Package <default>

Class Diagram Summary	
<default>	

<default>

Class Diagram <default>

Package EinAusgabe

Class Diagram Summary	
EinAusgabe	

Class Summary	
InitAkkScheibe	Ein Objekt dieser Klasse wird bei der Erzeugung einer Akkretionsscheibe benötigt.
InitOdeint	Ein Objekt dieser Klasse wird bei der Erzeugung eines Integrators benötigt.
InitPhysSystem	Ein Objekt dieser Klasse wird bei der Erzeugung eines Objekts der Klasse "PhysSystem" benötigt.
InitRechteSeiteDG	Ein Objekt dieser Klasse wird bei der Zusammenstellung der Kräfte benötig
Update	Diese abstrakte Klasse ermöglicht die Kommunikation zwischen der Simulation und einer GUI.

EinAusgabe

Class Diagram EinAusgabe

Class Summary	
InitAkkScheibe	Ein Objekt dieser Klasse wird bei der Erzeugung einer Akkretionsscheibe benötigt.
InitOdeint	Ein Objekt dieser Klasse wird bei der Erzeugung eines Integrators benötigt.
InitPhysSystem	Ein Objekt dieser Klasse wird bei der Erzeugung eines Objekts der Klasse "PhysSystem" benötigt.
InitRechteSeiteDG	Ein Objekt dieser Klasse wird bei der Zusammenstellung der Kräfte benötig
Update	Diese abstrakte Klasse ermöglicht die Kommunikation zwischen der Simulation und einer GUI.

EinAusgabe

Class InitAkkScheibe

---

class InitAkkScheibe

Ein Objekt dieser Klasse wird bei der Erzeugung einer Akkretionsscheibe benötigt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
InitAkkScheibe(const char * parameterfile, const char * teilchenfile) Ruft die Funktion "init" auf.

Method Summary	
`void`	**init**(const char * parameterfile, const char * teilchenfile) Initialisiert die Variablen.
`const char *`	**teilchenfile** ()

Constructor Detail

InitAkkScheibe

public **InitAkkScheibe**(const char * parameterfile, const char * teilchenfile)

Ruft die Funktion "init" auf.

Method Detail

init

5, InitAkkScheibe

public void **init**(const char * parameterfile, const char * teilchenfile)

Initialisiert die Variablen.

---

teilchenfile

public const char * **teilchenfile** ()

EinAusgabe

Class InitOdeint

---

class InitOdeint

Ein Objekt dieser Klasse wird bei der Erzeugung eines Integrators benötigt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
InitOdeint(const char * parameterfile) Ruft die Funktion "init" auf.

Method Summary	
`double`	**dt**()
`double`	**dtmin**()
`double`	**eps** ()
`void`	**eps** (double d)
`void`	**init**(const char * parameterfile) Initialisiert die Variablen.
`int`	**maxsteps** ()
`int`	**num_dumps** ()

Method Summary	
`void`	**num_dumps** (int i)
`Integrator`	**odeint** ()
`double`	**tend** ()
`void`	**tend** (double d)

Constructor Detail

InitOdeint

public **InitOdeint**(const char * parameterfile)

Ruft die Funktion "init" auf.

Method Detail

dt

public double **dt**()

---

dtmin

public double **dtmin**()

---

eps

public double **eps** ()

---

eps

public void **eps** (double d)

---

init

public void **init**(const char * parameterfile)

Initialisiert die Variablen.

---

maxsteps

public int **maxsteps** ()

---

num_dumps

public int **num_dumps** ()

---

num_dumps

public void **num_dumps** (int i)

---

odeint

public Integrator **odeint** ()

---

tend

public double **tend** ()

tend

public void **tend**(double d)

EinAusgabe

Class InitPhysSystem

class InitPhysSystem

Ein Objekt dieser Klasse wird bei der Erzeugung eines Objekts der Klasse "PhysSystem" benötigt

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

InitPhysSystem (const char * parameterfile, const char * teilchenfile)
Ruft die Funktion "init" auf.

Method Summary

`double`	**alpha** ()
`double`	**beta** ()
`double`	**h**()
`void`	**h**(double d)
`void`	**init**(const char * parameterfile, const char * teilchenfile) Initialisiert alle Variablen.
`Kern`	**kern**()
`double`	**m**()

Method Summary	
double	**M1**()
double	**M2**()
double	**max**()
double	**min**()
double	**nue** ()
double	**Porb**()
int	**rate**()

Constructor Detail

InitPhysSystem

public **InitPhysSystem** (const char * parameterfile, const char * teilchenfile)

Ruft die Funktion "init" auf.

Method Detail

alpha

public double **alpha** ()

---

beta

public double **beta** ()

---

h

public double **h**()

---

h

public void **h**(double d)

---

init

public void **init**(const char * parameterfile, const char * teilchenfile)

Initialisiert alle Variablen.

---

kern

public Kern **kern**()

---

m

public double **m**()

M1

public double **M1**()

M2

public double **M2**()

---

max

public double **max**()

---

min

public double **min**()

---

nue

public double **nue** ()

Porb

public double **Porb**()

---

rate

public int **rate**()

EinAusgabe

Class InitRechteSeiteDG

---

class InitRechteSeiteDG

Ein Objekt dieser Klasse wird bei der Zusammenstellung der Kräfte benötigt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
InitRechteSeiteDG (const char * parameterfile) Ruft die Funktion "init" auf.

Method Summary	
`void`	**init**(const char * parameterfile) Initialisiert die Variablen.
`vector<Forces>`	**kraefte**()

Constructor Detail

InitRechteSeiteDG

public **InitRechteSeiteDG** (const char * parameterfile)

Ruft die Funktion "init" auf.

Method Detail

init

public void **init**(const char * parameterfile)

Initialisiert die Variablen.

---

kraefte

public vector<Forces> **kraefte**()

EinAusgabe

Class Update

Direct Known Subclasses:UpdateQt

abstract class Update

Diese abstrakte Klasse ermöglicht die Kommunikation zwischen der Simulation und einer GUI. Dabei wird in der GUI eine konkrete Klasse von dieser abgeleitet, die den eigenen Anforderungen zugeschnitten ist und dessen Pointer dann der Akkretionsscheibe übergeben. Durch diese Klasse ist eine GUI von der eigentlichen Simulation entkoppelt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
Update () macht nichts !

Method Summary	
`abstract void`	**processEvents** () Diese Funktion ruft die "processEvent"−Funktion der GUI auf.
`abstract void`	**update** () Wird nach einer neuen Berechnung der Akkretionsscheibe aufgerufen.

Constructor Detail

Update

public **Update** ()

macht nichts !

Method Detail

processEvents

public abstractvoid **processEvents** ()

Diese Funktion ruft die "processEvent"−Funktion der GUI auf.

---

update

public abstractvoid **update** ()

Wird nach einer neuen Berechnung der Akkretionsscheibe aufgerufen.

Package Integratoren

Class Diagram Summary	
Integratoren	

Class Summary	
Odeint	Diese Klasse ist die einheitliche Schnittstelle der Integratoren.
RungeKutta	Dies ist die Implementation des Runge−Kutta−Integrators 5−ter Ordnung mi Schrittweitensteuerung.

Integratoren

Class Diagram Integratoren

Class Summary	
Odeint	Diese Klasse ist die einheitliche Schnittstelle der Integratoren.
RungeKutta	Dies ist die Implementation des Runge−Kutta−Integrators 5−ter Ordnung mi Schrittweitensteuerung.

Integratoren

Class Odeint

Direct Known Subclasses:RungeKutta

abstract class Odeint

Diese Klasse ist die einheitliche Schnittstelle der Integratoren.

Author:
Andreas Nagel
Version: 1.0
Stereotype Strategy:Strategy
date 4.7.2000

Method Summary	
`void`	**odeint** (AkkScheibe & , PhysSystem & , RechteSeiteDG &) Diese Funktion startet die Integration.
`void`	**stop** () Diese Funktion wird z.B.

Method Detail

odeint

public void **odeint** (AkkScheibe & , PhysSystem & , RechteSeiteDG &)

Diese Funktion startet die Integration. Die darin aufgerufene Stepper−Funktion hängt von de konkreten abgeleiteten Klasse ab, die für die Integration gewählt wurde.
Stereotype Template−Methode

---

stop

public void **stop** ()

Diese Funktion wird z.B. von der GUI aufgerufen, falls die Stop−Taste gedrückt wurde und

stoppt die Integration.

