⊞— ∫(f,a,b,n).. Simpson/Romberg Kernel

Identify each component

$$\begin{cases} \rho_a := 7850 \\ \alpha_c := 25 \\ \sigma := \frac{5.67}{10^8} \end{cases}$$?

$$\begin{cases} \Phi := 1.0 \\ \varepsilon_m := 0.7 \\ \varepsilon_f := 1.0 \end{cases}$$?

$$[k_{sh}] = [0.5888]$$

$$\begin{cases} ratio_{Am.b\_V} := 70 \cdot 10^2 \\ ratio_{Am\_V} := 107 \cdot 10^2 \\ k_{sh} := 0.9 \cdot \frac{ratio_{Am.b\_V}}{ratio_{Am\_V}} \end{cases}$$?

$$\theta_g(t) := 20 + 345 \cdot \log10(8 \cdot t + 1)$$

$$C_a := \begin{bmatrix} 425 + \frac{7.73}{10^1} \cdot \theta_a - \frac{1.69}{10^3} \cdot \theta_a^2 + \frac{2.22}{10^6} \cdot \theta_a^3 \\ 666 + \frac{13002}{738 - \theta_a} \\ 545 + \frac{17820}{\theta_a - 731} \\ 650 \end{bmatrix}$$

system segments specifications

$$join := \begin{bmatrix} 0 \\ 24.365 \\ 35.8 \\ 54.7 \\ 120 \end{bmatrix} \quad init := \begin{bmatrix} 20 \\ 600 \\ 735 \\ 900 \end{bmatrix} \quad pts := \begin{bmatrix} 200 \\ 100 \\ 100 \\ 1000 \end{bmatrix}$$

original 24.4 optimized ~24.365

collect the piecewise DE ... export vector algo style d(t,θ.a)

for $i \in 1 .. length(C_a)$

$$d_i := \frac{k_{sh}}{C_{a_i} \cdot \rho_a} \cdot ratio_{Am\_V} \cdot \left(\alpha_c \cdot \left(\theta_g(t) - \theta_a\right) + \Phi \cdot \varepsilon_m \cdot \varepsilon_f \cdot \sigma \cdot \left(\left(\theta_g(t)\right)^4 - \theta_a^4\right)\right)$$

$d(t, \theta_a) := d$ solving vector

collect the piecewise DE solutions from incremented "for loop"

$(s := 1) \equiv$ "auto-incrementing counter, accumulates sols in Res[s]"

for $i \in 2 .. length(join)$

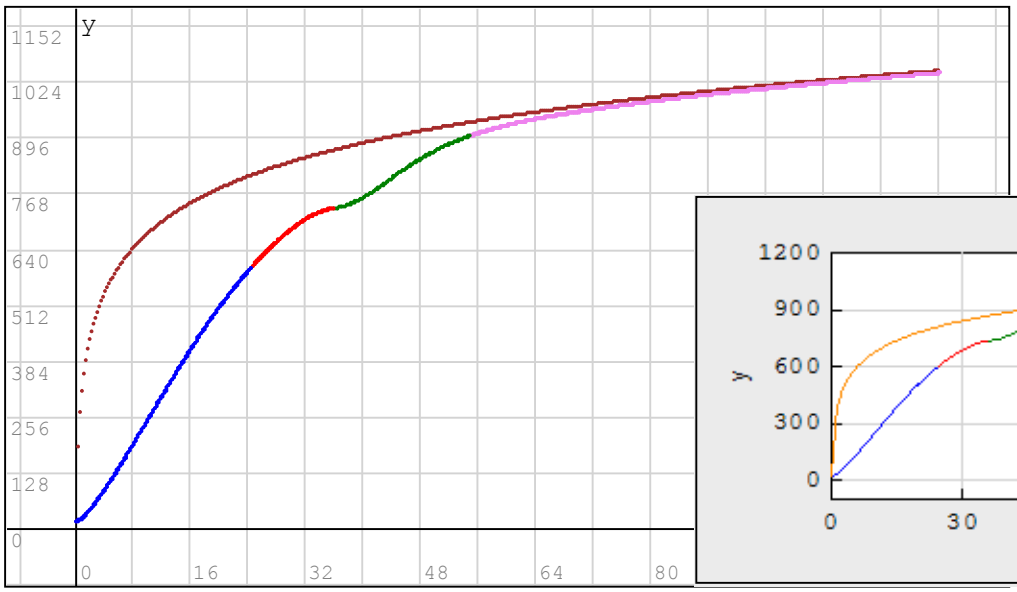
$Res_s := dn\_GearsBDF(init_s, join_s, join_i, pts_s, d(t, \theta_a)_s)$

