

SfePy Documentation

September 23, 2008

Contents

1	Notation	3
2	List of all terms	4
3	Introduction	6
3.1	Term call syntax	6
4	Terms in termsMass	6
4.1	dw_mass	6
4.2	dw_mass_scalar	7
4.3	dw_mass_scalar_fine_coarse	7
4.4	dw_mass_scalar_variable	7
4.5	dw_mass_vector	7
5	Terms in termsBasic	7
5.1	d_surface_dot	7
5.2	d_surface_integrate	8
5.3	d_volume	8
5.4	d_volume_dot	8
5.5	d_volume_wdot	8
5.6	de_average_variable	8
5.7	de_volume_average_mat	8
5.8	di_volume_integrate	9
5.9	di_volume_integrate_mat	9
5.10	dw_surface_integrate	10
5.11	dw_volume_integrate	10
5.12	dw_volume_wdot	10
5.13	dw_volume_wdot_dt	10
5.14	dw_volume_wdot_th	10
6	Terms in termsLaplace	11
6.1	d_diffusion	11
6.2	de_diffusion_velocity	11
6.3	dw_diffusion	11
6.4	dw_laplace	11
6.5	dw_permeability_r	11

7	Terms in termsNavierStokes	12
7.1	d_div	12
7.2	dq_grad	12
7.3	dq_lin_convect	12
7.4	dw_convect	12
7.5	dw_div	12
7.6	dw_div_grad	12
7.7	dw_grad	13
7.8	dw_grad_dt	13
7.9	dw_lin_convect	13
7.10	dw_st_grad_div	13
7.11	dw_st_pspg_c	13
7.12	dw_st_pspg_p	14
7.13	dw_st_supg_c	14
7.14	dw_st_supg_p	14
8	Terms in termsHyperElasticity	14
8.1	dw_tl_bulk_penalty	14
8.2	dw_tl_he_mooney_rivlin	14
8.3	dw_tl_he_neohook	15
9	Terms in termsPoint	15
9.1	dw_point_lspring	15
10	Terms in termsVolume	15
10.1	dw_volume_lvf	15
11	Terms in termsPiezo	15
11.1	dw_piezo_coupling	15
12	Terms in termsSurface	15
12.1	dw_surface_ltr	15
13	Terms in termsLinElasticity	16
13.1	d_lin_elastic	16
13.2	de_cauchy_strain	16
13.3	de_cauchy_stress	16
13.4	dw_lin_elastic	16
13.5	dw_lin_elastic_iso	16
13.6	dw_lin_viscous	17
13.7	dw_lin_viscous_th	17
14	Terms in termsBiot	17
14.1	d_biot_div	17
14.2	dw_biot_div	17
14.3	dw_biot_div_dt	18
14.4	dw_biot_div_th	18
14.5	dw_biot_grad	18
14.6	dw_biot_grad_dt	18
14.7	dw_biot_grad_th	19
15	Term caches in cachesFiniteStrain	19
15.1	finite_strain_tl	19

16	Term caches in cachesBasic	19
16.1	cauchy_strain	19
16.2	div_vector	19
16.3	grad_scalar	19
16.4	grad_vector	19
16.5	mat_in_qp	19
16.6	state_in_surface_qp	19
16.7	state_in_volume_qp	20
16.8	volume	20

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 ($\frac{1}{2}((\nabla \underline{u}) + (\nabla \underline{u})^T)$)
$\underline{\underline{F}}$	deformation gradient $F_{ij} = \frac{\partial x_i}{\partial X_j}$
J	$\det(F)$
$\underline{\underline{C}}$	right Cauchy-Green deformation tensor $C = F^T F$
$\underline{\underline{E}}(\underline{u})$	Green strain tensor $E_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_m}{\partial x_i} \frac{\partial u_m}{\partial x_j})$
$\underline{\underline{S}}$	second Piola-Kirchhoff stress tensor
\underline{f}	vector volume forces
f	scalar volume force (source)
ρ	density
ν	kinematic viscosity
c	any constant
$\delta_{ij}, \underline{\underline{I}}$	Kronecker delta, identity matrix

