TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
The University of Resources. Since 1765.

Faculty of Mechanical, Process and Energy Engineering
Institute for Mechanics and Fluid Dynamics (IMFD)
Chair Applied Mechanics – Solid Mechanics

Using the Non-local GTN-Model for simulation ductile damage in ABAQUS/Standard & ABAQUS/Explicit

Dinh Rinh Pham, Andreas Seupel, Geralf Hütter, Omar El Khatib, Bjoern Kiefer

May 17, 2022

Contents

1 Description

A non-local GTN-model for ductile damage was implemented in Abaqus. This model has been developed to circumvent spurious mesh dependency during simulation of ductile failure. Using this model, the stages of crack development in component, which are crack blunting, crack initiation and crack growth, are captured. The implementation makes use of the heat equation analogy so that nearly all built-in thermomechanical elements of Abaqus can be used. Details on theory and FEM implementation can be found in [2].

Furthermore, this non-local GTN-model implementation with Euler backward integration scheme for Abaqus/Standard, as described in [2], is transfered to VUMAT interface in Abaqus/-Explicit. ABAQUS/Explicit is favorable for highly dynamic problems or for problems with extreme non-linearity.

1.1 Features

- Mesh independency
- Capturing crack blunting, crack initiation and crack growth in ductile damage in large deformation analyses
- Support of all 3D, axisymmetric and 2D plain strain built-in elements of Abaqus
- Support of all parallelization capabilities of Abaqus

1.2 Requirements

- Abaqus version 2020 or 2021. For unknown reason, non-local GTN-model does not work with Abaqus 2022, in which the definition of thermal source in the heat equation is redefined.
- Intel Fortran compiler

The present implementation has been developed and tested intensively with Abaqus 2020 and Intel Fortran 19.0 under Debian GNU/Linux 10 and Intel Fortran Compiler Classic 2021.4 (Part of Intel OneAPI HPC Toolkit 2021) and Microsoft Visual Studio Community 2019 under Windows 10.

2 Usage

2.1 User-defined Material Property

2.1.1 Material definition in Abaqus/Standard

The present implementation requires to provide a user-defined material in Abaqus with at least 13 parameters as listed in Tab. 1. To provide the user a flexible way to define the strain hardening of material through a tabular form instead of a function, an option with user-defined material, which requires more than 13 parameters, is also implemented as described in Section 2.4. In addition, the internal length l_{nl} is an unknown significant parameter for each material in non-local GTN-model. This internal length l_{nl} will enter into model through conductivity property with keyword **Conductivity* in the material definition section, which takes the value *<Lintsqr>*=l_{nl}^2. Moreover, the critical porosity f_{c} is also another unknown parameter, while other parameters are determined through experiment or take some ad-hoc values from the literature. Procedures and examples for determining these parameters can be found in [1, 2, 3, 4].

To monitor the evolution of internal variables in material model and visualize the simulation results, 11 solution-dependent variables as listed in Tab. 2 are generated through keyword *Depvar in the material definition section.

Those mentioned above parameters and material properties shall be entered either in the input file (*.inp-file*) in text format or through Abaqus/CAE interface as following demonstration:

Input File Usage:

```
*Material, name=Material-1
*Conductivity
<Lintsqr>
*Depvar
 11,
*User Material, constants=13, unsymm
<Young>,<Nu>,<f0>,<fc>,<kappa>,<q1>,<q2>,<Damagecase>
<fn>,<S>,<En>,<SigmaY>,<n>
```

Abaqus/CAE Usage: Property module: material editor:

- **Thermal > Conductivity: Type: Isotropic**
- **General > Depvar: Number of solution-dependent state variables**
- **General > User Material: User material type: Mechanical**, toggle on **Use unsymmetric material stiffness matrix**

