SfePy Documentation

April 3, 2008

Contents

1 Notation

Ω	volume (sub)domain
Γ	surface (sub)domain
t	time
y	any function
$\underline{y}$	any vector function
$\underline{n}$	unit outward normal
q, s	scalar test function
p, r	scalar unknown or parameter function
$\bar{p}$	scalar parameter function
$\underline{v}$	vector test function
$\underline{w}$, $\underline{u}$	vector unknown or parameter function
$\underline{b}$	vector parameter function
$\underline{\underline{e}}(\underline{u})$	Cauchy strain tensor ($\frac{1}{2}((\nabla u) + (\nabla u)^T)$)
$\underline{f}$	vector volume forces
ρ	density
ν	kinematic viscosity
c	any constant
δ_{ij}, $\underline{\underline{I}}$	Kronecker delta, identity matrix

The suffix "$_0$" denotes a quatity related to a previous time step.
Term names are prefixed according to the following conventions:

dw	discrete weak	terms having a virtual (test) argument and zero or more unknown arguments, used for FE assembling
d	discrete	terms having all arguments known, the result is the scalar value of the integral
di	discrete integrated	like 'd' but the result is not a scalar (e.g. a vector)
dq	discrete quadrature	terms having all arguments known, the result are the values in quadrature points of elements
continued...		

... continued		
de	discrete element	terms having all arguments known, the result is a vector of integral averages over elements (element average of 'dq')

2 List of all terms

d_biot_div	$\int_{\Omega} r \ \alpha_{ij} e_{ij}(\underline{w})$
d_diffusion	$\int_{\Omega} K_{ij} \nabla_i \bar{p} \nabla_j r$
d_div	$\int_{\Omega} \bar{p} \ \nabla \cdot \underline{w}$
d_lin_elastic	$\int_{\Omega} D_{ijkl} \ e_{ij}(\underline{b}) e_{kl}(\underline{w})$
d_surface_dot	$\int_{\Gamma} pr$, $\int_{\Gamma} \underline{u} \cdot \underline{w}$
d_surface_integrate	$\int_{\Gamma} y$, for vectors: $\int_{\Gamma} \underline{y} \cdot \underline{n}$
d_volume	$\int_{\Omega} 1$
d_volume_dot	$\int_{\Omega} pr$, $\int_{\Omega} \underline{u} \cdot \underline{w}$
d_volume_integrate	$\int_{\Omega} y$
d_volume_wdot	$\int_{\Omega} ypr$, $\int_{\Omega} y\underline{u} \cdot \underline{w}$
de_cauchy_strain	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} \underline{\underline{e}}(\underline{w})$
de_cauchy_stress	vector of $\forall K \in \mathcal{T}_h$: $\int_{T_K} D_{ijkl} e_k l(\underline{w})$
de_diffusion_velocity	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} K_{ij} \nabla_j r$
di_volume_integrate_mat	$\int_{\Omega} m$
dq_grad	$(\nabla p)\|_{qp}$
dq_lin_convect	$((\underline{b} \cdot \nabla)\underline{u})\|_{qp}$
dw_biot_div	$\int_{\Omega} q \ \alpha_{ij} e_{ij}(\underline{u})$
dw_biot_div_dt	$\int_{\Omega} q \ \alpha_{ij} \frac{e_{ij}(\underline{u}) - e_{ij}(\underline{u_0})}{\Delta t}$
dw_biot_div_r	$\int_{\Omega} q \ \alpha_{ij} e_{ij}(\underline{w})$
dw_biot_div_th	$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t-\tau) \frac{\mathrm{d}e_{kl}(\underline{u}(\tau))}{\mathrm{d}\tau} \ \mathrm{d}\tau\right] q$
dw_biot_grad	$\int_{\Omega} p \ \alpha_{ij} e_{ij}(\underline{v})$
dw_biot_grad_r	$\int_{\Omega} r \ \alpha_{ij} e_{ij}(\underline{v})$
dw_biot_grad_th	$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t-\tau) \, p(\tau)) \ \mathrm{d}\tau\right] \ e_{ij}(\underline{v})$
dw_convect	$\int_{\Omega} ((\underline{u} \cdot \nabla)\underline{u}) \cdot \underline{v}$
dw_diffusion	$\int_{\Omega} K_{ij} \nabla_i q \nabla_j p$
dw_diffusion_r	$\int_{\Omega} K_{ij} \nabla_i q \nabla_j r$
dw_div	$\int_{\Omega} q \ \nabla \cdot \underline{u}$
dw_div_grad	$\int_{\Omega} \nu \ \nabla \underline{v} : \nabla \underline{u}$
continued...	

