Practicum 1

1 Inleiding

De doelstellingen van dit practicum zijn:

- Inleiding tot **MATLAB**, voor een mini-handleiding: zie appendix B van de cursus.
- Introductie van het begrip **conditie** (p.20).
- Voorbereiding op hoofdstuk 2 (stelsels lineaire vergelijkingen) en 3 (niet-lineaire benaderingen.)

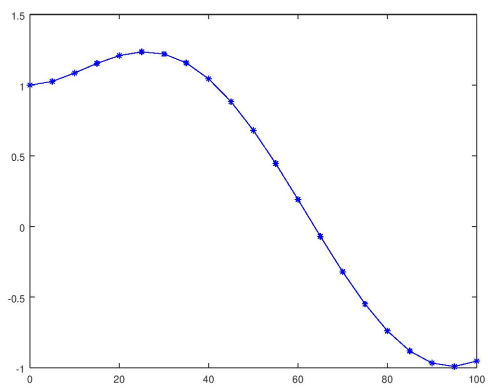
2 Interpolatie van datapunten

Stel, we beschouwen een systeem met als onderliggende functie $f(x) = \exp(-x/20) + \sin(x/20)$, MATLAB notatie: `f = @(x) exp(-x/20) + sin(x/20)`. We kennen die functie niet, maar we kunnen wel op bepaalde tijdstippen x_i, $i = 1, 2, \ldots, n$ meten.
Neem als waarde voor $\{x_i\}_{i=1}^n$ equidistante punten in het interval $[0, 100]$.

2.1 Data bekijken

We nemen $n = 21$ en visualiseren de functie $f(x)$ in $\{x_i\}$.
Maak eerst een figuurvenster met **figure** en gebruik daarna het commando **plot**:

```
figure % maakt een nieuw figuurvenster aan
plot(x,f(x), 'b*-')
```



2.2 Veelterminterpolatie

We kennen de functie $f(x)$ in n discrete meetpunten $\{x_i, f(x_i)\}_{i=1}^n$, maar we zouden toch uitspraken willen doen over andere waarden van $f(x)$, die niet gegeven zijn.

We zullen $f(x)$ benaderen als een functie $\hat{f}(x)$ door middel van interpolatie. Het principe is als volgt:

- We kiezen een functieruimte waarin de benadering $\hat{f}(x)$ leeft. Voor die ruimte kiezen we een basis. In dit geval kiezen we voor veeltermen en de basisfuncties x^i met $i = 0, 1, 2, \ldots$.
- We schrijven de benadering als $\hat{f}(x) = \sum_{j=0}^{d} c_j x^j$ met $d + 1$ onbekende coëfficiënten c_i.
- We bepalen de coëfficiënten via het stelsel

$$\begin{cases} \hat{f}(x_1) = f(x_1) \\ \vdots \\ \hat{f}(x_n) = f(x_n) \end{cases} .$$

Met dit principe gaan we nu aan de slag.

Opgave 1a: We benaderen $f(x)$ als $\hat{f}(x) = c_0 + c_1 x + c_2 x^2$.

1. Stel een stelsel lineaire vergelijkingen op waaruit je c_0, c_1, c_2 kan berekenen. (Om 3 onbekenden te bepalen heb je 3 datapunten nodig hiervoor. Welke datapunten kies je?)
2. Schrijf dit in de vorm $Ac = y$, met $A \in \mathbb{R}^{d+1\times d+1}$ een matrix en $c, y \in \mathbb{R}^{d+1}$ vectoren[1].
3. Los dit stelsel op (bereken c).
4. Evalueer $\hat{f}(x)$ in alle 21 punten die we hebben en vergelijk met de gegeven punten $\{x_i, f(x_i)\}_{i=1}^{21}$. Optioneel heb je het MATLAB-commando `polyval` om veeltermen te evalueren. Lees in de documentatie van `polyval` hoe je dit commando moet gebruiken. Als jij de coëfficiënten omgekeerd hebt geordend t.o.v. wat de documentatie vereist, kun je het commando `flipud` gebruiken.
5. Maak een figuur van de functie zelf en de benadering.

