Vektorn `c` innehåller nu polynomkoefficienterna. Plotta det interpolerande polynomet:

```
p=@(x) c(1)+c(2)*x+c(3)*x.^2+c(4)*x.^3;
ttv=0:0.1:1;  % obs: ej tv-vektorn, eftersom den skulle ge hackig kurva
plot(ttv,p(ttv));
p(tv) % detta ska ge värdena i yv
```

Andra MATLAB-kommandon: `polyfit`, `polyval`