C++ code 9.8.26: Implementation of class InteractionPropagator ⇢ GitLab

```cpp
InteractionPropagator::InteractionPropagator(double tau) {
  // We know that the discrete evolution operator for the non-linear part of the
  // method-of-lines ODE boils down to componentwise multiplication of the
  // vector with a single phase shift.  Thus, this phase shift can be
  // precomputed, here by means of a lambda function
  phase_multiplier_ = [tau](std::complex<double> z) {
    const std::complex<double> i(0, 1);
    return std::exp(-i * tau * std::norm(z)) * z;
  };
}

Eigen::VectorXcd InteractionPropagator::operator()(
    const Eigen::VectorXcd &mu) const {
  // Eigen's way of applying a function to all components of a vector.
  return mu.unaryExpr(phase_multiplier_);
}
```