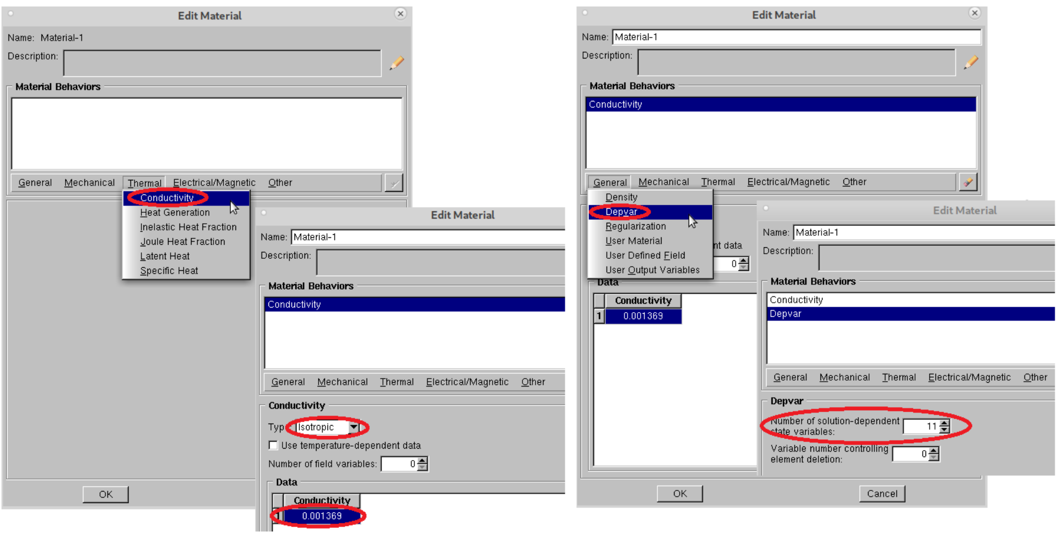

Fig. 1: Thermal conductivity and Depvar configuration

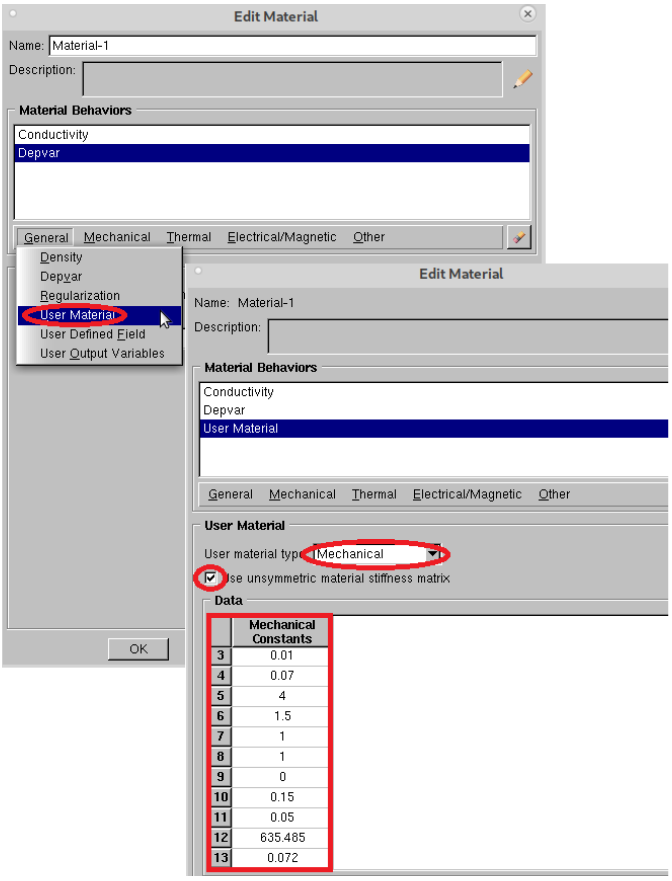

Fig. 2: User material configuration

Tab. 1: User material properties

#PROPS	PARAMETER	MEANING
1	E	Young's Modulus E
2	Nu	Poisson's ratio ν
3	fo	Initial porosity f_0
4	fc	Critical porosity f_c
5	kappa	void acceleration factor κ
6	q1	GTN parameter q_1
7	q2	GTN parameter q_2
8	Damagecase	1.0 -> Non-local GTN-model 0.0 -> Local GTN-model
9	fn	Nucleable porosity f_N
10	S	Void nucleation strain: standard variation S_N
11	En	Void nucleation strain: mean value EN
12	SigmaY	Initial yield stress
13	n	Power law exponent for strain hardening
14 ..	-	optional: as described in **Section 2.4** tabular form of stress-strain curve

Tab. 2: Solution-dependent state variables

#STATEV(SDV)	INTERNAL NAME	MEANING
1	EPSILON_MISES	Equivalent plastic strain
2	RR	Hardening (equivalent plastic strain of matrix material
3	-	Non-local driving force of damage (non-local volumetric plastic train)
4	DAMAGE	Damage variable related to void nucleation (DAMAGE=0 → no nucleation, DAMAGE=1 → necleation sites are active
5	-	Local source term of damage: volumetric plastic strain
6	FCSwitch	Indicator for exceeding critical void volume fraction and reaching total damage: =0 → under-critical, =2 → accelerated void growth regime, =4 → total damage state
7	-	Derivative of damage source term w.r.t strain increment
8	-	Exact void volume fraction f_c, when accelerated void growth sets in (numerical parameter)
9	FSTAR	Effective void volume fraction f^*
10	F_LOK	Void volume fraction f
11	T	Stress triaxiality

2.1.2 Material definition in Abaqus/Explicit

In ABAQUS/Explicit, 3 more material properties are required in material property definition. This material's density and specific heat need to be added in as requirement of transient heat transfer analysis. Furthermore, the Heat Generation option needs to be set to tell Abaqus that it needs call subroutine VHETVAL, for obtaining the source term $r = \varepsilon_{\mathrm{nl}} - \varepsilon_{\mathrm{l}}$. Thus, the following three keywords have to be set for input file usage:

Input File Usage:

```
*Material, name=Material-1
...
*Density
 7.85e-09,
*Specific Heat
 1.274e+02,
*Heat Generation
```

This can be set in Abaqus/CAE in the Material Editor as well in the same way as described before. The choice of the parameter Specific heat is discussed in Section 2.5.