The suffix "0" denotes a quantity 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
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

section	name	definition
(14.1)	d_biot_div	$\int_{\Omega} r \alpha_{ij} e_{ij}(\underline{w})$
(6.1)	d_diffusion	$\int_{\Omega} K_{ij} \nabla_i \bar{p} \nabla_j r$
(7.1)	d_div	$\int_{\Omega} \bar{p} \nabla \cdot \underline{w}$
(13.1)	d_lin_elastic	$\int_{\Omega} D_{ijkl} e_{ij}(\underline{b}) e_{kl}(\underline{w})$
(5.1)	d_surface_dot	$\int_{\Gamma} pr, \int_{\Gamma} \underline{u} \cdot \underline{w}$
(5.2)	d_surface_integrate	$\int_{\Gamma} y$, for vectors: $\int_{\Gamma} \underline{y} \cdot \underline{n}$
(5.3)	d_volume	$\int_{\Omega} 1$
(5.4)	d_volume_dot	$\int_{\Omega} pr, \int_{\Omega} \underline{u} \cdot \underline{w}$
(5.5)	d_volume_wdot	$\int_{\Omega} ypr, \int_{\Omega} y \underline{u} \cdot \underline{w}$
(5.6)	de_average_variable	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} y / \int_{T_K} 1$
(13.2)	de_cauchy_strain	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} \underline{e}(\underline{w}) / \int_{T_K} 1$
(13.3)	de_cauchy_stress	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} D_{ijkl} e_{kl}(\underline{w}) / \int_{T_K} 1$
(6.2)	de_diffusion_velocity	vector of $\forall K \in \mathcal{T}_h : \int_{T_K} K_{ij} \nabla_j r / \int_{T_K} 1$
(5.7)	de_volume_average_mat	$\forall K \in \mathcal{T}_h : \int_{T_K} m / \int_{T_K} 1$
(5.8)	di_volume_integrate	$\int_{\Omega} y, \int_{\Omega} \underline{y}$
(5.9)	di_volume_integrate_mat	$\int_{\Omega} m$
(7.2)	dq_grad	$(\nabla p) _{qp}$
(7.3)	dq_lin_convect	$((\underline{b} \cdot \nabla) \underline{u}) _{qp}$
(14.2)	dw_biot_div	$\int_{\Omega} q \alpha_{ij} e_{ij}(\underline{u})$
(14.3)	dw_biot_div_dt	$\int_{\Omega} q \alpha_{ij} \frac{e_{ij}(\underline{u}) - e_{ij}(\underline{u}_0)}{\Delta t}$
(14.4)	dw_biot_div_th	$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t - \tau) \frac{de_{kl}(\underline{u}(\tau))}{d\tau} d\tau \right] q$
(14.5)	dw_biot_grad	$\int_{\Omega} p \alpha_{ij} e_{ij}(\underline{v})$
<i>continued...</i>		