… *continued*	
dw_div_r	$\int_{\Omega} q \ \nabla \cdot \underline{w}$
dw_grad	$\int_{\Omega} p \ \nabla \cdot \underline{v}$
dw_grad_dt	$\int_{\Omega} \frac{p-p_0}{\Delta t} \nabla \cdot \underline{v}$
dw_laplace	$c \int_{\Omega} \nabla s \cdot \nabla r$ or $\sum_{K \in \mathcal{T}_h} \int_{T_K} c_K \ \nabla s \cdot \nabla r$
dw_lin_convect	$\int_{\Omega} ((\underline{b} \cdot \nabla) \underline{u}) \cdot \underline{v}$
dw_lin_elastic	$\int_{\Omega} D_{ijkl} \ e_{ij}(\underline{v}) e_{kl}(\underline{u})$
dw_lin_elastic_iso	$\int_{\Omega} D_{ijkl} \ e_{ij}(\underline{v}) e_{kl}(\underline{u})$ with $D_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda \ \delta_{ij}\delta_{kl}$
dw_lin_elastic_r	$\int_{\Omega} D_{ijkl} \ e_{ij}(\underline{v}) e_{kl}(\underline{w})$
dw_lin_viscous	$\int_{\Omega} D_{ijkl} \ e_{ij}(\underline{v}) \frac{e_{kl}(\underline{u}) - e_{kl}(\underline{u}_0)}{\Delta t}$
dw_lin_viscous_th	$\int_{\Omega} \left[\int_0^t \mathcal{H}_{ijkl}(t-\tau) \ \frac{\mathrm{d} e_{kl}(\underline{u}(\tau))}{\mathrm{d}\tau} \ \mathrm{d}\tau \right] e_{ij}(\underline{v})$
dw_mass	$\int_{\Omega} \rho \underline{v} \cdot \frac{\underline{u} - \underline{u}_0}{\Delta t}$
dw_mass_scalar	$\int_{\Omega} qp$
dw_mass_scalar_fine_coarse	$\int_{\Omega} q_h p_H$
dw_mass_scalar_r	$\int_{\Omega} qr$
dw_mass_scalar_variable	$\int_{\Omega} cqp$
dw_mass_vector	$\int_{\Omega} \rho \ \underline{v} \cdot \underline{u}$
dw_permeability_r	$\int_{\Omega} K_{ij} \nabla_j q$
dw_point_lspring	$\underline{f}^i = -k \underline{u}^i \quad \forall$ FE node i in region
dw_st_grad_div	$\gamma \int_{\Omega} (\nabla \cdot \underline{u}) \cdot (\nabla \cdot \underline{v})$
dw_st_pspg_c	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K \ ((\underline{b} \cdot \nabla) \underline{u}) \cdot \nabla q$
dw_st_pspg_p	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K \ \nabla p \cdot \nabla q$
dw_st_supg_c	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K \ ((\underline{b} \cdot \nabla) \underline{u}) \cdot ((\underline{b} \cdot \nabla) \underline{v})$
dw_st_supg_p	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K \ \nabla p \cdot ((\underline{b} \cdot \nabla) \underline{v})$
dw_surface_ltr	$\int_{\Gamma} \underline{v} \cdot \underline{\underline{\sigma}} \cdot \underline{n}$
dw_volume_integrate	$\int_{\Omega} q$
dw_volume_lvf	$\int_{\Omega} \underline{v} \cdot \underline{f}$
dw_volume_wdot	$\int_{\Omega} yqp$, $\int_{\Omega} y \underline{v} \cdot \underline{u}$
dw_volume_wdot_dt	$\int_{\Omega} yq \frac{p-p_0}{\Delta t}$, $\int_{\Omega} y \underline{v} \cdot \frac{\underline{u} - \underline{u}_0}{\Delta t}$
dw_volume_wdot_r	$\int_{\Omega} yqr$, $\int_{\Omega} y \underline{v} \cdot \underline{w}$
dw_volume_wdot_th	$\int_{\Omega} \left[\int_0^t \mathcal{G}(t-\tau) p(\tau) \ \mathrm{d}\tau \right] q$