2.2 Step Definition

2.2.1 Step Definition in Abaqus/Standard

In the nonlocal GTN model [2], the additional balance equation of Helmholtz-type

$$l_{\mathrm{nl}}^2 \Delta_{\mathrm{x}} \varepsilon_{\mathrm{nl}} = \varepsilon_{\mathrm{nl}} - \varepsilon_{\mathrm{l}} \qquad \forall x \in \Omega. \tag{1}$$

appears. This equation has identical structure as the steady state heat transfer equation and can thus be implemented in Abaqus via the heat analogy. For this purpose, a **couple temperature displacement, steady state** step has to be selected in Abaqus, whereby the non-local variable $\varepsilon_{\mathrm{nl}}$ takes the place of temperature variable. In particular, the step needs to be configured as below:

Input File Usage:

```
*Step, name=Step-1, nlgeom=YES, inc=%#INC
*Coupled Temperature-displacement, creep=none, steady
   state
%INITINC, %TTOT, %MININC, %MAXINC
```

The values %#INC, %INITINC, %TTOT, %MININC and %MAXINC are interpreted by Abaqus in the standard way as number of increment, initial time increment, total step time, minium and maximum time increment.

Recommended values in step configuration:

```
*Step, name=Step-1, nlgeom=YES, inc=5000
*Coupled Temperature-displacement, creep=none, steady
   state
0.001, 1., 1e-12, 0.01
```

Recommended values for severe nonlinearity with automatic stabilization:

```
*Step, name=Step-1, nlgeom=YES, inc=5000
*Coupled Temperature-displacement, creep=none, steady
   state, stabilize, factor=0.0002
0.001, 1., 1e-12, 0.01
```

Abaqus/CAE Usage: Step module:

- **Create Step: General: Coupled temp-displacement: Basic: Response: Steady State; Incrementation: Type: Automatic**

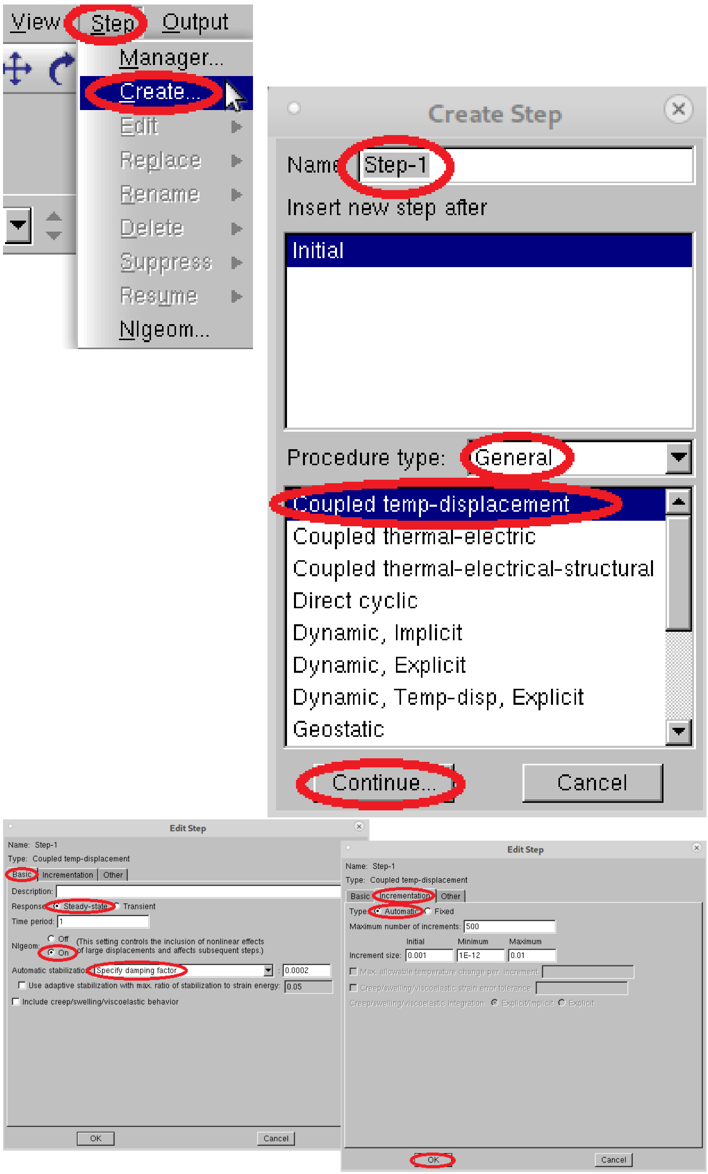

Fig. 3: Step configuration in Abaqus/Standard

In analyses with severe nonlinearities such as simulation for Compact-Tension-Test (CT-Test) or Middle-Tension-Test (MT-Test), the combination between automatic stabilization and solution control parameters could be used to obtain the solution. The recommended values for automatic stabilization are illustrated above. The non-standard values for some parameters in solution control parameters should be adjusted as below:

Input File Usage:

```
*Controls, reset
*Controls, parameters=time incrementation
, , , , , , , %IA, , ,
, , , , , , %DD,
*Controls, parameters=field, field=temperature
, %CA, , , , , ,
```

The values %IA, %DD and %CA are interpreted as maximum number of attempts allowed for an increment, increase factor when two consecutive increments converge in a small number of equilibrium iterations and convergence criterion for the ratio of the largest solution correction to the largest corresponding incremental solution value respectively in Abaqus.