Integratoren

Class RungeKutta

class RungeKuttaderived from Odeint

Dies ist die Implementation des Runge−Kutta−Integrators 5−ter Ordnung mit Schrittweitensteuerung.

Author:
Andreas Nagel
Version: 1.0
Stereotype Strategy:ConcreteStrategy
date 4.7.2000

Constructor Summary

RungeKutta (AkkScheibe & akk, InitOdeint & init)
Initialisiert die Membervariablen.

Destructor Summary

~RungeKutta ()
Löscht die für die Integration benötigten zusätzlichen Akkretionsscheiben.

Method Summary

`virtual void`	**stepper** (AkkScheibe & akk, PhysSystem & sys, RechteSeiteDG & rechteseite, double * dt_did, double * dt_next) Implementation des Runge−Kutta−Steppers.

Methods inherited from class Integratoren.Odeint

`odeint, stop`

Constructor Detail

RungeKutta

public **RungeKutta** (AkkScheibe & akk, InitOdeint & init)

Initialisiert die Membervariablen.

Method Detail

~RungeKutta

public **~RungeKutta** ()

Löscht die für die Integration benötigten zusätzlichen Akkretionsscheiben.

Method Detail

stepper

public virtualvoid **stepper** (AkkScheibe & akk, PhysSystem & sys, RechteSeiteDG & rechteseite double * dt_did, double * dt_next)

Implementation des Runge−Kutta−Steppers. Siehe Numerical Recipes in C (2−te Auflage) S. 719.

Association Links

to **Class** AkkScheibe

Hilfsakkretionsscheibe.

to **Class** AkkScheibe

Akkretionsscheibe für die Fehlerbestimmung des Integrationsschrittes.

to **Class** AkkScheibe

Enthält das Ergebnis nach dem 1−ten Step der Runge−Kutta−Integration.

to **Class** AkkScheibe

Enthält das Ergebnis nach dem 2−ten Step der Runge−Kutta−Integration.

to **Class** AkkScheibe

Enthält das Ergebnis nach dem 3−ten Step der Runge−Kutta−Integration.

to **Class** AkkScheibe

Enthält das Ergebnis nach dem 4−ten Step der Runge−Kutta−Integration.

to **Class** AkkScheibe

Enthält das Ergebnis nach dem 5−ten Step der Runge−Kutta−Integration.

Package Kraefte

Class Diagram Summary	
Kraefte	

Class Summary	
Artvisc	Ist zuständig für die Berechnung der künstlichen Viskositätskraft.
Externforce	Ist zuständig für die Berechnung der externen Kraft (Gravitationskraft).
Kraefte	Dies ist eine Kontainerklasse (siehe Composite−Pattern) für die Kräfte−Klassen.
Physvisc	Ist zuständig für die Berechnung der physikalischen Viskositätskraft.
Pressure	Ist zuständig für die Berechnung der inneren Druckkraft.
RechteSeiteDG	Diese Klasse ist die Kräfte−Schnittstelle nach außen und berechnet die recht Seite der DG.

Kraefte

Class Diagram Kraefte

Class Summary	
Artvisc	Ist zuständig für die Berechnung der künstlichen Viskositätskraft.
Externforce	Ist zuständig für die Berechnung der externen Kraft (Gravitationskraft).
Kraefte	Dies ist eine Kontainerklasse (siehe Composite−Pattern) für die Kräfte−Klassen.
Physvisc	Ist zuständig für die Berechnung der physikalischen Viskositätskraft.
Pressure	Ist zuständig für die Berechnung der inneren Druckkraft.
RechteSeiteDG	Diese Klasse ist die Kräfte−Schnittstelle nach außen und berechnet die recht Seite der DG.

Kraefte

Class Artvisc

---

class Artviscderived from RechteSeiteDG

Ist zuständig für die Berechnung der künstlichen Viskositätskraft.

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Leaf
date 4.7.2000

Constructor Summary
Artvisc() macht nichts !

Method Summary	
`virtual AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys) Instanziiert eine neue Akkretionsscheibe des Typs "PressartviscAkkScheibe" und gibt sie zurück.
`virtual void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Ruft die Funktion "add_artvisc" der Akkretionsscheibe auf.
`virtual Scheibe`	**scheibe** ()

Methods inherited from class Kraefte.RechteSeiteDG
`add, derivatives`

Constructor Detail

Artvisc

public **Artvisc**()

macht nichts !

Method Detail

createAkkScheibe

public virtual AkkScheibe * **createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys)

Instanziiert eine neue Akkretionsscheibe des Typs "PressartviscAkkScheibe" und gibt sie zurück.

---

deriv_ohne_init

public virtual void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Ruft die Funktion "add_artvisc" der Akkretionsscheibe auf.

---

scheibe

public virtual Scheibe **scheibe** ()

Kraefte

Class Externforce

---

class Externforcederived from RechteSeiteDG

Ist zuständig für die Berechnung der externen Kraft (Gravitationskraft).

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Leaf
date 4.7.2000

Constructor Summary

Externforce()
macht nichts !

Method Summary

`virtual AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys) Instanziiert eine neue Akkretionsscheibe des Typs "AkkScheibe" und gibt sie zurück.
`virtual void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Berechnet die Gravitationskraft, die auf jedes Teilchen wirkt und fügt sie der Gesamtkraft hinzu.
`virtual Scheibe`	**scheibe** ()

Methods inherited from class Kraefte.RechteSeiteDG

`add, derivatives`

Constructor Detail

Externforce

public **Externforce**()

macht nichts !

Method Detail

createAkkScheibe

public virtual AkkScheibe * **createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys)

Instanziiert eine neue Akkretionsscheibe des Typs "AkkScheibe" und gibt sie zurück.

---

deriv_ohne_init

public virtual void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Berechnet die Gravitationskraft, die auf jedes Teilchen wirkt und fügt sie der Gesamtkraft hinzu.

---

scheibe

public virtual Scheibe **scheibe** ()

Kraefte

Class Kraefte

---

class Kraefte derived from RechteSeiteDG

Dies ist eine Kontainerklasse (siehe Composite−Pattern) für die Kräfte−Klassen.

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Composite
date 4.7.2000

Constructor Summary
Kraefte() macht nichts !

Method Summary	
`virtual void`	**add** (RechteSeiteDG * kraft) Fügt eine neue Kraft (Leaf)− bzw.
`virtual AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys) Gibt die von der Funktion "scheibe()" berechnete Akkretionsscheibe zurück.
`void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Ruft einfach nacheinander dieselbe Funktion in den Kräfteklassen auf, wobei die einzelnen Kräfte einfach aufaddiert werden.
`virtual Scheibe`	**scheibe** () Berechnet die für die momentane Zusammenstellung der Kräfte benötig Akkretionsscheibe.

Methods inherited from class Kraefte.RechteSeiteDG
`derivatives`

Constructor Detail

Kraefte

public **Kraefte**()

macht nichts !

Method Detail

add

public virtual void **add** (RechteSeiteDG * kraft)

Fügt eine neue Kraft (Leaf)− bzw. eine neue Kontainer(Composite)−Klasse hinzu.

---

createAkkScheibe

public virtual AkkScheibe * **createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys)

Gibt die von der Funktion "scheibe()" berechnete Akkretionsscheibe zurück.

---

deriv_ohne_init

public void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Ruft einfach nacheinander dieselbe Funktion in den Kräfteklassen auf, wobei die einzelnen Kräfte einfach aufaddiert werden.

---

scheibe

public virtual Scheibe **scheibe** ()

Berechnet die für die momentane Zusammenstellung der Kräfte benötigte Akkretionsscheibe.

Kraefte

Class Physvisc

---

class Physviscderived from RechteSeiteDG

Ist zuständig für die Berechnung der physikalischen Viskositätskraft.

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Leaf
date 4.7.2000

Constructor Summary

Physvisc ()
macht nichts !