3 Introduction

Equations in SfePy are built using terms, which correspond directly to the integral forms of weak formulation of a problem to be solved. As an example, let us consider the Laplace equation:

$$c\Delta t = 0 \text{ in } \Omega, \quad t = \bar{t} \text{ on } \Gamma . \tag{1}$$

The weak formulation of (1) is: Find $t \in V$, such that

$$\int_{\Omega} c \, \nabla t : \nabla s = 0, \quad \forall s \in V_0 . \tag{2}$$

In the syntax used in SfePy input files, this can be written as

$$\texttt{dw\_laplace.i1.Omega(coef, s, t) = 0} , \tag{3}$$

which directly corresponds to the discrete version of (2): Find $\boldsymbol{t} \in V_h$, such that

$$\boldsymbol{s}^T (\int_{\Omega_h} c \, \boldsymbol{G}^T \boldsymbol{G}) \boldsymbol{t} = 0, \quad \forall \boldsymbol{s} \in V_{h0} ,$$

where $\nabla u \approx \boldsymbol{G u}$. The integral over the discrete domain Ω_h is approximated by a numerical quadrature, that is named `i1` in our case.

3.1 Term call syntax

In general, the syntax of a term call in SfePy is:

```
<term_name>.<i>.<r>( <arg1>, <arg2>, ... ),
```

where `<i>` denotes an integral name (i.e. a name of numerical quadrature to use) and `<r>` marks a region (domain of the integral). In the following, `<virtual>` corresponds to a test function, `<state>` to a unknown function and `<parameter>` to a known function arguments. We will now describe all the terms available in SfePy to date.

4 Terms in termsMass

4.1 dw_mass

Class: MassTerm
Description: Inertial forces term (constant density).
Definition:

$$\int_{\Omega} \rho \underline{v} \cdot \frac{\underline{u} - \underline{u}_0}{\Delta t}$$

Arguments:

material.rho	ρ
ts.dt	Δt
parameter	$\underline{u}_0$

Syntax: `dw_mass.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

4.2 dw_mass_scalar

Class: MassScalarTerm
Description: Scalar field mass matrix/rezidual.
Definition:

$$\int_\Omega qp$$

Syntax: dw_mass_scalar.<i>.<r>(<virtual>, <state>)

4.3 dw_mass_scalar_fine_coarse

Class: MassScalarFineCoarseTerm
Description: Scalar field mass matrix/rezidual for coarse to fine grid interpolation. Field p_H belong to the coarse grid, test field q_h to the fine grid.
Definition:

$$\int_\Omega q_h p_H$$

Syntax: dw_mass_scalar_fine_coarse.<i>.<r>(<virtual>, <state>, <iemaps>, <pbase>)

4.4 dw_mass_scalar_r

Class: MassScalarRTerm
Description: Scalar field mass rezidual — r is assumed to be known.
Definition:

$$\int_\Omega qr$$

Syntax: dw_mass_scalar_r.<i>.<r>(<virtual>, <parameter>)

4.5 dw_mass_scalar_variable

Class: MassScalarVariableTerm
Description: Scalar field mass matrix/rezidual with coefficient c defined in nodes.
Definition:

$$\int_\Omega cqp$$

Syntax: dw_mass_scalar_variable.<i>.<r>(<material>, <virtual>, <state>)

4.6 dw_mass_vector

Class: MassVectorTerm
Description: Vector field mass matrix/rezidual.
Definition:

$$\int_\Omega \rho \; \underline{v} \cdot \underline{u}$$

Syntax: dw_mass_vector.<i>.<r>(<material>, <virtual>, <state>)

