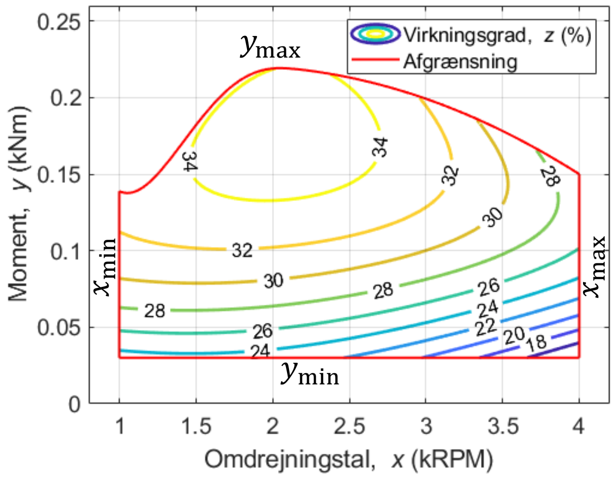
Bemærk følgende:

Hvis besvarelsen af en given delopgave inkluderer et plot, skal dette forsynes med passende titel og aksetekster (inkl. enhed hvis relevant) samt *grid*. I den enkelte delopgave kan der være anført yderligere krav til et givet plot, og disse skal naturligvis også efterkommes.

OPGAVE 1

Figur 1 viser et virkningsgradsdiagram for en mindre dieselmotor, dvs. virkningsgraden (i %) som funktion af motorakslens omdrejningstal (i antal 1000 omdrejninger/minut = kRPM) og momentbelastning (i kNm). Funktionssammenhængen er kun gældende inden for et afgrænset område, hvilket også fremgår af diagrammet.



Figur 1: Virkningsgrad for dieselmotor.

(1a)

Virkningsgraden, z, afhænger af omdrejningstal, x, og belastningsmoment, y, som følger:

$$z = 12{,}0 + 313y - 1466y^2 + 7{,}69x - 126xy + 878xy^2 - 2{,}42x^2 + 34{,}1x^2y - 213x^2y^2$$

Den kombination af omdrejningstal og belastningsmoment, der giver den maksimale virkningsgrad ønskes fundet vha. følgende numeriske metode: For hver kombination af x-værdierne 1,00, 1,01, 1,02, …, 4,00 og y-værdierne 0,030, 0,031, 0,032, …, 0,230 udregnes den tilsvarende z-værdi. Hver gang z udregnes for en ny kombination af x og y, undersøges det, om den beregnede z-værdi er større end den hidtil størst fundne, og hvis det er tilfældet, "noteres" de aktuelle værdier af både x, y og z. De sidst noterede værdier vil således angive det ønskede maksimum.

Følgende MATLAB-script, som kan downloades fra Digital Eksamen under navnet `maksvirkningsgrad.m`, har til formål at gennemføre ovenstående maksimeringsprocedure:

```
clear
z = @(x,y) 12.0 + 313*y - 1466*y^2 + 7.69*x - 126*x.*y ...
           + 878*x.*y^2 - 2.42*x^2 + 34.1*x^2.*y - 213*x^2.*y^2;
x = 1:0.01:4;
y = 0.03:0.001:0.23;
zmax = 100;
for i = 1:4
  for j = 3:23
    zny = z(x(i),y(j));
    if zny < zmax
      xmaxz = x;    % "Notér" aktuel x-værdi i xmaxz
      ymaxz = y;    % "Notér" aktuel y-værdi i ymaxz
      zmax = zny;   % "Notér" aktuel z-værdi i zmax
    end
  end
end
xmaxz, ymaxz, zmax  % Skriv funden løsning
```

Når scriptet er blevet kørt, skal variablen `zmax` indeholde den maksimalt fundne virkningsgrad (i %), og variablerne `xmaxz` og `ymaxz` skal indeholde de tilsvarende værdier af henholdsvis omdrejningstal (i kRPM) og belastningsmoment (i kNm). Scriptet er imidlertid behæftet med fejl, hvorfor de ønskede resultater ikke opnås.

> Ret scriptet, så det virker efter hensigten. I besvarelsen ønskes såvel det rettede script som dets output, når det køres, anført. Hvad er den maksimalt opnåelige virkningsgrad, og ved hvilken kombination af omdrejningstal og belastningsmoment opnås denne værdi ifølge scriptet?

