ONERA Fatigue Model

July 19, 2021

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

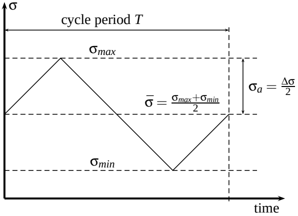
Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

ONERA Fatigue Model

- F1 : Uniaxial fatigue curves
  Woehler (N_f , $\sigma_a = \frac{\Delta\sigma}{2}$) curve from symmetric fatigue tests ($R = -1$)
- F2 : Loading ratio / mean-stress dependence
  Woehler curves with loading ratios $R > -1$, Haigh diagram
- F3 : Non-linear cumulation
  dependence of the loading path history on the damage results
- F4 : Temperature Effect
- F5 : Multiaxiality
- F6 : Creep damage
- F7 : Creep-Fatigue interaction

ONERA model (Chaboche)



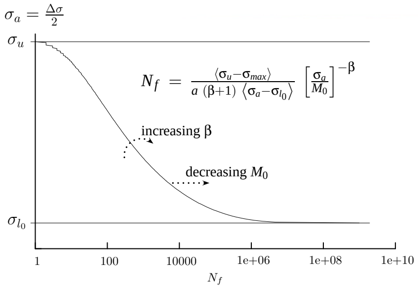
$$N_f = \frac{1}{(\beta+1)\,[1-\alpha(\sigma_a,\bar{\sigma})]}\left[\frac{\sigma_a}{M(\bar{\sigma})}\right]^{-\beta}$$

$$\text{with } \alpha = 1 - a\,\frac{\langle \sigma_a - \sigma_l(\bar{\sigma})\rangle}{\langle \sigma_u - \sigma_{max}\rangle}$$

$$N_f = \frac{\langle \sigma_u - \sigma_{max}\rangle}{a\,(\beta+1)\,\langle \sigma_a - \sigma_l(\bar{\sigma})\rangle}\left[\frac{\sigma_a}{M(\bar{\sigma})}\right]^{-\beta}$$

- Mean stress dependence: $\sigma_l(\bar{\sigma}) = \frac{\sigma_{l_0}}{1 + b_1\,\bar{\sigma}}$, $M(\bar{\sigma}) = \frac{M_0}{1 + b_2\,\bar{\sigma}}$
- Coefficients : σ_u, σ_{l_0}, β, M_0, b_1, b_2

F1: Calibration of symmetric fatigue results (1/2)



Active coefficients on symmetric fatigue tests ($R_\sigma = -1$)

- UTS : σ_u
- fatigue limit : σ_{l_0}
- slope : β
- position along N_f axis : M_0

F1: Calibration of symmetric fatigue results (2/2)

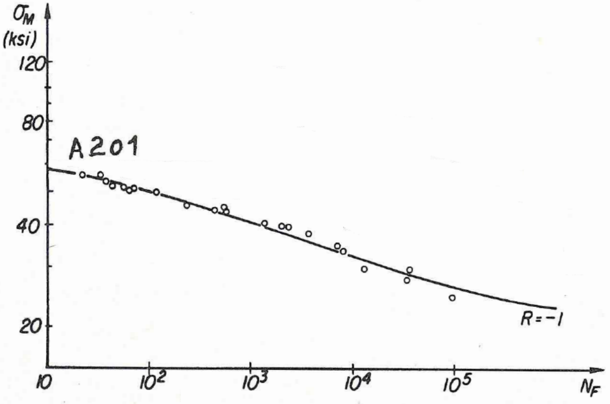
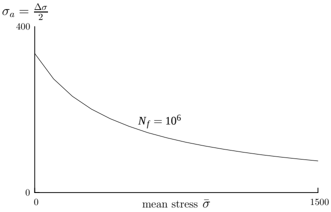


Fig. 15. Courbe des Woehler des aciers A 201 à 20 °C.

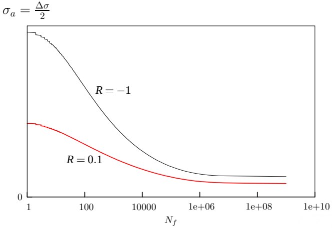
▸ demo ▸ reset ▸ open terminal

F2: Mean stress dependence (1/2)

Haigh diagram



Woehler curve



- b_1 : decrease of σ_l depending on mean stress $\bar{\sigma}$

$$\sigma_l(\bar{\sigma}) = \frac{\sigma_{l_0}}{1 + b_1 \bar{\sigma}}$$

- b_2 : translation to smaller N_f values

$$M(\bar{\sigma}) = \frac{M_0}{1 + b_2 \bar{\sigma}}$$

demo reset open terminal

F2: Mean stress dependence (2/2)

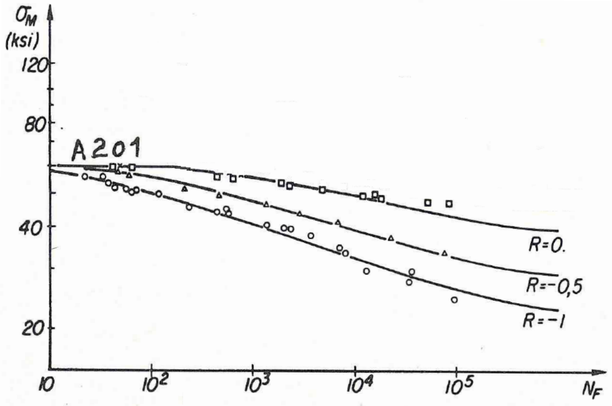
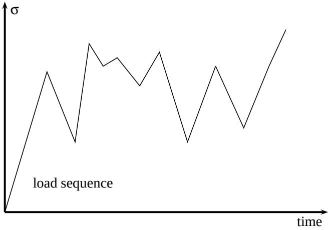