5 Terms in termsBasic

5.1 d_surface_dot

Class: DotProductSurfaceTerm
Description: Surface $L^2(\Gamma)$ dot product for both scalar and vector fields.
Definition:

$$\int_\Gamma pr, \; \int_\Gamma \underline{u} \cdot \underline{w}$$

Syntax: d_surface_dot.<i>.<r>(<parameter_1>, <parameter_2>)

5.2 d_surface_integrate

Class: IntegrateSurfaceTerm
Definition:

$$\int_{\Gamma} y, \text{ for vectors: } \int_{\Gamma} \underline{y} \cdot \underline{n}$$

Syntax: `d_surface_integrate.<i>.<r>( <parameter> )`

5.3 d_volume

Class: VolumeTerm
Description: Volume of a domain. Uses approximation of the parameter variable.
Definition:

$$\int_{\Omega} 1$$

Syntax: `d_volume.<i>.<r>( <parameter> )`

5.4 d_volume_dot

Class: DotProductVolumeTerm
Description: Volume $L^2(\Omega)$ dot product for both scalar and vector fields.
Definition:

$$\int_{\Omega} pr,\ \int_{\Omega} \underline{u} \cdot \underline{w}$$

Syntax: `d_volume_dot.<i>.<r>( <parameter_1>, <parameter_2> )`

5.5 d_volume_integrate

Class: IntegrateVolumeTerm
Definition:

$$\int_{\Omega} y$$

Syntax: `d_volume_integrate.<i>.<r>( <parameter> )`

5.6 d_volume_wdot

Class: WDotProductVolumeTerm
Description: Volume $L^2(\Omega)$ weighted dot product for both scalar and vector fields.
Definition:

$$\int_{\Omega} ypr,\ \int_{\Omega} y\underline{u} \cdot \underline{w}$$

Arguments:

material	weight function y

Syntax: `d_volume_wdot.<i>.<r>( <material>, <parameter_1>, <parameter_2> )`

5.7 di_volume_integrate_mat

Class: IntegrateVolumeMatTerm

Description: Integrate material parameter m over a domain. Uses approximation of y variable.
Definition:

$$\int_\Omega m$$

Arguments:

material	m (can have up to two dimensions)
parameter	y
shape	shape of material parameter
mode	'const' or 'vertex' or 'element_avg'

Syntax: `di_volume_integrate_mat.<i>.<r>( <material>, <parameter>, <shape>, <mode> )`

5.8 dw_volume_integrate

Class: IntegrateVolumeOperatorTerm
Definition:

$$\int_\Omega q$$

Syntax: `dw_volume_integrate.<i>.<r>( <virtual> )`

5.9 dw_volume_wdot

Class: WDotProductVolumeOperatorTerm
Description: Volume $L^2(\Omega)$ weighted dot product operator for scalar and vector (not implemented!) fields.
Definition:

$$\int_\Omega yqp,\ \int_\Omega y\underline{v}\cdot\underline{u}$$

Arguments:

material	weight function y

Syntax: `dw_volume_wdot.<i>.<r>( <material>, <virtual>, <state> )`

5.10 dw_volume_wdot_dt

Class: WDotProductVolumeOperatorDtTerm
Description: Volume $L^2(\Omega)$ weighted dot product operator for scalar and vector (not implemented!) fields.
Definition:

$$\int_\Omega yq\frac{p-p_0}{\Delta t},\ \int_\Omega y\underline{v}\cdot\frac{\underline{u}-\underline{u}_0}{\Delta t}$$

Arguments:

material	weight function y

Syntax: `dw_volume_wdot_dt.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

5.11 dw_volume_wdot_r

Class: WDotProductVolumeOperatorRTerm
Description: Volume $L^2(\Omega)$ weighted dot product operator for scalar and vector (not implemented!) fields (to use on a right-hand side).
Definition:

$$\int_\Omega yqr \,,\; \int_\Omega y \underline{v} \cdot \underline{w}$$

Arguments:

material	weight function y

Syntax: `dw_volume_wdot_r.<i>.<r>( <material>, <virtual>, <parameter> )`

5.12 dw_volume_wdot_th

Class: WDotProductVolumeOperatorTHTerm
Definition:

$$\int_\Omega \left[\int_0^t \mathcal{G}(t-\tau) p(\tau) \, \mathrm{d}\tau \right] q$$

