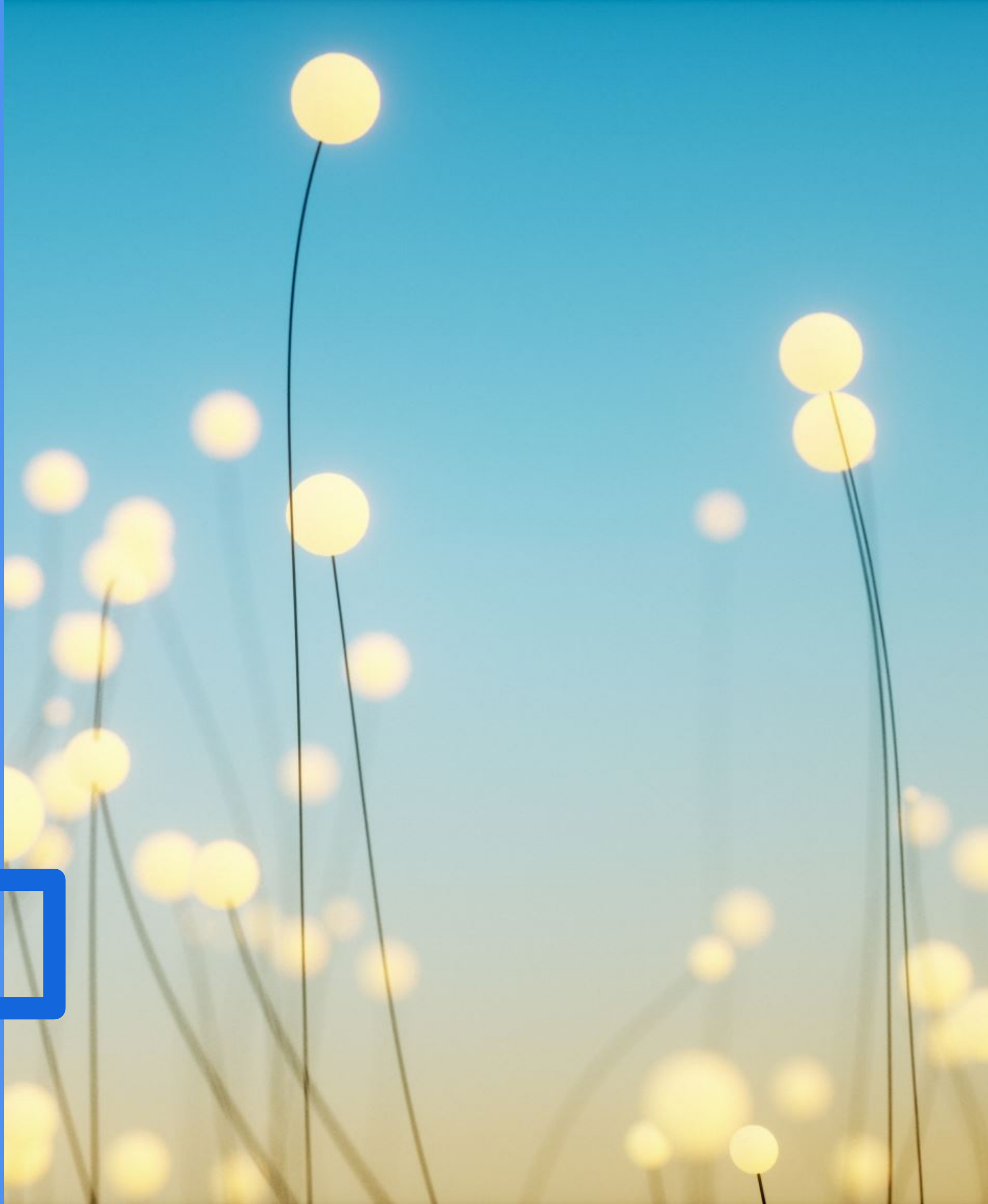
Introduction to computational plasticity using FORTRAN

Youngung Jeong

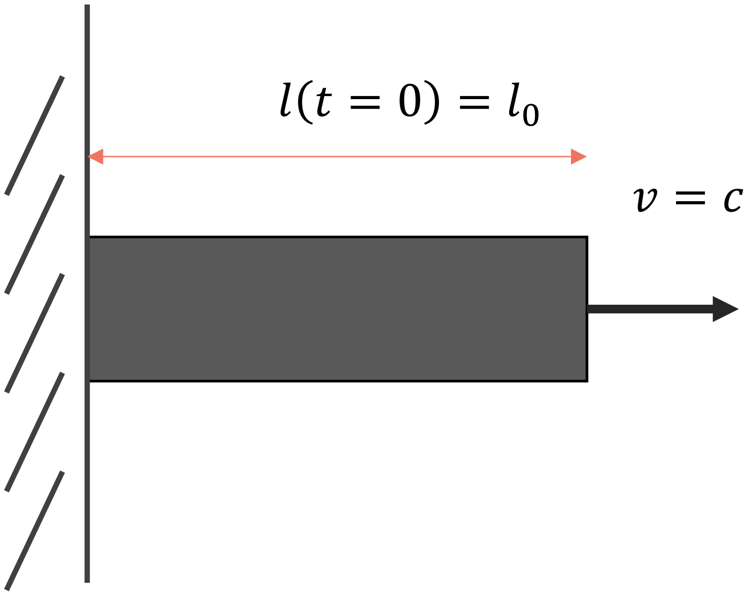
Changwon National Univ.



One dimensional elastic rod

- $d\varepsilon^{el} = \frac{dl}{l}$ with $\frac{dl}{dt} = c$
- Elastic constitutive law:
- $\mathbb{E}\varepsilon^{el} = \sigma$ (elastic stiffness $\mathbb{E} = 200\ [\mathrm{GPa}]$)
- Assuming $\mathbb{E}$ is constant, the above constitutive law can be expressed as:
- $d\varepsilon^{el} = \frac{1}{\mathbb{E}}\ d\sigma$

L vs σ?



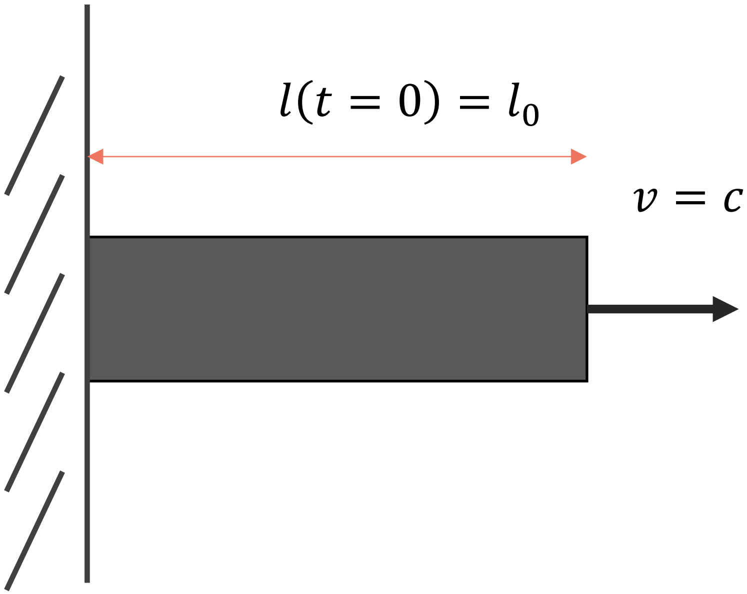
$$\frac{dv}{dt} = 0$$

One dimensional elastic rod

- $d\varepsilon^{el} = \frac{dl}{l}$ with $\frac{dl}{dt} = c$
- $d\varepsilon^{el} = \frac{1}{\mathbb{E}}\, d\sigma$
- $l = l_0 + ct$
- $\frac{1}{\mathbb{E}} d\sigma = \frac{dl}{l} \rightarrow \int_0^{\sigma} \frac{1}{\mathbb{E}} d\sigma = \int_{l_0}^{l} \frac{dl}{l} \rightarrow \frac{1}{\mathbb{E}} \sigma = \ln\left(\frac{l}{l_0}\right)$

