

TENTAMEN, DEL 2

SF1524

GRUNDLÄGGANDE NUMERISKA METODER OCH PROGRAMMERING

Fredag 13 mars 2020 kl 8.00-11.00

Rättas endast om del 1 är godkänd. Max antal poäng på denna del är 50. Betygsgräns: 10p D, 20p C, 30p B, 40p A. Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag.

Inga hjälpmedel är tillåtna (ej heller miniräknare).

P1. Följande datapunkter är givna:

t_i	0	1	2	4
y_i	e	$e^{\frac{3}{2}}$	$e^{\frac{9}{4}}$	$e^{\frac{15}{4}}$

- (7p) **a)** Datapunkterna antas passa väl till en kurva på formen $y(t) = Ce^{kt}$. Bestäm C och k med hjälp av minsta kvadratmetoden.
- (12p) **b)** Nu vill vi anpassa en kubisk spline $s(t)$ som interpolerar datapunkterna så att

$$s(t) = \begin{cases} a_0 + a_1t + a_2t^2 + a_3t^3, & 0 \leq t \leq 1, \\ b_0 + b_1t + b_2t^2 + b_3t^3, & 1 \leq t \leq 2, \\ c_0 + c_1t + c_2t^2 + c_3t^3, & 2 \leq t \leq 4. \end{cases}$$

Ställ upp det linjära ekvationssystem som behöver lösas för att bestämma $s(t)$ så att $s(t)$ också har kontinuerlig första- och andraderivata samt att andraderivatan är noll i ändpunkterna $t_0 = 0$ och $t_3 = 4$. Du behöver inte lösa ekvationssystemet.

P2. Trapetsmetoden kan användas för att approximera integralen

$$I = \int_a^b f(x)dx.$$

- (2p) **a)** Följande funktionshuvud ska användas i en tillämpning för att approximera värdet av en integral med trapetsmetoden:

```
function Th = trapets(f, int, h)
% Beräknar en approximation av integralen I av f(x) från
% a till b med Trapetsmetoden och steglängd
%
% f      - Funktionshandtag till integranden f(x),
%          t.ex. f = @(x) cos(x)
% int    - [a, b], t.ex. int = [0, 1]
% h      - Steglängd i trapetsmetoden, t.ex. h = 0.1
% Th     - Numerisk approximation till integralen av f(x)

Th = ?

end
```

Komplettera funktionen med Matlab-kod som beräknar Th.

- (8p) **b)** Skriv ett program som använder funktionen `trapets(f, int, h)` för att verifiera att noggrannhetsordningen för trapetsmetoden är 2 för integralen

$$\int_0^1 e^{\cos(2x)} dx.$$

Använd en lämplig uppskattning av felet. (Du kan anta att du har funktionen `trapets(f, int, h)` även om du inte har gjort uppgift a))

- (6p) **c)** Anta att integranden $f(x)$ är en glatt funktion (alla derivator är kontinuerliga) då har trapetsmetoden $T(h)$ noggrannhetsordning 2, dvs trunkeringsfelet är $E = |T(h) - I| \leq Ch^2$. För att visa detta kan felet skrivas som en summa av trunkeringsfel E_k från varje delintervall $[x_k, x_{k+1}] \subset [a, b]$, så kallade lokala trunkeringsfel:

$$E_k = \left| \frac{h}{2}(f(x_k) + f(x_{k+1})) - \int_{x_k}^{x_{k+1}} f(x)dx \right|$$

Visa att $E_k \leq Ch^3$, där C är en konstant oberoende av $h = x_{k+1} - x_k$.

P3. Vi önskar lösa begynnelsevärdesproblemet

$$u'(t) = -\sin(\alpha u(t) + \beta), \quad u(0) = 2, \quad (1)$$

där $\alpha = 10$ och $\beta = 1/8$.

- (3 p) **a)** Formulera metoden Euler Bakåt (Implicit Euler) för ekvation (1).
- (6 p) **b)** I varje tidssteg med Bakåt Euler behöver en icke-linjär ekvation lösas. Formulera Newtons metod för denna ekvation. Specificera en lämplig startpunkt för Newtoniterationerna i varje tidssteg.
- (6 p) **c)** Skriv ett Matlab-program som beräknar en approximation av $u(3)$, dvs. lösningen till ekvation (1) vid tiden $t = 3$. Matlabprogrammet ska använda Bakåt Euler för tidsdiskretiseringen och Newton i varje tidssteg, och ska skriva ut approximationen av $u(3)$ på skärmen. För full poäng krävs att Newtons metod är implementerad med kontroll av felet i varje iteration så att absolutbeloppet av felet är mindre än en given tolerans, tex 10^{-3} . Det får antas känt att Newtoniterationerna i varje tidssteg konvergerar.