Recommended values for solution control parameters:

```
*Controls, reset
*Controls, parameters=time incrementation
, , , , , , , 20, , ,
, , , , , , 4.,
*Controls, parameters=field, field=temperature
, 1., , , , , ,
```

Abaqus/CAE Usage: Step module:

- **Other > General > Solution Controls > Edit:** toggle on **Specify: Time Incrementation**
- **Other > General > Solution Controls > Edit:** toggle on **Specify: Field Equations: Specify individual fields:** Temperature

2.2.2 Step Definition for Abaqus/Explicit

For using the implementation in Abaqus/Explicit, the Step has to be defined as below:

Input File Usage:

```
*Step, name=Step-1, nlgeom=YES
*Dynamic Temperature-displacement, Explicit
, %TTOT
```

The values %TTOT are interpreted by Abaqus in the standard way as total step time, minium.

The total step time needs to be chosen such that it should statify not only the quasi-static condition but also the reasonable computational cost.

2.3 Mesh requirement

In contrast to the local GTN model, the present nonlocal modification leads to mesh convergent solutions (h-convergence). A sufficiently fine mesh in **damage zone** is required to reach this state of mesh convergence. Previous investigations have established the following requirements:

- Element size in damage zone $b_e \leqslant \frac{1}{4} l_{\mathrm{nl}}$
- Width of fine-meshed region in damage zone $w = 4\, l_{\mathrm{ln}}$

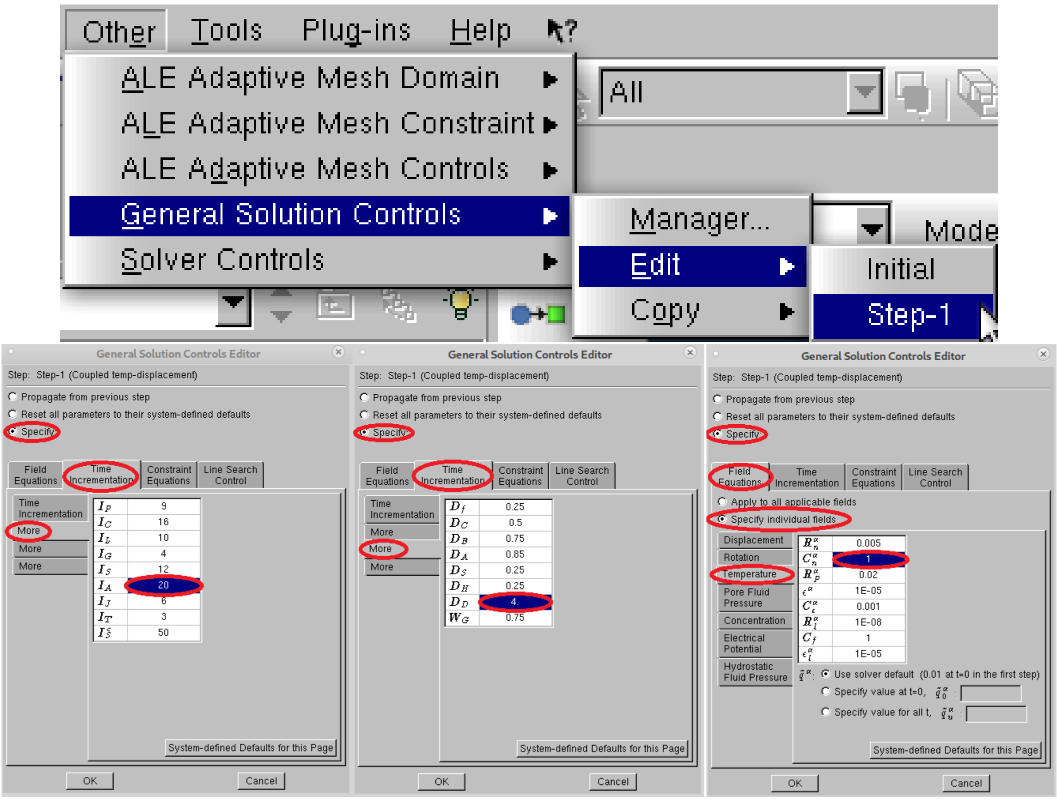

Fig. 4: Configuration and recommended values for solution control paramter in Abaqus/Standard

- Element dimension along crack front (in 3D problems) according to curvature of crack front
- Radius at the initial crack tip $r_t \sim (0.07 \ldots 0.7) l_{nl}$ (collapsed elements around crack tip may work as well)

The following element types in Abaqus have been tested successfully with the present implementation:

Abaqus/Standard

- Plane strain: CPE4T, CPE8RT. Element type CPE4RT is **not recommended**.
- Axisymmetric elements: CAX8RT. Element type CAX4RT is **not recommended**.
- 3D: C3D8T, C3D20RT