$$\sigma = \mathbb{E} \ln\left(\frac{l}{l_0}\right)$$

σ vs t ?



$$\frac{dv}{dt} = 0$$

One dimensional elastic rod

$$d\varepsilon^{el} = \frac{dl}{l} \text{ with } \frac{dl}{dt} = c$$

$$d\varepsilon^{el} = \mathbb{E}\, d\sigma$$

- $l = l_0 + ct$

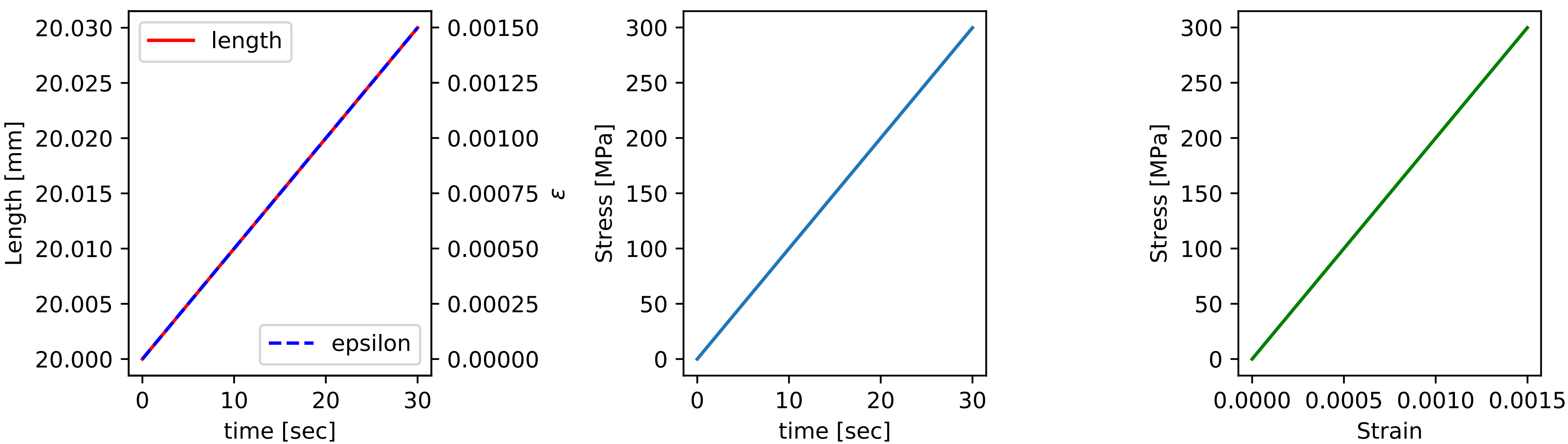
$$\frac{1}{\mathbb{E}} d\sigma = \frac{cdt}{l_0 + ct} \rightarrow \int_0^\sigma \frac{1}{\mathbb{E}} d\sigma = c \int_0^t \frac{dt}{l_0 + ct}$$

$$\rightarrow \frac{1}{\mathbb{E}} \sigma = \ln(l_0 + ct) - \ln(l_0)$$

$$\sigma = \mathbb{E} \ln\left(\frac{l_0 + ct}{l_0}\right)$$

Confirm following results:

- $l_0 = 20[mm]$
- $\frac{dl}{dt} = 0.001\ [mm]$
- $\mathbb{E} = 200\ [GPa] \quad (\text{equivalently}, 200000\ MPa)$
- Loading for 30 seconds

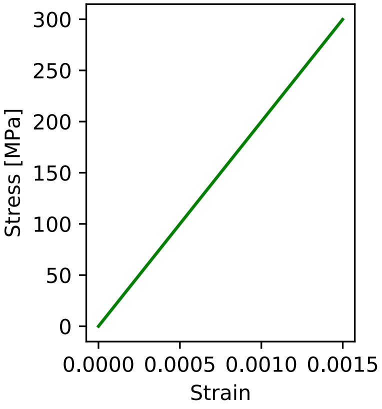


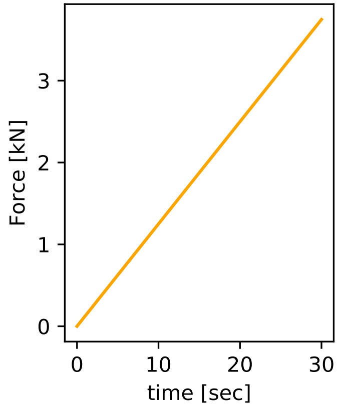
Now that you have stress, you could estimate the force as well.

- Let's say, the thickness is 1 [mm] and the width is 12.5 [mm]; The cross-sectional area amounts to 12.5 [mm^2]

$$F = A \cdot \sigma = A \,\mathbb{E}\, \ln\left(\frac{l_0 + ct}{l_0}\right)$$

$$[A \cdot \sigma] = [mm^2\ MPa] = \left[(10^{-3}m)^2\ 10^6 \frac{N}{m^2}\right] = [N]$$