Syntax: `dw_volume_wdot_th.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

6 Terms in termsLaplace

6.1 d_diffusion

Class: DiffusionIntegratedTerm
Description: Integrated general diffusion term with permeability K_{ij} constant or given in mesh vertices.
Definition:

$$\int_\Omega K_{ij} \nabla_i \bar{p} \nabla_j r$$

Syntax: `d_diffusion.<i>.<r>( <material>, <parameter_1>, <parameter_2> )`

6.2 de_diffusion_velocity

Class: DiffusionVelocityTerm
Description: Diffusion velocity averaged in elements.
Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} K_{ij} \nabla_j r$$

Syntax: `de_diffusion_velocity.<i>.<r>( <material>, <parameter> )`

6.3 dw_diffusion

Class: DiffusionTerm
Description: General diffusion term with permeability K_{ij} constant or given in mesh vertices.
Definition:

$$\int_\Omega K_{ij} \nabla_i q \nabla_j p$$

Syntax: `dw_diffusion.<i>.<r>( <material>, <virtual>, <state> )`

6.4 dw_diffusion_r

Class: DiffusionRTerm
Description: General diffusion term with permeability K_{ij} constant or given in mesh vertices. The argument r is a known field (to use on a right-hand side).
Definition:

$$\int_\Omega K_{ij} \nabla_i q \nabla_j r$$

Syntax: `dw_diffusion_r.<i>.<r>( <material>, <virtual>, <parameter> )`

6.5 dw_laplace

Class: LaplaceTerm
Description: Laplace term with c constant or constant per element.
Definition:

$$c \int_\Omega \nabla s \cdot \nabla r \text{ or } \sum_{K \in \mathcal{T}_h} \int_{T_K} c_K \; \nabla s \cdot \nabla r$$

Syntax: `dw_laplace.<i>.<r>( <material>, <virtual>, <state> )`

6.6 dw_permeability_r

Class: PermeabilityRTerm
Description: Special-purpose diffusion-like term with permeability K_{ij} constant or given in mesh vertices (to use on a right-hand side).
Definition:

$$\int_\Omega K_{ij} \nabla_j q$$

Syntax: `dw_permeability_r.<i>.<r>( <material>, <virtual>, <index> )`

7 Terms in termsNavierStokes

7.1 d_div

Class: DivIntegratedTerm
Description: Integrated divergence term (weak form).
Definition:

$$\int_\Omega \bar{p} \; \nabla \cdot \underline{w}$$

Syntax: `d_div.<i>.<r>( <parameter_1>, <parameter_2> )`

7.2 dq_grad

Class: GradQTerm
Description: Gradient term (weak form) in quadrature points.
Definition:

$$(\nabla p)|_{qp}$$

Syntax: `dq_grad.<i>.<r>( <state> )`

7.3 dq_lin_convect

Class: LinearConvectQTerm
Description: Linearized convective term evaluated in quadrature points.
Definition:

$$((\underline{b} \cdot \nabla)\underline{u})|_{qp}$$

Syntax: `dq_lin_convect.<i>.<r>( <parameter>, <state> )`

7.4 dw_convect

Class: ConvectTerm
Description: Nonlinear convective term.
Definition:

$$\int_{\Omega}((\underline{u} \cdot \nabla)\underline{u}) \cdot \underline{v}$$

Syntax: `dw_convect.<i>.<r>( <virtual>, <state> )`

7.5 dw_div

Class: DivTerm
Description: Divergence term (weak form).
Definition:

$$\int_{\Omega} q \ \nabla \cdot \underline{u}$$

Syntax: `dw_div.<i>.<r>( <virtual>, <state> )`

7.6 dw_div_grad

Class: DivGradTerm
Description: Diffusion term.
Definition:

$$\int_{\Omega} \nu \ \nabla \underline{v} : \nabla \underline{u}$$

Syntax: `dw_div_grad.<i>.<r>( <material>, <virtual>, <state> )`

7.7 dw_div_r

Class: DivRTerm
Description: Divergence term (weak form) with a known field (to use on a right-hand side).
Definition:

$$\int_{\Omega} q \ \nabla \cdot \underline{w}$$

Syntax: `dw_div_r.<i>.<r>( <virtual>, <parameter> )`

7.8 dw_grad

Class: GradTerm
Description: Gradient term (weak form).
Definition:

$$\int_{\Omega} p \; \nabla \cdot \underline{v}$$

Syntax: `dw_grad.<i>.<r>( <virtual>, <state> )`

7.9 dw_grad_dt

Class: GradDtTerm
Description: Gradient term (weak form) with time-discretized $\dot{p}$.
Definition:

$$\int_{\Omega} \frac{p - p_0}{\Delta t} \nabla \cdot \underline{v}$$

Arguments:

ts.dt	Δt
parameter	p_0

Syntax: `dw_grad_dt.<i>.<r>( <ts>, <virtual>, <state>, <parameter> )`

7.10 dw_lin_convect

Class: LinearConvectTerm
Description: Linearized convective term.
Definition:

$$\int_{\Omega} ((\underline{b} \cdot \nabla) \underline{u}) \cdot \underline{v}$$

Syntax: `dw_lin_convect.<i>.<r>( <virtual>, <parameter>, <state> )`

7.11 dw_st_grad_div

Class: GradDivStabilizationTerm
Description: Grad-div stabilization term (γ is a global stabilization parameter).
Definition:

$$\gamma \int_{\Omega} (\nabla \cdot \underline{u}) \cdot (\nabla \cdot \underline{v})$$

Syntax: `dw_st_grad_div.<i>.<r>( <material>, <virtual>, <state> )`

7.12 dw_st_pspg_c

Class: PSPGCStabilizationTerm
Description: PSPG stabilization term, convective part (τ is a local stabilization parameter).
Definition:

$$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K \; ((\underline{b} \cdot \nabla) \underline{u}) \cdot \nabla q$$

Syntax: `dw_st_pspg_c.<i>.<r>( <material>, <virtual>, <parameter>, <state> )`

7.13 dw_st_pspg_p

Class: PSPGPStabilizationTerm
Description: PSPG stabilization term, pressure part (τ is a local stabilization parameter), alias to Laplace term dw_laplace.
Definition:

$$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K \ \nabla p \cdot \nabla q$$

Syntax: `dw_st_pspg_p.<i>.<r>( <material>, <virtual>, <state> )`

7.14 dw_st_supg_c

Class: SUPGCStabilizationTerm
Description: SUPG stabilization term, convective part (δ is a local stabilization parameter).
Definition:

$$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K \ ((\underline{b} \cdot \nabla) \underline{u}) \cdot ((\underline{b} \cdot \nabla) \underline{v})$$

Syntax: `dw_st_supg_c.<i>.<r>( <material>, <virtual>, <parameter>, <state> )`

7.15 dw_st_supg_p

Class: SUPGPStabilizationTerm
Description: SUPG stabilization term, pressure part (δ is a local stabilization parameter).
Definition:

$$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K \ \nabla p \cdot ((\underline{b} \cdot \nabla) \underline{v})$$

Syntax: `dw_st_supg_p.<i>.<r>( <material>, <virtual>, <parameter>, <state> )`

8 Terms in termsPoint

8.1 dw_point_lspring

Class: LinearPointSpringTerm
Description: Linear springs constraining movement of FE nodes in a reagion; use as a relaxed Dirichlet boundary conditions.
Definition:

$$\underline{f}^i = -k \underline{u}^i \quad \forall \text{ FE node } i \text{ in region}$$

Syntax: `dw_point_lspring.<i>.<r>( <material>, <virtual>, <state> )`

9 Terms in termsVolume

9.1 dw_volume_lvf

Class: LinearVolumeForceTerm
Description: Linear volume forces (weak form).
Definition:

$$\int_\Omega \underline{v} \cdot \underline{f}$$

Syntax: `dw_volume_lvf.<i>.<r>( <material>, <virtual> )`

10 Terms in termsSurface

10.1 dw_surface_ltr

Class: LinearTractionTerm
Description: Linear traction forces (weak form), where, depending on dimension of 'material' argument, $\underline{\underline{\sigma}} \cdot \underline{n}$ is $\bar{p}\underline{\underline{I}} \cdot \underline{n}$ for a given scalar pressure, $\underline{f}$ for a traction vector, and itself for a stress tensor.
Definition:

$$\int_{\Gamma} \underline{v} \cdot \underline{\underline{\sigma}} \cdot \underline{n}$$