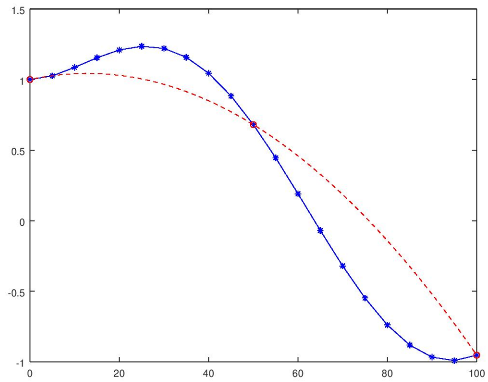
Om meerdere plots te maken in een figuurvenster kan je het commando `hold on` gebruiken. Bijvoorbeeld:

```
figure
plot(x,f(x), 'b*-')
hold on
plot(x,polyval(flipud(c),x), 'r--')
```

Onze keuze voor benaderende functie is duidelijk te simpel om de data goed te benaderen, we willen een hogeregraadsveelterm.

Opgave 1b: Formuleer het probleem voor algemene d, het principe is hetzelfde als in Opgave 1a. Kies een graad

[1] De matrix A heeft een speciale structuur. We noemen dit soort matrices *Vandermonde-matrices* MATLAB heeft gespecialiseerde methoden om hiermee te rekenen en om ze op te stellen. Je kunt dit opzoeken met het commando `lookfor vandermonde`.



1. Stel een stelsel lineaire vergelijkingen op waaruit je $c_0, c_1, \ldots, c_d$ kan berekenen.
2. Schrijf dit in de vorm $Ac = y$, met $A \in \mathbb{R}^{d+1\times d+1}$ een matrix en $c, y \in \mathbb{R}^{d+1}$ vectoren.
3. Los dit stelsel op, i.e., bereken c. Schrijf je code zodanig dat die gemakkelijk is om aan te passen naar een andere graad van de veelterm.
4. Evalueer de benadering in alle 21 punten die we hebben en vergelijk met de gegeven punten $\{x_i, f(x_i)\}_{i=1}^{21}$.

Wat is de maximale graad waarvoor het stelsel een unieke oplossing heeft? Hint: men kan bewijzen dat

$$\det\left(\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{d+1} & x_{d+1}^2 & \ldots & x_{d+1}^d \end{bmatrix}\right) = \prod_{1\leq i<j\leq d+1} (x_j - x_i).$$

Deze determinant kan enkel nul worden als twee punten x_j en x_i met $i \neq j$ samenvallen.

Bereken de veeltermbenadering $\hat{f}$ van f voor enkele graden $d = 2, 3, \ldots$. Zie je dat de benaderende functie beter aansluit bij de opgegeven functie in de 21 opgegeven punten naarmate de graad stijgt? Wat gebeurt er als je een hogere graad neemt, dichter bij de maximale graad?

2.3 Conditie van een probleem

Theoretisch gezien geeft de methode die we hebben uitgevoerd een veelterm die door alle opgegeven meetpunten $(x_i, f(x_i))$ gaat. Het stelsel dat we oplossen drukt immers uit dat $\hat{f}(x_i) = f(x_i)$ voor alle opgegeven punten x_i.

Toch zien we dat dit in de praktijk niet meer van toepassing is als we het stelsel numeriek oplossen. Dit komt doordat de getallen in de computer niet exact worden voorgesteld. Om dit te beprijpen, kijken we naar het begrip *conditie*. Herinner je dat het conditiegetal van een matrix gegeven wordt als $\kappa(A) = \|A\|\|A^{-1}\|$. We formuleren de toepassing van het conditiegetal voor een algemeen lineair stelsel.