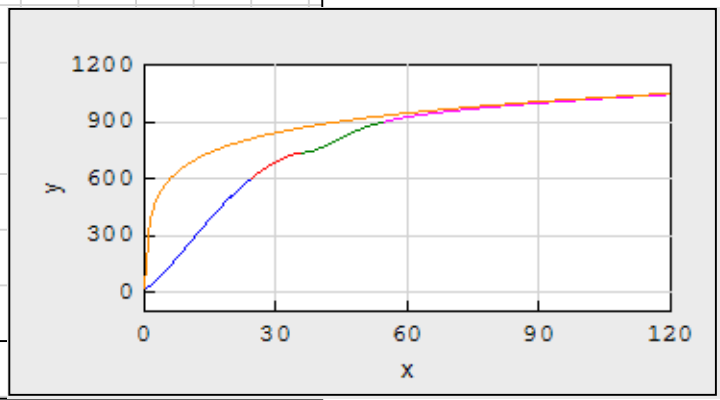
$s := s + 1$

$t0 := time(1)$

$time(1) - t0 = 1.7\ s$

$$plot := \begin{cases} Res_1 \\ Res_2 \\ Res_3 \\ Res_4 \\ "" \\ \theta_g(x) \quad join_1 \le x \le join_5 \end{cases}$$





strip from repeated junctions

```
res_2 := submatrix(Res_2, 2, rows(Res_2), 1, 2)
res_3 := submatrix(Res_3, 2, rows(Res_3), 1, 2)
res_4 := submatrix(Res_4, 2, rows(Res_4), 1, 2)
global := stack(Res_1, res_2, res_3, res_4)
Global := augment(col(global, 1), col(global, 2))
```

$$global = \begin{bmatrix} 0 & 20 \\ 0.12 & 20.31 \\ 0.24 & 21.05 \\ 0.37 & 22.08 \\ 0.49 & 23.32 \\ 0.61 & 24.74 \\ 0.73 & 26.31 \\ \vdots & \end{bmatrix}$$

$f(x) := linterp(col(global, 1), col(global, 2), x)$

```
inc(Low, step, High, fnct(■)) := | U := Low, step .. High
                                 | for i ∈ 1 .. rows(U)
                                 |   f_i := eval(fnct(U_i))
                                 | augment(U, f)
```