Syntax: `dw_surface_ltr.<i>.<r>( <material>, <virtual> )`

11 Terms in termsLinElasticity

11.1 d_lin_elastic

Class: LinearElasticIntegratedTerm
Description: Integrated general linear elasticity term.
Definition:

$$\int_{\Omega} D_{ijkl}\ e_{ij}(\underline{b}) e_{kl}(\underline{w})$$

Syntax: `d_lin_elastic.<i>.<r>( <material>, <parameter_1>, <parameter_2> )`

11.2 de_cauchy_strain

Class: CauchyStrainTerm
Description: Cauchy strain tensor averaged in elements.
Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} \underline{\underline{e}}(\underline{w})$$

Syntax: `de_cauchy_strain.<i>.<r>( <parameter> )`

11.3 de_cauchy_stress

Class: CauchyStressTerm
Description: Cauchy stress tensor averaged in elements.
Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} D_{ijkl} e_k l(\underline{w})$$

Syntax: `de_cauchy_stress.<i>.<r>( <material>, <parameter> )`

11.4 dw_lin_elastic

Class: LinearElasticTerm
Description: General linear elasticity term, with D_{ijkl} given in the usual matrix form exploiting symmetry: in 3D it is 6×6 with the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it is 3×3 with the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_{\Omega} D_{ijkl}\ e_{ij}(\underline{v}) e_{kl}(\underline{u})$$

Syntax: `dw_lin_elastic.<i>.<r>( <material>, <virtual>, <state> )`

11.5 dw_lin_elastic_iso

Class: LinearElasticIsotropicTerm
Description: Isotropic linear elasticity term.
Definition:

$$\int_\Omega D_{ijkl}\ e_{ij}(\underline{v}) e_{kl}(\underline{u}) \mbox{ with } D_{ijkl} = \mu (\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}) + \lambda \ \delta_{ij}\delta_{kl}$$

Syntax: `dw_lin_elastic_iso.<i>.<r>( <material>, <virtual>, <state> )`

11.6 dw_lin_elastic_r

Class: LinearElasticRTerm
Description: General linear elasticity term with a known field (to use on a right-hand side).
Definition:

$$\int_\Omega D_{ijkl}\ e_{ij}(\underline{v}) e_{kl}(\underline{w})$$

Syntax: `dw_lin_elastic_r.<i>.<r>( <material>, <virtual>, <parameter> )`

11.7 dw_lin_viscous

Class: LinearViscousTerm
Description: General linear viscosity term, with D_{ijkl} given in the usual matrix form exploiting symmetry: in 3D it is 6×6 with the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it is 3×3 with the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_\Omega D_{ijkl}\ e_{ij}(\underline{v}) \frac{e_{kl}(\underline{u}) - e_{kl}(\underline{u}_0)}{\Delta t}$$

Arguments:

ts.dt	Δt
material	D_{ijkl}
virtual	$\underline{v}$
state	$\underline{u}$ (displacements of current time step)
parameter	$\underline{u}_0$ (known displacements of previous time step)

Syntax: `dw_lin_viscous.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

11.8 dw_lin_viscous_th

Class: LinearViscousTHTerm
Definition:

$$\int_\Omega \left[\int_0^t \mathcal{H}_{ijkl}(t-\tau)\,\frac{\mathrm{d} e_{kl}(\underline{u}(\tau))}{\mathrm{d}\tau} \mathrm{d}\tau \right]\,e_{ij}(\underline{v})$$

Syntax: `dw_lin_viscous_th.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

12 Terms in termsBiot

12.1 d_biot_div

Class: BiotDivRIntegratedTerm
Description: Integrated Biot divergence-like term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_{\Omega} r \ \alpha_{ij} e_{ij}(\underline{w})$$

