

NumericalPropagationWithSRP 4.5.1

De Wiki

Aller à : [navigation](#), [rechercher](#)

[NumericalPropagationWithSRP 4.5.1](#)

```
public static void main(String[] args) throws PatriusException,
IOException, ParseException, URISyntaxException {

    // Patrius Dataset initialization (needed for example to get the UTC
time)
    PatriusDataset.addResourcesFromPatriusDataset() ;

    // Recovery of the UTC time scale using a "factory" (not to duplicate
such unique object)
    final TimeScale TUC = TimeScalesFactory.getUTC();

    // Date of the orbit given in UTC time scale)
    final AbsoluteDate date = new AbsoluteDate("2010-01-01T12:00:00.000",
TUC);

    // Getting the frame with wich will defined the orbit parameters
// As for time scale, we will use also a "factory".
    final Frame GCRF = FramesFactory.getGCRF();

    // Initial orbit
    final double sma = 7000.e+3;
    final double exc = 0.;
    final double per = sma*(1.-exc);
    final double apo = sma*(1.+exc);
    final double inc = FastMath.toRadians(98.);
    final double pa = FastMath.toRadians(0.);
    final double raan = FastMath.toRadians(0.);
    final double anm = FastMath.toRadians(0.);
    final double MU = Constants.WGS84_EARTH_MU;

    final ApsisRadiusParameters par = new ApsisRadiusParameters(per, apo,
inc, pa, raan, anm, PositionAngle.MEAN, MU);
    final Orbit iniOrbit = new ApsisOrbit(par, GCRF, date);

    // Mass model using an Assembly

    final AssemblyBuilder builder = new AssemblyBuilder();

    // Initial mass (mandatory to take into account mass for atmospheric
force computation)
    final double dryMass = 100.;
    builder.addMainPart("MAIN");
    builder.addProperty(new MassProperty(dryMass), "MAIN");
```

```

//SPECIFIC
    final double ka = 1.0;
    final double ks = 0.0;
    final double kd = 0.0;
    builder.addProperty(new RadiativeProperty(ka, ks, kd), "MAIN");
    builder.addProperty(new RadiativeIRProperty(ka, ks, kd), "MAIN");
    final double radius = 10.;
    builder.addProperty(new RadiativeSphereProperty(radius), "MAIN");
//SPECIFIC

    final Assembly assembly = builder.returnAssembly();

//SPECIFIC
    // Sun ephemeris
    CelestialBody sun = new MeeusSun();

    // Definition of the Earth ellipsoid for later SRP computation
    final Frame ITRF = FramesFactory.getITRF();
    final double AE = Constants.WGS84_EARTH_EQUATORIAL_RADIUS;
    final GeometricBodyShape EARTH = new ExtendedOneAxisEllipsoid(AE,
Constants.WGS84_EARTH_FLATTENING, ITRF, "EARTH");

    // Direct SRP data
    final double dRef = 1.4959787E11;
    final double pRef = 4.5605E-6;
    DirectRadiativeModel rm = new DirectRadiativeModel(assembly);
    SolarRadiationPressureEllipsoid radPres = new
SolarRadiationPressureEllipsoid(dRef, pRef, sun, EARTH, rm);

    // Rediffused DRP data
    final int inCorona = 1;
    final int inMeridian = 10;
    final IEmissivityModel inEmissivityModel = new KnockeRiesModel();
    final boolean inAlbedo = true;
    final boolean inIr = true;
    final double coefAlbedo = 1.;
    final double coefIr = 1.;
    RediffusedRadiativeModel rdm = new RediffusedRadiativeModel(inAlbedo,
inIr, coefAlbedo, coefIr, assembly);
    RediffusedRadiationPressure reDiff = new
RediffusedRadiationPressure(sun, GCRF, inCorona, inMeridian,
inEmissivityModel, rdm);
//SPECIFIC

    final MassProvider mm = new MassModel(assembly);

    // We create a spacecraftstate
    final SpacecraftState iniState = new SpacecraftState(iniOrbit, mm);

    // Initialization of the Runge Kutta integrator with a 2 s step
    final double pasRk = 2.;

