Algorithm 1 Domain integration on cut element by triangulation

Given: `grainPart`, `method='subdivision'`, `quadratureOrder`
`cutDomain` $\leftarrow$ find the cut domain which belongs to `grainPart`
for all connected components of `cutDomain` **do**
    Triangulate the connected set (e.g. CDT)
    for all triangles **do**
        Calculate (constant) Jacobian J_{q}
        for all quadrature point x_{qp} in triangle **do**
            $x = \chi_2(x_{\mathrm{qp}})$ ▷ map from integration domain to physical domain
            $x_{\mathrm{ref}} = \chi^{-1}(x)$ ▷ map from physical domain to reference domain
            $\nabla_x f(x_{\mathrm{ref}}) = \mathbf{J}^{-1}(x_{\mathrm{ref}}) \cdot \nabla_{x_{\mathrm{ref}}} f(x_{\mathrm{ref}})$ ▷ gradient in physical coord.
            Form $B(x_{\mathrm{ref}})$ from $\nabla_x f(x_{\mathrm{ref}})$ ▷ strain-displacement matrix
            Update $\mathbf{B}^\top \mathbf{C} \mathbf{B} \det J_{\mathrm{q}} w_{\mathrm{qp}}$ ▷ additivity of integral
        end for
    end for
end for

Algorithm 2 Monolithic creation of the stiffness matrix

```
1: Given: storeRequest (true/false)          ▷ store locally assembled matrices/vectors
2: for all subdomains do
3:     Determine local (subdomain) → global mapping
4:     for all grains do
5:         for all elements do
6:             Calculate K^elem
7:             Determine element → local (subdomain) mapping
8:             if storeRequest then
9:                 Assemble K^elem to K^local
10:                Store K^local
11:            end if
12:            Assemble K^elem to K^global
13:        end for
14:    end for
15: end for
```

- Original algorithm

Algorithm 3 Find elements with local maximum stress

1: Given: element set E, element connectivity N
2: $S = \{e \in E \mid \sigma_1 > \sigma_c\}$ ▷ possibly failing elements
3: **while** $S \neq \emptyset$ **do** ▷ until all admissible elements are placed into a cluster
4: $e = \arg\max_{e \in S} \sigma_1(S)$ ▷ element with locally largest stress value
5: $Q = \{e\}$ ▷ admissible elements in the current layer
6: $S_n = \{e\}$ ▷ container for admissible elements in a cluster
7: **while** $Q \neq \emptyset$ **do** ▷ until there are admissible elements locally
8: $Q = \{e \in N(Q) \mid \sigma_c \leq \sigma(e) \leq \sigma_1(Q)\}$ ▷ criterion for neighbouring elements
9: $S_n = S_n \cup Q$ ▷ accumulate elements in the local cluster
10: **end while**
11: $S = S \setminus S_n$ ▷ remove cluster elements from the whole
12: **end while**

- Create the clusters S_n (including the elements in σ_1-decreasing order) of unconnected element sets where the stress criterion is locally fulfilled
- Breadth-first search on the element connectivity graph

- Modified algorithm

Algorithm 4 Find elements with local maximum stress

1: Given: element set E, element connectivity N, cut elements C
2: $S = \{e \in E \mid \sigma_e > \sigma_c\}$ ▷ possibly failing elements, other stress criterion
3: **while** $S \neq \emptyset$ **do** ▷ until all admissible elements are placed into a cluster
4: $e = \arg\max_{e \in S} \sigma_e(S)$ ▷ element with locally largest stress value
5: $Q = \{e\}$ ▷ admissible elements in the current layer
6: $S_n = \{e\}$ ▷ container for admissible elements in a cluster
7: **while** $Q \neq \emptyset$ **do** ▷ until there are admissible elements locally
8: $Q = \{e \in N(Q) \wedge e \in C \mid \sigma_c \leq \sigma(e) \leq \sigma_e(Q)\}$ ▷ criterion for neighbouring elements
9: $S_n = S_n \cup Q$ ▷ accumulate elements in the local cluster
10: **end while**
11: $S = S \setminus S_n$ ▷ remove cluster elements from the whole
12: **end while**

- Create the clusters S_n (including the elements in σ_1-decreasing order) of unconnected element sets where the stress criterion is locally fulfilled
- Breadth-first search on the element connectivity graph