Fig. 15. Courbes des Woehler des aciers A 201 à 20 °C.

demo reset open terminal

F3: Non-linear cumulation (1/4)

- For complex loading paths



- Linear cumulation (Miner's rule) is not conservative
  - sequence effect: mildly damaging cycles have a greater effect if they occur after strongly damaging one
  - cycles with amplitude below the fatigue limit do have an effect if prior fatigue damage occurred
- Needs a multi-axial rainflow procedure (see tools)

F3: Non-linear cumulation (2/4)

Fatigue damage evolution equation:

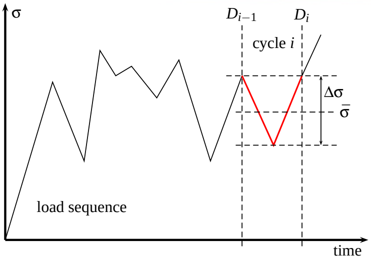
$$\delta D = \left[1 - (1 - D)^{\beta+1}\right]^{\alpha(\sigma_a, \bar{\sigma})} \left[\frac{\sigma_a}{M(\bar{\sigma})\,(1 - D)}\right]^{\beta} \delta N$$

$$\text{with } \alpha = 1 - a\,\frac{\langle \sigma_a - \sigma_l(\bar{\sigma}) \rangle}{\langle \sigma_u - \sigma_{max} \rangle}$$

- integration of D between 0 and 1

$$N_f = \frac{1}{(\beta + 1)\,[1 - \alpha(\sigma_a, \bar{\sigma})]} \left[\frac{\sigma_a}{M(\bar{\sigma})}\right]^{-\beta}$$

F3: Non-linear cumulation (3/4)



integration between D_{i-1} and D_i

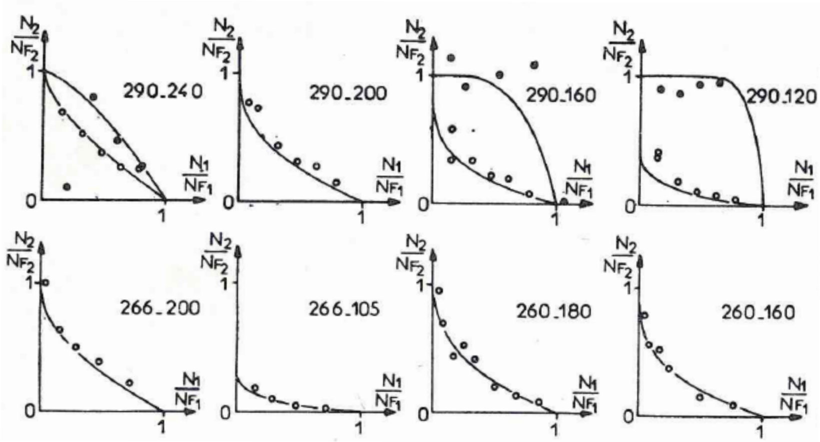
- above the fatigue limit: $\sigma^a > \sigma_l(\bar{\sigma})$, $\alpha < 1$

$$\left[1-(1-D_i)^{\beta+1}\right]^{1-\alpha} - \left[1-(1-D_{i-1})^{\beta+1}\right]^{1-\alpha} = \frac{1}{N_f(\Delta\sigma,\bar{\sigma},\sigma_{max})}$$

- below: $\sigma^a \leq \sigma_l(\bar{\sigma})$, $\alpha = 1$

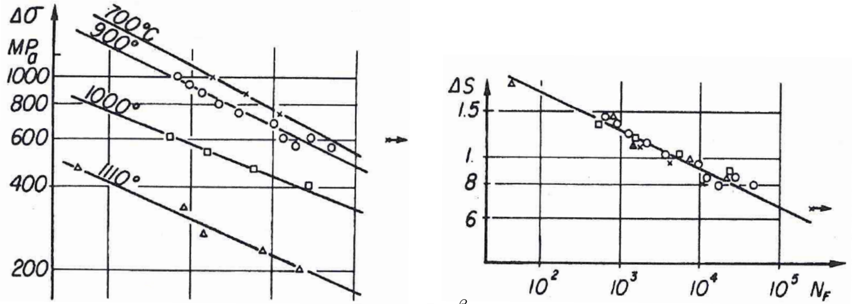
$$ln\left[\frac{1-(1-D_i)^{\beta+1}}{1-(1-D_{i-1})^{\beta+1}}\right] = (\beta+1)\left(\frac{\sigma^a}{M(\bar{\sigma})}\right)^{\beta}$$

F3: Non-linear cumulation (4/4)



▸ demo ▸ reset ▸ open terminal

F4: Temperature Effect



$$N_f = \frac{\langle 1 - S_{max} \rangle}{a\,(\beta + 1)\,\left\langle \frac{\Delta S}{2} - \sigma_l(\bar{S}) \right\rangle} \left[\frac{\frac{\Delta S}{2}}{M(\bar{S})} \right]^{-\beta} \qquad \text{where} \qquad S(T) = \frac{\sigma(T)}{\sigma_u}$$

