**1c)**

e\_mach\*konditionstalet hos matrisen ger felet enligt sauer p.88-89. Normen av residualen som beräknas, alltså felet. Vi får beräknat e\_mac =~10^-16, vilket är konsekvent med felet som specificeras för decimal64 lagringsprotokollet. Störningen är maskinfelet, konditionstalet är felförstärkningsfaktorn.

**2a)**

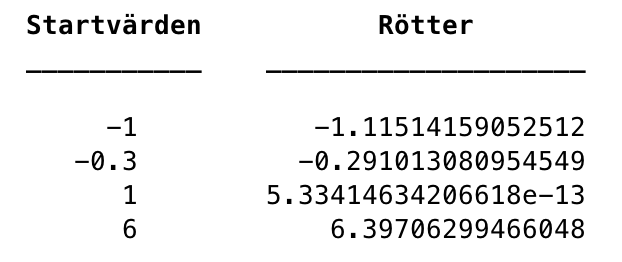
| **f()** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 61x | 61 | 122 | 183 | 244 | 305 | 366 | 427 |
| ((-)/(-))^7 | 0.0086 | 0.35 | 3.64 | 20.61 | 82.04 | 259.2 | 695.0 |
| x\*e^x | 7.3 | 5.4 | 2.99 | 1.5 | 0.67 | 0.30 | 0.13 |

Från studium av funktionerna samt bekräftelse genom några beräknade värden i tabellen är x\*e^x insignifikant vid större värden och polynomkvoten insignifikant vid småvärden. Den minsta positiva roten verkar vara kring 0 och den största roten borde ges när polynomkvoten är lika med den linjära ekvationen, eftersom x\*e^x är insignifikant. För att förenkla: dominerande termen i täljaren delat på dominerande termen i nämnaren. ~(x/3)^7 = 61x ⇒ x ~ 5.1.

Använda medelvärdessatsen för att grovlokalisera rötter. 8

Behöver beräknas

**2c)**

****

**2d)**

1.24 i Sauer ger:

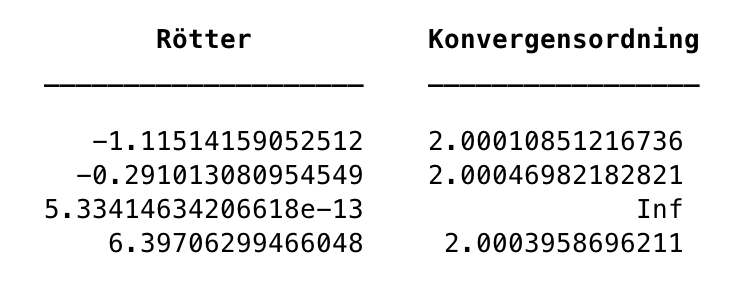
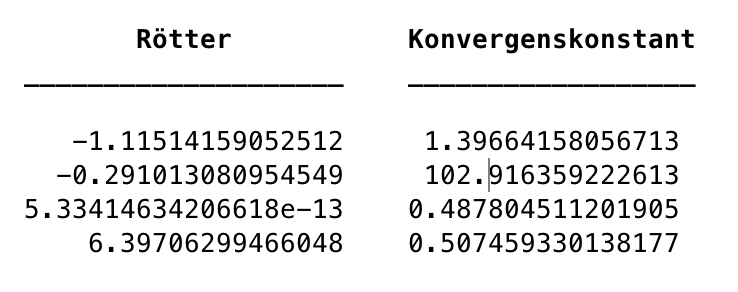
e(i+1) = e(i)^2 \* |f ‘ ‘ (c(i)) / 2 \* f ‘ (x(i)) |

där c(i) motsvarar ett värde mellan x(i) och roten r

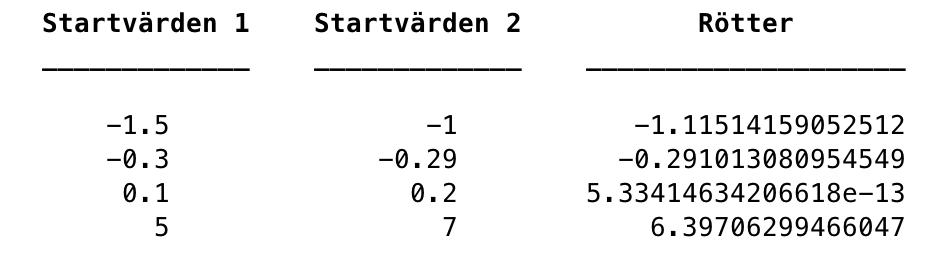
Tumregeln blir:

e(i+1) = M \* e(i)^2

**2e)**

****

**3b)**

****

**3c)**

Sauer s. 61

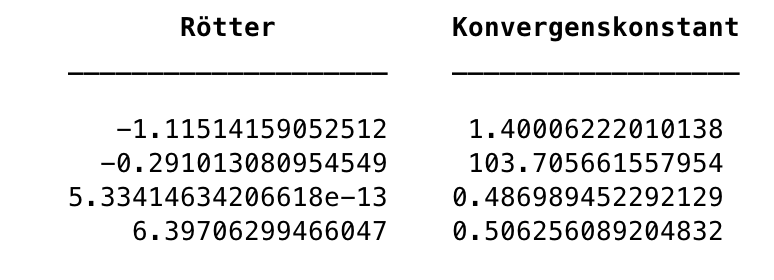
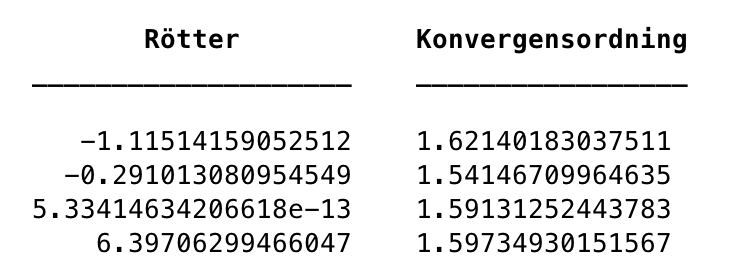
error(i+1) = | f ‘ ‘ (r) / 2\*f ‘ (r) | \* error(i)\*error(i-1)

Vilket ger tumregeln:

K = error(i) / error(i-1)\*error(i-2)

Sekantmetodens konvergens är mellan linjär och kvadratisk, vilket innebär mellan 1 och 2. Kring 1.62

**3d)**

****

**3e)**

newton kommer fram till rätt svar efter färre antal iterationer och har lättare att hitta de flesta rötter.

**4a)**

Splines koefficienter = 4\*12 = 48

**4b)**

De fyra ansatserna som krävde 3 koefficienter är D), E), F), G).

D), E), F) är ett andragradspolynom y = c1 + c2\*x+c3\*x^2

G) är givet som y = c1 + c2\*cos(w\*x) + c3\*sin(w\*x)

**4c)**

Metod C), SPLINES är bäst pga den använder data från hela året och splines använder kubisk polynom. Soltiden blir 1111.4 min, dag 174 vilket motsvarar 22 Juni 2024

**4d)**

Metod G) är bäst pga periodtiden är w = 2\*pi / 365 som tar hänsyn till årets alla dagar och periodiciteten hos soltimmarna.

Soltiden blir 1091.1 min, dag 184 vilket motsvarar 1 Juli 2024

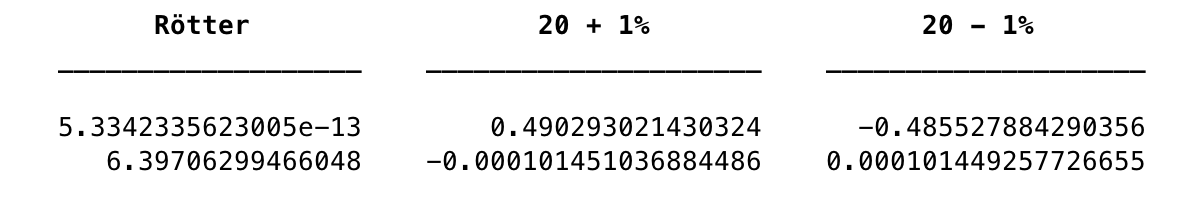
**4e)**

Genom att modifiera Ansats D) och istället använda mätpunkter nära julafton, vilket i detta fall 1 dec, 31 dec, 1 jan. Därmed får vi en effektiv metod med 3 koefficienter.