... continued		
(14.6)	dw_biot_grad_dt	$\int_{\Omega} \frac{p-p_0}{\Delta t} \alpha_{ij} e_{ij}(\underline{v})$
(14.7)	dw_biot_grad_th	$\int_{\Omega} \left[\int_0^t \alpha_{ij}(t-\tau) p(\tau) d\tau \right] e_{ij}(\underline{v})$
(7.4)	dw_convect	$\int_{\Omega} ((\underline{u} \cdot \nabla) \underline{u}) \cdot \underline{v}$
(6.3)	dw_diffusion	$\int_{\Omega} K_{ij} \nabla_i q \nabla_j p$
(7.5)	dw_div	$\int_{\Omega} q \nabla \cdot \underline{u}$
(7.6)	dw_div_grad	$\int_{\Omega} \nu \nabla \underline{v} : \nabla \underline{u}$
(7.7)	dw_grad	$\int_{\Omega} p \nabla \cdot \underline{v}$
(7.8)	dw_grad_dt	$\int_{\Omega} \frac{p-p_0}{\Delta t} \nabla \cdot \underline{v}$
(6.4)	dw_laplace	$c \int_{\Omega} \nabla s \cdot \nabla r$ or $\sum_{K \in \mathcal{T}_h} \int_{T_K} c_K \nabla s \cdot \nabla r$
(7.9)	dw_lin_convect	$\int_{\Omega} ((\underline{b} \cdot \nabla) \underline{u}) \cdot \underline{v}$
(13.4)	dw_lin_elastic	$\int_{\Omega} D_{ijkl} e_{ij}(\underline{v}) e_{kl}(\underline{u})$
(13.5)	dw_lin_elastic_iso	$\int_{\Omega} D_{ijkl} e_{ij}(\underline{v}) e_{kl}(\underline{u})$ with $D_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda \delta_{ij}\delta_{kl}$
(13.6)	dw_lin_viscous	$\int_{\Omega} D_{ijkl} e_{ij}(\underline{v}) \frac{e_{kl}(\underline{u}) - e_{kl}(\underline{u}_0)}{\Delta t}$
(13.7)	dw_lin_viscous_th	$\int_{\Omega} \left[\int_0^t \mathcal{H}_{ijkl}(t-\tau) \frac{de_{kl}(\underline{u}(\tau))}{d\tau} d\tau \right] e_{ij}(\underline{v})$
(4.1)	dw_mass	$\int_{\Omega} \rho \underline{v} \cdot \frac{\underline{u} - \underline{u}_0}{\Delta t}$
(4.2)	dw_mass_scalar	$\int_{\Omega} q p$
(4.3)	dw_mass_scalar_fine_coarse	$\int_{\Omega} q_h p_H$
(4.4)	dw_mass_scalar_variable	$\int_{\Omega} c q p$
(4.5)	dw_mass_vector	$\int_{\Omega} \rho \underline{v} \cdot \underline{u}$
(6.5)	dw_permeability_r	$\int_{\Omega} K_{ij} \nabla_j q$
(11.1)	dw_piezo_coupling	$\int_{\Omega} g_{kij} e_{ij}(\underline{u}) \nabla_k q, \int_{\Omega} g_{kij} e_{ij}(\underline{v}) \nabla_k p$
(9.1)	dw_point_lspring	$\underline{f}^i = -k \underline{u}^i \quad \forall \text{ FE node } i \text{ in region}$
(7.10)	dw_st_grad_div	$\gamma \int_{\Omega} (\nabla \cdot \underline{u}) \cdot (\nabla \cdot \underline{v})$
(7.11)	dw_st_pspg_c	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K ((\underline{b} \cdot \nabla) \underline{u}) \cdot \nabla q$
(7.12)	dw_st_pspg_p	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \tau_K \nabla p \cdot \nabla q$
(7.13)	dw_st_supg_c	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K ((\underline{b} \cdot \nabla) \underline{u}) \cdot ((\underline{b} \cdot \nabla) \underline{v})$
(7.14)	dw_st_supg_p	$\sum_{K \in \mathcal{T}_h} \int_{T_K} \delta_K \nabla p \cdot ((\underline{b} \cdot \nabla) \underline{v})$
(5.10)	dw_surface_integrate	$\int_{\Gamma} q$
(12.1)	dw_surface_ltr	$\int_{\Gamma} \underline{v} \cdot \underline{\underline{\sigma}} \cdot \underline{n}$
(8.1)	dw_tl_bulk_penalty	$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$
(8.2)	dw_tl_he_mooney_rivlin	$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$
(8.3)	dw_tl_he_neohook	$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$
(5.11)	dw_volume_integrate	$\int_{\Omega} q$
(10.1)	dw_volume_lvf	$\int_{\Omega} \underline{f} \cdot \underline{v}$ or $\int_{\Omega} f q$
(5.12)	dw_volume_wdot	$\int_{\Omega} y q p, \int_{\Omega} y \underline{v} \cdot \underline{u}$
(5.13)	dw_volume_wdot_dt	$\int_{\Omega} y q \frac{p-p_0}{\Delta t}, \int_{\Omega} y \underline{v} \cdot \frac{\underline{u} - \underline{u}_0}{\Delta t}$
continued...		