- Normalized coefficients : σ_{l_0}, M, b_1, b_2

demo reset open terminal

F5: Multiaxiality

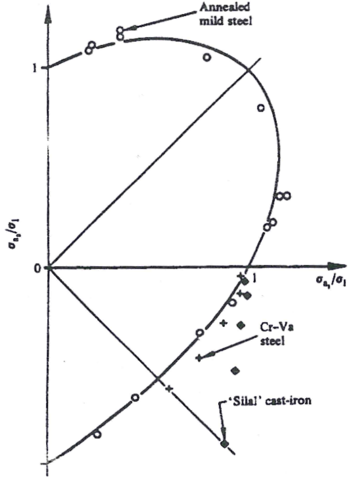


Fig. Sines criterion in reversed loading (σ_l means $\sigma_{l0} = \sigma_l(0)$)

F6: Creep damage (1/2)

Rabotnov-Kachanov model:

$$\dot{D}_C = \left(\frac{\sigma_{eq}}{A}\right)^{r} \frac{1}{(1 - D_C)^{k}}$$

where: $$\sigma_{eq} = (1 - \alpha - \beta)\, J(\underset{\sim}{\sigma}) + \alpha\, \sigma_p + \beta\, \bar{\sigma}$$

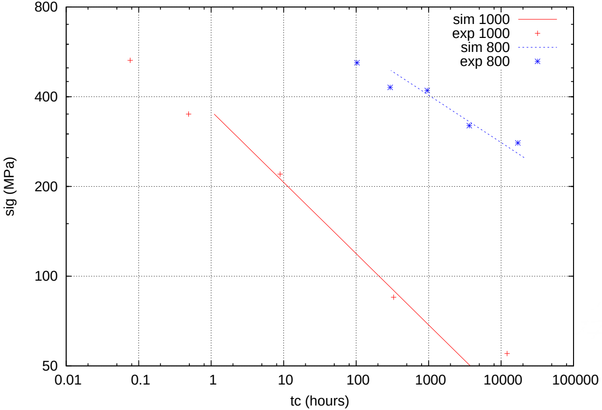
and:

- $J(\underset{\sim}{\sigma})$ Mises invariant of σ
- σ_p first principal stress
- $\bar{\sigma}$ mean trace of σ
- r, k, A material parameters that depend on temperature

Integration between 0 and 1 yields the critical time t_C for creep initiation:

$$t_C = \frac{1}{k+1} \left(\frac{\sigma_{eq}}{A}\right)^{-r}$$

F6: Creep damage (2/2)



▸ demo ▸ reset ▸ open terminal

F7: Creep-Fatigue interaction (1/2)

For each cycle damage goes from D_i to D_f:

- damage increases from D_i to D_c due to creep

$$C = \frac{1}{N_c} = (1 - D_i)^{k+1} - (1 - D_c)^{k+1}$$

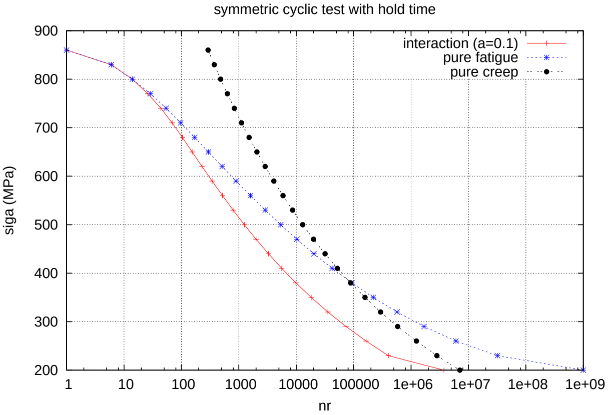
- damage increases from D_c to D_f due to fatigue

$$F = \frac{1}{N_f} = \left[1 - (1 - D_f)^{\beta+1}\right]^{1-\alpha} - \left[1 - (1 - D_c)^{\beta+1}\right]^{1-\alpha}$$

where:

- C increase of damage on one cycle due to pure creep
- F increase of damage on one cycle due to pure fatigue
- $\alpha = 1 - a \frac{\langle \sigma_a - \sigma_l(\bar{\sigma}) \rangle}{\langle \sigma_u - \sigma_{max} \rangle}$
- coefficient a controls the amount of of cumulation non-linearity:
  $0 \leq a \leq 1$
  $a \approx 1$: quasi-linear
  $a \approx 0$: strongly non-linear

F7: Creep-Fatigue interaction (2/2)



demo reset open terminal

Fatigue models summary

Inputs required by the fatigue model:

- From the constitutive model used in FE analysis:
  - cyclic hardening modeling (σ_a)
  - mean stress relaxation modeling ($\bar{\sigma}$)

- Dedicated post-processing tools:
  - calculation of the amplitude of a multiaxial stress path
  - multiaxial rainflow algorithm

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Manson-Coffin LCF model

- classical strain-based fatigue model
- doesn't account for non-zero mean stress effects

$$\frac{\Delta\epsilon}{2} = A\, N_f^{-\alpha} + B\, N_f^{-\beta}$$

$$\frac{\Delta\epsilon}{2} = \frac{\Delta\epsilon^p}{2} + \frac{\Delta\epsilon^e}{2} = \frac{\Delta\epsilon^p}{2} + \frac{\Delta\sigma}{2\,E}$$