Method Summary

`virtual AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys) Instanziiert eine neue Akkretionsscheibe des Typs "PhysviskAkkScheibe" und gibt sie zurück.
`virtual void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Ruft die Funktion "add_physvisc" der Akkretionsscheibe auf.
`virtual Scheibe`	**scheibe** ()

Methods inherited from class Kraefte.RechteSeiteDG

`add, derivatives`

Constructor Detail

Physvisc

public **Physvisc** ()

macht nichts !

Method Detail

createAkkScheibe

public virtual AkkScheibe * **createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys)

Instanziiert eine neue Akkretionsscheibe des Typs "PhysviskAkkScheibe" und gibt sie zurück

---

deriv_ohne_init

public virtual void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Ruft die Funktion "add_physvisc" der Akkretionsscheibe auf.

---

scheibe

public virtual Scheibe **scheibe** ()

Kraefte

Class Pressure

---

class Pressurederived from RechteSeiteDG

Ist zuständig für die Berechnung der inneren Druckkraft.

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Leaf
date 4.7.2000

Constructor Summary

Pressure ()
macht nichts !

Method Summary

`virtual AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys) Instanziiiert eine neue Akkretionsscheibe des Typs "PressArtviscAkkScheibe" und gibt sie zurück.
`virtual void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Ruft die Funktion "add_press" der Akkretionsscheibe auf.
`virtual Scheibe`	**scheibe** ()

Methods inherited from class Kraefte.RechteSeiteDG

`add, derivatives`

Constructor Detail

Pressure

public **Pressure** ()

macht nichts !

Method Detail

createAkkScheibe

public virtual AkkScheibe * **createAkkScheibe** (InitAkkScheibe & iniakk, PhysSystem & sys)

Instanziiiert eine neue Akkretionsscheibe des Typs "PressArtviscAkkScheibe" und gibt sie zurück.

---

deriv_ohne_init

public virtual void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Ruft die Funktion "add_press" der Akkretionsscheibe auf.

---

scheibe

public virtual Scheibe **scheibe** ()

Kraefte

Class RechteSeiteDG

Direct Known Subclasses:Artvisc, Externforce, Kraefte, Physvisc, Pressure

abstract class RechteSeiteDG

Diese Klasse ist die Kräfte−Schnittstelle nach außen und berechnet die rechte Seite der DG.

Author:
Andreas Nagel
Version: 1.0
Stereotype Composite:Component
date 4.7.2000

Method Summary	
`virtual void`	**add** (RechteSeiteDG *) Falls diese Funktion nicht reimplementiert wird gibt sie folgendes aus: "add in leaf geht nicht!"
`abstract AkkScheibe *`	**createAkkScheibe** (InitAkkScheibe & , PhysSystem &) Diese abstrakte Funktion gibt nach der Zusammenstellung der Kräfte die benötigte Akkretionsscheibe zurück, die für die Berechnung der Kräfte benötigt wird.
`abstract void`	**deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys) Ist eine abstrakte Funktion und wird in den abgeleiteten Klassen implementiert.
`void`	**derivatives** (AkkScheibe & akk, PhysSystem & sys) Berechnet die rechte Seite der DG.
`abstract Scheibe`	**scheibe** () Diese abstrakte Funktion gibt in den abgeleiteten Kräfte−Klassen die für die berechnete Kraft benötigte Akkretionsscheibe an.

Method Detail

add

38, RechteSeiteDG

public virtual void **add** (RechteSeiteDG *)

Falls diese Funktion nicht reimplementiert wird gibt sie folgendes aus: "add in leaf geht nicht

---

createAkkScheibe

public abstract AkkScheibe * **createAkkScheibe** (InitAkkScheibe & , PhysSystem &)

Diese abstrakte Funktion gibt nach der Zusammenstellung der Kräfte die benötigte Akkretionsscheibe zurück, die für die Berechnung der Kräfte benötigt wird.

---

deriv_ohne_init

public abstract void **deriv_ohne_init** (AkkScheibe & akk, PhysSystem & sys)

Ist eine abstrakte Funktion und wird in den abgeleiteten Klassen implementiert. Wie der Nam schon andeutet, wird dort die rechte Seite der DG berechnet ohne die Funktion "init_force" der Akkretionsscheibe aufzurufen.

---

derivatives

public void **derivatives** (AkkScheibe & akk, PhysSystem & sys)

Berechnet die rechte Seite der DG. Diese Template Methode (siehe Template−Pattern) ruft zuerst die "force_init"−Funktion der Akkretionsscheibe auf (setzt z.B. die Gesamtkraft auf 0 und dann die eigentliche Berechnungsfunktion "deriv_ohne_init".
Stereotype Template Methode

---

scheibe

public abstract Scheibe **scheibe** ()

Diese abstrakte Funktion gibt in den abgeleiteten Kräfte−Klassen die für die berechnete Kraf benötigte Akkretionsscheibe an.

Package PhysikalischeObjekte

Class Diagram Summary	
PhysikalischeObjekte	

Class Summary	
AkkScheibe	Implementation der Akkretionsscheibe.
Kernel	Ist die Elternklasse aller Kernel und ist abstrakt.
Kernel1	Implementation eines konkreten Kernels.
PhysSystem	Das Physikalisches System
PhysviscAkkScheibe	Diese Akkretionsscheibe wird für die Berechnung der physikalischen Viskositätskraft benötigt.
PhysviscTeilchen	Physvisc−Teilchen der Physvisc−Akkretionsscheibe.
PhysviscWW	Physvisc−Wechselwirkungsteilchen der Physvisc−Akkretionsscheibe und des Physvisc−Teilchens.
PressArtviscAkkScheibe	Diese Akkretionsscheibe wird für die Berechnung der Druckkraft und/oder der künstlichen Viskositätskraft benötigt.
PressArtviscTeilchen	PressArtviscTeilchen der PressArtvisc−Akkretionsscheibe.
PressArtviscWW	PressArtvisc−Wechselwirkungsteilchen der PressArtvisc−AkkScheibe und des PressArtvisc−Teilchen.
SphAkkScheibe	Die Sph−Akkretionsscheibe ist die Elternklasse aller Akkretionsscheiben mi Sph−Kräften.
SphTeilchen	Sph−Teilchen der Sph−AkkScheibe.
SphWW	Sph−Wechselwirkungsteilchen der Sph−AkkScheibe und des Sph−Teilchen.
Teilchen	Teilchen der AkkScheibe

PhysikalischeObjekte

Class Diagram PhysikalischeObjekte

Class Summary	
AkkScheibe	Implementation der Akkretionsscheibe.
Kernel	Ist die Elternklasse aller Kernel und ist abstrakt.
Kernel1	Implementation eines konkreten Kernels.
PhysSystem	Das Physikalisches System
PhysviscAkkScheibe	Diese Akkretionsscheibe wird für die Berechnung der physikalischen Viskositätskraft benötigt.
PhysviscTeilchen	Physvisc−Teilchen der Physvisc−Akkretionsscheibe.
PhysviscWW	Physvisc−Wechselwirkungsteilchen der Physvisc−Akkretionsscheibe und des Physvisc−Teilchens.
PressArtviscAkkScheibe	Diese Akkretionsscheibe wird für die Berechnung der Druckkraft und/oder der künstlichen Viskositätskraft benötigt.
PressArtviscTeilchen	PressArtviscTeilchen der PressArtvisc−Akkretionsscheibe.
PressArtviscWW	PressArtvisc−Wechselwirkungsteilchen der PressArtvisc−AkkScheibe und des PressArtvisc−Teilchen.
SphAkkScheibe	Die Sph−Akkretionsscheibe ist die Elternklasse aller Akkretionsscheiben mi Sph−Kräften.
SphTeilchen	Sph−Teilchen der Sph−AkkScheibe.
SphWW	Sph−Wechselwirkungsteilchen der Sph−AkkScheibe und des Sph−Teilchen.
Teilchen	Teilchen der AkkScheibe

PhysikalischeObjekte

Class AkkScheibe

Direct Known Subclasses:SphAkkScheibe

class AkkScheibe

Implementation der Akkretionsscheibe. Diese Klasse ist eine Kontainerklasse für Objekte des Typs Teilchen und enthält Operationen, die diese Teilchen verwalten.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
AkkScheibe () macht nichts!
AkkScheibe (InitAkkScheibe & initakk, PhysSystem & sys) Der Konstruktor ruft die Template−Funktion _init auf.