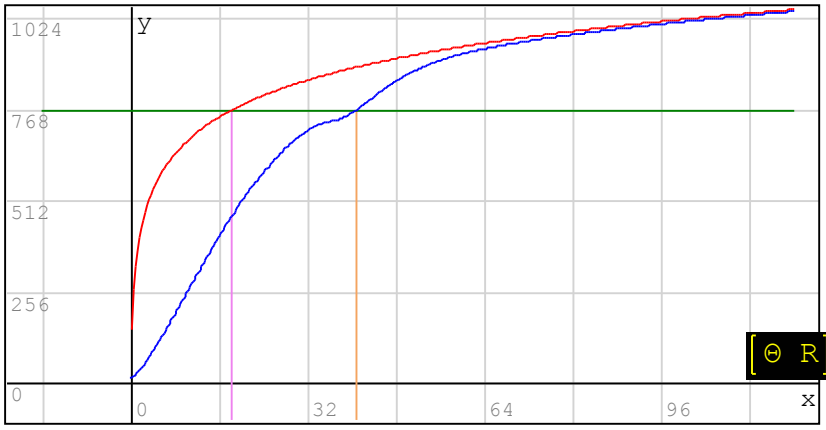
recast for regular tabulation

$New := inc(0, 0.25, 120, f(x))$

$$New = \begin{bmatrix} 0 & 20 \\ 0.25 & 21.11 \\ 0.5 & 23.47 \\ 0.75 & 26.57 \\ 1 & 30.2 \\ 1.25 & 34.24 \\ 1.5 & 38.61 \\ 1.75 & 43.27 \\ 2 & 48.17 \\ 2.25 & 53.29 \\ 2.5 & 58.59 \\ 2.75 & 64.05 \\ 3 & 69.66 \\ 3.25 & 75.4 \\ 3.5 & 81.26 \\ 3.75 & 87.22 \\ 4 & 93.28 \\ 4.25 & 99.43 \\ 4.5 & 105.65 \\ \vdots & \end{bmatrix}$$

$user := 768$

```
Θ := solve(θ_g(x) - user, x, 0, 120)
R := solve(f(x) - user, x, 0, 120)
```

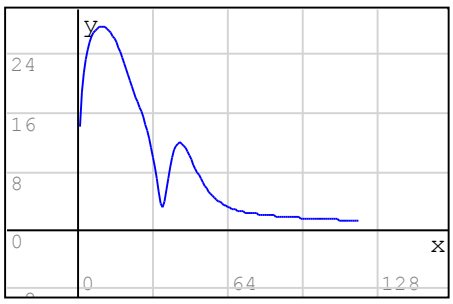


[Θ R]=[18.3 40.8]

$h := 0.1$

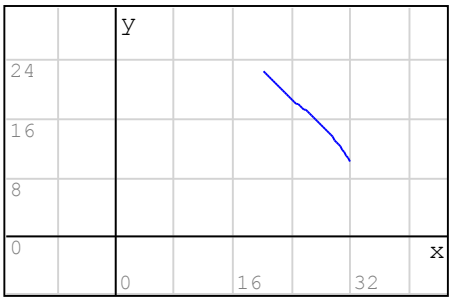
$$D1(x, h, f) := \frac{-f(x+2\cdot h)+8\cdot f(x+h)-8\cdot f(x-h)+f(x-2\cdot h)}{12\cdot h}$$

$$\lambda(x) := \begin{cases} 1 & \text{if } (0 \le x) \wedge (x \le 120) \\ "" & \text{otherwise} \end{cases}$$



$D1(x, h, f)\cdot\lambda(x)$

$$\lambda(x) := \begin{cases} 1 & \text{if } (20 \le x) \wedge (x \le 32) \\ "" & \text{otherwise} \end{cases}$$



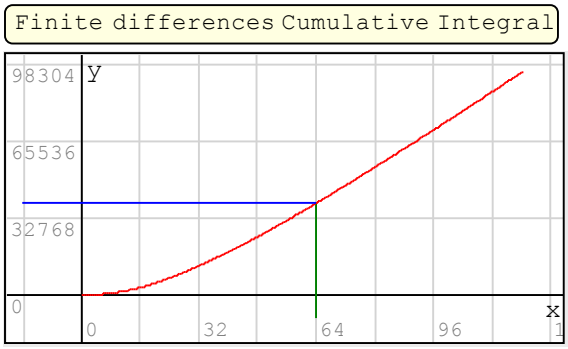
$D1(x, h, f)\cdot\lambda(x)$

$$\int_{0}^{10} f(x)\,dx = 1233 \qquad \int(f, 0, 10, 20) = 1233 \qquad \text{rows}(New) = 481$$

```
Σ(data) := | "Finite differences Cumul Integration"
           | vx := col(data, 1)
           | vy := col(data, 2)
           | Y_1 := 0
           | for i ∈ 2 .. rows(vx)
           |     Δx_i := vx_i - vx_(i-1)
           |     Y_i := Y_(i-1) + (1/2)·((vy_(i-1) + vy_i)·Δx_i)
           | augment(vx, Y)
```

data := New desired := 256

track := row(Σ(data), desired) = [64 39114]