A, α (plastic), B, β (elastic) fatigue model coefficients, E Young's modulus

A ramberg-osgood equation calibrated from cyclic hardening experimental curves can be used to obtain $\Delta\epsilon^p$ from $\Delta\sigma$ and transform standard S-N curves to ($\Delta\epsilon$,N_f) plots

$$\frac{\Delta\sigma}{2} = K \left(\frac{\Delta\epsilon^p}{2}\right)^n$$

Note that (K, n) are linked to manson coefficients:

$$n = \frac{\beta}{\alpha} \quad , \quad K = \frac{EB}{A^n}$$

Manson-Coffin demo

- calibration of (A, α) on $(N_f, \frac{\Delta\epsilon^p}{2})$ points using the ***plastic*** function item and the ***opt_plastic*** otimization item
- calibration of (B, β) on $(N_f, \frac{\Delta\epsilon^e}{2})$ points using the ***elastic*** function item and the ***opt_elastic*** otimization item
- check the whole $(N_f, \frac{\Delta\epsilon}{2})$ curve using the ***woehler_manson*** external simulation item built with Zpost and the **strain_control.z7p** script
- the ***woehler_manson_ramberg*** simulation item recalculates the plastic strain from the stress input using the ramberg-osgood equation given in the previous slide
- use the multiplot item to compare manson-coffin with the onera model calibration done previously (see onera R=-1 calibration)

demo reset open terminal

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Multiaxial loading path amplitude (1/2)

- solve the Smallest Enscribed Hypersphere problem (SEH)

$$\min_{\underset{\sim}{\boldsymbol{X}}} \left\{ R(\underset{\sim}{\boldsymbol{X}}) = \max_{i} \left\{ J(\underset{\sim}{\boldsymbol{\sigma}}^{i} - \underset{\sim}{\boldsymbol{X}}) \right\} \right\}$$

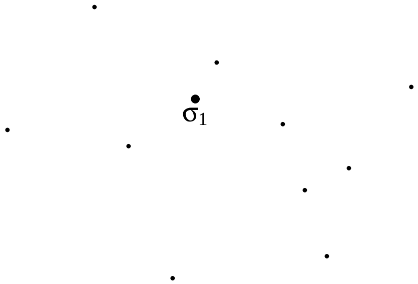
- generic geometrical problem with applications in many fields (pattern recognition, protein analysis, political science, ray-tracing ...)
- still, efficient algorithms are fairly recent (Bend Gartner, "Fast and Robust Smallest Enclosing Balls" 1999)

Multiaxial loading path amplitude (2/2)

- Iteratively solve optimization problems $SEH(\mathcal{S})$ on subsets $\mathcal{S}$ of $\mathcal{P}$
- At each iteration, solution of $SEH(\mathcal{S})$ is found by solving a linear system with size the number of points in $\mathcal{S}$
- Fast convergence to the solution of $SEH(\mathcal{P})$

