Norton behaviour description

- file : Norton.mfront
- author : Helfer Thomas
- date : 23 / 11 / 06

This viscoplastic behaviour is fully determined by the evolution of the equivalent viscoplastic strain p as a function f of the Von Mises stress σ_{eq} :

$$\dot{p} = f\left(\sigma_{\mathrm{eq}}\right) = A\,\sigma_{\mathrm{eq}}^{E}$$

where :

- A is the Norton coefficient .
- E is the Norton exponent .

A and E are declared as material properties .

Source code

```
@Parser IsotropicMisesCreep;
@Behaviour Norton;
@Author Helfer Thomas;
@Date 23/11/06;
@Description{
  This viscoplastic behaviour is fully determined by the evolution
  of the equivalent viscoplastic strain "\(p\)" as a function "\(f\)"
  of the Von Mises stress "\(\sigmaeq\)":
  "\["
  "\dot{p}=f\paren{\sigmaeq}=A\,\sigmaeq^{E}"
  "\]"
  where:

  - "\(A\)" is the Norton coefficient.
  - "\(E\)" is the Norton exponent.

  "\(A\)" and "\(E\)" are declared as material properties.
}

@UMATFiniteStrainStrategies[umat] {None,LogarithmicStrain1D};

//! The Norton coefficient
@MaterialProperty real A;
```

```
A.setEntryName("NortonCoefficient");

//! The Norton coefficient
@MaterialProperty real E;
E.setEntryName("NortonExponent");

@FlowRule{
  /*!
   * The return-mapping algorithm used to integrate this behaviour
   * requires the definition of \(f\) and \(\deriv{f}{\sigmaeq}\) (see
   * @simo_computational_1998 and @helfer_generateur_2013 for
   * details).
   *
   * We introduce an auxiliary variable called `tmp` to
   * limit the number of call to the `pow` function
   */
  const real tmp = A*pow(seq,E-1);
  f       = tmp*seq;
  df_dseq = E*tmp;
}
```