

C++ code 9.8.29: Definition of class **SplitStepPropagator** → [GitLab](#)

```
2 class SplitStepPropagator : public Propagator {
3     using SparseMatrixXcd = Eigen::SparseMatrix<std::complex<double>>;
4     using SparseMatrixXd = Eigen::SparseMatrix<double>;
5
6     public:
7         // @brief Forwards the arguments to the constructors of the underlying
8         // propagators KineticPropagator (semi-step) and InteractionPropagator
9         // (full-step).
10        // @param A stiffness matrix of shape  $N \times N$ 
11        // @param M complex mass matrix of shape  $N \times N$ 
12        // @param tau size of the timestep to perform by Strang splitting
13        //
14        SplitStepPropagator(const SparseMatrixXd &A, const SparseMatrixXcd &M,
15                           double tau);
16        /** @brief Performs the propagation according Strang splitting between
17        the
18        kinetic (semi-step) and interaction (full-step) propagator.
19        /** @param mu vector of length N containing nodal values
20        before the timestep
21        /** @return vector of length N containing the nodal values
22        after the timestep
23        /**
24        Eigen::VectorXcd operator()(const Eigen::VectorXcd &mu) const override;
25
26    private:
27        // Kinetic propagator for semi step  $\Psi^{0,\frac{\tau}{2}}$ 
28        KineticPropagator kineticPropagator_;
29        // Interaction propagator for full step
30        InteractionPropagator interactionPropagator_;
31    };
```