Individual project

Individual project is the form of final control together with oral exam. Individual projects can be done in groups of maximum two students. The results of all project assignments should be summarized in a report.

Formal presentation of the individual project can be regarded as taking exam. You can either take an oral exam or defend your individual project. In the latter case the project presentation will include oral discussion with each student about theoretical aspects of the used methods and approaches.

Task variants for individual project:

	Equation	**Boundary conditions**
1	$2\frac{\partial^2 u}{\partial x^2}+3\frac{\partial^2 u}{\partial x\partial y}+2\frac{\partial^2 u}{\partial y^2}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}+5u=2$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
2	$\frac{\partial^2 u}{\partial x^2}+3\frac{\partial^2 u}{\partial x\partial y}+20\frac{\partial^2 u}{\partial y^2}+xyu=x$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
3	$4\frac{\partial^2 u}{\partial x^2}-2\sin x\frac{\partial^2 u}{\partial x\partial y}+4\frac{\partial^2 u}{\partial y^2}+2\frac{\partial u}{\partial x}-\frac{\partial u}{\partial y}+u=\cos x$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
4	$\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 u}{\partial y^2}-x\frac{\partial u}{\partial x}-y\frac{\partial u}{\partial y}+u=1$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
5	$(1+x^2)\frac{\partial^2 u}{\partial x^2}+(y^2+x^2)\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 u}{\partial y^2}+x^2\frac{\partial u}{\partial x}+y^2\frac{\partial u}{\partial y}=x^2+y^2$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
6	$\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 u}{\partial y^2}-\frac{\partial u}{\partial x}=\sin(x^2+y^2)$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
7	$\frac{\partial^2 u}{\partial x^2}-2x\frac{\partial^2 u}{\partial x\partial y}+2\frac{\partial^2 u}{\partial y^2}-\lvert y\rvert\frac{\partial u}{\partial x}=\lvert x\rvert$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].
8	$\cos^2 y\frac{\partial^2 u}{\partial x^2}-2\sin x\cos y\frac{\partial^2 u}{\partial x\partial y}+\sin^2 x\frac{\partial^2 u}{\partial y^2}=1$	$u(x,y)\big\|_\Gamma=0$, where Γ is the boundary of the unit square [0,1]x[0,1].

Contents of the project

1. **Discretization of PDE.** Using finite difference method, perform a discretization of the given boundary-value problem.
   1.1. Obtain a linear system of equations with respect to $u_{i,j}$
   1.2. Choose the ordering algorithm to label the internal nodes $u_{i,j}$, $i,j=1,2,...,n$ in order to obtain variables with one index v_k, $k=1,2,...,n^2$ (for example, use the method of natural ordering). Write a Matlab program for the chosen ordering algorithm.
   1.3. Apply this algorithm to label the unknowns of the system and obtain the resulting matrix and the vector of the right-hand side.
   1.4. Write a Matlab program which generates the matrix A (of the size $n^2\times n^2$) and the right-hand side vector b (of the size n^2) for a given n, where n is the number of internal nodes in the direction of each coordinate axis.

2. **Sparse formats.** Develop an algorithm that transfers a matrix from a dense format into a given sparse format and vice versa. Write a Matlab program for this algorithm and (**optional task, for additional points**) for an algorithm of matrix-by-vector multiplication, when a matrix is stored in a given sparse format. Use the programs you wrote for the related PA and your variants for it.

   a) CSR (Compressed Sparse Row)
   b) CSC (Compressed Sparse Column)
   c) MSR (Modified Sparse Row)
   d) MSC (Modified Sparse Column)
   e) Ellpack-Itpack

3. **Norms and condition numbers.** Calculate the norms and condition numbers of the resulting matrix A (of the size $n^2 \times n^2$) for n=3, 5, 10, 50. Determine how the norms and condition numbers change depending on the dimension of the problem. Consider the following six types of matrix norms: $\|\cdot\|_1$, $\|\cdot\|_\infty$, $\|\cdot\|_2$, $\|\cdot\|_E$, $\|\cdot\|_{l_1}$, $\|\cdot\|_M$.

4. **Matrix properties and factorizations.** Study the properties of the resulting matrix. Is it symmetric (Hermitian) or positive definite? Calculate its eigenvalues for small values of *n*, plot the eigenvalues. Find basic matrix factorizations: QR, SVD, Schur, LU, Cholessky (if applicable).

Note 1. Test each solution method below for several dimensions of the linear system (n =3, 5, 10, 50, where n is the number of internal nodes in the directions of the coordinate axes). Vary the value of the accuracy for the solution.

