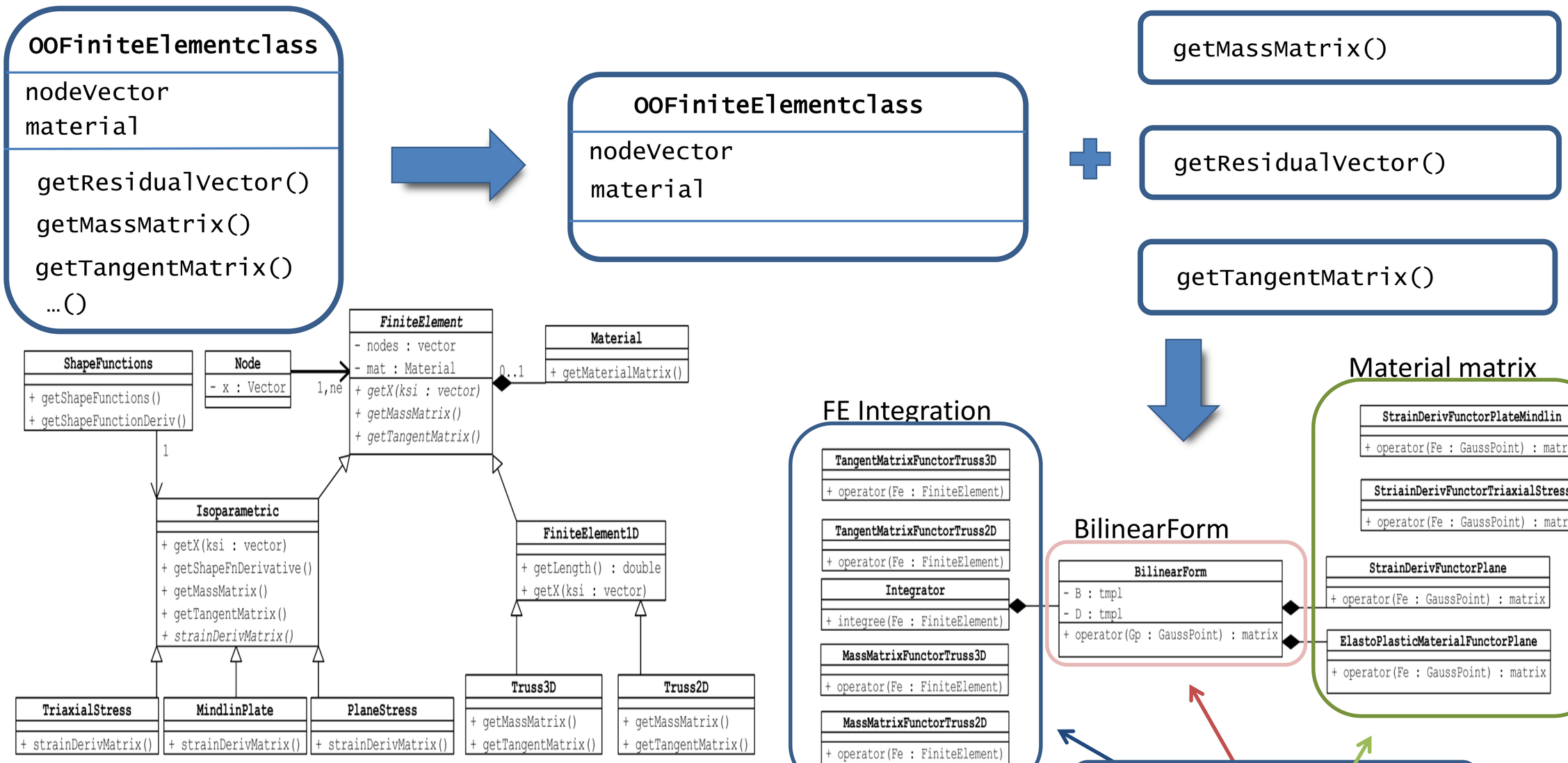
STRESS CONSTRAINED STRUCTURAL TOPOLOGY OPTIMIZATION WITH FUNCTOR-ORIENTED FINITE ELEMENT IMPLEMENTATION

