1 Introduction

... 

1.3 Scripting vs programming

**Question 1:**

1.3.1 What is a script?

1.3.2 Characteristics of a script

1 Introduction

... 

1.3 Scripting vs programming

... 

**Question 2:**

1.3.3 Why not stick to Java, C/C++ or Fortran?

1.4 Scripts yield short code

1 Introduction

... 

1.5 Performance issues

**Question 3:**

1.5.1 Scripts can be slow

1.5.2 Scripts may be fast enough

1 Introduction

... 

1.5 Performance issues

... 

**Question 4:**

1.5.3 When scripting is convenient

1.5.4 When to use C, C++, Java, Fortran
2 What is Scientific Computing?

Question 5:

2.1 Introduction to Scientific Computing

Question 6:

2.2 Specifics of computational problems

Question 7:

2.3 Mathematical model and general strategy

Question 8:

3.1 Sources of approximation error

3.1.1 Error sources that are under our control
3 Approximation in Scientific Computing
  3.1 Sources of approximation error

Question 9:
  3.1.2 Errors created during the calculations

Question 10:
  3.1.3 Forward error and backward error
(example excluded)

Question 11:
  3.2 Floating-Point Numbers

Question 12:
  3.3 Normalised floating-point numbers
3 Approximation in Scientific Computing

Question 13:

3.4 IEEE (Normalised) Arithmetics

5 Solving Systems of Linear Equations

Question 14:

5.1 Systems of Linear Equations

5 Solving Systems of Linear Equations

Question 15:

5.2 Classification

5.2.1 Problem Transformation

5 Solving Systems of Linear Equations

Question 16:

5.3 Triangular linear systems
5 Solving Systems of Linear Equations

\textbf{Question 17:}
5.4 Elementary Elimination Matrices

5 Solving Systems of Linear Equations

\textbf{Question 18:}
5.5 Gauss Elimination and LU Factorisation
5.6 Number of operations in GEM

5 Solving Systems of Linear Equations

\textbf{Question 19:}
5.7 LU-factorisation with GEM using row permutations

5 Solving Systems of Linear Equations

\textbf{Question 20:}
5.8 Reliability of the LU-factorisation with partial pivoting
Question 21:
6 BLAS (Basic Linear Algebra Subroutines)
   6.1 Motivation
   6.2 BLAS implementations

7 Numerical Solution of Differential Equations

Question 22:
   7.1 Ordinary Differential Equations

Question 23:
   7.2 Partial Differential Equations overview

Question 24:
   7.3 2nd order PDEs
Question 25:

7.4.1 Finite Difference Method (FDM)

Question 26:

7.4.2 Finite element Method (FEM)

Question 27:

7.5 Sparse matrix storage schemes

7.5.1 Triple storage format

Question 28:

8.1 Problem setup: Iterative methods for solving systems of linear equations with sparse matrices

8.2 Jacobi method
8 Iterative methods

Question 29:
  8.3 Conjugate Gradient Method
  8.4 Preconditioning

9. Some examples

Question 30:
  9.1 Google PageRank problem

9. Some examples

Question 31:
  9.2 Graph partitioning

10. Parallel Computing

Question 32:
  10.1 Motivation
10. Parallel Computing

*Question 33:*
10.2 Design and Evaluation of Parallel Programs

10. Parallel Computing

*Question 34:*
10.3 Assessing parallel programs

10. Parallel Computing

*Question 35:*
11 Parallel programming models