<i>... continued</i>			
(5.14)	dw_volume_wdot_th	$\int_{\Omega} \left[\int_0^t \mathcal{G}(t - \tau) p(\tau) \, d\tau \right]$	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:

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

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

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

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

$$\text{dw_laplace.i1.Omega}(\text{coef}, \mathbf{s}, \mathbf{t}) = 0, \quad (3)$$

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

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

where $\nabla u \approx \mathbf{G}u$. 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:

$$\langle \text{term_name} \rangle . \langle \mathbf{i} \rangle . \langle \mathbf{r} \rangle (\langle \text{arg1} \rangle, \langle \text{arg2} \rangle, \dots),$$

where $\langle \mathbf{i} \rangle$ denotes an integral name (i.e. a name of numerical quadrature to use) and $\langle \mathbf{r} \rangle$ marks a region (domain of the integral). In the following, $\langle \text{virtual} \rangle$ corresponds to a test function, $\langle \text{state} \rangle$ to a unknown function and $\langle \text{parameter} \rangle$ 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>, <pbases>)`

4.4 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>)`

4.5 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_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>)

5.6 de_average_variable

Class: AverageVariableTerm

Description: Variable y averaged in elements.

Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} y / \int_{T_K} 1$$

Syntax: de_average_variable.<i>.<r>(<parameter>)

5.7 de_volume_average_mat

Class: AverageVolumeMatTerm

Description: Material parameter m averaged in elements. Uses approximation of y variable.

Definition:

$$\forall K \in \mathcal{T}_h : \int_{T_K} m / \int_{T_K} 1$$

Arguments:

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

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

5.8 di_volume_integrate

Class: `IntegrateVolumeTerm`

Definition:

$$\int_{\Omega} y, \int_{\Omega} \underline{y}$$

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

5.9 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 parameter
mode	'const' or 'vertex' or 'element_avg'

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

5.10 dw_surface_integrate

Class: IntegrateSurfaceOperatorTerm

Definition:

$$\int_{\Gamma} q$$

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

5.11 dw_volume_integrate

Class: IntegrateVolumeOperatorTerm

Definition:

$$\int_{\Omega} q$$

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

5.12 dw_volume_wdot

Class: WDotProductVolumeOperatorTerm

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

Definition:

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

Arguments:

material	weight function y
----------	---------------------

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

5.13 dw_volume_wdot_dt

Class: WDotProductVolumeOperatorDtTerm

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

Definition:

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

Arguments:

material	weight function y
----------	---------------------

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

5.14 dw_volume_wdot_th

Class: WDotProductVolumeOperatorTHTerm

Definition:

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

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

$$\forall K \in \mathcal{T}_h : \int_{T_K} K_{ij} \nabla_j r / \int_{T_K} 1$$

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_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>)`

6.5 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_grad

Class: GradTerm

Description: Gradient term (weak form).

Definition:

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

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

7.8 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>)

7.9 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>)

7.10 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>)

7.11 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>)

7.12 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>)

7.13 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>)

7.14 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 termsHyperElasticity

8.1 dw_tl_bulk_penalty

Class: BulkPenaltyTerm

Description: Hyperelastic bulk penalty term. Stress $S_{ij} = K(J - 1) J C_{ij}^{-1}$.

Definition:

$$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$$

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

8.2 dw_tl_he_mooney_rivlin

Class: MooneyRivlinTerm

Description: Hyperelastic Mooney-Rivlin term. Effective stress $S_{ij} = \kappa J^{-\frac{4}{3}} (C_{kk} \delta_{ij} - C_{ij} - \frac{2}{3} I_2 C_{ij}^{-1})$.

Definition:

$$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$$

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

8.3 dw_tl_he_neohook

Class: NeoHookeanTerm

Description: Hyperelastic neo-Hookean term. Effective stress $S_{ij} = \mu J^{-\frac{2}{3}}(\delta_{ij} - \frac{1}{3}C_{kk}C_{ij}^{-1})$.