P. Tauzowski[1], B. Błachowski[1], and J. Logo[2]

[1]*Institute of Fundamental Technological Research, Polish Academy of Sciences*
[2]*Budapest University of Technology and Economics, Budapest, Hungary*

Decomposing classical OO classes into functor classes



Classical programming of finite elements contains usual class, which duty is not only to approximate some physical field of interest (displacements, accelerations or temperature), but also definition of matrices necessary for particular analysis. It often leads to sophisticated class hierarchy of finite elements. In our approach matrices necessary for FE analysis are in separate classes. Hierarchy of these classes can be developed almost separately from declaration of the finite element class. Also finite elements hierarchy is much smaller, because each class represents one kind of matrix computed in FE analysis. In our opinion the functor is best suited object for this kind of approach. The functor represents one subroutine and it can also be invoked as function. The study presents application of functor oriented programming to finite element analysis. Functor represents one subroutine and also it can be invoked as function:

`functor(FiniteElement)`.

```
template< class Gp, class B, class D >
class FnBilinearForm
{
    B m_B;
    D m_D;
public:
     FnBilinearForm( const B &b, const D &d ):m_B( b ), m_D( d ) { }
    ~FnBilinearForm() {  }
     const matrix& operator()( Gp *gp, const mvector &xi, const matrix &J );
};

template< class FEi, class B, class D >
const matrix& FnBilinearForm<FEi,B,D>::operator()( Gp *gp, const mvector &xi, const matrix &J )
{
    return m_value.dTrBDB( m_B( gp, xi, J ), mD( gp, xi, J ) );
};
```

```
template <class T>
class TElemFunctor
{
public:
  TElemFunctor( const DTvec &dv):m_dofs( dv){ }
  virtual TElemFunctor* Clone() const = 0;
  const T& GetValue() const { return m_value; }

  virtual const T& operator()  (  const CFEInstance &ielem  ) = 0;
protected:
    T m_value;
};
```

Topology optimization formulation

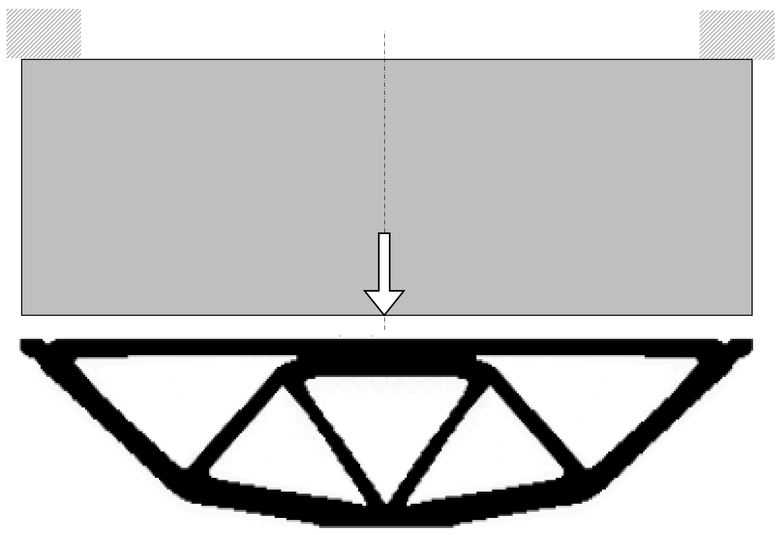
Find minimum of $V(\boldsymbol{\rho}) = \sum_e V_e(\rho_e)$

Subject to $\mathbf{Ku} - \mathbf{f} = \mathbf{0}$

$|\boldsymbol{\sigma}_{red}| \leq \sigma_0 \mathbf{I}_\rho$

$\mathbf{I}_\rho \leq \boldsymbol{\rho} \leq \rho_{max} \mathbf{I}_\rho$

Benchmark example



Topology Optimization Algorithm