

```

/** This file is part of the MOOSE framework
/** https://www.mooseframework.org
/**
/** All rights reserved, see COPYRIGHT for full restrictions
/** https://github.com/idaholab/moose/blob/master/COPYRIGHT
/**
/** Licensed under LGPL 2.1, please see LICENSE for details
/** https://www.gnu.org/licenses/lgpl-2.1.html

#include "ComputeElasticityTensor.h"
#include "libmesh/libmesh.h"
#include <algorithm>
#include <cmath>
#include "libmesh/point.h"
#include "libmesh/utility.h"

registerMooseObject("SolidMechanicsApp", ComputeElasticityTensor);

InputParameters
ComputeElasticityTensor ::validParams()
{
  InputParameters params = ComputeElasticityTensorBase::validParams();
  params.addClassDescription("Compute an isotropic elasticity tensor for elastic constants that "
    "change as a function of material properties");
  params.addRequiredParam<Real>("K", "K for the material.");
  params.addParam<Real>("pa", 98100, "pa for the material.");
  params.addParam<Real>("n", 1, "n for the material.");

  return params;
}

ComputeElasticityTensor ::ComputeElasticityTensor(const InputParameters & parameters)
: ComputeElasticityTensorBase(parameters),

  _K(getParam<Real>("K")),
  _pa(getParam<Real>("pa")),
  _n(getParam<Real>("n")),

  _S30(declareProperty<Real>("S30")),
  _S30_old(getMaterialPropertyOld<Real>(_base_name + "S30")),

  _stress_old(getMaterialPropertyOld<RankTwoTensor>(_base_name + "stress"))
{
  // all tensors created by this class are always isotropic
  issueGuarantee(_elasticity_tensor_name, Guarantee::ISOTROPIC);

```

```
}
```

```
void  
ComputeElasticityTensor ::initQpStatefulProperties()
```

```
{  
  _S30[_qp] = 0.0;  
}
```

```
void  
ComputeElasticityTensor ::computeQpElasticityTensor()
```

```
{  
  Point direction_dummy;  
  Real ps3 = -RankTwoScalarTools::maxPrincipal(_stress_old[_qp], direction_dummy);
```

```
  Real Et = _K * _pa * pow(_S30_old[_qp] / _pa, _n);
```

```
  _Cijkl.fillSymmetricIsotropicEandNu(Et, nu);  
  _elasticity_tensor[_qp] = _Cijkl;
```

```
  if (ps3 > _S30_old[_qp])
```

```
  {  
    _S30[_qp] = ps3;  
  }
```

```
  else
```

```
  {  
    _S30[_qp] = _S30_old[_qp];  
  }
```

```
}
```