Het stelsel kan worden geschreven in de vorm $Ac = y$. De matrix A en het rechterlid y zijn (berekend uit) de gegevens. Deze kunnen we hoedanook niet exact voorstellen in een computer omdat onze getalvoorstelling eindig is. In het geheugen van de computer zitten dus een matrix $\tilde{A} := A + \Delta A$ en een rechterlid $\tilde{y} := y + \Delta y$. We hebben dus het stelsel

$$(A + \Delta A)\tilde{c} = y + \Delta y. \tag{1}$$

Het beste wat we nu nog kunnen hopen is dat dit licht gewijzigde stelsel exact wordt opgelost. In wat volgt, maken we de (optimistische maar eigenlijk onrealistische) vereenvoudiging dat $\Delta A = 0$. Als c de exacte oplossing is van $Ac = y$ en $\tilde{c}$ de oplossing van het ruizige stelsel (1), dan kunnen we de relatieve fout op de oplossing langs boven begrenzen als

$$\frac{\|c - \tilde{c}\|}{\|c\|} \leq \kappa(A)\frac{\|\Delta y\|}{\|y\|} + \mathcal{O}(\|\Delta y\|^2) \quad \text{waarin} \quad \kappa(A) = \|A\|\|A^{-1}\|. \tag{2}$$

De term $\mathcal{O}(\|\Delta y\|^2)$ zullen we verwaarlozen als $\|\Delta y\|$ klein is t.o.v. $\|y\|$. In het geval $\|\Delta y\| \to 0$ wordt deze bovengrens scherp. We kunnen over het algemeen dus niet verwachten dat een algoritme dat het stelsel oplost een nauwkeuriger resultaat bereikt dan wat geschat wordt met deze bovengrens.

Concreet nu voor ons geval: de vector y bestaat uit de meetpunten $[f(x_1), \ldots, f(x_n)]$. Hierop zit ruis die veroorzaakt kan worden door het inlezen in een computer met een eindige getalvoorstelling; of in de praktijk kan die ruis ook komen door inexacte metingen van een fysische grootheid. De nauwkeurigheid van de oplossing c kunnen we schatten met (2).

Opgave 1c:

1. Bereken het conditiegetal $\kappa(A)$, waar A de matrix is voor de $d+1$ punten uit de vorige oefening. Hiervoor heb je het MATLAB-commando `cond`.

2. We simuleren nu dat $y_i = f(x_i)$ fysische metingen zijn met een relatieve meetfout zit van 10^{-6}. Wiskundig gezegd: nu werken we met het rechterlid

   $$\tilde{y} = [y_i(1 + \varepsilon_i)]_{i=1}^n \quad \text{met} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \quad \text{en} \quad \sigma = 10^{-6}.$$

   In MATLAB kun je dit doen door een vector `epsilon` te maken met `epsilon = 1e-6 * randn(size(y))` en vervolgens

   ```
   ytilde = y .* (1 + epsilon)
   ```

   te definiëren. Stel dan het stelsel op en bereken de coëfficiënten $\tilde{c}$ als oplossing van het stelsel $A\tilde{c} = \tilde{y}$.

3. Stel nu dat de oplossing c zoals berekend in de vorige oefening de exacte oplossing van het stelsel $Ac = y$ is[2]. Wijken de coëfficiënten $\tilde{c}$ sterk af van c (let op dat je dezelfde datapunten gebruikt bij de interpolatie)? Kijk ook na of de ongelijkheid (2) voldaan is.

4. Is het een goed idee om de graad van de veelterm hoog te kiezen? (Denk ook eens terug aan Opgave 1b, voor $d = 20$.)

[2] Eigenlijk is dit niet zo, want deze oplossing is berekend op een computer met een eindige getalvoorstelling. Toch zal deze vector c meer op de theoretisch exacte oplossing lijken in vergelijking met de oplossing $\tilde{c}$ het stelsel $A\tilde{c} = \tilde{y}$. In dat stelsel hadden we namelijk opzettelijk een meetfout aangebracht die vele malen groter is dan de machineprecisie.