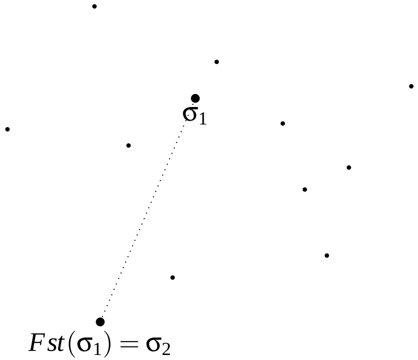
SEH example

- Initialization: first point σ_1 in the loading path $\mathcal{P}$



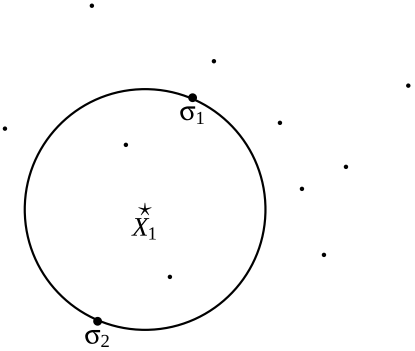
SEH example

- Farthest of σ_1 is $Fst(\sigma_1) = \sigma_2$



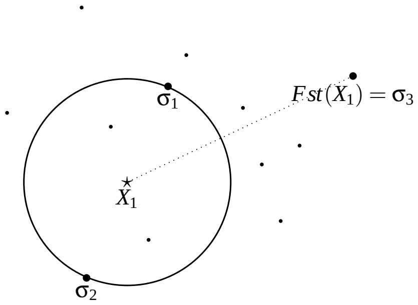
SEH example

- Solve $SEH(\sigma_1, \sigma_2)$: solution is H_1 with center X_1



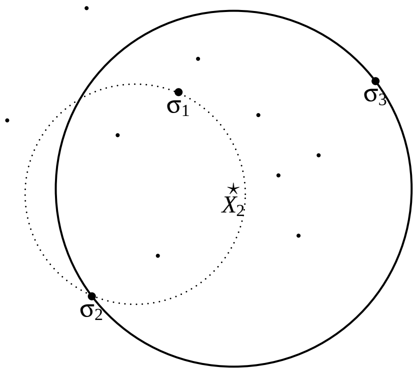
SEH example

- Farthest of X_1 is $Fst(X_1) = \sigma_3$



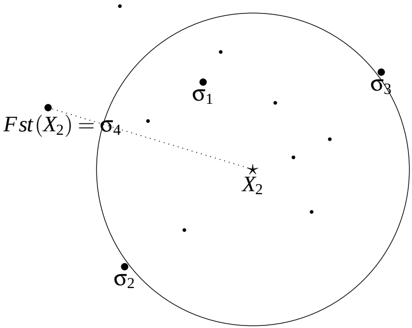
SEH example

- Solve $SEH(\sigma_1, \sigma_2, \sigma_3)$: solution is H_2 with center X_2



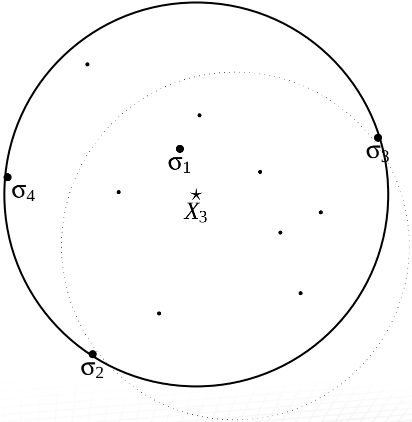
SEH example

- Farthest of X_2 is $Fst(X_2) = \sigma_4$



SEH example

- Solve $SEH(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$: solution is H_3 with center X_3
- all points are in H_3, end

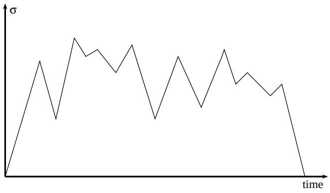


Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Multiaxial rainflow algorithm (1/4)

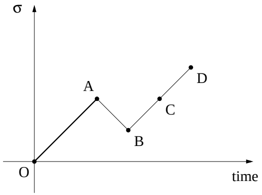
- extract sub-cycles from a complex stress path for (non-)linear damage cumulation



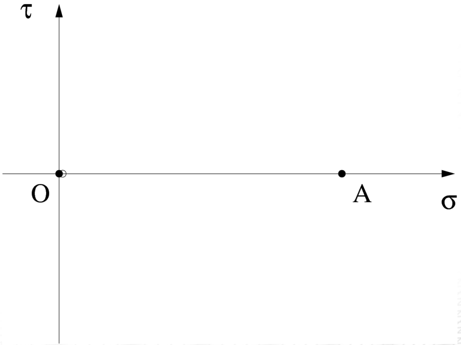
- generalize the 1D rainflow method to multiaxial stress states (6 independent components)
- use the concept of active surface used in some plasticity models (Melnikov, Semenov)

Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

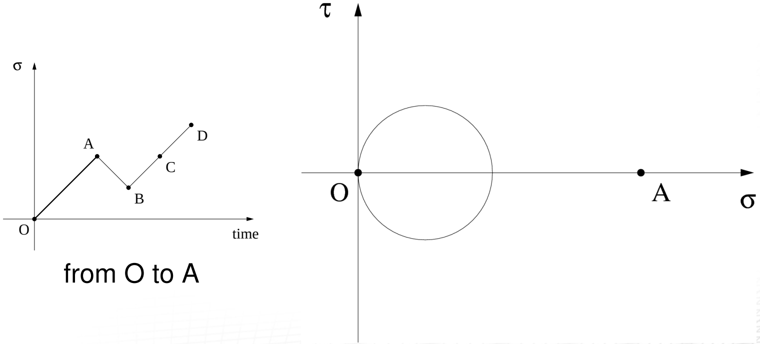


from O to A



Multiaxial rainflow algorithm (2/4)

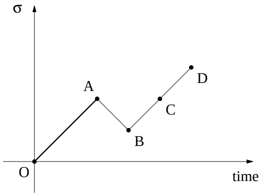
- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one