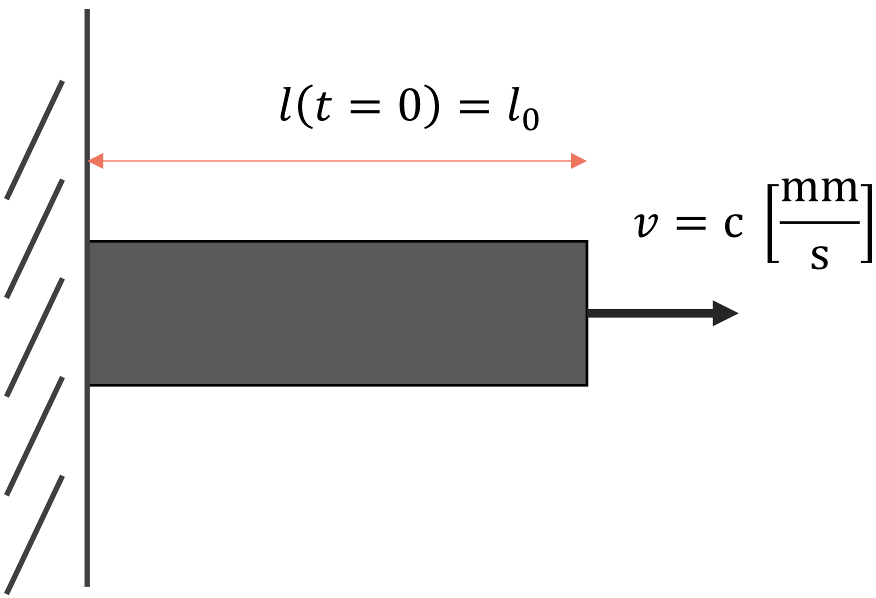


One dimensional Newtonian rod

- $d\varepsilon = \frac{dl}{l}$ with $\frac{dl}{dt} = c$
- Newtonian fluid's constitutive law:
- $\eta \frac{d\varepsilon}{dt} = \sigma$
- Assuming η is constant (Newtonian fluid), the above constitutive law can be expressed as:
- $\eta \frac{d\varepsilon}{dt} = \sigma \rightarrow \eta \frac{\frac{dl}{l}}{dt} = \sigma \rightarrow \frac{\eta}{l} \frac{dl}{dt} = \sigma$

L vs σ?

The cross-sectional area amounts to 12.5 [mm^2]



$$\frac{dv}{dt} = 0$$

One dimensional Newtonian rod

$$\frac{\eta}{l}\frac{dl}{dt} = \sigma \qquad \frac{\eta}{l_0 + ct}c = \sigma$$

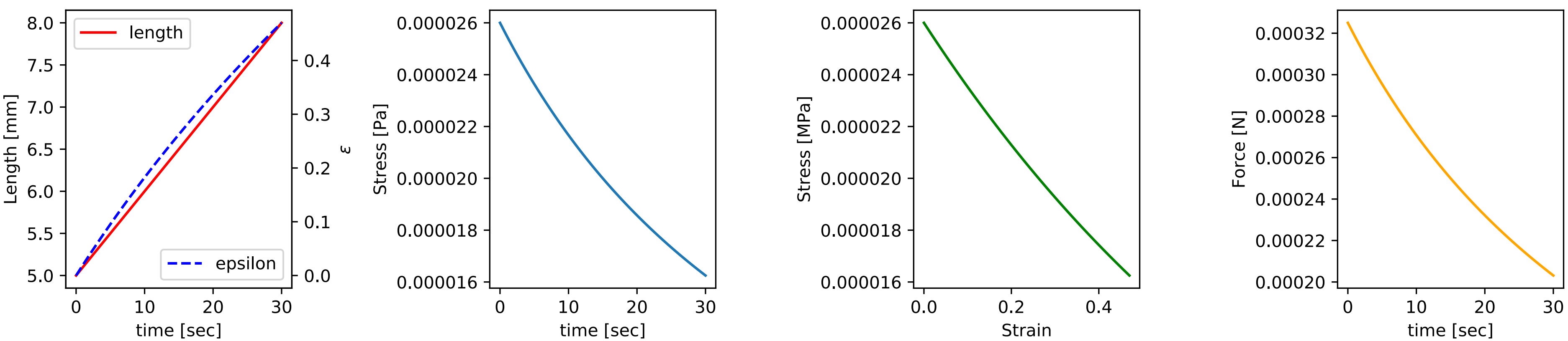
Water's viscosity is known as
$\eta = 1.3\times10^{-3}$ [Pa · s] at $10°C$

$$\left[\frac{\eta}{l}\frac{dl}{dt}\right] = \left[\frac{Pa\cdot s}{mm}\ \left[\frac{mm}{s}\right]\right] = [Pa]$$

Newtonian fluid (water) under tension

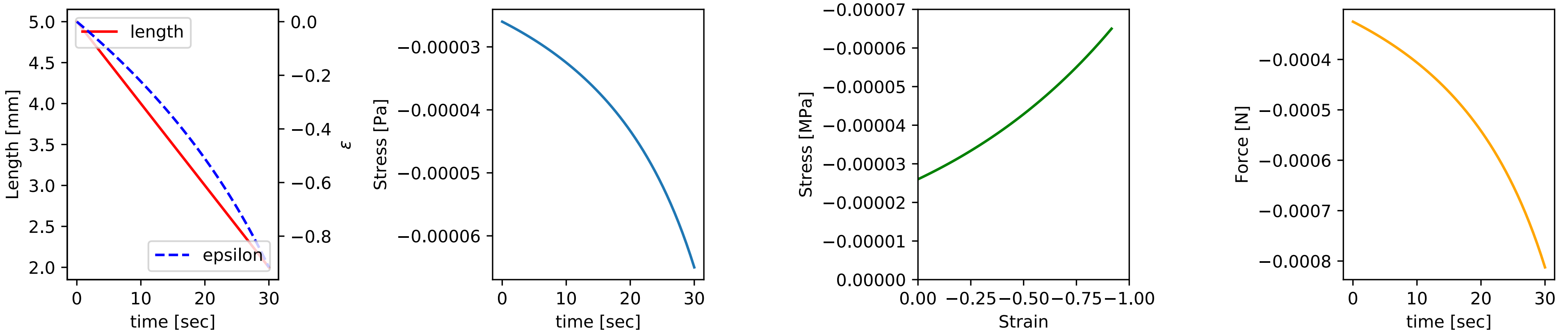
$$l_0 = 5.\,[mm]$$
$$\frac{dl}{dt} = 0.1\ \left[\frac{mm}{sec}\right]$$
$$\eta = 1.3\times10^{-3}\ [Pa\cdot s]$$

The cross-sectional area amounts to 12.5 [mm^2]



Newtonian fluid (water) under compression

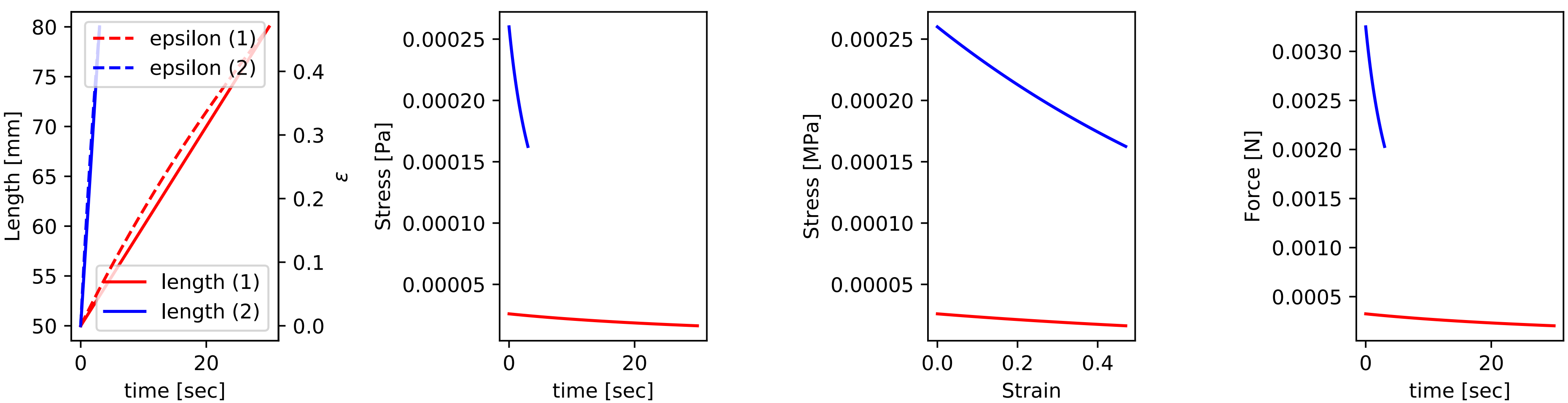
$$l_0 = 5.\,[mm]$$
$$\frac{dl}{dt} = -0.1\ \left[\frac{mm}{sec}\right]$$
$$\eta = 1.3\times10^{-3}\ [Pa \cdot s]$$