Abaqus/Explicit

- Plane strain: CPE4RT
- Axisymmetric elements: CAX4RT
- 3D: C3D20RT

2.4 Strain hardening

There exist 2 options to define stress-strain curve in the present implementation, either via a one-parametric law or by a tabular form. While the tabular form is prefered, when the experimental data is available, the one-parametric law is used for the case that only yield stress *(σ_y, $R_{p0.2}$)* and ultimate yield strength R_m are known. Both options are interchangeable, when the experimental data can be fitted through the one-parametric law or the other way around. The implicit expression of one-parametric law can be presented as below, refer to [3] for more detail:

$$\bar{\sigma} = \sigma_{\mathrm{y}} \left[\frac{\bar{\sigma}}{\sigma_{\mathrm{y}}} + \frac{E}{\sigma_{\mathrm{y}}} \bar{\varepsilon} \right]^{1/n} . \tag{2}$$

The one-parametric law is implemented in UMAT as default option and is activated with the argument *constants=13*, which is given in the line of keyword **User Material* in the *.inp-file*. Correspondingly, the 13 constants are required in the following data lines in the *.inp-file* as illustrated in **Section 2.1** or below . This option requires 2 parameters: initial yield stress *<SigmaY>* and power law exponent *<n>* as inputs. Those are #PROPS 12 and #PROPS 13 in **Tab. 1**.

Input File Usage:

```
*User Material, constants=13, unsymm
<Young>,<Nu>,<f0>,<fc>,<kappa>,<q1>,<q2>,<Damagecase>
<fn>,<S>,<En>,<SigmaY>,<n>
```

If more than 13 #PROPS are provided in the data lines, the routine assumes that the stress-strain curve is described in tabular form, whereby pairs of true stress(*%t_stress*) and plastic strain (*%pl_strain*) are appended one after another following the 13^{th} parameter <n> in the second data line under keyword **User Material*. Subsequently, one should change the value in argument *constants* accordingly. For the below illustration, this argument takes value of 21 (*constants=21*) and 4 pairs of true stress and plastic strain are required. Once the tabular form option is activated, the value of properties *<SigmaY>,<n>* will be ignored.

Input File Usage:

```
*User Material, constants=21, unsymm
<Young>,<Nu>,<f0>,<fc>,<kappa>,<q1>,<q2>,<Damagecase>
<fn>,<S>,<En>,<SigmaY>,<n>,%t_stress,%pl_strain,%t_stress
%pl_strain,%t_stress,%pl_strain,%t_stress,%pl_strain
```

Obviously, both options can also be configured in Abaqus/CAE in the same manner as described in **Section 2.1**.

2.5 Further requirements in Abaqus/Explicit

2.5.1 Value of Specific Heat

The implementation for Abaqus/Standard is based on the mathematical equivalence between the Helmholtz PDE and the PDE of stationary heat transfer. The time integration algorithm of Abaqus/Explicit allows to use only the instationary heat equation. In terms of the aforementioned mathematical equivalence, the Helmholtz for the nonlocal strain has thus to be extended as

$$l_{\mathrm{nl}}^2 \Delta_{\mathrm{x}} \varepsilon_{\mathrm{nl}} - (\varepsilon_{\mathrm{nl}} - \varepsilon_{\mathrm{l}}) = \rho c \dot{\varepsilon}_{\mathrm{nl}} \qquad \forall x \in \Omega. \tag{3}$$

Here, ρ and c are material density and a (nominal) specific heat, respectively. In order to recover the original form, the value of c, which is interpreted as a purely numerical parameter in this context, has to be chosen sufficiently small. However, in the explicit time integration scheme, the value of c limits the allowed time increment

$$\Delta t \leqslant min(\frac{2}{\omega_{\mathrm{max}}}, \frac{2}{\lambda_{\mathrm{max}}}), \tag{4}$$

where ω_{max} is the highest frequency in the system of equations of the mechanical solution response and λ_{max} is the largest eigenvalue in the system of equations of the thermal solution response. Exact values of these eigenvalues are not available, but the common estimates for both parts are

$$\Delta t_{\mathrm{mech,min}} \approx \frac{L_{\mathrm{min}}}{c_d}, \tag{5}$$

$$\Delta t_{\mathrm{therm,min}} \approx \frac{L_{\mathrm{min}}^2}{2\alpha}. \tag{6}$$

related to the smallest element dimension L_{min} in the mesh to the material properties of longitudinal wave speed $c_d = \sqrt{E/\rho}$ and thermal diffusivity $\alpha = \frac{k}{\rho c}$. The latter is defined through thermal conductivity k (being identical to l_{nl}^2 in the present heat analogy), density ρ and specific heat c. Thus, the smallest possible value of c, which avoids additional limitations on the allowed time increment $\Delta t_{\mathrm{mech,min}} = \Delta t_{\mathrm{therm,min}}$, is

$$c = \frac{2 l_{\mathrm{nl}}^2}{L_{\mathrm{min}}\sqrt{E\rho}} \tag{7}$$

For the parameters for steel in the examples ($E = 200000\,\mathrm{MPa}$, $\rho = 7.85 \cdot 10^{-9}$ t/mm^3 $l_{\mathrm{nl}} = 0.1\,\mathrm{mm}$) with the given mesh recommendation $L_{\mathrm{min}}/l_{\mathrm{nl}} = 0.25$ a value of $c \gtrsim 20$ s/(t/mm^3).