Method Summary	
`virtual AkkScheibe *`	**clone** () Ruft die gleichnamige Template−Funktion auf.
`virtual void`	**entferne_teilchen** (int i) Ruft die Template−Funktion _erase auf.
`virtual void`	**force_init** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`virtual void`	**odeint_assign** (AkkScheibe & akk) Die Eigenschaften der AkkScheibe "akk" werden übernommen.
`virtual void`	**odeint_init** (AkkScheibe & akk) Ruft die gleichnamige Template−Funktion auf.

Method Summary	
`virtual Teilchen &`	**operator[]**(int n) Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern.
`virtual void`	**output**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`void`	**processEvents** () Diese Funktion löst einen ProcessEvent der GUI aus.
`void`	**set_update** (Update * update) Das in der GUI erzeugte "update"−Objekt wird am Anfang der Simulation über diese Funktion übergeben.
`void`	**t**(double t) Ändert die aktuelle Zeit der Akkretionsscheibe.
`double`	**t**()
`virtual int`	**tanz**() Gibt die Anzahl der Teilchen in der Akkretionsscheibe zurück.
`void`	**update** () Aktualisiert die GUI, d.h.

Constructor Detail

AkkScheibe

public **AkkScheibe** ()

macht nichts!

AkkScheibe

public **AkkScheibe** (InitAkkScheibe & initakk, PhysSystem & sys)

Der Konstruktor ruft die Template−Funktion _init auf.

Method Detail

clone

public virtual AkkScheibe * **clone** ()

Ruft die gleichnamige Template−Funktion auf.

---

entferne_teilchen

public virtual void **entferne_teilchen** (int i)

Ruft die Template−Funktion _erase auf.

---

force_init

public virtual void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

odeint_assign

public virtual void **odeint_assign** (AkkScheibe & akk)

Die Eigenschaften der AkkScheibe "akk" werden übernommen. Diese Funktion wird im RungeKutta::stepper aufgerufen, um das Ergebnis des Integrationsschrittes in die eigentliche Akkretionsscheibe zu übernehmen.

---

odeint_init

public virtual void **odeint_init** (AkkScheibe & akk)

Ruft die gleichnamige Template−Funktion auf.

---

operator[]

public virtual Teilchen & **operator[]**(int n)

Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern. (Bsp.: akk[4] ist das 4−te Teilchen der Akkretionsscheibe "akk".)

---

output

public virtual void **output**(PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

processEvents

public void **processEvents** ()

Diese Funktion löst einen ProcessEvent der GUI aus. Diese Funktion kann überall im Programm aufgerufen werden, wobei zu häufiges Aufrufen zu Performance−Verlust führt. (siehe RechteSeite::derivatives)

---

set_update

public void **set_update** (Update * update)

Das in der GUI erzeugte "update"−Objekt wird am Anfang der Simulation über diese Funktio übergeben. (siehe SimulationQt::startstop)

---

t

public void **t**(double t)

Ändert die aktuelle Zeit der Akkretionsscheibe. (z.B. nach einem Integrationsschritt)

---

t

public double **t**()

Output Gibt die aktuelle Zeit der Akkretionsscheibe aus.

---

tanz

public virtual int **tanz** ()

Gibt die Anzahl der Teilchen in der Akkretionsscheibe zurück.

---

update

public void **update** ()

Aktualisiert die GUI, d.h. die Funktion wird nach jeder Neuberechnung der Akkretionsscheib
aufgerufen. (siehe Odeint::odeint)

Association Links

to **Class** Update

Wird zum Aktualisieren einer GUI benutzt. Eine Zustandsänderung der AkkScheibe kann übe
dieses Objekt nach "außen" weitergegeben werden.

PhysikalischeObjekte

Class Kernel

Direct Known Subclasses:Kernel1

abstract class Kernel

Ist die Elternklasse aller Kernel und ist abstrakt.

Author:
Andreas Nagel
Version: 1.0
Stereotype Strategy:Strategy
date 4.7.2000

Constructor Summary
Kernel(InitPhysSystem & initphys) Definiert "_h".

Method Summary	
`double`	**dW_fak**() Gibt die Variable "_dW_fak" zurück.
`double`	**W_fak**() Gibt die Variable "_W_fak" zurück.
`abstract double`	**W_ohne_fak** (double dist, valarray<double> & dW) Dies ist eine abstrakte Klasse.

Constructor Detail

Kernel

public **Kernel**(InitPhysSystem & initphys)

Definiert "_h".

Method Detail

dW_fak

public double **dW_fak**()

Gibt die Variable "_dW_fak" zurück.

W_fak

public double **W_fak**()

Gibt die Variable "_W_fak" zurück.

W_ohne_fak

public abstract double **W_ohne_fak** (double dist, valarray<double> & dW)

Dies ist eine abstrakte Klasse. Sie gibt "W_ohne_fak" zurück und schreibt in den Parameter "dW" das Ergebnis "dW_ohne_fak".
Input dist: Abstandsbetrag. dW: Abstandsvektor.
Output dW: dW_ohne_fak

PhysikalischeObjekte

Class Kernel1

class Kernel1derived from Kernel

Implementation eines konkreten Kernels.

Author:
Andreas Nagel
Version: 1.0
Stereotype Strategy:ConcreteStrategy
date 4.7.2000

Constructor Summary
Kernel1 (InitPhysSystem & initphys) Initialisiert die Variablen "_W_fak" und _dW_fak" aus Kernel.

Method Summary	
`virtual double`	**W_ohne_fak** (double dist, valarray<double> & dW) Konkrete Implementierung der abstrakten Funktion der Elternklasse.

Methods inherited from class PhysikalischeObjekte .Kernel
`dW_fak, W_fak`

Constructor Detail

Kernel1

public **Kernel1** (InitPhysSystem & initphys)

Initialisiert die Variablen "_W_fak" und _dW_fak" aus Kernel.

Method Detail

W_ohne_fak

public virtual double **W_ohne_fak** (double dist, valarray<double> & dW)

Konkrete Implementierung der abstrakten Funktion der Elternklasse.

PhysikalischeObjekte

Class PhysSystem

---

class PhysSystem

Das Physikalisches System

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
PhysSystem (InitPhysSystem & initsys, Kernel * kernel) Initialisiert das Physikalische System.

Method Summary	
`double`	**alpha** ()
`double`	**beta** ()
`double`	**G**()
`double`	**h**()
`Kernel *`	**kernel** ()
`double`	**L1**()
`double`	**m**()

Method Summary	
double	**M1**()
double	**M2**()
double	**max**()
double	**min**()
double	**nue** ()
double	**Omega** ()
double	**Porb**()
valarray<double>	**R1**()
valarray<double>	**R2**()
double	**Stardist**()

Constructor Detail

PhysSystem

public **PhysSystem** (InitPhysSystem & initsys, Kernel * kernel)

Initialisiert das Physikalische System.

Method Detail

alpha

public double **alpha** ()

---

beta

public double **beta** ()

G

public double **G**()

---

h

public double **h**()

---

kernel

public Kernel * **kernel** ()

L1

public double **L1**()

---

m

public double **m**()

M1

public double **M1**()

M2

public double **M2**()

---

max

public double **max**()

---

min

public double **min**()

---

nue

public double **nue** ()

Omega

public double **Omega** ()

Porb

public double **Porb**()

R1

public valarray<double> **R1** ()

R2

public valarray<double> **R2** ()

Stardist

public double **Stardist**()

Association Links

to **Class** Kernel

Pointer auf das verwendete Kernelobjekt.

PhysikalischeObjekte

Class PhysviscAkkScheibe

---

class PhysviscAkkScheibederived from PressArtviscAkkScheibe

Diese Akkretionsscheibe wird für die Berechnung der physikalischen Viskositätskraft benötigt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
PhysviscAkkScheibe () macht nichts !
PhysviscAkkScheibe (InitAkkScheibe & initakk, PhysSystem & sys) Ruft die Template−Funktion "_init" in Sph−AkkScheibe auf.

Destructor Summary
~PhysviscAkkScheibe () Löscht das Gitter und die Wechselwirkungsliste.