Let's compare two different speed of compression.

$$l_0 = 50\ [mm]$$
$$\eta = 1.3 \times 10^{-3}\ [Pa \cdot s]$$

$$\frac{dl}{dt} = -0.1\ \left[\frac{mm}{sec}\right] \qquad \frac{dl}{dt} = -1.0\ \left[\frac{mm}{sec}\right]$$



My Python cheat sheet

```
def eps(t,l0,vel):
    return np.log(l0+vel*t)-np.log(l0)
def length(t,l0,vel):
    return l0+vel*t
```

```
def elasticity(t,E,l0,c):
    return E*np.log((l0+c*t)/l0)
```

```
def newtonian(t,eta,l0,c):
    return eta*c/(l0+c*t)
```

Corn starch

https://youtu.be/Vx2DjGwnd44

A reasonable consideration (*VERY* phenomenological constitutive model)

Newtonian fluid's constitutive law:

$$\eta \frac{d\varepsilon}{dt} = \sigma$$

non-Newtonian fluid

$$\eta(\sigma) \frac{d\varepsilon}{dt} = \sigma$$

with

$$\eta(\boldsymbol{\sigma}) = \eta_0 + \alpha\boldsymbol{\sigma}$$

Continued

- $d\varepsilon = \frac{dl}{l}$ with $\frac{dl}{dt} = c$

$$\eta \frac{d\varepsilon}{dt} = \sigma \rightarrow (\eta_0 + \alpha\sigma)\frac{\frac{dl}{l}}{dt} = \sigma$$

$$\rightarrow \frac{\eta_0}{l}\frac{dl}{dt} + \frac{\alpha\sigma}{l}\frac{dl}{dt} = \sigma \qquad \rightarrow \frac{\eta_0 c}{l} + \frac{\alpha\sigma c}{l} = \sigma$$

$$\rightarrow \frac{\eta_0 c}{l} + \frac{\alpha c\sigma}{l} = \sigma \qquad \rightarrow \frac{\eta_0 c}{l} = \sigma\left(1 - \frac{\alpha c}{l}\right)$$

$$\rightarrow \frac{\eta_0 c}{(l - \alpha c)} = \sigma$$

Okay, analytical solution was easily found.

A reasonable consideration (*quite* demanding phenomenological constitutive model)

Newtonian fluid's constitutive law:

$$\eta \frac{d\varepsilon}{dt} = \sigma$$

non-Newtonian fluid

$$\eta(\sigma) \frac{d\varepsilon}{dt} = \sigma$$

with a viscosity that is exponentially related with stress?

$$\eta(\boldsymbol{\sigma}) = \eta_0 \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)$$

Continued

- $d\varepsilon = \frac{dl}{l}$ with $\frac{dl}{dt} = c$

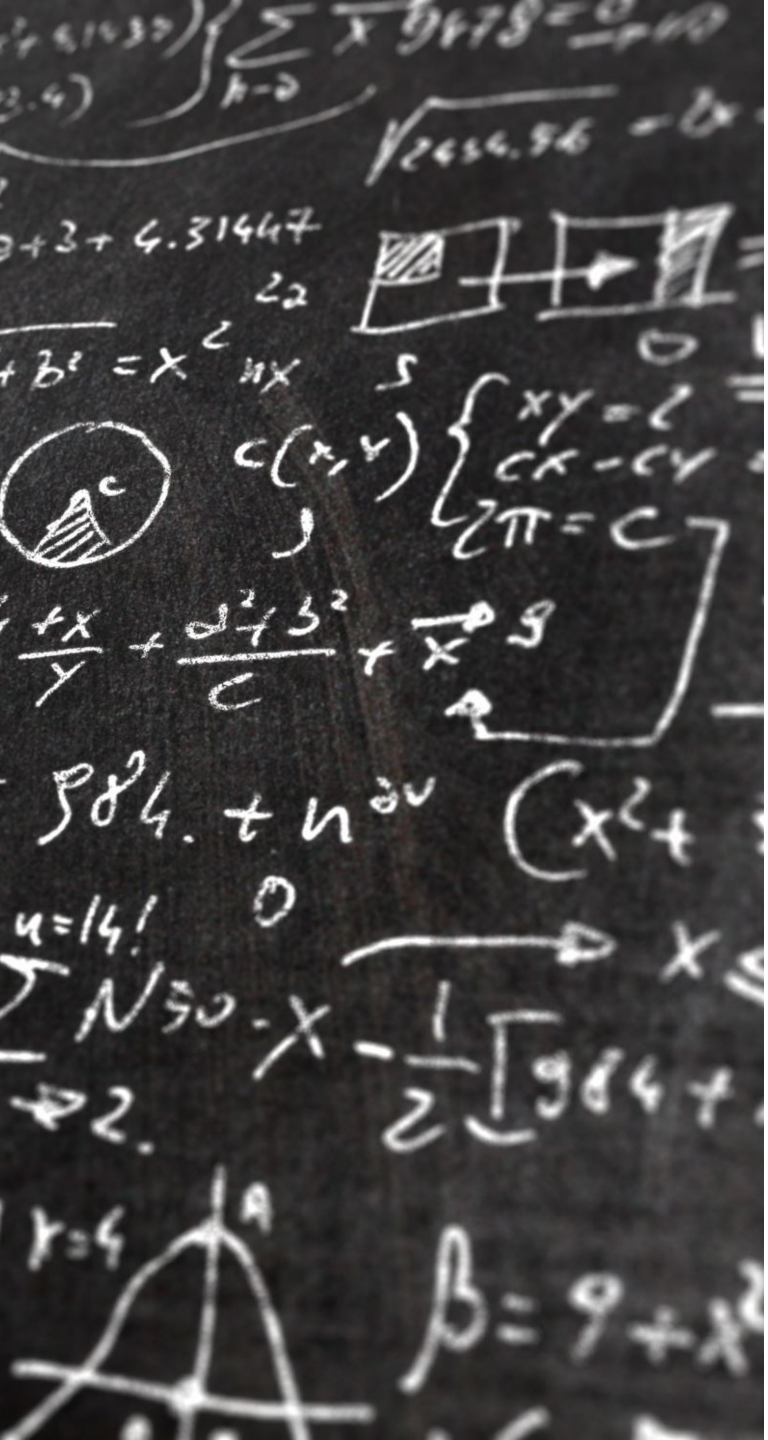
$$\eta(\sigma)\frac{d\varepsilon}{dt} = \sigma \rightarrow \eta_0 \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)\frac{\frac{dl}{l}}{dt} = \sigma$$

I gave up looking for the analytical solution of $\boldsymbol{\sigma}(l)$…

Numerical approach?

$$\eta_0 \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)\frac{d\varepsilon}{dt} = \sigma$$

$$d\varepsilon = \frac{dl}{l} \qquad \frac{dl}{dt} = c$$



Newton-Raphson method

How to solve?

