

C++ code 9.2.29: Sub-problem (9-2.I), Auxliary class **S DIRK2Timestepper** → [GitLab](#)

```
2 class S DIRK2Timestepper {
3   public:
4     // Disabled constructor
5     S DIRK2Timestepper() = delete;
6     S DIRK2Timestepper(const S DIRK2Timestepper &) = delete;
7     S DIRK2Timestepper(S DIRK2Timestepper &&) = delete;
8     S DIRK2Timestepper &operator=(const S DIRK2Timestepper &) = delete;
9     S DIRK2Timestepper &operator=(const S DIRK2Timestepper &&) = delete;
10    // Main constructor; precomputations are done here
11    explicit S DIRK2Timestepper(const If::assemble::DofHandler &dofh, double tau,
12                                double cool_coeff);
13    // Destructor
14    virtual ~S DIRK2Timestepper() = default;
15
16    /* Class member functions */
17    // Discrete evolution operator for S DIRK-2
18    Eigen::VectorXd discreteEvolutionOperator(const Eigen::VectorXd &mu) const;
19
20   private:
21     double tau_; // step size (in time)
22     // Sparses matrices holding Galerkin matrices
23     Eigen::SparseMatrix<double> A_; // Element matrix (Laplace + bdy mass)
24     double lambda_; // coefficient for S DIRK-2 Butcher tableau
25     // For fixed step-size in time, the linear system of equations implicitly
26     // defining the Runge-Kutta increments are independent of time. We can thus
27     // precompute the LU decompositions for more efficiency. S DIRK-2 is diagonally
28     // implicit: precomputing the LU decomposition for the two stages
29     // independently makes for an easy implementation.
30     Eigen::SparseLU<Eigen::SparseMatrix<double>> solver;
31 };
```