Method Summary	
`virtual void`	**add_artvisc** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.
`virtual void`	**add_physvisc** (PhysSystem & sys) Diese Funktion müßte eigentlich auch als Template−Funktion definiert werden, damit neuere Akkretionsscheiben auf diese Funtion zugreifen können.

57, PhysviscAkkScheibe

Method Summary	
`virtual void`	**add_press** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.
`virtual PhysviscAkkScheibe *`	**clone** () Ruft die gleichnamige Template−Funktion in AkkScheibe auf.
`virtual void`	**entferne_teilchen** (int i) Ruft die Template−Funktion "_erase" in AkkScheibe auf.
`virtual void`	**force_init** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.
`virtual void`	**odeint_assign** (AkkScheibe & akk) Die Eigenschaften der PhysviscAkkScheibe "akk" werden übernommen
`virtual void`	**odeint_init** (AkkScheibe & akk) Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.
`virtual PhysviscTeilchen &`	**operator[]**(int n) Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern.
`virtual void`	**output**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion in AkkScheibe auf.
`virtual int`	**tanz**() Gibt die Anzahl der Physvisc−Teilchen der Physvisc−Akkretionsscheibe zurück.

Methods inherited from class PhysikalischeObjekte .AkkScheibe
`processEvents, set_update, t, t, update`

Constructor Detail

PhysviscAkkScheibe

public **PhysviscAkkScheibe** ()

57, PhysviscAkkScheibe

macht nichts !

PhysviscAkkScheibe

public **PhysviscAkkScheibe** (InitAkkScheibe & initakk, PhysSystem & sys)

Ruft die Template−Funktion "_init" in Sph−AkkScheibe auf.

Method Detail

~PhysviscAkkScheibe

public **~PhysviscAkkScheibe** ()

Löscht das Gitter und die Wechselwirkungsliste.

Method Detail

add_artvisc

public virtualvoid **add_artvisc** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.

---

add_physvisc

public virtualvoid **add_physvisc** (PhysSystem & sys)

Diese Funktion müßte eigentlich auch als Template−Funktion definiert werden, damit neuere Akkretionsscheiben auf diese Funtion zugreifen können. Der aktuelle gcc.2.95.2 macht aber auf Intel−Basis Probleme (aber nicht auf Sparc−Rechnern !), daher muß man auf einen neueren gcc (Version 3.0 ?) warten, der dieses Problem nicht mehr besitzt. Die Berechnung der physikalischen Viskosität ist ein vierstufiger Prozeß: 1.) Zuerst wird für jedes Wechselwirkungsteilchen die Funktion "sigma" aufgerufen. 2.) Danach wird für jedes Physvisc−Teilchen die Funktion "sigma" aufgerufen. 3.) Jetzt wird für jedes Wechselwirkungsteilchen die Funktion "physvisc" aufgerufen. 4.) Als letzten Schritt wird für jedes Physvisc−Teilchen die Funktion "add_physvisc" aufgerufen, die die nun berechnete

physikalische Viskosität der Gesamtkraft hinzufügt.

---

add_press

public virtual void **add_press** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.

---

clone

public virtual PhysviscAkkScheibe * **clone** ()

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

entferne_teilchen

public virtual void **entferne_teilchen** (int i)

Ruft die Template−Funktion "_erase" in AkkScheibe auf.

---

force_init

public virtual void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.

---

odeint_assign

public virtual void **odeint_assign** (AkkScheibe & akk)

Die Eigenschaften der PhysviscAkkScheibe "akk" werden übernommen.

---

odeint_init

public virtual void **odeint_init** (AkkScheibe & akk)

Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.

---

operator[]

public virtual PhysviscTeilchen & **operator[]**(int n)

Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern. (Bsp.: physviscakk[4] ist das 4−te Physvisc−Teilchen der Physvisc−Akkretionsscheibe "physviscakk".)

---

output

public virtual void **output**(PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

tanz

public virtual int **tanz**()

Gibt die Anzahl der Physvisc−Teilchen der Physvisc−Akkretionsscheibe zurück.

PhysikalischeObjekte

Class PhysviscTeilchen

---

class PhysviscTeilchenderived from PressArtviscTeilchen

Physvisc−Teilchen der Physvisc−Akkretionsscheibe.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
PhysviscTeilchen () Definiert die Variablen "_dvdt_physvisc" und "_sigma".

Method Summary	
`void`	**add_artvisc** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscTeilchen auf.
`void`	**add_physvisc** (PhysSystem & sys) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.
`void`	**add_press** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscTeilchen auf.
`void`	**force_init** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.
`void`	**init**() Ruft die gleichnamige Template−Funtion in SphTeilchen auf.
`void`	**sigma** (PhysSystem & sys) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.

62, PhysviscTeilchen

Method Summary	
`valarray<double>`	**sigma** () Gibt den sigma−Tensor zurück.
`void`	**ww**(PhysviscWW * ww) Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.
`int`	**wwanz** () Gibt die Anzahl der Wechselwirkungsobjekte zurück.

Methods inherited from class PhysikalischeObjekte .PressArtviscTeilchen
`cs, p, T, ww`

Methods inherited from class PhysikalischeObjekte .SphTeilchen
`rho, ww`

Methods inherited from class PhysikalischeObjekte .Teilchen
`b, b, max_v_koord, r, r, v, v`

Constructor Detail

PhysviscTeilchen

public **PhysviscTeilchen** ()

Definiert die Variablen "_dvdt_physvisc" und "_sigma".

Method Detail

add_artvisc

public void **add_artvisc** (PhysSystem & sys)

62, PhysviscTeilchen

63, PhysviscTeilchen

Ruft die gleichnamige Template−Funktion in PressArtviscTeilchen auf.

---

add_physvisc

public void **add_physvisc** (PhysSystem & sys)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Sie berechnet die physikalische Viskositätskraft des Teilchens und fügt sie der Gesamtkraft hinzu.

---

add_press

public void **add_press** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscTeilchen auf.

---

force_init

public void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscAkkScheibe auf.

---

init

public void **init**()

Ruft die gleichnamige Template−Funtion in SphTeilchen auf.

---

sigma

public void **sigma** (PhysSystem & sys)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Diese Funktion berechnet den sigma−Tensor.

---

sigma

63, PhysviscTeilchen

public valarray<double> **sigma** ()

Gibt den sigma−Tensor zurück.

---

ww

public void **ww**(PhysviscWW * ww)

Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.

---

wwanz

public int **wwanz** ()

Gibt die Anzahl der Wechselwirkungsobjekte zurück.

PhysikalischeObjekte

Class PhysviscWW

class PhysviscWWderived from PressArtviscWW

Physvisc−Wechselwirkungsteilchen der Physvisc−Akkretionsscheibe und des Physvisc−Teilchens

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

PhysviscWW ()
Definiert die Variablen "_physvisc" und "_sigma".

Method Summary

`valarray<double>`	**dW_ohne_fak** (PhysviscTeilchen * t) Ruft die gleichnamige Template−Funktion in SphWW auf.
`void`	**init**(PhysviscTeilchen * t1, PhysviscTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys) Ruft die gleichnamige Template−Funktion in SphWW auf.
`valarray<double>`	**physvisc** () Gibt die Variable "_physvisc" zurück.
`void`	**physvisc** (PhysSystem & sys) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.
`valarray<double>`	**physvisc** (PhysviscTeilchen * teilchen) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.
`double`	**pi**() Gibt die Variable "__pi" aus PressArtviscWW zurück.

66, PhysviscWW

Method Summary	
`void`	**pi**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscWW auf.
`double`	**press** () Ruft die gleichnamige Template−Funktion auf.
`void`	**press** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion in PressArtviscWW auf.
`void`	**sigma** (PhysSystem & sys) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.
`valarray<double>`	**sigma** (PhysviscTeilchen * teilchen) Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden.

Methods inherited from class PhysikalischeObjekte .PressArtviscWW
`dW_ohne_fak, init`

Methods inherited from class PhysikalischeObjekte .SphWW
`dW_ohne_fak, init, W_ohne_fak`

Constructor Detail

PhysviscWW

public **PhysviscWW** ()

Definiert die Variablen "_physvisc" und "_sigma".