```

```

        final FirstOrderIntegrator integrator = new
ClassicalRungeKuttaIntegrator(pasRk);

        // Initialization of the propagator
        final NumericalPropagator propagator = new
NumericalPropagator(integrator);
        propagator.resetInitialState(iniState);

        // Adding additional state (change name add to set for V3.3)
        propagator.setMassProviderEquation(mm);

        // Forcing integration using cartesian equations
        propagator.setOrbitType(OrbitType.CARTESIAN);

//SPECIFIC
        // Adding SRP forces
        propagator.addForceModel(radPres);
        propagator.addForceModel(reDiff);
//SPECIFIC

        // Propagating 5 periods
        final double dt = 5.*iniOrbit.getKeplerianPeriod();
        final AbsoluteDate finalDate = date.shiftedBy(dt);
        final SpacecraftState finalState = propagator.propagate(finalDate);
        final Orbit finalOrbit = finalState.getOrbit();

        // Printing new date and semi major axis
        System.out.println();
        System.out.println("Initial semi major axis =
"+iniOrbit.getA()/1000.+ " km");
        System.out.println("New date = "+finalOrbit.getDate().toString(TUC)+
deg");
        System.out.println("Final semi major axis =
"+finalOrbit.getA()/1000.+ " km");
        // Printing mass
        System.out.println();
        System.out.println("Mass = "+finalState.getMass("MAIN")+ " kg");

    }

}

```

Récupérée de

« http://patrius.cnes.fr/index.php?title=NumericalPropagationWithSRP_4.5.1&oldid=2703 »

Menu de navigation

Outils personnels

- 18.222.56.71

- [Discussion avec cette adresse IP](#)
- [Créer un compte](#)
- [Se connecter](#)

Espaces de noms

- [Page](#)
- [Discussion](#)

Variantes

Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Historique](#)
- [Exporter en PDF](#)

Plus

Rechercher

PATRIUS

- [Welcome](#)

Evolutions

- [Main differences between V4.15 and V4.14](#)
- [Main differences between V4.14 and V4.13](#)
- [Main differences between V4.13 and V4.12](#)
- [Main differences between V4.12 and V4.11](#)
- [Main differences between V4.11 and V4.10](#)
- [Main differences between V4.10 and V4.9](#)
- [Main differences between V4.9 and V4.8](#)
- [Main differences between V4.8 and V4.7](#)
- [Main differences between V4.7 and V4.6.1](#)
- [Main differences between V4.6.1 and V4.5.1](#)
- [Main differences between V4.5.1 and V4.4](#)
- [Main differences between V4.4 and V4.3](#)

- [Main differences between V4.3 and V4.2](#)
- [Main differences between V4.2 and V4.1.1](#)
- [Main differences between V4.1.1 and V4.1](#)
- [Main differences between V4.1 and V4.0](#)
- [Main differences between V4.0 and V3.4.1](#)

User Manual

- [User Manual 4.15](#)
- [User Manual 4.14](#)
- [User Manual 4.13](#)
- [User Manual 4.12](#)
- [User Manual 4.11](#)
- [User Manual 4.10](#)
- [User Manual 4.9](#)
- [User Manual 4.8](#)
- [User Manual 4.7](#)
- [User Manual 4.6.1](#)
- [User Manual 4.5.1](#)
- [User Manual 4.4](#)
- [User Manual 4.3](#)
- [User Manual 4.2](#)
- [User Manual 4.1](#)
- [User Manual 4.0](#)
- [User Manual 3.4.1](#)
- [User Manual 3.3](#)

Tutorials

- [Tutorials 4.15](#)
- [Tutorials 4.14](#)
- [Tutorials 4.13.5](#)
- [Tutorials 4.12.1](#)
- [Tutorials 4.8.1](#)
- [Tutorials 4.5.1](#)
- [Tutorials 4.4](#)
- [Tutorials 4.1](#)
- [Tutorials 4.0](#)

Links

- [CNES freeware server](#)

Navigation

- [Accueil](#)
- [Modifications récentes](#)

- [Page au hasard](#)
- [Aide](#)

Outils

- [Pages liées](#)
 - [Suivi des pages liées](#)
 - [Pages spéciales](#)
 - [Adresse de cette version](#)
 - [Information sur la page](#)
 - [Citer cette page](#)
- Dernière modification de cette page le 17 août 2020 à 09:09.
- [Politique de confidentialité](#)
 - [À propos de Wiki](#)
 - [Avertissements](#)
 - 