2.5.2 Quasi-static condition

The nonlocal GTN model is rate-independent. Thus, the only time scale arises from the inertia which, however, limit the allowed time increment as outlined in the previous section. For performing quasi-static simulations in reasonable computation times, the load should thus be applied in short Step Time, or (equivalently) Mass Scaling is activated. However, both methods lead to an articial amplification of oscillations. In the attached examples, a reasonable trade-off was found with a step time 0.0055 s for NTT-Test, an 0.05 s for SENB-Test.

2.5.3 Mass Scaling & Time Scaling

Mass scaling and time scaling are often used in Abaqus/Explicit for computational efficiency. However, these options should be used with caution to avoid non-physical and noise in the output results. The following shows the setting for both options, which will deliver the correct outputs for the examples included in the download-material, while the simulation time reduces dramatically to half of the normal simulation time for the same hardware configuration.

Mass Scaling

Input File Usage:

```
*Fixed Mass Scaling, dt=1e-08, type=below min, factor=1.5
```

Abaqus/CAE Usage: Step module:

- **Create Step: General, Dynamic, Temp-disp, Explicit: Mass scaling: Use scaling definitions below: Create: Semi-automatic mass scaling, Scale: At beginning of step**

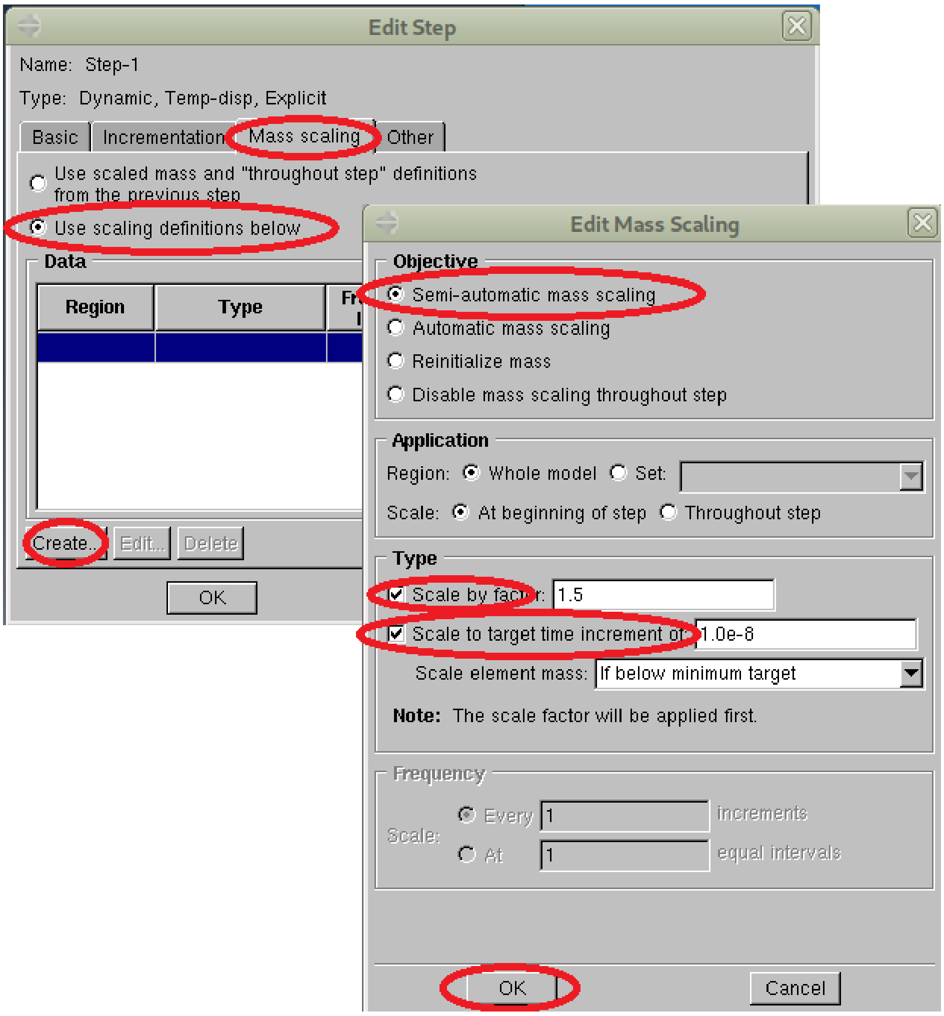

Fig. 5: Mass scaling setting during step configuration

Time Scaling

Input File Usage:

```
*Dynamic Temperature-displacement, Explicit, element by
   element, scale factor=1.5
```

Abaqus/CAE Usage: Step module:

- **Create Step: General, Dynamic, Temp-disp, Explicit: Incrementation: Automatic, Stable increment estimator: Element-by-element, Time scaling factor: 1.5**