Note 2. The programs for the iterative methods below can make use of the input matrix in a sparse format (apart from FOM, GMRES and Lanczos methods, which require additional procedures for solving simple linear systems). Optional task, for additional points: when using different solution methods for a linear system , try to enter your matrix in a sparse format and use a function for a matrix by vector multiplication from task 2. For the matrices of large dimensions, compare the CPU time when using conventional and sparse formats.

Note 3. If your matrix is not symmetric or Hermitian, consider a preconditioned system for the methods that require Hermitian (symmetric) or Hermitian (symmetric) positive definite matrix.

5. **Classic iterative methods.** Write programs for the following iterative methods. Use the programs you wrote for the related PA and your variants for it.

   5.1. Check the convergence condition of each iterative method for the matrix A of the obtained system of linear equations (for n=3, 5, 10).
   5.2. Solve the obtained system of linear equations by these iterative methods. Determine the number of iterations. Find optimal value of the relaxation parameter ω for SOR and SSOR methods.

   a) Simple iteration method,
   b) Jacobi method (J),
   c) Gauss-Seidel method (GS),
   d) Successive Over Relaxation method (SOR),
   e) Symmetric Successive Over Relaxation method (SSOR).

6. **Krylov subspace methods: FOM.** Write a program for a Full Orthogonalization (FOM) method:

   6.1. Write a subprogram for an algorithm of Arnoldi orthogonalization, which constructs an orthonormal basis for Krylov subspace $K_m(A, r_0)$ and thus reduces initial linear system $Ax = b$ to a linear system $H_m y = \beta e_1$ with upper Hessenberg matrix H_m.

   6.2. Using a subprogram for Arnoldi orthogonalization, write a program for FOM method, where you solve the resulting linear system $H_m y = \beta e_1$ internally in Matlab. You can choose any version of restarted FOM.

   6.3. Check the applicability condition of the given method. Solve the obtained system of linear equations by this method (for n=3, 5, 10). Determine the number of iterations (for restarted FOM). Compare results for several values of m.

7. **Krylov subspace methods: GMRES.** Write a program for a Generalized Minimal Residual (GMRES) method:

   7.1. Write a subprogram for an algorithm of Arnoldi orthogonalization, which constructs an orthonormal basis for Krylov subspace $K_m(A, r_0)$, and thus reduces initial linear system $Ax = b$ to a linear system $\overline{H_m} y = \beta e_1$ with upper Hessenberg rectangular matrix $\overline{H_m}$.

   7.2. **Optional task, for additional points**: Write a subprogram that transforms the given matrix $\overline{H_m}$ in the upper Hessenberg form to an upper triangular rectangular matrix $\overline{R_m}$ with the help of orthogonal (Givens) rotations, and thus reduces the linear system $\overline{H_m} y = \beta e_1$ to a linear system $\overline{R_m} y = \overline{g_m}$ or a linear system $R_m y = g_m$ with upper triangular square matrix R_m.

   7.3. Using a subprogram for Arnoldi orthogonalization, write a program for GMRES method, where you solve the resulting linear system $R_m y = g_m$ (in case of using Givens rotations) or the linear system $\overline{H_m} y = \beta e_1$ internally in Matlab. You can choose any version of restarted GMRES.

   7.4. Check the applicability condition of the given method. Solve the obtained system of linear equations by this method (for n=3, 5, 10). Determine the number of iterations (for restarted GMRES). Compare results for several values of m.

8. **Other projection methods.** Explore four or more other projection methods available in Matlab. You can find more details on the available methods in Matlab help (see *Iterative Methods for Linear Systems* and *Systems of Linear Equations* in Matlab Documentation). Study the algorithms that the chosen methods use. Check the applicability conditions of the chosen methods. Solve the obtained system of linear equations by these methods (for n=3, 5, 10). Determine the number of iterations.

   a) pcg (preconditioned conjugate gradients)
   b) lsqr (least squares)
   c) minres (minimum residual)
   d) symmlq (symmetric LQ)
   e) bicg (biconjugate gradient)
   f) bicgstab (biconjugate gradient stabilized)
   g) bicgstabl (biconjugate gradient stabilized (l))
   h) cgs (conjugate gradient squared)
   i) gmres (generalized minimum residual)
   j) qmr (quasi-minimal residual)

k) tfqmr (transpose-free quasi-minimal residual)

9. **Preconditioning.** Improve the convergence of the previous methods by using preconditioners. You can find more details on the preconditioners available in Matlab by checking Matlab help (see *Iterative Methods for Linear Systems* in Matlab Documentation). Study the types of the preconditioners the chosen methods can use. Apply preconditioners and check the results.

10. **Conclusions.** Compare the results of solving the given linear system using the above solution methods. Summarize your results in a table that contains the dimension of the problem, the dimension of the Krylov subspace, the accuracy, the relative error, the number of iterations and the CPU time. Make conclusions.