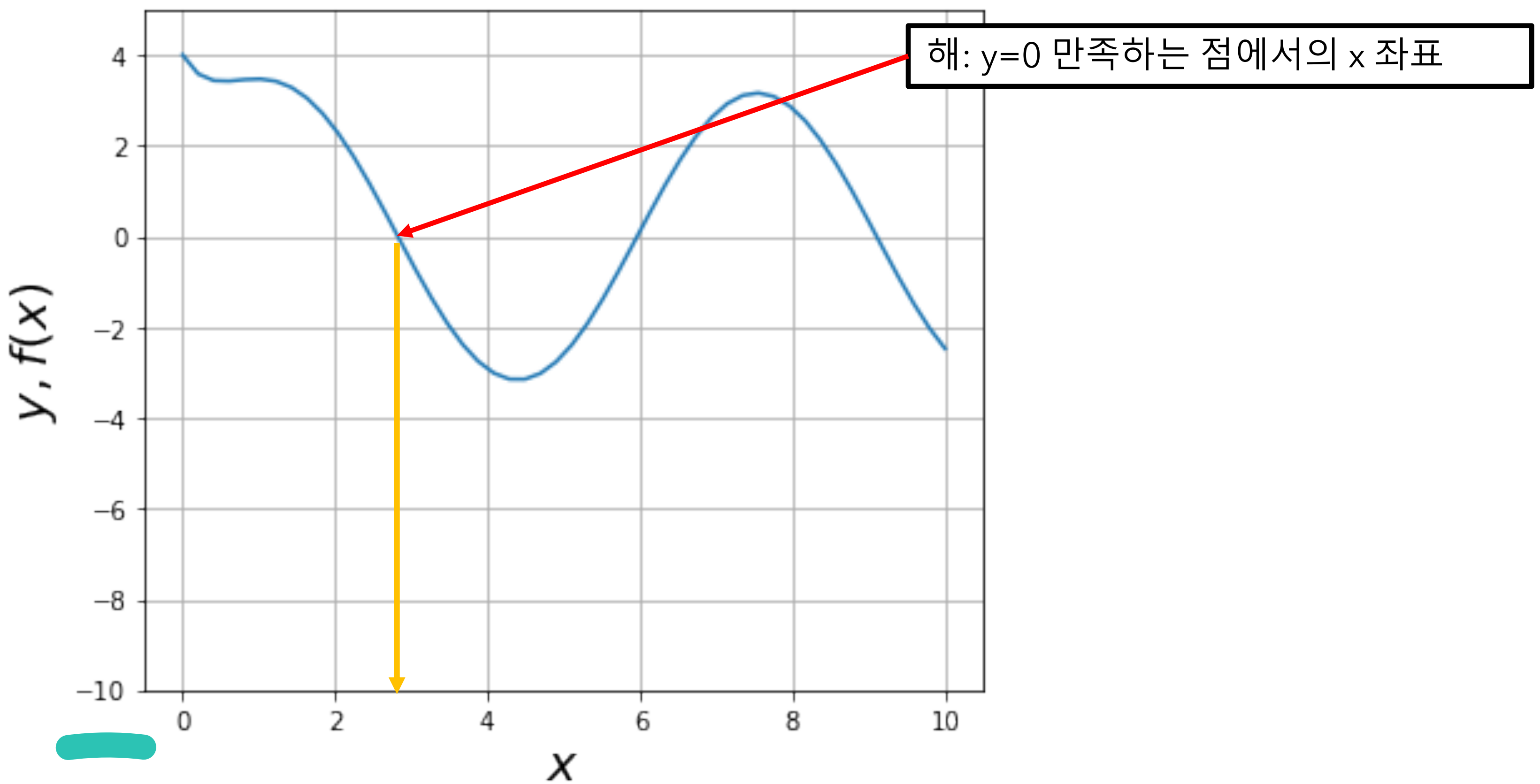
$$\eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{d\varepsilon}{dt} = \sigma$$