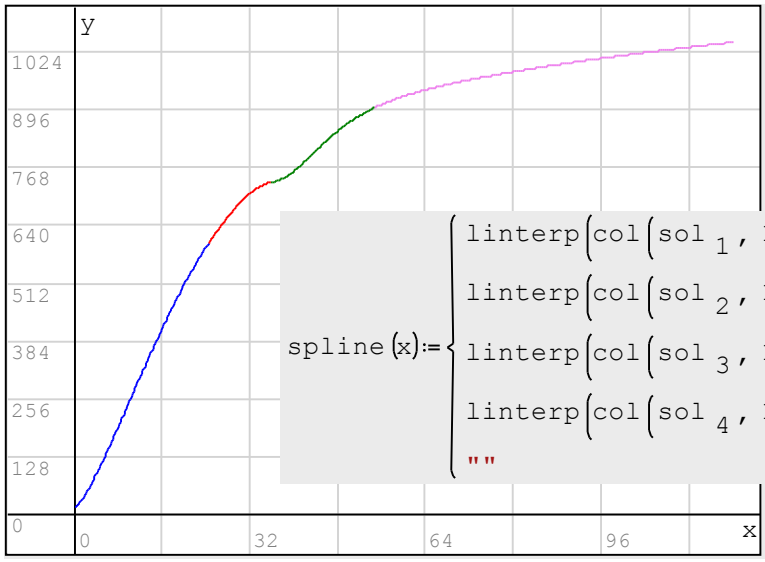
QA [Quality Assurance] requirements

$$d(t,\theta_a):=\begin{cases}\dfrac{22167995058257\left(2500000000000\left(5\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)-\theta_a\right)+3969\cdot\left(625\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)^4-\theta_a{}^4\right)\right)}{55244051176926200\left(500\cdot\left(100\cdot\left(425000+773\cdot\theta_a\right)-169\cdot\theta_a{}^2\right)+111\cdot\theta_a{}^3\right)} \\ \dfrac{21\cdot\left(738-\theta_a\right)\cdot\left(2500000000000\left(5\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)-\theta_a\right)+3969\cdot\left(625\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)^4-\theta_a{}^4\right)\right)}{157000000000000\left(2167+111\cdot\left(738-\theta_a\right)\right)} \\ \dfrac{63\cdot\left(-731+\theta_a\right)\cdot\left(2500000000000\left(5\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)-\theta_a\right)+3969\cdot\left(625\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)^4-\theta_a{}^4\right)\right)}{392500000000000\left(3564+109\cdot\left(-731+\theta_a\right)\right)} \\ \dfrac{63\cdot\left(2500000000000\left(5\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)-\theta_a\right)+3969\cdot\left(625\cdot\left(4+69\cdot\log10\left(1+8\cdot t\right)\right)^4-\theta_a{}^4\right)\right)}{5102500000000000}\end{cases}$$

$$sol:=\begin{cases}dn\_GearsBDF\left(20,0,24.365,120,d\left(t,\theta_a\right)_1\right) \\ dn\_GearsBDF\left(600,24.365,35.8,120,d\left(t,\theta_a\right)_2\right) \\ dn\_GearsBDF\left(735,35.8,54.7,120,d\left(t,\theta_a\right)_3\right) \\ dn\_GearsBDF\left(900,54.7,120,1200,d\left(t,\theta_a\right)_4\right)\end{cases}$$