(1b)
Det afgrænsede område i figur 1 opfylder følgende betingelser:

$$x_{\min} \leq x \leq x_{\max} \;\wedge\; y_{\min} \leq y \leq y_{\max}$$

hvor

$$x_{\min} = 1, \quad x_{\max} = 4, \quad y_{\min} = 0{,}03$$

$$y_{\max} = \begin{cases} a_1\,(x-x_0)^4 + b_1\,(x-x_0)^3 + c_0\,(x-x_0)^2 + s_0\,(x-x_0) + y_0 & \text{for } x_{\min} \leq x \leq x_0 \\ c_0\,(x-x_0)^2 + s_0\,(x-x_0) + y_0 & \text{for } x_0 < x \leq x_{\max} \end{cases}$$

$$a_1 = 0{,}1637,\; b_1 = 0{,}237,\; c_0 = -0{,}01425,\; s_0 = -0{,}01113,\; x_0 = 2{,}164,\; y_0 = 0{,}2185$$

Denne opgave handler om at skrive en MATLAB-funktion, `driftomr`, som kan afgøre, om en given kombination af omdrejningstal, *x*, og belastningsmoment, *y*, ligger inden for afgrænsningerne. Her følger en ufærdig version af funktionen (kan downloades fra Digital Eksamen under navnet `driftomr.m`):

```
function resultat = driftomr(x,y)
a1 = 0.1637; b1 = 0.237; c0 = -0.01425;
s0 = -0.01113; x0 = 2.164; y0 = 0.2185;
xmin = 1; xmax = 4; ymin = 0.03;
ymax1 = @(x) a1*(x-x0)^4 + b1*(x-x0)^3 + c0*(x-x0)^2 + s0*(x-x0) + y0;
ymax2 = @(x) c0*(x-x0)^2 + s0*(x-x0) + y0;
if <UDFYLD>
 <UDFYLD FLERE LINJER>
end
```

Funktionen skal sætte `resultat` lig med `1`, hvis punktet (`x`, `y`) ligger inden for det afgrænsede område, og sætte `resultat` lig med `0`, hvis punktet (`x`, `y`) ligger udenfor. Hvis man fx tester funktionen med kommandoen `iomr = driftomr(2,0.1)` vil man af figur 1 kunne se, at det skal resultere i, at `iomr` bliver lig med `1`.

> Færdiggør funktionen ved at erstatte **`<UDFYLD>`** og **`<UDFYLD FLERE LINJER>`** med MATLAB-kode, der får funktionen til at fungere som beskrevet ovenfor. Afprøv derefter funktionen ved efter tur at kalde den med følgende otte kombinationer af *x* og *y*:

x	0,9	1,5	2,5	2,5	3,0	4,0	4,0	4,1
y	0,01	0,2	0,03	0,2	0,2	0,1	0,15	0,1

> Besvarelsen skal indeholde den færdiggjorte funktion samt MATLAB-kommandoer og resultater opnået ved afprøvningerne.

OPGAVE 2

Betragt igen som i opgave 1 virkningsgrad, *z* (%), som funktion af omdrejningstal, *x* (kRPM), og belastningsmoment, *y* (kNm) for en mindre dieselmotor:

$$z = 12{,}0 + 313y - 1466y^2 + 7{,}69x - 126xy + 878xy^2 - 2{,}42x^2 + 34{,}1x^2y - 213x^2y^2$$

Den mekaniske effekt, P (kW), der leveres via motorakslen udregnes ved følgende funktion af *x* og *y*:

$$P = 104{,}7xy$$

Opgaven drejer sig om at finde værdier af *x* og *y*, hvor virkningsgraden er 34% og den mekaniske effekt er 40 kW. Dvs., at følgende ligningssystem skal løses:

$$12{,}0 + 313y - 1466y^2 + 7{,}69x - 126xy + 878xy^2 - 2{,}42x^2 + 34{,}1x^2y - 213x^2y^2 - 34 = 0$$
$$104{,}7xy - 40 = 0$$

(2a)
Ligningssystemet kan løses ad numerisk vej ved at anvende MATLAB-funktionen `newtmult`, der har været anvendt i undervisningen. For at anvende `newtmult` er det nødvendigt først at skrive en MATLAB-funktion, der for givne værdier af de ubekendte, *x* og *y*, kan udregne ligningssystemets venstresider (i form af en vektor `f`) samt den tilsvarende Jacobi-matrix (`J`). Her følger en delvist færdig udgave af en sådan funktion, her navngivet `jacobifunk`:

```
function [J,f] = jacobifunk(X)
x = X(1); y = X(2);
f = [12.0+313*y-1466*y^2+7.69*x-126*x*y+878*x*y^2-2.42*x^2+34.1*x^2*y-213*x^2*y^2-34
     <UDFYLD>];
J = [<UDFYLD>, <UDFYLD>
     <UDFYLD>, <UDFYLD>];
```

Funktionen kan downloades fra Digital Eksamen under navnet `jacobifunk.m`.

Færdiggør funktionen `jacobifunk.m` ved at erstatte tekstangivelserne **<UDFYLD>** med korrekt MATLAB-kode. Angiv den færdige funktion som svar.

