

Ikarus::NonLinearElastic  
::calculateMatrixImpl

Ikarus::NonLinearElastic  
::calculateScalarImpl

Ikarus::NonLinearElastic  
::calculateVectorImpl

Ikarus::NonLinearElastic  
::strainFunction

```
graph LR; A["Ikarus::NonLinearElastic::calculateMatrixImpl"] --> D["Ikarus::NonLinearElastic::strainFunction"]; B["Ikarus::NonLinearElastic::calculateScalarImpl"] --> D; C["Ikarus::NonLinearElastic::calculateVectorImpl"] --> D;
```

The diagram illustrates a design pattern where three separate implementation methods (calculateMatrixImpl, calculateScalarImpl, and calculateVectorImpl) are all directed towards a single, central function (strainFunction). This suggests that the strainFunction method likely acts as a dispatcher or a base function that delegates the specific calculation tasks to the appropriate implementation based on the input or context.