$$join:=\begin{bmatrix}0\\24.365\\35.8\\54.7\\120\end{bmatrix}$$

$$init:=\begin{bmatrix}20\\600\\735\\900\end{bmatrix}$$

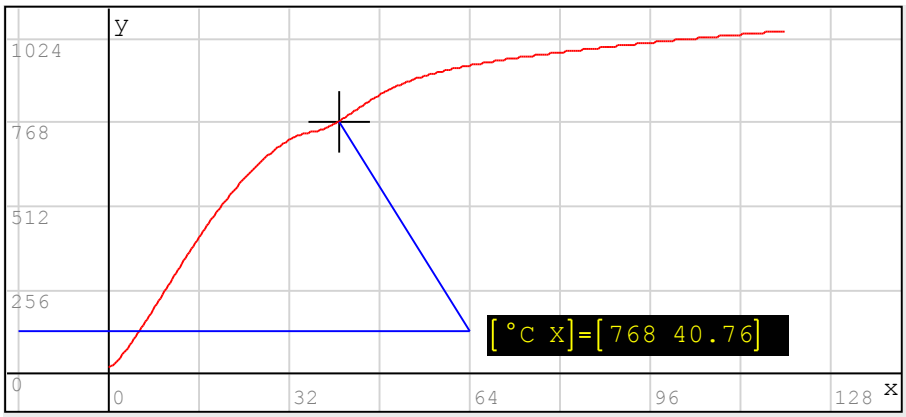


$$spline(x):=\begin{cases}linterp\left(col\left(sol_1,1\right),col\left(sol_1,2\right),x\right) & \text{if } (0\le x)\wedge(x\le 24.365) \\ linterp\left(col\left(sol_2,1\right),col\left(sol_2,2\right),x\right) & \text{if } (24.365\le x)\wedge(x\le 35.8) \\ linterp\left(col\left(sol_3,1\right),col\left(sol_3,2\right),x\right) & \text{if } (35.8\le x)\wedge(x\le 54.7) \\ linterp\left(col\left(sol_4,1\right),col\left(sol_4,2\right),x\right) & \text{if } (54.7\le x)\wedge(x\le 120) \\ "" & \text{otherwise}\end{cases}$$

°C := 768

X := solve(spline(x) - °C, x, 0, 120) = 40.76

$$\text{pointer} := \begin{bmatrix} -16 & 128 \\ 64 & 128 \\ 64 & 128 \\ X & °C \end{bmatrix}$$



$$\begin{cases} \text{pointer} \\ \text{spline}(x) \\ [X\ °C\ "+"\ 40\ "black"] \end{cases}$$

Solving differential equation Eurocode 1993-1-2 4.2.5 Steel temperature

$$\begin{cases} 425 + \frac{7.73}{10^1}\cdot\theta_a - \frac{1.69}{10^3}\cdot\theta_a^2 + \frac{2.22}{10^6}\cdot\theta_a^3 & \text{if } (20 \le \theta_a) \wedge (\theta_a \le 600) \\ 666 + \frac{13002}{738 - \theta_a} & \text{if } (600 \le \theta_a) \wedge (\theta_a \le 735) \\ 545 + \frac{17820}{\theta_a - 731} & \text{if } (735 \le \theta_a) \wedge (\theta_a \le 900) \\ 650 & \text{otherwise} \end{cases}$$

is it that only those change from Euro codes to codes ?

$$\begin{bmatrix} k_{sh} \\ \rho_a \\ ratio_{Am\_V} \\ \alpha_c \\ \Phi \cdot \varepsilon_m \cdot \varepsilon_f \cdot \sigma \end{bmatrix} = \begin{bmatrix} 0.5888 \\ 7850 \\ 10700 \\ 25 \\ 3.969 \cdot 10^{-8} \end{bmatrix}$$

Ctrl +. to evaluate
export in QA sanity

$$d(t, \theta_a)$$