Method Detail

dW_ohne_fak

66, PhysviscWW

public valarray<double> **dW_ohne_fak** (PhysviscTeilchen * t)

Ruft die gleichnamige Template−Funktion in SphWW auf.

---

init

public void **init**(PhysviscTeilchen * t1, PhysviscTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in SphWW auf.

---

physvisc

public valarray<double> **physvisc** ()

Gibt die Variable "_physvisc" zurück.

---

physvisc

public void **physvisc** (PhysSystem & sys)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Berechnet die Variable "_physvisc".

---

physvisc

public valarray<double> **physvisc** (PhysviscTeilchen * teilchen)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Sie gibt je nach der angegebenen Variablen "teilchen" den positiven bzw. negativen Wert von "_physvisc" zurück.

---

pi

public double **pi**()

Gibt die Variable "__pi" aus PressArtviscWW zurück.

---

pi

public void **pi**(PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscWW auf.

---

press

public double **press** ()

Ruft die gleichnamige Template−Funktion auf.

---

press

public void **press** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in PressArtviscWW auf.

---

sigma

public void **sigma** (PhysSystem & sys)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Die Funktion berechnet die Variable "_sigma".

---

sigma

public valarray<double> **sigma** (PhysviscTeilchen * teilchen)

Diese Funktion sollte in Zukunft als Template−Funktion implementiert werden. Die Funktion gibt je nach der angegebenen Variablen "teilchen" den positiven bzw. negativen Wert von "_sigma" zurück.

Association Links

to **Class** PhysviscTeilchen

Die Physvisc−Wechselwirkung bezieht sich auf 2 Physvisc−Teilchen, dessen Reihenfolge nicht vertauscht werden darf.

PhysikalischeObjekte

Class PressArtviscAkkScheibe

Direct Known Subclasses:PhysviscAkkScheibe

class PressArtviscAkkScheibederived from SphAkkScheibe

Diese Akkretionsscheibe wird für die Berechnung der Druckkraft und/oder der künstlichen Viskositätskraft benötigt.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

PressArtviscAkkScheibe ()
macht nichts!

PressArtviscAkkScheibe (InitAkkScheibe & initakk, PhysSystem & sys)
Ruft die Template−Funktion "_init" in Sph−AkkScheibe auf.

Destructor Summary

~PressArtviscAkkScheibe ()
Löscht das Gitter und die Wechselwirkungsliste.

Method Summary

`virtual void`	**add_artvisc** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`virtual void`	**add_press** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`virtual PressArtviscAkkScheibe *`	**clone** () Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

71, PressArtviscAkkScheibe

Method Summary	
`virtual void`	**entferne_teilchen** (int i) Ruft die Template−Funktion "_erase" in AkkScheibe auf.
`virtual void`	**force_init**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.
`virtual void`	**odeint_assign** (AkkScheibe & akk) Die Eigenschaften der PressArtviscAkkScheibe "akk" werden übernommen.
`virtual void`	**odeint_init**(AkkScheibe & akk) Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.
`virtual PressArtviscTeilchen &`	**operator[]**(int n) Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern.
`virtual void`	**output**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion in AkkScheibe auf.
`virtual int`	**tanz**() Gibt die Anzahl der PressArtvisc−Teilchen der PressArtvisc−Akkretionsscheibe zurück.

Methods inherited from class PhysikalischeObjekte .AkkScheibe
`processEvents, set_update, t, t, update`

Constructor Detail

PressArtviscAkkScheibe

public **PressArtviscAkkScheibe** ()

macht nichts!

PressArtviscAkkScheibe

71, PressArtviscAkkScheibe

public **PressArtviscAkkScheibe** (InitAkkScheibe & initakk, PhysSystem & sys)

Ruft die Template−Funktion "_init" in Sph−AkkScheibe auf.

Method Detail

~PressArtviscAkkScheibe

public **~PressArtviscAkkScheibe** ()

Löscht das Gitter und die Wechselwirkungsliste.

Method Detail

add_artvisc

public virtual void **add_artvisc** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

add_press

public virtual void **add_press** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

clone

public virtual PressArtviscAkkScheibe * **clone** ()

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

entferne_teilchen

public virtual void **entferne_teilchen** (int i)

Ruft die Template−Funktion "_erase" in AkkScheibe auf.

---

force_init

public virtual void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.

---

odeint_assign

public virtual void **odeint_assign** (AkkScheibe & akk)

Die Eigenschaften der PressArtviscAkkScheibe "akk" werden übernommen.

---

odeint_init

public virtual void **odeint_init** (AkkScheibe & akk)

Ruft die gleichnamige Template−Funktion in Sph−AkkScheibe auf.

---

operator[]

public virtual PressArtviscTeilchen & **operator[]** (int n)

Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern. (Bsp.: pressartviscakk[4] ist das 4−te PressVisc−Teilchen der PressArtviscAkkretionsscheibe "pressartviscakk".)

---

output

public virtual void **output** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

tanz

public virtual int **tanz** ()

74, PressArtviscAkkScheibe

Gibt die Anzahl der PressArtvisc−Teilchen der PressArtvisc−Akkretionsscheibe zurück.

74, PressArtviscAkkScheibe

PhysikalischeObjekte

Class PressArtviscTeilchen

Direct Known Subclasses:PhysviscTeilchen

class PressArtviscTeilchenderived from SphTeilchen

PressArtviscTeilchen der PressArtvisc−Akkretionsscheibe.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

PressArtviscTeilchen ()
Definiert die Variablen "_dvdt_press" und "_dvdt_artvisc".

Method Summary

`void`	**add_artvisc** (PhysSystem & sys) Ruft die gleichnamige Funktion auf.
`void`	**add_press** (PhysSystem & sys) Ruft die gleichnamige Funktion auf.
`double`	**cs** () Gibt die Schallgeschwindigkeit zurück.
`void`	**force_init** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`void`	**init**() Ruft die gleichnamige Template−Funktion in SphTeilchen auf.
`double`	**p**() Gibt den Druck aus der polytropen Zustandsgleichnung zurück.
`double`	**T**() Gibt die Temperatur zurück.

Method Summary	
`void`	**ww**(PressArtviscWW * ww) Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.
`int`	**wwanz** () Gibt die Anzahl der Wechselwirkungsobjekte zurück.

Methods inherited from class PhysikalischeObjekte .SphTeilchen
`rho, ww`

Methods inherited from class PhysikalischeObjekte .Teilchen
`b, b, max_v_koord, r, r, v, v`

Constructor Detail

PressArtviscTeilchen

public **PressArtviscTeilchen** ()

Definiert die Variablen "_dvdt_press" und "_dvdt_artvisc".

Method Detail

add_artvisc

public void **add_artvisc** (PhysSystem & sys)

Ruft die gleichnamige Funktion auf.

---

add_press

public void **add_press** (PhysSystem & sys)

77, PressArtviscTeilchen

Ruft die gleichnamige Funktion auf.

---

cs

public double **cs** ()

Gibt die Schallgeschwindigkeit zurück.

---

force_init

public void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

init

public void **init**()

Ruft die gleichnamige Template−Funktion in SphTeilchen auf.

---

p

public double **p**()

Gibt den Druck aus der polytropen Zustandsgleichnung zurück.

T

public double **T**()

Gibt die Temperatur zurück.

---

ww

public void **ww**(PressArtviscWW * ww)

77, PressArtviscTeilchen

Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.

---

wwanz

public int **wwanz** ()

Gibt die Anzahl der Wechselwirkungsobjekte zurück.

PhysikalischeObjekte

Class PressArtviscWW

Direct Known Subclasses:PhysviscWW

class PressArtviscWWderived from SphWW

PressArtvisc−Wechselwirkungsteilchen der PressArtvisc−AkkScheibe und des PressArtvisc−Teilchen.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

PressArtviscWW()
Definiert die Variable "__press".

Method Summary

`valarray<double>`	**dW_ohne_fak** (PressArtviscTeilchen * t) Ruft die gleichnamige Template−Funktion in SphWW auf.
`void`	**init**(PressArtviscTeilchen * t1, PressArtviscTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys) Ruft die gleichnamige Template−Funktion in SphWW auf.
`void`	**pi**(PhysSystem & sys) Ruft die gleichnamig Template−Funktion auf.
`double`	**pi**() Gibt die Variable "__pi" zurück.
`double`	**press** () Gibt den Druckterm "__press" zurück.
`void`	**press** (PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.

