MA3227 Numerical Analysis II

Lecture 10: Module Summary

Simon Etter

Semester II, AY 2020/2021

Module Summary

Exam details

- Date: Thursday, 6 May 2021, 9-11am,
- Venue: S17-05-12
- Open book. In particular, you are allowed to bring your own laptop to access the lecture slides and any other static resources (e.g. Wikipedia, but not WhatsApp or Facebook).

  Please let me know if you do not have access to a laptop.
- Pen-and-paper only. No coding or Julia-specific questions.
- Exam will test understanding, not memorisation and mindless execution of algorithms. In particular, most questions will not follow any pattern that you have seen before.

Module Summary

Topics

- Bisection and Newton method
- Finite differences
- Sparse LU factorisation
- Krylov subspace methods
- Runge-Kutta methods
- Monte Carlo methods

Not independently examinable:

- Big O notation

Module Summary

Topic blueprint

- Problem statement
- Background: existence and uniqueness, applications
- Algorithm: structure, derivation/motivation, runtime & convergence
- Miscellaneous

Module Summary

Root-finding

- Problem statement:

$$\text{Given } f : \mathbb{R}^n \to \mathbb{R}^n, \text{ find } x \in \mathbb{R}^n \text{ such that } f(x) = 0.$$

Bisection method

- Algorithm structure:

  Given $f : \mathbb{R} \to \mathbb{R}$ and an initial bracketing interval $[a_0, b_0]$,
  construct a sequence of increasingly tighter
  bracketing intervals $[a_k, b_k]$.

- Existence: bracketing theorem.
  Uniqueness: not guaranteed.
- Runtime: $O(1)$ per bisection step.
  Convergence: $|b_k - a_k| = 2^{-k} \, |b_0 - a_0|$.
- Miscellaneous:
  - Converges to machine precision in at most 63 iterations.
  - Optimal root-finder (in some sense).
  - Only applies to one-dimensional root-finding.

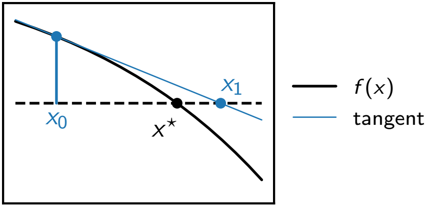
Module Summary

Newton's method

- Algorithm structure:

  Given differentiable $f : \mathbb{R} \to \mathbb{R}$ and an initial guess x_0,
  construct sequence $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_{k-1}}$

- Existence & uniqueness: not guaranteed.
- Derivation/motivation: Roots of tangents.



Module Summary

Newton's method (continued)

- Runtime: $O(1)$ per iteration.

  Local convergence: $\quad x_{k+1} - x^\star = O\big((x_k - x^\star)^2\big) \quad$ if $f'(x^\star) \neq 0$.

  Global conv.: not guaranteed in general, provable in special cases.
- Miscellaneous:
  - Deciding when to terminate can be difficult.
  - Works in any dimension.
- Exam prep: study convergence proof.

Module Summary

Finite differences

- Problem statement:

  Given $f : [0,1]^d \to \mathbb{R}$, determine $u : [0,1]^d \to \mathbb{R}$ such that

  $$-\Delta u(x) = f(x) \qquad \text{for all } x \in [0,1]^d.$$

- Existence & uniqueness: not discussed in this module.
- Algorithm: Solve $-\Delta_n^{(d)} u_n = f$.
- Derivation/motivation:
  - Represent functions as vectors of point values.
  - Replace derivatives with (nested) finite differences.
- Runtime: One sparse linear system solve.
- Convergence theory:
  - Stability & consistency lemma
  - Stability via Fourier analysis
  - Consistency via Taylor series
  - I will not examine the energy method.
- I will not examine Kronecker product.
- Exam prep: study convergence theory.

Module Summary

Sparse LU factorisation

- Problem statement: Solve $Ax = b$.
- Algorithm structure, existence & uniqueness: obvious / elementary.
- Convergence: Not applicable.
- Runtime:

  $$O\left(\sum_{k=1}^{n} \operatorname{nnz}\big(L[:,k]\big)\,\operatorname{nnz}\big(U[k,:]\big)\right).$$

  Sparsity pattern can be predicted using fill path theorem:

  $$(L+U)[i,j] \neq 0 \quad \Longleftrightarrow \quad \exists \text{ fill path } j \to i \text{ in } G(A)$$

- Miscellaneous:
  - Terminology: sparsity pattern / structure, fill-in
  - Permutations for improving performance
  - Nested dissection permutation
- Exam prep: Practise application of fill path theorem.

