

Ikarus::KirchhoffLoveShell
::calculateMatrixImpl

Ikarus::KirchhoffLoveShell
::calculateScalarImpl

Ikarus::KirchhoffLoveShell
::calculateVectorImpl

Ikarus::KirchhoffLoveShell
::displacementFunction

```
graph LR; A[Ikarus::KirchhoffLoveShell::calculateMatrixImpl] --> D[Ikarus::KirchhoffLoveShell::displacementFunction]; B[Ikarus::KirchhoffLoveShell::calculateScalarImpl] --> D; C[Ikarus::KirchhoffLoveShell::calculateVectorImpl] --> D;
```

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