80, PressArtviscWW

Methods inherited from class PhysikalischeObjekte .SphWW
dW_ohne_fak, init, W_ohne_fak

Constructor Detail

PressArtviscWW

public **PressArtviscWW**()

Definiert die Variable "__press".

Method Detail

dW_ohne_fak

public valarray<double> **dW_ohne_fak** (PressArtviscTeilchen * t)

Ruft die gleichnamige Template−Funktion in SphWW auf.

---

init

public void **init**(PressArtviscTeilchen * t1, PressArtviscTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys)

Ruft die gleichnamige Template−Funktion in SphWW auf.

---

pi

public void **pi**(PhysSystem & sys)

Ruft die gleichnamig Template−Funktion auf.

---

pi

public double **pi**()

80, PressArtviscWW

Gibt die Variable "__pi" zurück.

---

press

public double **press** ()

Gibt den Druckterm "__press" zurück.

---

press

public void **press** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

Association Links

to **Class** PressArtviscTeilchen

Die PressArtvisc−Wechselwirkung bezieht sich auf 2 PressArtvisc−Teilchen, dessen Reihenfolge nicht vertauscht werden darf.

PhysikalischeObjekte

Class SphAkkScheibe

Direct Known Subclasses:PressArtviscAkkScheibe

class SphAkkScheibederived from AkkScheibe

Die Sph−Akkretionsscheibe ist die Elternklasse aller Akkretionsscheiben mit Sph−Kräften.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

SphAkkScheibe ()
macht nichts !

SphAkkScheibe (InitAkkScheibe & initakk, PhysSystem & sys)
Der Konstruktor ruft die Template−Funktion _init auf.

Destructor Summary

~SphAkkScheibe ()
Löscht das Gitter und die Wechselwirkungsliste.

Method Summary

`virtual SphAkkScheibe *`	**clone** () Ruft die gleichnamige Template−Funktion in AkkScheibe auf.
`virtual void`	**entferne_teilchen** (int i) Ruft die Template−Funktion _erase in AkkScheibe auf.
`virtual void`	**force_init**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.

Method Summary	
`virtual void`	**odeint_assign** (AkkScheibe & akk) Die Eigenschaften der Sph−AkkScheibe "akk" werden übernommen.
`virtual void`	**odeint_init** (AkkScheibe & akk) Ruft die gleichnamige Template−Funktion in AkkScheibe auf.
`virtual SphTeilchen &`	**operator[]**(int n) Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern.
`virtual void`	**output**(PhysSystem & sys) Ruft die Template−Funktion _output in AkkScheibe auf.
`virtual int`	**tanz**() Gibt die Anzahl der Sph−Teilchen in der Sph−Akkretionsscheibe zurück

Methods inherited from class PhysikalischeObjekte .AkkScheibe
`processEvents, set_update, t, t, update`

Constructor Detail

SphAkkScheibe

public **SphAkkScheibe** ()

macht nichts !

SphAkkScheibe

public **SphAkkScheibe** (InitAkkScheibe & initakk, PhysSystem & sys)

Der Konstruktor ruft die Template−Funktion _init auf.

Method Detail

~SphAkkScheibe

public **~SphAkkScheibe** ()

Löscht das Gitter und die Wechselwirkungsliste.

Method Detail

clone

public virtual SphAkkScheibe * **clone** ()

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

entferne_teilchen

public virtual void **entferne_teilchen** (int i)

Ruft die Template−Funktion _erase in AkkScheibe auf.

---

force_init

public virtual void **force_init** (PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

odeint_assign

public virtual void **odeint_assign** (AkkScheibe & akk)

Die Eigenschaften der Sph−AkkScheibe "akk" werden übernommen. Diese Funktion wird im RungeKutta::stepper aufgerufen, um das Ergebnis des Integrationsschrittes in die eigentliche Akkretionsscheibe zu übernehmen.

---

odeint_init

public virtual void **odeint_init** (AkkScheibe & akk)

Ruft die gleichnamige Template−Funktion in AkkScheibe auf.

---

operator[]

public virtual SphTeilchen & **operator[]** (int n)

Der Zugriff auf die einzelnen Teilchen ist durch diese Funktion genauso wie bei einem Array durch Klammern. (Bsp.: sphakk[4] ist das 4−te Sph−Teilchen der SphAkkretionsscheibe "sphakk".)

---

output

public virtual void **output** (PhysSystem & sys)

Ruft die Template−Funktion _output in AkkScheibe auf.

---

tanz

public virtual int **tanz** ()

Gibt die Anzahl der Sph−Teilchen in der Sph−Akkretionsscheibe zurück.

PhysikalischeObjekte

Class SphTeilchen

Direct Known Subclasses:PressArtviscTeilchen

class SphTeilchenderived from Teilchen

Sph−Teilchen der Sph−AkkScheibe.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

SphTeilchen ()
macht nichts!

Method Summary

`void`	**force_init**(PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`void`	**init**() Ruft die gleichnamige Template−Funktion auf.
`double`	**rho**() Gibt die Dichte zurück.
`void`	**ww**(SphWW * ww) Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.
`int`	**wwanz** () Gibt die Anzahl der Wechselwirkungsobjekte zurück.

Methods inherited from class PhysikalischeObjekte .Teilchen

`b, b, max_v_koord, r, r, v, v`

87, SphTeilchen

Constructor Detail

SphTeilchen

public **SphTeilchen** ()

macht nichts!

Method Detail

force_init

public void **force_init**(PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

---

init

public void **init**()

Ruft die gleichnamige Template−Funktion auf.

---

rho

public double **rho**()

Gibt die Dichte zurück.

---

ww

public void **ww**(SphWW * ww)

Fügt ein neues Wechselwirkungsobjekt zu den anderen hinzu.

---

wwanz

87, SphTeilchen

public int **wwanz** ()

Gibt die Anzahl der Wechselwirkungsobjekte zurück.

PhysikalischeObjekte

Class SphWW

Direct Known Subclasses:PressArtviscWW

class SphWW

Sph−Wechselwirkungsteilchen der Sph−AkkScheibe und des Sph−Teilchen.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

SphWW()
Definiert die Orts− und Geschwindigkeits−Differenzvektoren und die Kernel−Ableitung.

Method Summary

`valarray<double>`	**dW_ohne_fak** (SphTeilchen * t) Ruft die gleichnamige Template−Funktion auf.
`void`	**init**(SphTeilchen * t1, SphTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys) Ruft die gleichnamige Template−Funktion auf.
`double`	**W_ohne_fak** () Gibt den Kernel ohne den Faktor zurück.

Constructor Detail

SphWW

public **SphWW**()

Definiert die Orts− und Geschwindigkeits−Differenzvektoren und die Kernel−Ableitung.

Method Detail

dW_ohne_fak

public valarray<double> **dW_ohne_fak** (SphTeilchen * t)

Ruft die gleichnamige Template−Funktion auf.

---

init

public void **init**(SphTeilchen * t1, SphTeilchen * t2, valarray<double> r, double dist, PhysSystem & sys)

Ruft die gleichnamige Template−Funktion auf.

W_ohne_fak

public double **W_ohne_fak** ()

Gibt den Kernel ohne den Faktor zurück.

Association Links

to **Class** SphTeilchen

Die Sph−Wechselwirkung bezieht sich auf 2 Sph−Teilchen, dessen Reihenfolge nicht vertauscht werden darf.

PhysikalischeObjekte

Class Teilchen

Direct Known Subclasses:SphTeilchen

class Teilchen

Teilchen der AkkScheibe

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

Teilchen ()
Initialisiert den Ort, die Geschwindigkeit und die Beschleunigung.

Method Summary

`valarray<double>`	**b**() Gibt die Beschleunigung zurück.
`void`	**b**(valarray<double> b) Setzt die Beschleunigung des Teilchens.
`void`	**init**() Setzt die Beschleunigung auf 0.
`double`	**max_v_koord** () Gibt die maximale Geschwindigkeitskoordinate aus.
`valarray<double>`	**r**() Gibt den Ort zurück.
`void`	**r**(valarray<double> r) Setzt den Ort des Teilchens.
`valarray<double>`	**v**() Gibt die Geschwindigkeit zurück.