Definition:

$$\int_{\Omega} S_{ij}(\underline{u}) \delta E_{ij}(\underline{u}; \underline{v})$$

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

9 Terms in termsPoint

9.1 dw_point_lspring

Class: LinearPointSpringTerm

Description: Linear springs constraining movement of FE nodes in a region; 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>)

10 Terms in termsVolume

10.1 dw_volume_lvf

Class: LinearVolumeForceTerm

Description: Vector or scalar linear volume forces (weak form) — a right-hand side source term.

Definition:

$$\int_{\Omega} \underline{f} \cdot \underline{v} \text{ or } \int_{\Omega} f q$$

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

11 Terms in termsPiezo

11.1 dw_piezo_coupling

Class: PiezoCouplingTerm

Description: Piezoelectric coupling term.

Definition:

$$\int_{\Omega} g_{kij} e_{ij}(\underline{u}) \nabla_k q, \int_{\Omega} g_{kij} e_{ij}(\underline{v}) \nabla_k p$$

Syntax: dw_piezo_coupling.<i>.<r>(<material>, <virtualstate_i, jstate—virtual_i)—

12 Terms in termsSurface

12.1 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>)

13 Terms in termsLinElasticity

13.1 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>)

13.2 de_cauchy_strain

Class: CauchyStrainTerm

Description: Cauchy strain tensor averaged in elements.

Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} \underline{e}(\underline{w}) / \int_{T_K} 1$$

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

13.3 de_cauchy_stress

Class: CauchyStressTerm

Description: Cauchy stress tensor averaged in elements.

Definition: vector of

$$\forall K \in \mathcal{T}_h : \int_{T_K} D_{ijkl} e_{kl}(\underline{w}) / \int_{T_K} 1$$

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

13.4 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>)

13.5 dw_lin_elastic_iso

Class: LinearElasticIsotropicTerm

Description: Isotropic linear elasticity term.

Definition:

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

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

13.6 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>)

13.7 dw_lin_viscous_th

Class: LinearViscousTHTerm

Definition:

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

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

14 Terms in termsBiot

14.1 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>)

14.2 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>)`

14.3 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>)`

14.4 dw_biot_div_th

Class: BiotDivTHTerm

Definition:

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

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

14.5 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>)`

14.6 dw_biot_grad_dt

Class: BiotGradDtTerm

Description: Biot gradient-like term (weak form) with time-discretized \dot{p} and 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:

$$\int_{\Omega} \frac{p - p_0}{\Delta t} \alpha_{ij} e_{ij}(\underline{v})$$

Arguments:

ts.dt	Δt
parameter	p_0

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

14.7 dw_biot_grad_th

Class: BiotGradTHTerm

Definition:

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

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

15 Term caches in cachesFiniteStrain

15.1 finite_strain_tl

Class: FiniteStrainTLDataCache

cache = term.get_cache('finite_strain_tl', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16 Term caches in cachesBasic

16.1 cauchy_strain

Class: CauchyStrainDataCache

cache = term.get_cache('cauchy_strain', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16.2 div_vector

Class: DivVectorDataCache

cache = term.get_cache('div_vector', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16.3 grad_scalar

Class: GradScalarDataCache

cache = term.get_cache('grad_scalar', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16.4 grad_vector

Class: GradVectorDataCache

cache = term.get_cache('grad_vector', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16.5 mat_in_qp

Class: MatInQPDataCache

cache = term.get_cache('mat_in_qp', <index>)

data = cache(<data name>, <ig>, <ih>, mat, ap, assumed_shapes, mode_in)

16.6 state_in_surface_qp

Class: StateInSurfaceQPDataCache

cache = term.get_cache('state_in_surface_qp', <index>)

data = cache(<data name>, <ig>, <ih>, state)

16.7 state_in_volume_qp

Class: StateInVolumeQPDataCache

```
cache = term.get_cache( 'state_in_volume_qp', <index> )
```

```
data = cache( <data name>, <ig>, <ih>, state )
```

16.8 volume

Class: VolumeDataCache

```
cache = term.get_cache( 'volume', <index> )
```

```
data = cache( <data name>, <ig>, <ih>, region, field )
```