

Scientific Computing – Prüfungsfragen ab WS 23/24

25. Jänner 2024

1 Object-oriented scientific computing

- Allgemeinwissen (keine separate Prüfungsfrage): pointer/reference, call by value/call by reference, allocation on stack/heap, new/delete and memory leaks,
- Deklaration einer Template-Klasse: Bedeutung des Template-Parameters, Klassifikation der Member-Objekte/Funktionen, Konstruktoren, Destruktor, Zuweisungsoperator; Zugriff auf als `private/protected` klassifizierte Member; was bedeutet `const` in diesem Zusammenhang?
- `Vector<T>`: Erklären Sie die unterschiedliche Performance von direkter Überladung des

```
Vector<T> operator+(const Vector<T>& a, const Vector<T>& b)
```

und einer Funktion

```
void Add(const Vector<T>& a, const Vector<T>& b, Vector<T>& c)
```

anhand der Zuweisung $c = a + b$. Was würde passieren, wenn Sie in der Deklaration einzelne Referenzierungsoperatoren `&` vergessen?

- Vererbung: Was ist eine abstrakte Basisklasse? Was kann man damit nicht, warum? Was ist eine abgeleitete Klasse? Erklären Sie weiters die Begriffe `public`, `private`, `protected` und `virtual`, `override`. Was ist eine *pure virtual function*? Wie wirkt sich ein vergessenes `virtual` beim Überladen von Funktionen aus?

2 Numerical linear algebra

- Lineare Algebra: Kern/Nullraum/Rang einer Matrix, Zusammenhang Singulärwertzerlegung; Definition einer Pseudoinversen aus der SVD und Anwendung aus über-/unterbestimmte Systeme.
- Eigenschaften einer Norm. Vektornormen - Beispiele. Matrixnorm - Kompatibilität, induzierte Norm, Submultiplikativität + Beispiele
- Gleitkommadarstellung: Darstellung eines 64bit double, Wertebereich, Fließkommagenauigkeit.
- Datenstabilität und Konditionszahl, Kondition einer Matrix
- LU Zerlegung: Algorithmus (Idee), Anwendung der Inversen, Pivotsuche, Komplexität, Vor-/Nachteile
- QR Zerlegung: Algorithmus (Idee), Anwendung der Inversen, Komplexität, Stabilität, Vor-/Nachteile
- Überbestimmte Systeme: Normalengleichung, Lösung über QR-Zerlegung, Alternative SVD, Anwendung auf Regressionsprobleme.
- LAPACK: Warum? Wie sehen LAPACK-Routinen aus (Benennung, welche Typen werden übergeben)?
- `SparseMatrix<T>`: Datenstruktur, wie erfolgt der Zugriff auf einzelne Einträge bzw. Anwendung (`Apply`), welche Komplexität? Welche Methoden zur Lösung dünnbesetzter Gleichungssysteme gibt es? Direkte Löser: vgl. Komplexität Berechnung der Inversen und LU-Faktorisierung für Bandmatrizen, Idee Nested Dissection.

- Iterative Methoden: Idee der residuenbasierten Verfahren, Varianten (Richardson, Jacobi, Gauss-Seidel, SOR), Konvergenzabschätzung
- Iterative Methoden: CG-Verfahren, Definition Krylov-Raum
- Krylov-Räume und Arnoldi-Iteration; Anwendungsbereiche (soweit im Stoffumfang der LVA)
- Iterative Methoden: GMRES-Verfahren (unter Verwendung Arnoldi-Iteration)

3 Applied computational methods

- Eigenwertproblem für allgemeines $\mathbf{A} \in \mathbb{C}^{n \times n}$, Eigenschaften, Shift, Ähnlichkeitstransformation, Schur-Form, Spezialfall für \mathbf{A} symmetrisch/hermitesch, das verallgemeinerte Eigenwertproblem für symmetrische reelle Matrizen.
- Der QR-Algorithmus, Bedeutung der Arnoldi-Iteration in diesem Zusammenhang
- Inverse Iteration
- Nichtlineare Probleme: Newton-Iteration, Algorithmus (Idee), Konvergenz, Dämpfung.
- Nichtlineare Ausgleichsprobleme, das Gauß-Newton-Verfahren.