Method Summary	
`void`	**v**(valarray<double> v) Setzt die Geschwindigkeit des Teilchens.

Constructor Detail

Teilchen

public **Teilchen** ()

Initialisiert den Ort, die Geschwindigkeit und die Beschleunigung.

Method Detail

b

public valarray<double> **b**()

Gibt die Beschleunigung zurück.

---

b

public void **b**(valarray<double> b)

Setzt die Beschleunigung des Teilchens.

---

init

public void **init**()

Setzt die Beschleunigung auf 0. Dies wird am Anfang der Kräfteberechnung ausgeführt.

---

max_v_koord

public double **max_v_koord** ()

Gibt die maximale Geschwindigkeitskoordinate aus.

---

r

public valarray<double> **r**()

Gibt den Ort zurück.

---

r

public void **r**(valarray<double> r)

Setzt den Ort des Teilchens.

---

v

public valarray<double> **v**()

Gibt die Geschwindigkeit zurück.

---

v

public void **v**(valarray<double> v)

Setzt die Geschwindigkeit des Teilchens.

Package Simulationen

Class Diagram Summary	
Simulationen	

Class Summary	
AkkDisplayQt	Dieses Qt−Widget zeigt die Akkretionsscheibe an.
Simulation	Ist die Elternklasse aller für die Steuerung verantwortlichen Klassen.
SimulationConsole	Steuert die Simulation, wobei die Ein− bzw Ausgaben über eine Konsole bzw.
SimulationQt	Steuert die Simulation über eine graphische Oberfläche (mit Hilfe der Qt−Library).
UpdateQt	Diese Klasse ist speziell auf die GUI−Klasse "SimulationQt" abgestimmt un ist für die Kommunikation zwischen der Simulation und der GUI−Klasse verantwortlich.

Simulationen

Class Diagram Simulationen

Class Summary	
AkkDisplayQt	Dieses Qt−Widget zeigt die Akkretionsscheibe an.
Simulation	Ist die Elternklasse aller für die Steuerung verantwortlichen Klassen.
SimulationConsole	Steuert die Simulation, wobei die Ein− bzw Ausgaben über eine Konsole bzw.
SimulationQt	Steuert die Simulation über eine graphische Oberfläche (mit Hilfe der Qt−Library).
UpdateQt	Diese Klasse ist speziell auf die GUI−Klasse "SimulationQt" abgestimmt un ist für die Kommunikation zwischen der Simulation und der GUI−Klasse verantwortlich.

Simulationen

Class AkkDisplayQt

---

class AkkDisplayQtderived from QWidget

Dieses Qt−Widget zeigt die Akkretionsscheibe an.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

AkkDisplayQt(QWidget * parent, const char * name)
Setzt die minimale Größe des Widget auf 500x500, setzt den Hintergrund schwarz initialisiert das Pixmap.

Destructor Summary

~AkkDisplayQt()
Löscht das Pixmap.

Method Summary

`void`	**updateAkk**(AkkScheibe * akk, double phi, double theta) Diese Funktion berechnet ein neues Pixmap und ruft danach die Funktion "repaint" (bzw.

Constructor Detail

AkkDisplayQt

public **AkkDisplayQt**(QWidget * parent, const char * name)

Setzt die minimale Größe des Widget auf 500x500, setzt den Hintergrund schwarz initialisier

das Pixmap.

Method Detail

~AkkDisplayQt

public **~AkkDisplayQt**()

Löscht das Pixmap.

Method Detail

updateAkk

public void **updateAkk**(AkkScheibe * akk, double phi, double theta)

Diese Funktion berechnet ein neues Pixmap und ruft danach die Funktion "repaint" (bzw. paintEvent) auf.

Association Links

to **Class** AkkScheibe

Ist die darzustellende Akkretionsscheibe.

Simulationen

Class Simulation

Direct Known Subclasses:SimulationConsole, SimulationQt

class Simulation

Ist die Elternklasse aller für die Steuerung verantwortlichen Klassen.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

Simulation ()
Initialisiert die Membervariablen auf 0.

Destructor Summary

~Simulation ()
löscht die Membervariablen.

Constructor Detail

Simulation

public **Simulation** ()

Initialisiert die Membervariablen auf 0.

Method Detail

~Simulation

public **~Simulation** ()

löscht die Membervariablen.

Association Links

to **Class** AkkScheibe

Die in der Simulation benutzte Akkretionsscheibe.

to **Class** PhysSystem

Die in der Simulation benutztes physikalisches System.

to **Class** RechteSeiteDG

Die in der Simulation benutzte rechte Seite der DG, die aus vorher bestimmten Kräften zusammengesetzt ist.

to **Class** Odeint

Der in der Simulation benutzte Integrator.

to **Class** Kernel

Der in der Simulation benutzte Kernel.

to **Class** InitAkkScheibe

Wird für die Initialisierung der Akkretionsscheibe benötigt.

to **Class** InitPhysSystem

Wird für die Initialisierung des physikalischen Systems benötigt.

to **Class** InitOdeint

Wird für die Initialisierung der Integrators benötigt.

to **Class** InitRechteSeiteDG

Wird für die Initialisierung der rechten Seite der DG benötigt.

Simulationen

Class SimulationConsole

---

class SimulationConsolederived from Simulation

Steuert die Simulation, wobei die Ein− bzw Ausgaben über eine Konsole bzw. ein Terminal erfolg

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary
SimulationConsole (const char * parameterfile, const char * teilchenfile) Initialisiert die Membervariablen.

Method Summary	
`void`	**run**(int argc, char * * argv) Startet die Simulation.

Constructor Detail

SimulationConsole

public **SimulationConsole** (const char * parameterfile, const char * teilchenfile)

Initialisiert die Membervariablen.

Method Detail

run

public void **run**(int argc, char * * argv)

Startet die Simulation.

Simulationen

Class SimulationQt

class SimulationQt derived from QWidget, Simulation

Steuert die Simulation über eine graphische Oberfläche (mit Hilfe der Qt−Library).

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Association Links

to **Class** UpdateQt

Dieser Pointer zeigt auf das Update−Objekt, das die Simulation mit der GUI synchronisiert.

to **Class** AkkDisplayQt

Zeigt auf ein Qt−Widget, das die Akkretionsscheibe darstellt.

Simulationen

Class UpdateQt

class UpdateQtderived from Update

Diese Klasse ist speziell auf die GUI−Klasse "SimulationQt" abgestimmt und ist für die Kommunikation zwischen der Simulation und der GUI−Klasse verantwortlich.

Author:
Andreas Nagel
Version: 1.0
date 4.7.2000

Constructor Summary

UpdateQt ()

Destructor Summary

~UpdateQt ()
Löscht die Membervariablen "_sim".

Method Summary

`virtual void`	**processEvents** () Diese Funktion ruft die Funktion "processEvents" der GUI auf.
`void`	**set_widget** (SimulationQt * sim) Setzt die Membervariable "_sim".
`virtual void`	**update** () Ruft die Funktion "updateAkk" der GUI−Klasse "_sim" auf, die die graphisch dargestellte Akkretionsscheibe neu darstellt.

Constructor Detail

104, UpdateQt

UpdateQt

public **UpdateQt** ()

Method Detail

~UpdateQt

public **~UpdateQt** ()

Löscht die Membervariablen "_sim".

Method Detail

processEvents

public virtual void **processEvents** ()

Diese Funktion ruft die Funktion "processEvents" der GUI auf.

---

set_widget

public void **set_widget** (SimulationQt * sim)

Setzt die Membervariable "_sim".

---

update

public virtual void **update** ()

Ruft die Funktion "updateAkk" der GUI−Klasse "_sim" auf, die die graphisch dargestellte Akkretionsscheibe neu darstellt. Diese Funktion wird von der Simulation aufgerufen, wenn sie eine Akkretionsscheibe berechnet hat.

Association Links

104, UpdateQt

105, UpdateQt

to **Class** SimulationQt

Ist die GUI−Klasse, zu der die Informationen der Simulation weitergegeben werden.

105, UpdateQt