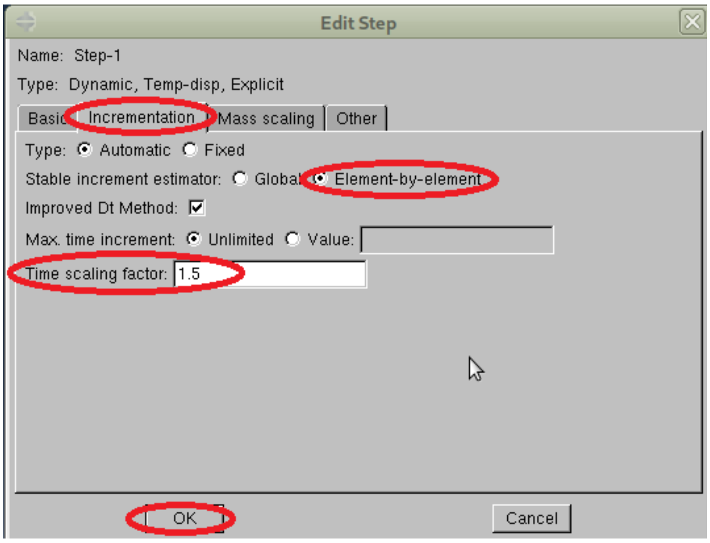

Fig. 6: Time scaling setting during step configuration

2.5.4 Hourglass control

All available elements (CPE4RT, CAX4RT) in Abaqus/Explicit employ a reduced integration scheme, thus necessitating hourglass control. From both available options ("Enhanced" and "Relax Stiffness"), default option "Relax Stiffness" is recommended, which have been verified to give virtually the same results for the provided examples as CPE8RT or CAX8RT elements in Abaqus/Standard (with the same mesh).

Input File Usage:

```
*Section Controls, name=EC-1, hourglass=RELAX STIFFNESS
1., 1., 1.
```

2.5.5 Contact Definition

For those simulations using contact during analysis such as Single-Edge-Notched-Bend-Test (SEND-Test) or Compact-Tension-Test (CT-Test) or Small-Punch-Test (SPT), these exists the difference of contact definition between Abaqus/Standard and Abaqus/Explicit, which should be taken care of.

2.6 Local GTN-Model in Abaqus/Standard

Within the implementation through UMAT subroutine in Abaqus/Standard, users do have an option to activate the local GTN-model for ductile damge, where the value of 1.0 is given to PROPS #8, as listed in **Tab. 1** for variable *<Damagecase>*. This local GTN-model is the basis as well as motivation for developing the nonlocal GTN-model because of its well-known disadvantage - the spurious mesh dependency. This local GTN-model performs well with respect to the available Gurson-model in Abaqus/Standard and Abaqus/Explicit.

2.6.1 Material definition

Only 2 keywords **Depvar* and **User material* are needed for local GTN-model under material property definition.

Input File Usage:

```
*Material, name=Material-1
*Depvar
 11,
*User Material, constants=13, unsymm
<Young>,<Nu>,<f0>,<fc>,<kappa>,<q1>,<q2>,1.0
<fn>,<S>,<En>,<SigmaY>,<n>
```

2.6.2 Step Definition

The analysis procedure should be changed to **General, Static**. Subsequently, the element type should be adjusted with repect to static procedure. The following element types are well tested with this local GTN-model: CPE8R, CPE4R, CAX8R, CAX4R.

Input File Usage:

```
*Step, name=Step-1, nlgeom=YES, inc=%#INC
*Static
%INITINC, %TTOT, %MININC, %MAXINC
```

3 Examples

Notched-tensile test and single-edge-notched bend test are 2 standard experiments for fracture toughness. These 2 experiments will be provided as examples for running simulation with UMAT subroutine in Abaqus/Standard and VUMAT subroutine in Abaqus/Explicit. The *.inp-file* can be generated directly from Abaqus/CAE using the attached cae files. Then all necessary files for each example should be located in the same folder for running simulation with command line as mentioned in **Section 4**. Analogously, the *CAE-file* and source code *UMAT_GTN.f or VUMAT_GTN.f* should be located in the same folder for running simulation though Abaqus/CAE interface. The simulation results are given for comparison.

Tab. 3: Examples for Abaqus/Standard

Example	Strain hardening	necessary files
NOTCHED_TENSILE_TEST	power law	R4_MESH_POWERLAW.inp, UMAT_GTN.f

Example	Strain hardening	necessary files
NOTCHED_TENSILE_TEST	tabular form	R4_MESH_TABLE.inp, UMAT_GTN.f
SENB_TEST	power law	SENB_POWERLAW.inp, UMAT_GTN.f
SENB_TEST	tabular form	SENB_TABLE.inp, UMAT_GTN.f

Tab. 4: Examples for Abaqus/Explicit

Example	Strain hardening	necessary files
NOTCHED_TENSILE_TEST	power law	R4_MESH_POWERLAW_EX.inp, VUMAT_GTN.f
NOTCHED_TENSILE_TEST	tabular form	R4_MESH_TABLE_EX.inp, VUMAT_GTN.f
SENB_TEST	power law	SENB_POWERLAW_EX.inp, VUMAT_GTN.f
SENB_TEST	tabular form	SENB_TABLE_EX.inp, VUMAT_GTN.f

4 Job Execution

Simulations can be started the usual way, either via command line or from within Abaqus/CAE. It has to be ensured only that the source file `UMAT_GTN.f` and `VUMAT_GTN.f` is provided. In the command line, this is done as