**4f)**

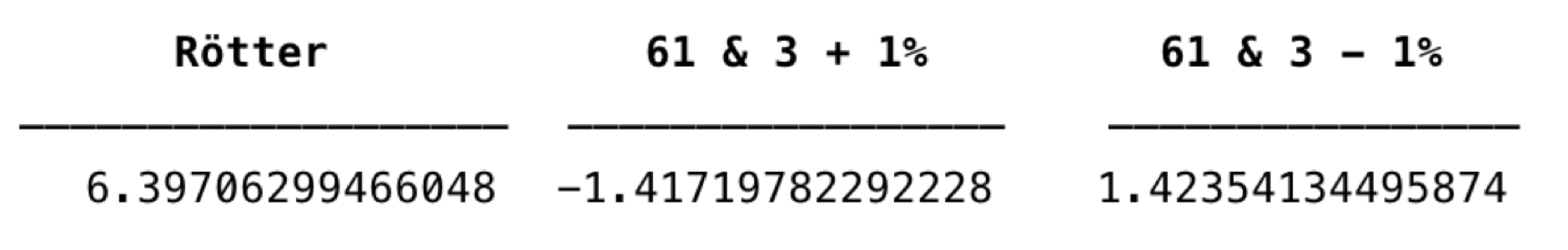
Splines är antagligen generellt den bästa eftersom den har flest koefficienter och alltså bäst kan anpassas efter datan. Däremot är det ett stort arbete att ta fram alla koefficienterna, och kan således vara olämpligt om det finns många datapunkter, eller om det måste göras med begränsad datorkraft. För täta dataset är minsta kvadratmetoden bra. Den viktigaste är att olika metoder är bra för olika tillämpningar. Ju bättre modell man har, desto mindre beräkningar behöver man göra. Den periodiska ansatsen är bra eftersom vi vet att soltiden är periodisk över året. Den kanske skulle kunna sägas vara bäst eftersom det är förhållandevis få beräkningar och modell välgrundat i verkligheten.

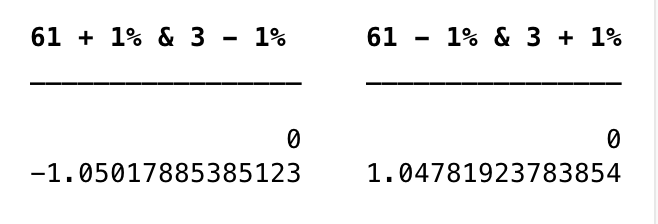
**5a)**

****

Minsta roten ökar med 0.5 % och den största roten minskar minimalt. De påverkas ej åt samma håll.

**5c)**

****

****

**6b)**

Vi valde övre gräns till 1000 då det gav 9 korrekta decimaler.

**6c)**

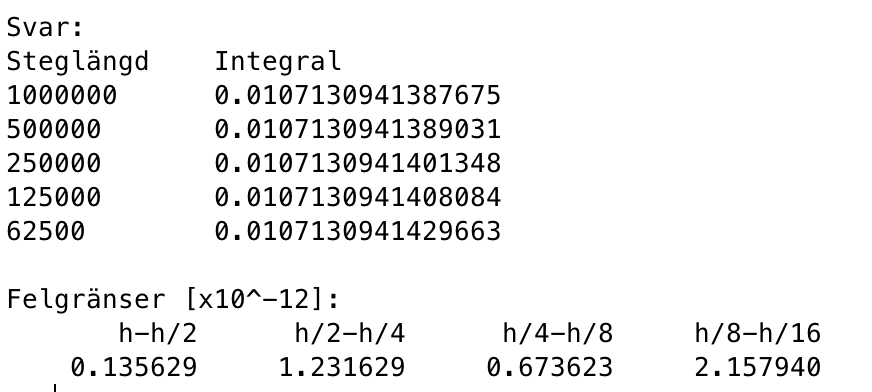
Eliminera genom att beräkna gränsvärdet när funktionen går mot noll vilket ger

B = 1/625

**6d)**

Trapetsregeln kräver att integralen är jämn. Då darrandet är eliminerat behövs inga fler förberedelser.

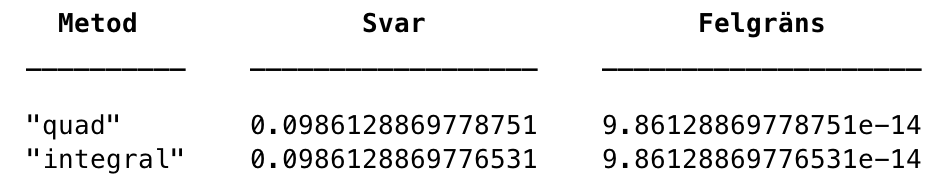
**6e)**

****

**6f)**

Felgränsen beräknas genom att beräkna integralen med olika steglängder och ta skillnaden mellan dem. Desto mindre steglängden blir, desto större blir felgränsen.

**7a)**

****

**7b)**

Genom att börja med en låg feltolerans och sedan öka denna för att få ett mer exakt svar kan man spara tid på arbetet. Vidare genom att dela upp funktionen där det sker snabba förändringar, exempelvis genom att notera asymptoter.