Our goal is to find a certain σ that gives

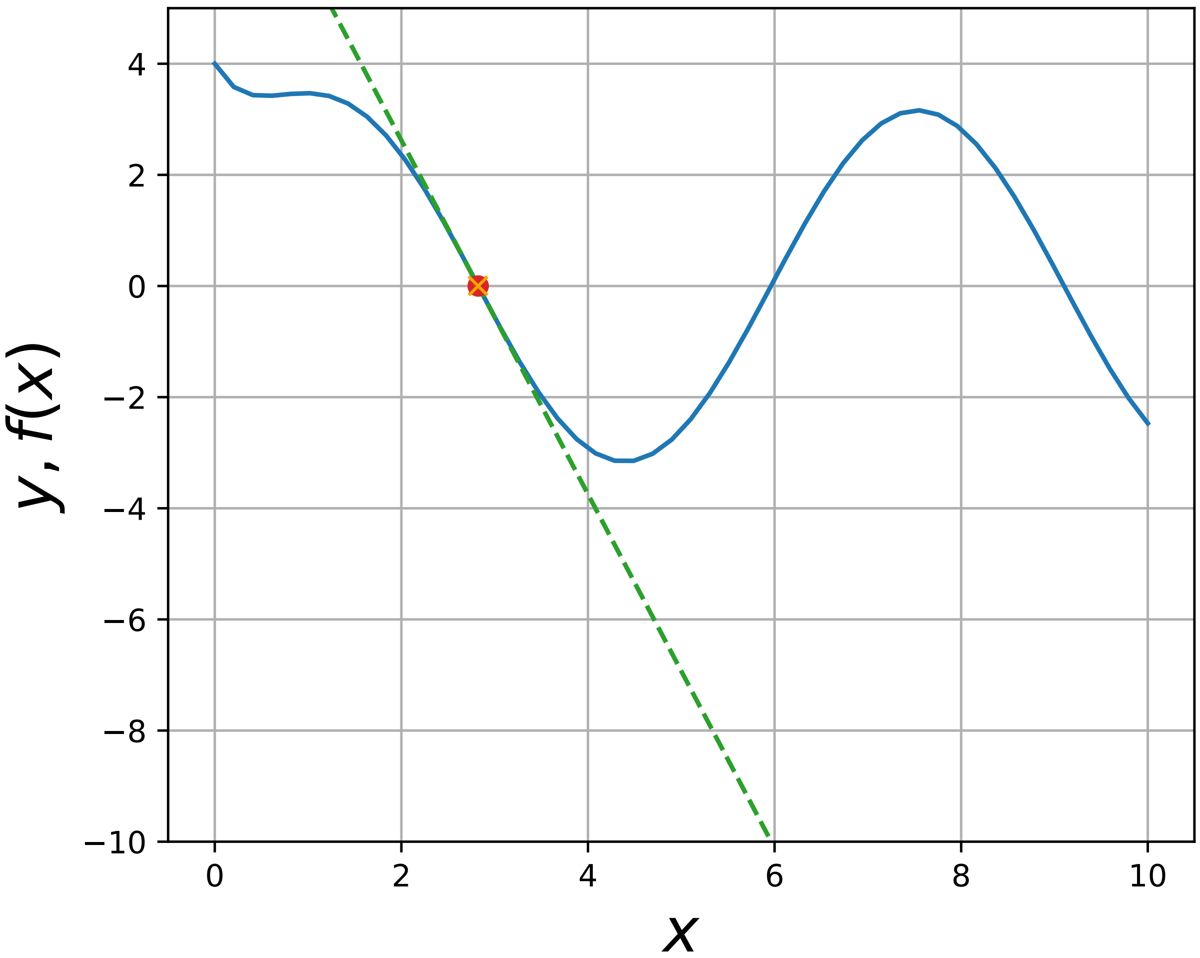
$$f(\sigma) = 0$$

Example NR

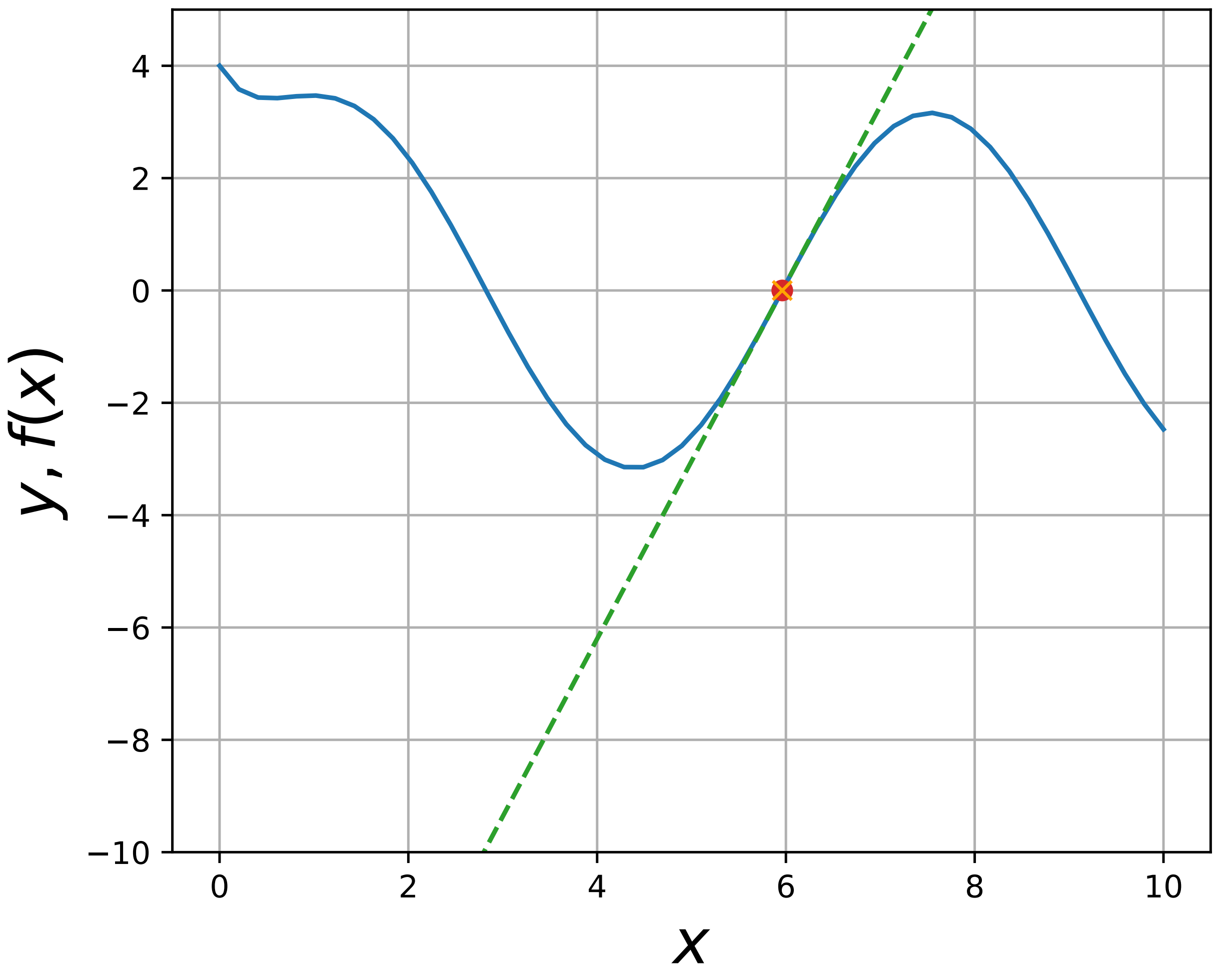
- $f(x) = y = \cos x + 3\sin x + 3\exp(-2x) = 0$ 의 해(즉, y=0 일때의 x값)를 찾아보자.



Visual illustration of NR (ex 1)



Visual illustration of NR (ex 2)



Newton Raphson Method - Algorithm

- 1. Guess x value and let's name it as x_0 where the subscript 0 means 'initial'.
- 2. Obtain new guess x_1 by following the below tasks.
  - Estimate $f(x_0)$ and $\frac{\partial f}{\partial x}$. In case $\frac{\partial f}{\partial x}$ is a function of x. For the first attempt, use x_0.
  - Obtain the next guess x_1 by drawing a tangent line at the point of $(x_0, f(x_0))$ and obtain its intercept with x-axis. You can do it by defining the line function derived from the tangent line, i.e.,

$$y = \frac{\partial f}{\partial x}(x_0)\times(x - x_0) + f(x_0)$$

Find the intercept of the line with x-axis, i.e., $y = 0$, which gives x_1:

$$0 = \frac{\partial f}{\partial x}(x_0)\times(x_1 - x_0) + f(x_0) \rightarrow x_1 - x_0 = -\frac{f(x_0)}{\frac{\partial f}{\partial x}(x_0)} \rightarrow x_1 = x_0 - \frac{f(x_0)}{\frac{\partial f}{\partial x}(x_0)}$$

- 3. We are using this intercept as the new x.

  And repeat 2-1/2-2 steps until $f(x_n) \approx 0$.

이 페이지를 확인하세요:
https://youngung.github.io/nr_example/

NR summary

- $x_{n+1} = x_n - \frac{f(x_n)}{\frac{\partial f}{\partial x}(x_n)}$
- Repeat the above until $f(x_n) < \text{tolerance}$
- Of course, you can do it manually, step-by-step. Usually, people make computer do the repetitive and tedious tasks.