- abaqus job=*<.inp_file_name>* user=UMAT_GTN.f

for Abaqus/Standard. Identically, the job can be started in Abaqus/Explicit with the following command line:

- abaqus job=*<.inp_file_name>* user=VUMAT_GTN.f double

The command line option "double" are explicitly required for reducing round-off error once the number of increment exceeds 300000. In Abaqus/CAE, the source file has to be entered in the Job dialogue as shown below:

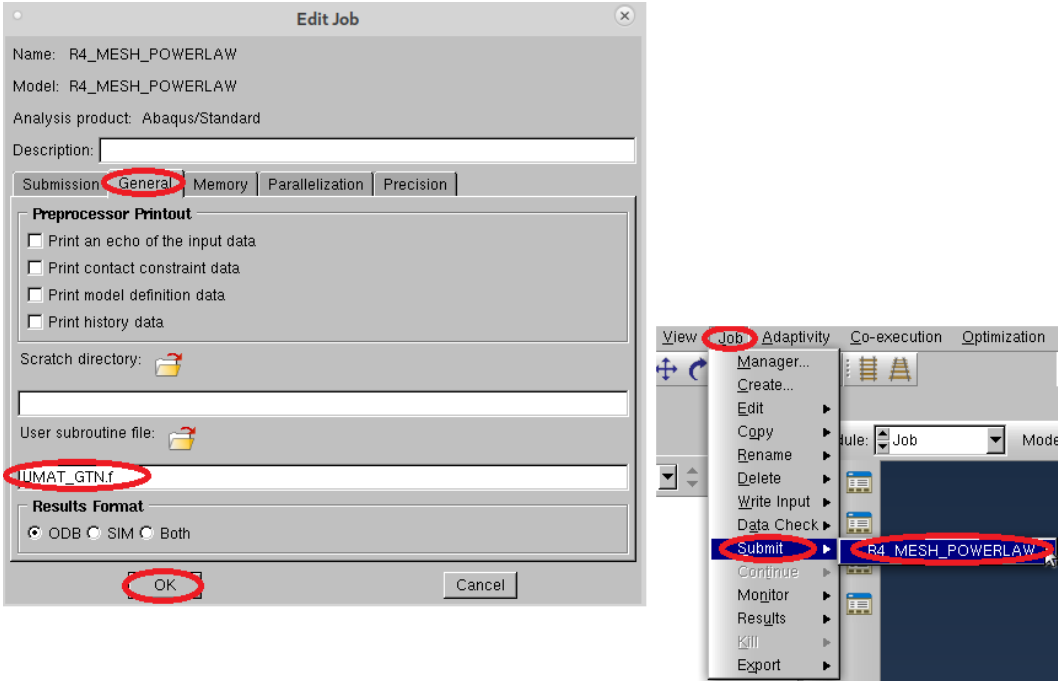

Fig. 7: Including UMAT into simulation and job submission in Abaqus/Standard

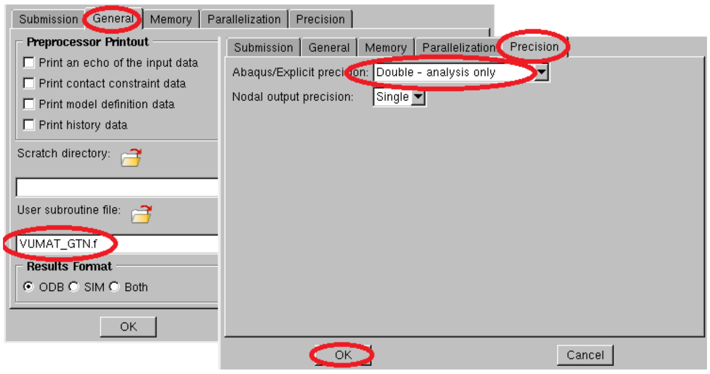

Fig. 8: Including VUMAT and "double" precision option in Abaqus/Explicit

5 Version History

Date	GIT-Version	Released	Editor	Remarks
01.05.2021	96cd5056	-	AS	First establishment for GTN-model in UMAT subroutine
25.11.2021	2b75583e	v0.90	RP	Nonlocal GTN-Model in Fortran fixed-form
17.05.2022	4a637acb 7f1ffa1b	v1.0	RP	Added: VUMAT for Abaqus/Explicit Support for local GTN-model in UMAT

References

[1] G. Hütter, T. Linse, U. Mühlich and M. Kuna: Simulation of Ductile Crack Initiation and Propagation by means of a Non-local GTN-model under Small-Scale Yielding, International Journal of Solids and Structures, 50 (2013), 662–671.

[2] A. Seupel, G. Hütter, and M. Kuna: On the identification and uniqueness of constitutive parameters for a non-local GTN-model, Engineering Fracture Mechanics, 229 (2020), 106817.

[3] R.D. Pham, O. El Khatib, A. Seupel, G. Hütter and B. Kiefer: Non-iterative determination of the parameters for the Gurson model from two standard tests, DVM-Bericht 254 (2022), 43-52

[4] O. El Khatib, G. Hütter, D.R. Pham, A. Seupel, M. Kuna and B. Kiefer : A non-iterative parameter calibration procedure for the GTN model based on standardized experiments, *in preparation*.