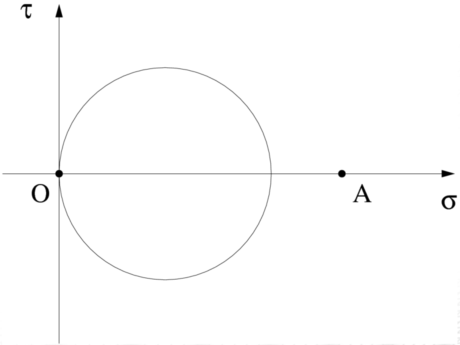
from O to A

Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

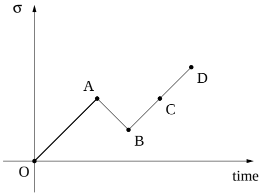


from O to A

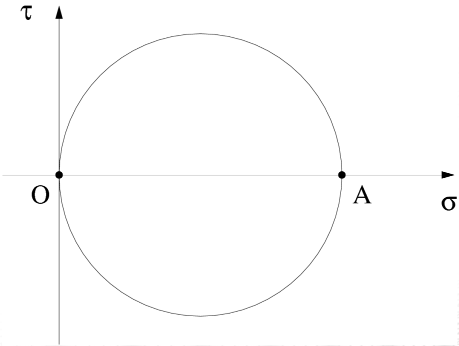


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

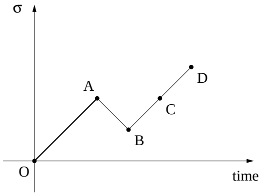


at point A

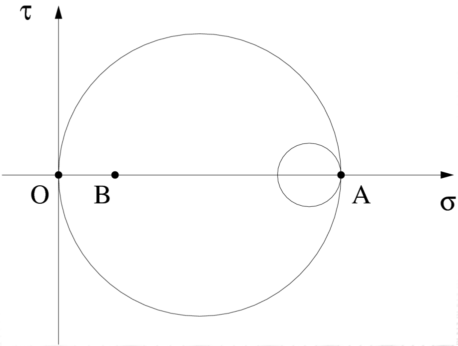


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

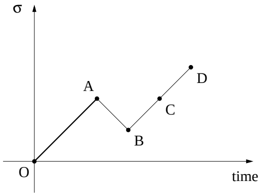


unloading from A to B

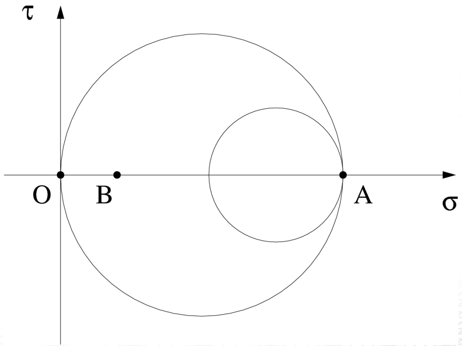


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

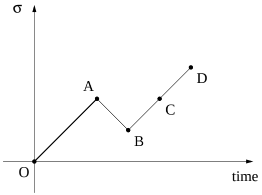


unloading from A to B

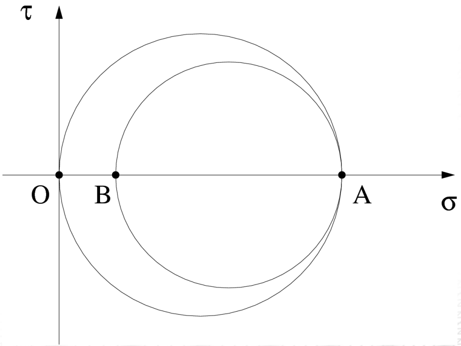


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

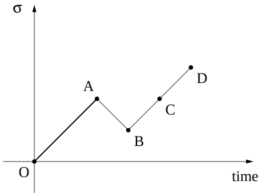


at point B

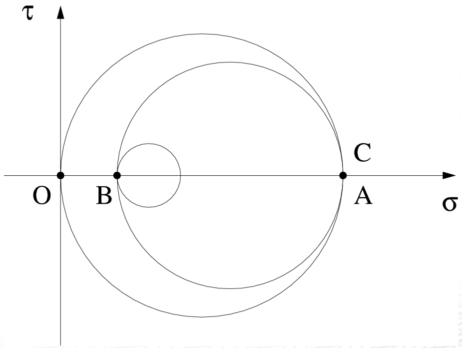


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

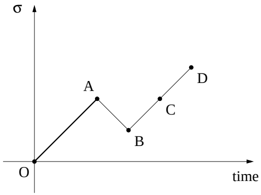


from B to C

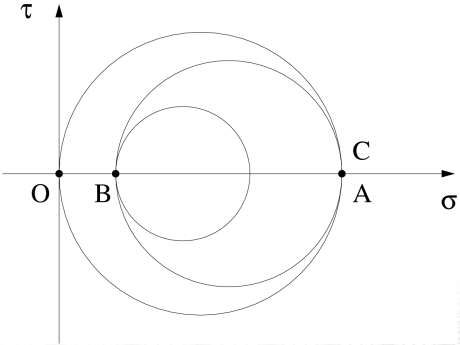


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

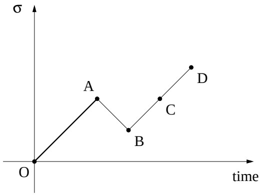


from B to C

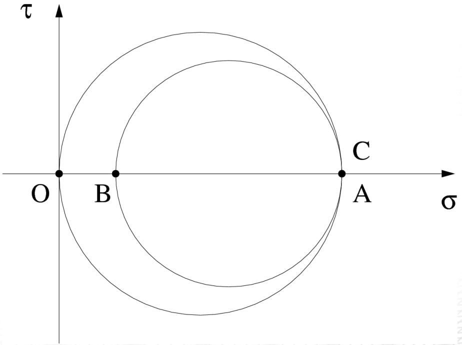


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

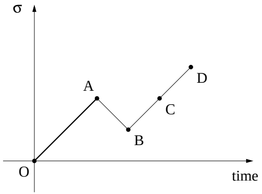


at point C

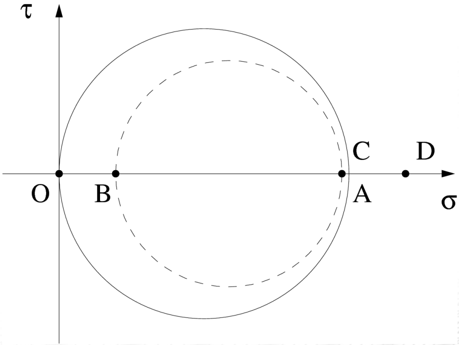


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

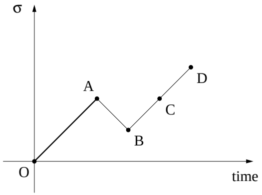


extra loading from C to D

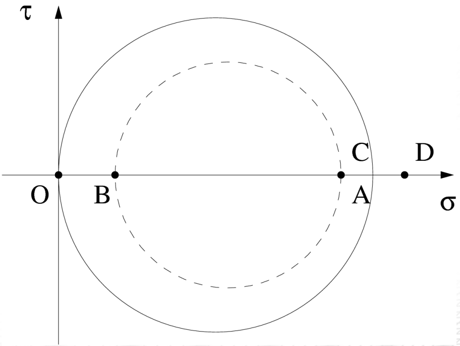


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one

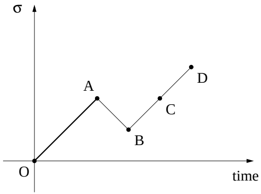


extra loading from C to D

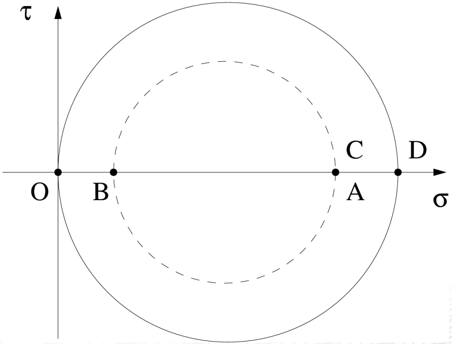


Multiaxial rainflow algorithm (2/4)

- follow the stress path and recursively builds active surfaces included in one another
