Laboration 2: Numerisk Integration, Polynominterpolation, Minstakvadratmetoden

1 Mål av denna laboration

Lab 2 är tänkt som en förberedelse inför föreläsningen 5 och 6 och handlar om numerisk integration, polynominterpolation och minstakvadratmetoden.

Polynominterpolation: Vi kommer att titta på interpolation av funktioner med olika polynom samt studera Runges fenomen.

Minstakvadratmetoden: Vi kommer att undersöka hur en funktion kan anpassas till givna data.

Numerisk integration: Vi kommer att testa trapets- och simpsons regel som används för att approximera en intergral. Vi ska studera hur man kan bestämma noggrannhetsordningen av en gränsvärdesprocess och hur adaptivitet kan hjälpa oss att integrera noggrannare.

2 Polynominterpolation

MATLAB-filer: *Polynominterpolation.m, Rungesfenomen.m*

2.1 Kort introduktion

I detta avsnitt lär vi oss hur man kan finna en funktion $F(x)$ som passerar genom n-givna punkter $(x_i, y_i), i = 1, 2, .., n$ dvs finna $F(x)$ så att $F(x_i) = y_i$. $F(x)$ kallas interpolant.
Vi fokuserar oss på polynom som interpolationsfunktioner

$$F(x) \in \{\sum_{j=0}^{n-1} a_j x^j | a_j \in \mathbf{R}\}$$

2.2 Interpolation

1. I *Polynominterpolation.m* har vi nio givna temperaturvärden för olika tidpunkter under ett dygn. Kör programmet. I första plotten interpolerar vi de nio givna mätvärdena med ett polynom av grad 8, $F(x) = \sum_{j=0}^{8} c_j x^j$. Hur får vi matrisen A som kallas Vandermondematris i detta fall?

2. Styckvis linjär interpolation
   I andra delen av *Polynominterpolation.m* interpolerar vi mätvärdena styckvis linjärt. I styckvis linjär interpolation kräver man att $F(x)$ ska vara linjär över varje delintervall:

$$F(x) = a_i + b_i x \quad a_i, b_i \in \mathbf{R}, \quad x \in [x_i, x_{i+1}]$$
$$F(x_i) = y_i, \quad F(x_{i+1}) = y_{i+1}$$

   Vad beräknas i for-loopen som tillhör denna figur? Beskriv varje rad med en kort kommentar.

3. Sista plotten erhölls med de inbyggda MATLAB funktionerna *polyfit* och *polyval*. Hur använder man dem för att interpolera de 9 givna mätvärdena med ett polynom av grad 8? Jamför *p* och *coeff* i programmet.

4. Bestäm andragradspolynomet genom punkterna (9, -8), (10, 1) och (11, 12).

2.3 Runges fenomen

I *Rungesfenomen.m* vill man interpolera $f = e^x/(1 + 9x^2)$ i vissa punkter. Man väljer en interpolation med 7 resp 13 ekvidistanta punkter. Vad uppstår i bägge fallen?
Dessa svängningar vid randen kallas Runges fenomen och visar upp en varning att interpolation av hög grad med ekvidistanta delintervaller inte alltid är optimal. Man kan minska felet genom att fördela mätpunkterna på ett bättre sätt. Ändra i programmet mätpunkterna i x till

$$x_i = cos(\frac{n-i}{n-1}\pi), \quad i = 1, \cdots, 13 \quad n = 13$$

3 Minstakvadratmetoden

MATLAB-filer: *Minstrakvadratmetoden.m*

3.1 Kort introduktion

I detta avsnitt lär vi oss att hitta en god approximation $F(x)$ till givna punkter $(x_i, y_i), \quad i = 1, \cdots, n$ när ett linjärt ekvationssystem $Ax = b$ som beskriver $F(x)$ är överbestämd. Det vill säga att för $Ax = b$ med A en (mxn)-matris och b en vektor med n element gäller $n < m$.
För att minimera felet löser man $A^T Ax = A^T b$ (minstakvadratmetoden). MATLAB beräknar minstakvadratlösningen med vänsterdivisionen, $x = A\backslash b$.

3.2 Exempel

1. Givet är följande uppsättning punkter:

x	2	4	7	9
y	3	6	7	10

   I *Minstrakvadratmetoden.m* bestäms den räta linjen $y = a + bx$ som i minstrakvadratmetodens mening bäst approximerar mätvärdena. Vilka värde beräknas för a och b?