Syntax: `d_biot_div.<i>.<r>( <material>, <parameter_1>, <parameter_2> )`

12.2 dw_biot_div

Class: BiotDivTerm
Description: Biot divergence-like term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_{\Omega} q \ \alpha_{ij} e_{ij}(\underline{u})$$

Syntax: `dw_biot_div.<i>.<r>( <material>, <virtual>, <state> )`

12.3 dw_biot_div_dt

Class: BiotDivDtTerm
Description: Biot divergence-like rate term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_{\Omega} q \ \alpha_{ij} \frac{e_{ij}(\underline{u}) - e_{ij}(\underline{u_0})}{\Delta t}$$

Syntax: `dw_biot_div_dt.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

12.4 dw_biot_div_r

Class: BiotDivRTerm
Description: Biot divergence-like term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$. The argument $\underline{w}$ is a known field (to use on a right-hand side).
Definition:

$$\int_{\Omega} q \ \alpha_{ij} e_{ij}(\underline{w})$$

Syntax: `dw_biot_div_r.<i>.<r>( <material>, <virtual>, <parameter> )`

12.5 dw_biot_div_th

Class: BiotDivTHTerm
Definition:

$$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t-\tau) \frac{\mathrm{d} e_{kl}(\underline{u}(\tau))}{\mathrm{d}\tau} \ \mathrm{d}\tau \right] q$$

Syntax: `dw_biot_div_th.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

12.6 dw_biot_grad

Class: BiotGradTerm
Description: Biot gradient-like term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$.
Definition:

$$\int_{\Omega} p \ \alpha_{ij} e_{ij}(\underline{v})$$

Syntax: `dw_biot_grad.<i>.<r>( <material>, <virtual>, <state> )`

12.7 dw_biot_grad_r

Class: BiotGradRTerm
Description: Biot gradient-like term (weak form) with α_{ij} given in vector form exploiting symmetry: in 3D it has the indices ordered as $[11, 22, 33, 12, 13, 23]$, in 2D it has the indices ordered as $[11, 22, 12]$. The argument r is a known field (to use on a right-hand side).
Definition:

$$\int_{\Omega} r \ \alpha_{ij} e_{ij}(\underline{v})$$

Syntax: `dw_biot_grad_r.<i>.<r>( <material>, <virtual>, <parameter> )`

12.8 dw_biot_grad_th

Class: BiotGradTHTerm
Definition:

$$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t-\tau) \, p(\tau)) \, \mathrm{d}\tau \right] \, e_{ij}(\underline{v})$$

Syntax: `dw_biot_grad_th.<i>.<r>( <ts>, <material>, <virtual>, <state>, <parameter> )`

13 Term caches in cachesBasic

13.1 cauchy_strain

Class: CauchyStrainDataCache

```
cache = term.getCache( 'cauchy_strain', <index> )
data = cache( <data name>, <ig>, <ih>, state )
```

13.2 div_vector

Class: DivVectorDataCache

```
cache = term.getCache( 'div_vector', <index> )
data = cache( <data name>, <ig>, <ih>, state )
```

13.3 grad_scalar

Class: GradScalarDataCache

```
cache = term.getCache( 'grad_scalar', <index> )
data = cache( <data name>, <ig>, <ih>, state )
```

13.4 mat_in_qp

Class: MatInQPDataCache

```
cache = term.getCache( 'mat_in_qp', <index> )
data = cache( <data name>, <ig>, <ih>, mat, ap, assumedShapes, modeIn )
```

13.5 state_in_surface_qp

Class: StateInSurfaceQPDataCache

```
cache = term.getCache( 'state_in_surface_qp', <index> )
data = cache( <data name>, <ig>, <ih>, state )
```

13.6 state_in_volume_qp

Class: StateInVolumeQPDataCache

```
cache = term.getCache( 'state_in_volume_qp', <index> )
data = cache( <data name>, <ig>, <ih>, state )
```

13.7 volume

Class: VolumeDataCache

```
cache = term.getCache( 'volume', <index> )
data = cache( <data name>, <ig>, <ih>, region, field )
```