(2b)
Anvend funktionen `newtmult` til at løse ligningssystemet, idet der anvendes startværdier på $x = 1$ og $y = 0{,}03$, og idet der udføres så mange iterationer, at fejlen på løsningen er højest 0,002% (`newtmult.m` kan downloades fra Digital Eksamen). Besvarelsen skal indeholde såvel de fundne løsningsværdier af x og y som de MATLAB-kommandoer, der er anvendt til at finde løsningen. Derudover ønskes det kommenteret, om den ønskede nøjagtighed af løsningen er opnået.

PS! Såfremt delopgave 2a ikke er besvaret, kan følgende alternative (ikke korrekte, men dog anvendelige) version af funktionen `jacobifunk` benyttes:

```
function [J,f] = jacobifunk(X)
x = X(1); y = X(2);
f = [12.0+313*y-1466*y^2+7.69*x-126*x*y+878*x*y^2-2.42*x^2+34.1*x^2*y-213*x^2*y^2-34
     x+y-2.72591];
J = [0, 101
     1,   1];
```

(2c)
Den i delopgave 2b fundne løsning til ligningssystemet er blot én ud af de to, der faktisk findes. Den anden løsning kan findes vha. Newton-Raphsons metode ved at benytte startværdier på $x = 2$ og $y = 0{,}2$. Efter fire iterationer vil man nå frem til løsningsværdierne $x = 1{,}822289$ kRPM og $y = 0{,}2096506$ kNm, hvis approksimative relative fejl er er beregnet til 0,0047%.

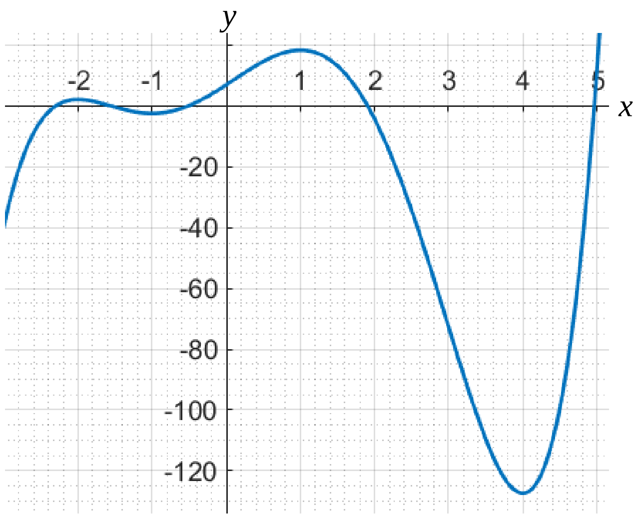
Hvad er løsningsværdierne for x og y, hvis man kun medtager sikkert bestemte decimaler (efter afrunding)?

OPGAVE 3

Lad der være givet følgende femtegradspolynomium:

$$p(x) = 0{,}4x^5 - x^4 - 6x^3 + 2x^2 + 16x + 7$$

Grafen for dette polynomium er vist i figur 2, og heraf fremgår det, at det har fem rødder, nemlig for *x* cirka lig med -2,3, -1,6, -0,5, 1,9 og 5,0.



Figur 2: Graf svarende til $y = p(x)$.

(3a)

Anvend Newton-Raphsons metode til at finde en rod i polynomiet, idet der benyttes en startværdi på $x_0 = 0$, og idet der foretages fire iterationer. I besvarelsen skal både beregningsresultater og beregningsformler for de enkelte iterationer angives (på tabelform i fx Excel). Desuden skal den fundne rod angives med sikkert bestemte decimaler (afrundet).

(3b)

Igen skal Newton-Raphsons metode anvendes til at finde en rod i polynomiet, men nu med start i $x_0 = 3$ og ved gennemførelse af fem iterationer. Her følger et ufærdigt MATLAB-script, som er beregnet til formålet (kan downloades fra Digital Eksamen under navnet `NewtonRaphsonPolynomium.m`):

```
clear, format long
p = @(x) 0.4*x^5 - x^4 - 6*x^3 + 2*x^2 + 16*x + 7;
pdiff = @(x) 2*x^4 - 4*x^3 - 18*x^2 + 4*x + 16;
x = <UDFYLD>;
disp(x)
for i = 1:<UDFYLD>
  x = <UDFYLD>;
  disp(x)
end
```

Færdiggør scriptet ved at erstatte tekstangivelserne **<UDFYLD>** med korrekt MATLAB-kode og kør derefter scriptet. Besvarelsen skal indeholde det færdige script, det output, der produceres, når det køres samt den fundne rod med sikkert bestemte decimaler (afrundet).