How to solve?

Let's consider $\frac{d\varepsilon}{dt}$ is given as $\dot{\varepsilon}$

$$\eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{d\varepsilon}{dt} = \sigma$$

$$f(\sigma) = \sigma - \eta_0 \dot{\varepsilon} \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)$$

$$\frac{\partial f(\sigma)}{\partial \sigma} = 1 - \frac{\eta_0 \dot{\varepsilon}}{\alpha} \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)$$

Now, if you have a reasonable guess on σ (say, σ_0), let's estimate next guess σ_1 and so on.

$$\sigma_{n+1} = \sigma_n - \frac{f(\sigma_n)}{\frac{\partial f}{\partial x}(\sigma_n)}$$

Cheat sheet

```fortran
c     non newtonian
      real function calc_f(sigma, eta0, alpha, edot)
      implicit none
      real sigma, et0, alpha, edot, eta0
      calc_f = sigma - eta0 * edot * exp(sigma / alpha)
      return
      end function

c     ---------------
      real function calc_df(sigma, eta0, alpha, edot)
      implicit none
      real sigma, eta0, alpha, edot
      calc_df = 1. - eta0*edot/alpha * exp(sigma/alpha)
      return
      end function

c     ---------------
      program main
      implicit none
      real s, tol, calc_f, calc_df, f, df, edot,alpha ,eta0
      integer kount
      parameter(tol=1e-5)

c     Input conditions
      eta0 = 13.
      alpha = 2.
      edot = 1e-3
c     -- File
      open(3,file='nr.txt',status='unknown')
c     --
      s = 1.                          ! initial guess
      f = tol * 2.                    ! work-around
      kount = 0

c     -- Newton-Raphson loop
      do while(abs(f)>tol .and. kount < 10)
         f=calc_f(s,eta0,alpha,edot)
         df = calc_df(s,eta0,alpha,edot)
         write(3,'(i2.2,3e11.3)')kount, s, f, df
         s = s - f/df
         kount = kount + 1
      enddo

      close(3)
      end
```

Continued

- $d\varepsilon = \frac{dl}{l}$ with $\frac{dl}{dt} = c$

$$\eta(\sigma)\frac{d\varepsilon}{dt} = \sigma \rightarrow \eta_0 \exp\left(\frac{\boldsymbol{\sigma}}{\alpha}\right)\frac{\frac{dl}{l}}{dt} = \sigma$$

I gave up looking for the analytical solution of $\boldsymbol{\sigma}(l)$…

We might be able to use NR method to solve the above in combination with Euler method!

Euler + Newton-Raphson

$$\eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{dl}{l}}{dt} = \sigma \;\rightarrow\; \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{\Delta l}{l}}{\Delta t} = \sigma \;\rightarrow\; 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{\Delta l}{l}}{\Delta t}$$

$$l_{(n+1)} = l_{(n)} + \Delta l$$

$$t_{(n+1)} = t_{(n)} + \Delta t$$

$$t_0 = 0$$
$$l_0 = 0$$

Δt is, as usual, fixed as constant

Note that $\frac{dl}{dt} = c$. If we apply Euler approximation,

$$\frac{\Delta l}{\Delta t} = c \quad \rightarrow \Delta l = c\Delta t$$

$$\rightarrow 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{c\Delta t}{l}}{\Delta t}$$

$$\rightarrow 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{c}{l}$$

We apply the Newton-Raphson method to as below function:

$$\rightarrow f(\sigma_{(n)}) = \sigma_{(n)} - \eta_0 \exp\left(\frac{\sigma_{(n)}}{\alpha}\right)\frac{c}{l_{(n)}}$$

$$\rightarrow \frac{\partial f(\sigma)}{\partial \sigma} = 1 - \frac{\eta_0}{\alpha}\exp\left(\frac{\sigma_{(n)}}{\alpha}\right)\frac{c}{l_{(n)}}$$

- Outer loop over time
- Inner loop over NR search

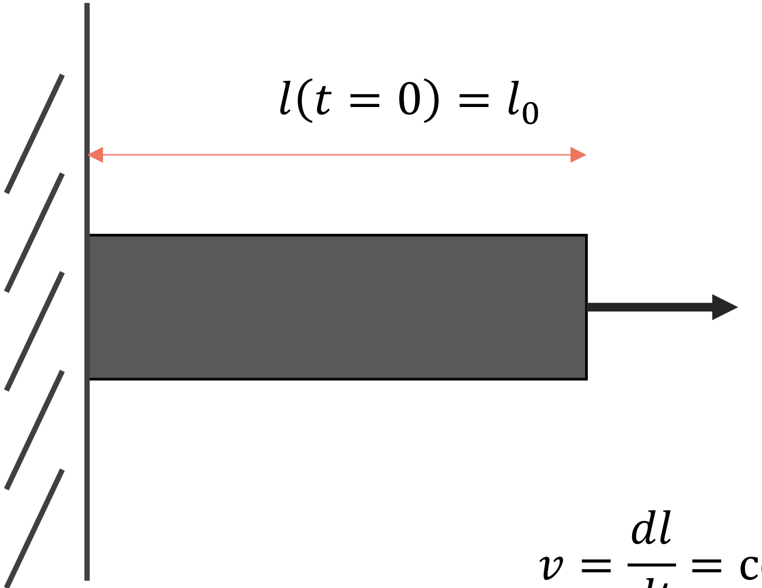
Euler + Newton-Raphson (Cheat sheet)

```
c     non newtonian
      real function calc_f(sigma, eta0, alpha, c, l)
      implicit none
      real sigma, et0, alpha, c,l, eta0
      calc_f = sigma - eta0  * exp(sigma / alpha) * c / l
      return
      end function

c     ---------------
      real function calc_df(sigma, eta0, alpha, c, l)
      implicit none
      real sigma, eta0, alpha, c,l
      calc_df = 1. - eta0/alpha * exp(sigma/alpha) *c/l
      return
      end function
```

```
      program main
      implicit none
      real dt,alpha,eta0,vel,l,t,calc_f,calc_df,tol,f,df,dl,sigma
      integer kount, i
      character*12 cdt
      parameter(tol=1e-5)

c     input
      dt = 1.
      alpha=300.
      eta0=30.
      vel=0.0001

      do i=1,iargc()
         call getarg(i,cdt)
         read(cdt,'(e20.13)')dt
      enddo
c
      dl = vel* dt
c
      l = 10.                           ! initial length
      t = 0.
c
      sigma=0.                          ! the very initial guess on stress

c     file
      open(2,file='euler_nr.txt')

      do while(t<30.01)
c     solve the equation to obtain sigma
         f = tol *2.                    ! work-around
         kount = 0
         do while(abs(f)>tol .and. kount < 10)
            f = calc_f(sigma,eta0,alpha,vel,l)
            df = calc_df(sigma,eta0,alpha,vel,l)
            sigma = sigma - f/df
            kount = kount + 1
         enddo
c         write(*,*) t, l, sigma, kount
         write(2,*) t, l, sigma, kount
         l=l+dl
         t=t+dt
      enddo

      close(2)
      end
```

