NumericalPropagationtWithFixedStepHandler 4.1

```java
public class NumericalPropagationWithFixedStepHandler {

    public static void main(String[] args) throws PatriusException {

        // 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 = 7200.e+3;
        final double exc = 0.01;
        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);

        // We create a spacecratftstate
        final SpacecraftState iniState = new SpacecraftState(iniOrbit);

        // Initialization of the Runge Kutta integrator with a 2 s step
        final double pasRk = 2.;
        final FirstOrderIntegrator integrator = new
ClassicalRungeKuttaIntegrator(pasRk);
```

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

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

//SPECIFIC
        // Creation of a fixed step handler
        final ArrayList<SpacecraftState> listOfStates = new
ArrayList<SpacecraftState>();
        PatriusFixedStepHandler myStepHandler = new PatriusFixedStepHandler()
{
            private static final long serialVersionUID = 1L;
            public void init(SpacecraftState s0, AbsoluteDate t) {
                // Nothing to do ...
            }
            public void handleStep(SpacecraftState currentState, boolean
isLast)
                    throws PropagationException {
                // Adding S/C to the list
                listOfStates.add(currentState);
            }
        };
        // The handler frequency is set to 10S
        propagator.setMasterMode(10., myStepHandler);
//SPECIFIC

        // Propagating 100s
        final double dt = 101.;
        final AbsoluteDate finalDate = date.shiftedBy(dt);
        final SpacecraftState finalState = propagator.propagate(finalDate);

        // Display data at each step
        System.out.println(iniState.getDate().toString(TUC)+" ; LV =
"+FastMath.toDegrees(iniState.getLv())+ " deg");
        for (SpacecraftState sc : listOfStates) {
            System.out.println(sc.getDate().toString(TUC)+" ; LV =
"+FastMath.toDegrees(sc.getLv())+ " deg");
        }
        System.out.println(finalState.getDate().toString(TUC)+" ; LV =
"+FastMath.toDegrees(finalState.getLv())+ " deg");

    }

}
```

Récupérée de
«

Menu de navigation

Outils personnels

- 18.118.19.123
- 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

Rechercher Lire

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

User Manual

Tutorials

Links

- [CNES freeware server](#)

- Dernière modification de cette page le 18 décembre 2018 à 10:28.