(3c)

Hvis man anvender Newton-Raphsons metode med start i $x_0 = 3{,}9$, altså nærmest den største rod på ca. 5,0, vil man efter nogle iterationer ende med at finde roden -2,303216, som er den mindste af polynomiets fem rødder.

Giv en principiel forklaring på, hvorfor Newton-Raphsons metode ikke finder den rod, der ligger nærmest startværdien. Underbyg forklaringen med en skitse eller evt. en enkel beregning.

(3d)

Hvis man anvender Newton-Raphsons metode med start i $x_0 = -1$, kan man ikke iterere sig frem til en rod. Forklar hvorfor? Underbyg forklaringen med en relevant beregning.

(3e)

Anvend den indbyggede MATLAB-funktion `fzero` til at foretage en nærmere bestemmelse af den rod i polynomiet, der ligger omkring -1,6. Besvarelsen skal indeholde såvel anvendte MATLAB-kommandoer som den fundne rodværdi.

(3f)

Benyt den indbyggede MATLAB-funktion `roots` til at bestemme samtlige rødder i polynomiet. Besvarelsen skal indeholde såvel anvendte MATLAB-kommandoer som fundne rodværdier.

OPGAVE 4

SIR-modellen er en simpel, men populær matematisk model til beskrivelse udviklingen i visse epidemityper. Modellen gælder for situationer, hvor infektionen får lov at brede sig i en population uden modforanstaltninger. Populationen bestående af N individer opdeles i antal modtagelige, S, antal inficerede, I, og antal raskmeldte (inkl. døde), R. De enkelte individer vil over tid overgå fra at være modtagelige til at være inficerede for til sidst at være raskmeldte (eller døde).

Antallet af individer i de tre grupper som funktion af tiden t, dvs. $S(t)$, $I(t)$ og $R(t)$, opnås ifølge SIR-modellen ved at løse følgende koblede system af differentialligninger med tilhørende begyndelsesbetingelser:

$$\frac{dS}{dt} = -\frac{\beta}{N}SI, \qquad S(0) = S_0$$
$$\frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I, \qquad I(0) = I_0$$
$$\frac{dR}{dt} = \gamma I, \qquad R(0) = R_0$$

I modellen indgår to overførselskonstanter: β vedrører overførselsraten fra modtagelig til inficeret, og γ vedrører overførselsraten fra inficeret til raskmeldt (eller død).

I denne opgave regnes tiden t i enheden uger, og der betragtes en population på $N = 1000000$ individer, hvoraf der til starttidspunktet $t = 0$ er en enkelt inficeret, mens resten er modtagelige, dvs. $S_0 = 999999$, $I_0 = 1$ og $R_0 = 0$. Der regnes med følgende værdier af modellens konstanter: $\beta = 3{,}4\ \text{uge}^{-1}$, $\gamma = 1{,}3\ \text{uge}^{-1}$.

(4a)

Anvend Eulers metode til at løse differentialligningssystemet fra $t = 0$ til 12 uger med tidsskridt på $h = 0{,}25$ uge. I besvarelsen ønskes både talresultater og beregningsformler anført på tabelform (fx vha. Excel):

iteration	t	S	I	R	dS/dt	dI/dt	dR/dt
0							
1							
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Til hvilket tidspunkt er der ifølge løsningen flest inficerede individer, og hvor mange drejer det sig om?

(4b)

For at opnå en mere nøjagtig numerisk løsning af differentialligningssystemet kan en fjerdeordens Runge-Kutta-metode anvendes. Metoden kan gennemføres vha. MATLAB-funktionen `rk4system`, som er blevet introduceret ved undervisningen. Funktionen kan downloades fra Digital Eksamen under navnet `rk4system.m`.

> Løs igen differentialligningssystemet fra $t = 0$ til 12 uger med tidsskridt på $h = 0{,}25$ uge, men nu ved brug funktionen `rk4system`. Besvarelsen skal indeholde de MATLAB-kommandoer, der er blevet anvendt til løsning af opgaven samt de talresultater, som funktionen `rk4system` returnerer. Til hvilket tidspunkt er der ifølge løsningen flest inficerede individer, og hvor mange drejer det sig om?

(4c)
Hvis man løser differentialligningssystemet vha. Eulers metode med et tidsskridt på 1 uge, opnås følgende resultater efter henholdsvis 12 og 13 iterationer:

Iteration	t	S	I	R
⋮	⋮	⋮	⋮	⋮
12	12	197803	387546	414651
13	13	-62833	144373	918461

Ifølge løsningen estimeres antallet af modtagelige individer, S, efter 13 uger til at være -62833, hvilket naturligvis ikke er et gyldigt antal.

> Forklar, hvorfor Eulers metode kommer frem til et negativt estimat af S, og hvad man kan gøre for at undgå, at Eulers metode giver et negativt resultat.