Euler + Newton-Raphson

The cross-sectional area amounts to 12.5 [mm^2]



$$v = \frac{dl}{dt} = \cos(t) - 0.5 \qquad \frac{dv}{dt} = \sin(t)$$

Euler + Newton-Raphson

$$\eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{dl}{l}}{dt} = \sigma \;\rightarrow\; \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{\Delta l}{l}}{\Delta t} = \sigma \;\rightarrow\; 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{\Delta l}{l}}{\Delta t}$$

$$l_{(n+1)} = l_{(n)} + \Delta l$$

$$t_{(n+1)} = t_{(n)} + \Delta t$$

$$t_0 = 0$$
$$l_0 = 0$$

Δt is, as usual, fixed as constant

Note that $\frac{dl}{dt} = \cos(t) - 0.5$. If we apply Euler approximation,

$$\frac{\Delta l}{\Delta t} = \cos(t) - 0.5 \rightarrow \Delta l = \{\cos(t) - 0.5\}\Delta t$$

$$\rightarrow 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\frac{\{\cos(t) - 0.5\}\Delta t}{l}}{\Delta t}$$

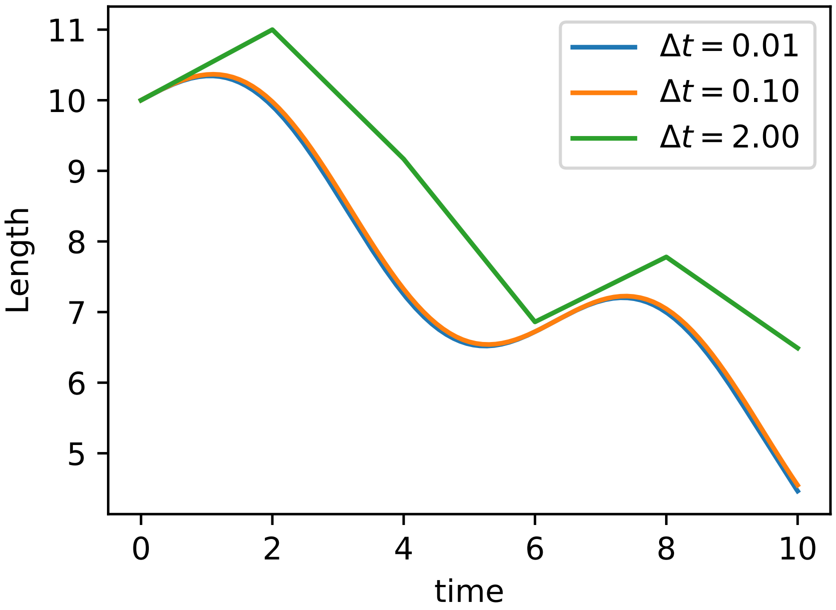
$$\rightarrow 0 = \sigma - \eta_0 \exp\left(\frac{\sigma}{\alpha}\right)\frac{\cos(t) - 0.5}{l}$$

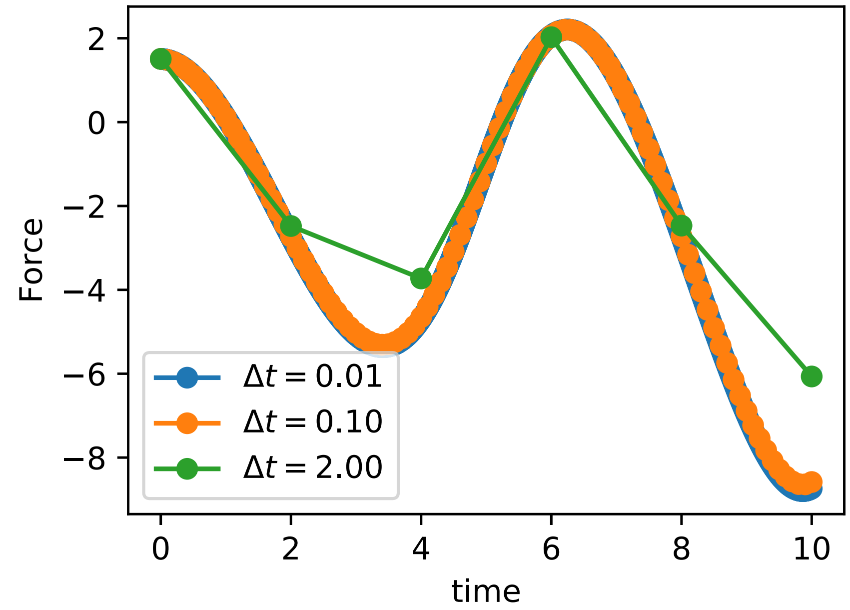
We apply the Newton-Raphson method to as below function:

$$\rightarrow f\left(\sigma_{(n)}\right) = \sigma_{(n)} - \eta_0 \exp\left(\frac{\sigma_{(n)}}{\alpha}\right)\frac{\cos t_{(n)} - 0.5}{l_{(n)}}$$

$$\rightarrow \frac{\partial f(\sigma)}{\partial \sigma} = 1 - \frac{\eta_0}{\alpha}\exp\left(\frac{\sigma_{(n)}}{\alpha}\right)\frac{\cos t_{(n)} - 0.5}{l_{(n)}}$$

Results





Euler + Newton-Raphson (Cheat sheet)

```fortran
c     non newtonian
      real function calc_f(sigma, eta0, alpha, t, l)
      implicit none
      real sigma, et0, alpha, t,l, eta0
      calc_f = sigma - eta0  * exp(sigma / alpha) * (cos(t)-0.5) / l
      return
      end function

c     ---------------
      real function calc_df(sigma, eta0, alpha, t, l)
      implicit none
      real sigma, eta0, alpha, t,l
      calc_df = 1. - eta0/alpha * exp(sigma/alpha) *(cos(t)-0.5)/l
      return
      end function
```

```fortran
      program main
      implicit none
      real dt,alpha,eta0,l,t,calc_f,calc_df,tol,f,df,dl,sigma
      integer kount, i
      character*12 cdt
      parameter(tol=1e-5)

c     input
      dt = 1.
      alpha=300.
      eta0=30.

      do i=1,iargc()
         call getarg(i,cdt)
         read(cdt,'(e20.13)')dt
      enddo
c
c      dl = vel* dt
c
      l = 10.                          ! initial length
      t = 0.
c
      sigma=0.                          ! the very initial guess on stress

c     file
      open(2,file='euler_nr.txt')

      do while(t<10.01)
         dl = (cos(t)-0.5)*dt
c     solve the equation to obtain sigma
         f = tol *2.                   ! work-around
         kount = 0
         do while(abs(f)>tol .and. kount < 10)
            f = calc_f(sigma,eta0,alpha,t,l)
            df = calc_df(sigma,eta0,alpha,t,l)
            sigma = sigma - f/df
            kount = kount + 1
         enddo
c         write(*,*) t, l, sigma, kount
         write(2,*) t, l, sigma, kount
         l=l+dl
         t=t+dt
      enddo

      close(2)
      end
```