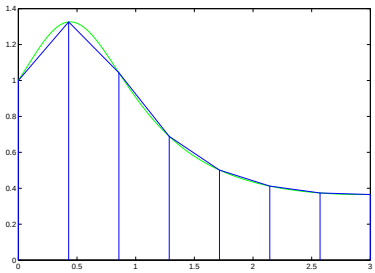

Figure 1: Approximation av en funktion med styckvis linjära funktioner

2. Bestäm minstakvadratlösningen till det överbestämda ekvationssystemet.

$$\begin{aligned} x + y &= 1 \\ 2x + y &= 3 \\ x + 2y &= 2 \\ 2x - y &= 2 \end{aligned}$$

4 Numerisk integration

MATLAB-filer: *Trapetsregel.m, f62a.m, Simpsonsregel.m, peakfkt.m, WithoutAdaptivity.m, With-Adaptivity.m, NoggrannhetTrapets.m, NoggrannhetSimpson.m*

4.1 Kort introduktion

I detta avsnitt lär vi oss hur man kan använda olika numeriska metoder för att approximera ett integralvärde, $A = \int_a^b f(x)\mathrm{d}x$.
Idén vid numerisk integrering är att intervallet $[a, b]$ delas in i delintervall och på varje sådant delintervall $[x_i, x_{i+1}]$ approximeras integranden med en enklare funktion som kan integreras exakt. Integralens värde kan t.ex. approximeras med summan av paralleltrapetsareor som visas i Figur 1. Detta motsvarar exakt integration av en styckvis linjär approximation av f.

4.2 Trapetsregeln

1. *Trapetsregel.m* visar hur följande integralvärde beräknas med trapetsregeln.

$$\int_0^3 \frac{e^x}{1 + 2x^3}\mathrm{d}x$$

   I trapetsregeln approximeras integralen på ett delintervall $[x_i, x_{i+1}]$ med trapetsregeln

$$A_i = (x_{i+1} - x_i) \cdot \frac{(f(x_i) + f(x_{i+1}))}{2}$$

   Väljer man ekvidistanta delintervaller dvs $(x_{i+1} - x_i) = h$ och summerar upp alla så blir det approximerade interalvärdet $A(h) = h \cdot (\sum_{i=1}^{n+1} f(x_i) - \frac{f(x_1)+f(x_{n+1})}{2})$.
   Ladda ner *f62a.m* och kör *Trapetsregel.m*. Hur kan approximationen förbättras?

2. Lös med samma program

$$\int_0^1 \frac{4}{1+x^2}\mathrm{d}x$$

Den exakta lösningen till denna integral är π.

4.3 Simpsons regel

I trapetsregeln approximerades integranden med styckvis linjära funktioner. Simpsons regel använder polynom av grad två på varje delintervall. Detta innebär att vi behöver tre stödpunkter. Om man integrerar polynomet fås följande approximation för varje delintervall:

$$A_i(h) = \frac{h}{6}(f(x_i) + 4f(\frac{x_i + x_{i+1}}{2}) + f(x_{i+1}))$$

som summeras ihop för att erhålla integralen.
Kör *Simpsonsregel.m* och jamför med trapetsregeln. Vad händer i andra for-loopen?

4.4 Noggrannhetsordning och trunkeringsfel

Trunkeringsfelet $e(h)$ är felet när en gränsvärdeprocess $A(h)$ konvergerar mot en storlek A men avbryts.

$$e(h) = A(h) - A$$

Om trunkeringsfelet av en metod som motsvarar en gränvärdeprocess kan skattas med

$$e(h) = c_1h^p + c_2h^q + c_3h^r + ... \quad p < q < r$$

så defineras p som noggrannhetsordning av metoden och vi uppskattar $e(h) \sim c_1 \cdot h^p$.
Kör *NoggrannhetTrapets.m* och *NoggrannhetSimpson.m* som visar noggrannhetsordning av trapets- respektiv simpsons regel. Integralapproximeringen är beräknad med olika steglängder h och differanser mellan två trunkeringsfel är plottad med loglog-funktionen.
Varför använder man loglog-funktionen? Vilken noggranngetsordning har trapets- respektiv simpsons regel?

4.5 Adaptivitet

Vi vill beräkna

$$\int_{-10}^{10} 0.51 + 25 \cdot e^{-81(11x+45)^2}\mathrm{d}x.$$

Integranden finns implementerad i *peakfkt.m*.

1. Beräkna integralvärdet med hjälp av *WithoutAdaptivity.m*. Använd 100 delvinterall. Vad har gått fel? Hur kan man lösa problemet?
2. Kör nu *WithAdaptivity.m*. Vilket värde får man nu? Vad behövde åtgärdas?
3. Vilket värde får du med MATLABs inbyggda funktion *quad*? För att få hjälp kör *help quad* i MATLABs terminalfönster.