Module Summary

Krylov subspace methods

- Problem statement: Solve $Ax = b$.
- Algorithm structure:

$$\text{Compute} \qquad x_n = p(A)\,b \qquad \text{where} \qquad p = \underset{p \in \mathcal{P}_n}{\arg\min} \, \|b - A\,p(A)\,b\|.$$

- Motivation: Polynomials require only addition and multiplication.
- Runtime:

$$\begin{aligned} \text{GMRES:} \quad & O(n\,\mathrm{nnz}(A) + n^{2}\,\mathrm{len}(b)) \\ \text{MinRes, CG:} \quad & O(n\,\mathrm{nnz}(A) + n^{1}\,\mathrm{len}(b)) \end{aligned}$$

- Convergence:
  - Exact result once degree$(p) \geq$ [# distinct eigenvalues] $- 1$.
  - Exponential convergence with known rate if eigenvalues are contained in $[c, d] \subset (0, \infty)$.

Module Summary

Krylov subspace methods (continued)

- Krylov subspace methods:
  - GMRES: General A, minimise residual 2-norm
  - MinRes: Symmetric A, minimise residual 2-norm
  - Conjugate gradients: spd A, minimise residual A^{-1}-norm / error A-norm
- Performance tricks:
  - Restarted GMRES
  - Preconditioning

Module Summary

Krylov subspace methods (continued)

- Conjugate gradients vs LU factorisation for $-\Delta_n^{(d)}\, u_n = f$:

	Runtime		**Memory**	
	LU	CG	LU	CG
$d = 1$	$O(N)$	$O(N^2)$	$O(N)$	$O(N)$
$d = 2$	$O(N^{3/2})$	$O(N^{3/2})$	$O(N \log(N))$	$O(N)$
$d = 3$	$O(N^2)$	$O(N^{4/3})$	$O(N^{4/3})$	$O(N)$

$N = n^d$ denotes the number of unknowns.

Module Summary

Runge-Kutta methods

- Problem statement:

$$\text{Given} \qquad f : \mathbb{R}^n \to \mathbb{R}^n, \qquad y_0 \in \mathbb{R}^n \qquad \text{and} \qquad T > 0,$$

$$\text{determine } y : [0, T) \to \mathbb{R}^n \text{ such that}$$

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = f\big(y(t)\big) \quad \text{for all } t \in [0, T).$$

- Existence & uniqueness: Picard-Lindelöf theorems (global and local)
  Continuity of solutions
- Algorithm structure:
  - Split $[0, T]$ into many small intervals $[t_{k-1}, t_k]$
  - On each small interval, compute

$$\tilde{y}(t_k) = \tilde{y}(t_{k-1}) + \sum_{j=1}^{s} f(\tilde{y}_j^{(k)})\, w_j\, (t_k - t_{k-1})$$

- Runtime: $O(ns)$ Convergence: $O(n^{-p})$
  Convergence theory is based on error bound and comparing Taylor series of exact and numerical solutions (order of consistency).

Module Summary

Runge-Kutta methods (continued)

- Miscellaneous
  - Butcher tableaus
  - Stability and implicit Runge-Kutta schemes
  - I will not examine adaptive step size control
- Exam prep:
  - Practise Butcher tableaus
  - Practise determining orders of consistency (but not beyond third-order consistency)
  - Practise determining stability functions, and make sure you understand their meaning

Module Summary

Monte Carlo methods

- Algorithm structure:

$$\tilde{\mathbb{E}}_N[x] \;=\; \frac{1}{N}\sum_{k=1}^{N} X_k \;\approx\; \mathbb{E}[X].$$

- Convergence: $O(N^{-1/2})$
  Convergence theory is based on central limit theorem.
- Motivation: avoid complicated sums, beat the curse of dimensionality.
- Sampling theorems:
  - Inverse transform sampling: fast if applicable
  - Rejection sampling: usually slow, but flexible
  - Importance sampling: "rejection sampling without rejecting", variance reduction.
- Exam prep: recap sampling theorems and their various tradeoffs.

Module Summary

Recommended exam preparation

- Go over all the lecture slides once more. Check with your friends or myself if anything seems unclear.
- Work through the “exam prep” points mentioned above.
- Work through the tutorials and assignments once more.

Good luck and don’t panic!