- a cycle is detected when the size of the current surface exceeds the previous one



extra loading from C to D

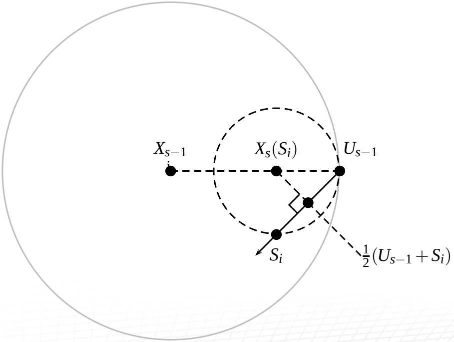


Multiaxial rainflow algorithm (3/4)

An active surface s is defined by:

- $\underset{\sim}{\boldsymbol{U}}_s$ its closing point created when unloading occurs
- $\underset{\sim}{\boldsymbol{X}}_s$ its center

Update of center: $\underset{\sim}{\boldsymbol{X}}_s(\underset{\sim}{\boldsymbol{S}}_i)$ intersection of the mediatrix of $(\underset{\sim}{\boldsymbol{U}}_{s-1}, \underset{\sim}{\boldsymbol{S}}_i)$ with $(\underset{\sim}{\boldsymbol{X}}_{s-1}, \underset{\sim}{\boldsymbol{U}}_{s-1})$

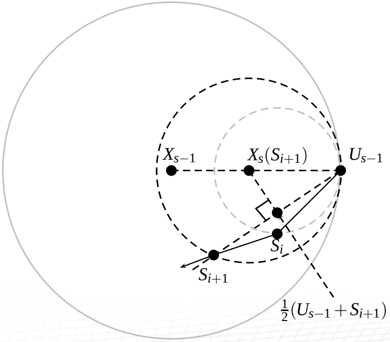


Multiaxial rainflow algorithm (3/4)

An active surface s is defined by:

- $\underset{\sim}{\boldsymbol{U}}_s$ its closing point created when unloading occurs
- $\underset{\sim}{\boldsymbol{X}}_s$ its center

Update of center: $\underset{\sim}{\boldsymbol{X}}_s(\underset{\sim}{\boldsymbol{S}}_{i+1})$ intersection of the mediatrix of $(\underset{\sim}{\boldsymbol{U}}_{s-1}, \underset{\sim}{\boldsymbol{S}}_{i+1})$ with $(\underset{\sim}{\boldsymbol{X}}_{s-1}, \underset{\sim}{\boldsymbol{U}}_{s-1})$

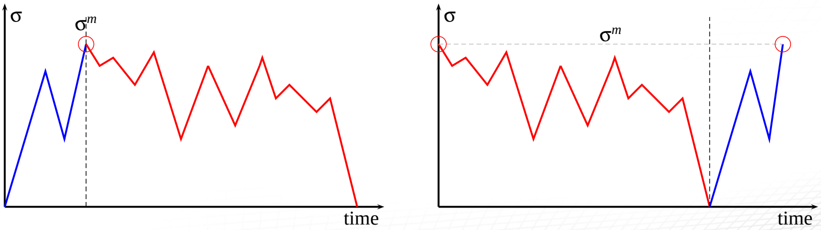


Multiaxial rainflow algorithm (4/4)

Initialization:

- $\underset{\sim}{\boldsymbol{X}}_0 = \underset{\sim}{\boldsymbol{X}}_{SEH}$ center of the SEH calculated for the whole loading path
- $\underset{\sim}{\boldsymbol{U}}_0 = \underset{\sim}{\boldsymbol{0}}$
- to avoid residuals reorganize the path to begin by the maximum point $\underset{\sim}{\boldsymbol{\sigma}}_m$ such that:

$$J(\underset{\sim}{\boldsymbol{\sigma}}_m - \underset{\sim}{\boldsymbol{X}}_{SEH}) = \max_{i \in \mathcal{P}} \left\{ J(\underset{\sim}{\boldsymbol{\sigma}}_i - \underset{\sim}{\boldsymbol{X}}_{SEH}) \right\}$$



Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

**process onera

- (multiaxial) rainflow sub-cycles extraction
- (nonlinear) cumulation

```
**output_number ...
**material_file mat
**process onera
 [*cycle beg1-end1/rep1 ... begn-endn/repn]
 [*preload beg1-end1/rep1 ... begn-endn/repn]
  *mode NLC_ONERA | LC
  *fatigue fatigue_rainflow 2
 [*creep creep 2]
 [*reverse 3]

% material file
***post_processing_data
 **process onera
   a 0.1   % (0<a<1)
***return
```

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

**process fatigue_rainflow

- number of cycles to failure according to the ONERA fatigue model

```
**material_file mat
**process fatigue_rainflow
  *var sig
 [*mean_stress (standard|variant)]
 [*mode with_a]
 [*normalized_coeff]
 [*reverse 3]

% material file
***post_processing_data
 **process fatigue_rainflow
    sigma_u 600.0  sigma_l 0.2  M 40.0  beta 2.0
  [ a 0.1 ]
  [ b1 0.1  b2 0.1  sigma_n 550.0  sigma_p 0.01 ]
***return
```

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

**process manson_coffin

- number of cycles to failure according to the Manson-Coffin model

```
**material_file mat
**process manson_coffin
  *var sig
% name of the plastic strain tensor (epi) or cyclic_hardening
% to derive epi from sig using a ramberg-osgood equation
  *plastic epi | cyclic_hardening
[ *infinity_is 1.0e+08 ]
[ *full_output ] % additional ee_alt, ep_alt output

% material file
***post_processing_data
 **process manson_coffin
  A 0.01 alpha 0.1
  B 0.01 beta  0.1
  young 200000.
***return
```

Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Stress-controlled script

- Input file (file *stress*):

```
*frequency 20.0
*temperature 950.0
*material endo.mat
*stress 220.0 2000.0 50.0
*load_factor -1.0
[*creep]
[*delay]
```

- Command:

```
Zrun -zp stress_control.z7p -s ZLanguage.args stress
```

- Output (file *stress.out*):

```
# nr dsig/2 dsig sig_max sig_min nf nc
3.111629e+06 2.200000e+02 4.400000e+02 2.200000e+02 -2.200000e+02 4.440855e+08 1.284159
...
```

- Use in SimOpt: demo terminal

Strain-controlled script

- Input file (file *strain*):

```
*frequency 20.0
*temperature 950.0
*material endo.mat
*behavior comportement.mat
*strain 3.e-3 1.e-2 5.e-4
*load_factor -1.0
[*creep]
[*delay]
```

- Command:

```
Zrun -zp strain_control.z7p -s ZLanguage.args strain
```

- Output (file *strain.out*):

```
1.402353e+06 2.705294e+02 5.410588e+02 -8.940000e-04 6.000000e-03 2.223231e+07 1.588618e
...
```

- Use in SimOpt: demo terminal

Goodman diagram script

- Input file (file *goodman*):

```
*frequency 1.
*target 1.e7
*temperature 950.0
*material mat.mat
*steady_stress 0.0 700.0 100.0
[*creep]
[*delay]
```

- Command:

```
Zrun -zp goodman.z7p -s ZLanguage.args goodman
```

- Output (file *goodman.out*):

```
# ss sa nr
0.000000e+00 3.141640e+02 1.000003e+07
1.000000e+02 3.088142e+02 1.000019e+07
...
```

- Use in SimOpt: demo terminal

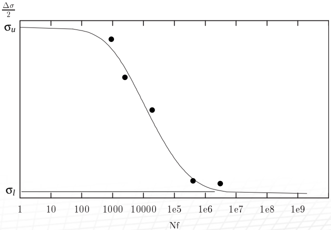
Plan

1. ONERA Model
2. Manson-Coffin Model
3. Basic Tools
   - Multiaxial stress amplitude (SEH)
   - Multiaxial rainflow
4. Z-post input commands
   - process onera
   - process fatigue_rainflow
   - process manson_coffin
   - Calibration scripts
5. Experimental database

Calibration experimental database (1/2)

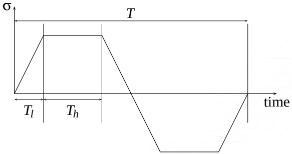
For each temperature spanning the operating temperature range, ie. at least 4 temperature levels including T_{min} (room temperature) and T_{max}:

- E1: Creep tests with at least 3 points in the (σ, t_f) diagram
- E2: *Pure fatigue* symmetric tests ($R_\sigma = -1$)
  - High frequency tests (50 Hz) to uncouple from creep
  - If not possible lower frequency tests ($>$0.5 Hz)
  - Several frequencies if possible
  - Characterization of the full S-N curve: 3 points in the LCF regime ($N_f < 10^5$), 2 points in the HCF regime (fatigue limit)



Calibration experimental database (2/2)

- E3: Non symmetric fatigue tests to characterize the mean stress effect
  - A least 1 non symmetric loading ratio ($R\sigma \neq -1$), and ideally 2 to characterize the full Haigh diagram.
  - For each loading ratio, S-N curve as in E2
- E4: Creep-fatigue interaction tests
  - Symmetric $R_\sigma = -1$ tests with hold time
  - Typically $T = 220\ s$ with $T_l = 10\ s$ and $T_h = 90\ s$ (check T_h with creep tests in E1)
  - S-N curve in the low cycle regime (at least 3 points)



Simplifications:

- E1 and E4 are not needed at low temperature
- E3 can be skipped for some temperatures
- Cyclic tests at $R_\epsilon = -1$ available from material behavior tests and performed up to failure can be used to build the S-N curves