Solving the discrete system

after discretization by any discretization method, one has to solve the resulting discrete system in order to determine the approximate solution

- For finite difference methods, one solves for the grid function u_{jn}

- For finite element methods, one solves for the coefficients c_j

- For linear PDEs, the discrete system is often a system of linear algebraic equations

  – direct solvers are mostly limited (because of storage and CPU costs) to problems in one and two dimensions

    - Cholesky factorization for symmetric, positive definite linear systems

    - Gauss elimination for general linear systems

    - super LU is the most efficient implementation of direct methods

– iterative solvers

  - Gauss-Seidel and conjugate gradients methods for symmetric, positive definite linear systems

  - GMRES, multigrid, and bi-cg methods for general systems

- For nonlinear problems, one has to linearize the equations before applying a linear system solver

  – use Newton's method, quasi-Newton methods, nonlinear conjugate gradient methods, ...

  – all of these methods are iterative in character, so that a code would be structured as follows

    outer loop – `nonlinear iteration`

        inner loop – `linear solve` (may be direct or iterative)

  – thus, for each iteration of, e.g., Newton's method, one has to solve a linear system