Combinatorial Problems in Scientific Computing

Introduction

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen University

Contents

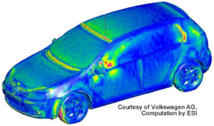
Outline

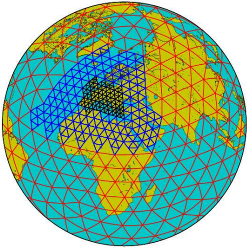
Motivation

- Newton's Method
- SuiteSparse Matrix Collection

Admin

Scientific computing uses mathematical modelling to numerically simulate / predict / optimize reality, e.g, automotive engineering, geophysics, ...





Numerical simulations are implemented as computer programs.

Sparsity is induced by missing data dependences, e.g, today's temperature in Hong Kong is probably independent of yesterday's temperature in Aachen, ...

Numerical programs exhibit structure, e.g, iteration, recursion, ...

Efficiency and robustness requires solution of various combinatrial problems (ordering, coloring, elimination, ...) on sparse matrices and graphs.

Key ingredients of scientific computing are the numerical approximation of solutions of

- systems of nonlinear equations

$$F(x) = 0, \quad F : \boldsymbol{R}^n \to \boldsymbol{R}^n$$

  for given implementations of the residual $y = F(x)$ and

- convex unconstrained nonlinear optimization problems

$$\text{argmin}_x f(x), \quad f : \boldsymbol{R}^n \to \boldsymbol{R}$$

  for given implementations of the objective $y = f(x)$

by Newton's method. The method uses differentiation and linear algebra $\Rightarrow$ plenty of structure and sparsity to exploit.

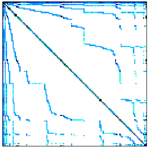
Ingredients of Newton's Method

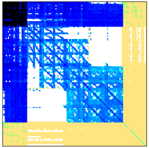
The content of this course inspired by the ingredients of Newton's method, namely

- solution of systems of linear equations involving
  - sparse matrix-vector products
  - sparse matrix chain products
  - sparse direct linear solvers
- algorithmic differentiation involving
  - accumulation of sparse Jacobians and Hessians
  - data-flow reversal

The discussion of these topics will be based on introductions to essential background of Newton's method and of algorithmic differentiation.

We use generic residuals $F(x)[=0]$ and objectives $[\min_x] f(x)$ replicating given sparsity patterns, e.g, provided by the SuiteSparse Matrix Collection; see sparse.tamu.edu.





... not a virus!

A dense matrix has n^2 nonzero entries (obviously, ...).

A matrix is sparse if the number of nonzero entries is $O(n)$.

Consider linear system $A \cdot x = b$ with regular system matrix $A \in \boldsymbol{R}^{n \times n}$ and solved as

$$L \cdot \underbrace{U \cdot x}_{:=z} = b$$

by

- $A = L \cdot U$ decomposition at cost of $O(n^3)$
- forward substitution for $L \cdot z = b$ at cost of $O(n^2)$
- backward substitution for $U \cdot x = z$ at cost of $O(n^2)$

Obviously, treatment of a lower triangular A as dense, e.g,

$$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$$

is suboptimal.

Householder Reflection of $a \in \boldsymbol{R}^n$:

$$v = a + \text{sign}(a_0) \cdot \|a\| \cdot e_0$$

$$H = I - 2 \cdot \frac{v \cdot v^T}{v^T \cdot v}$$

$$a = H \cdot a$$

better at cost of $O(n)$ (instead of $O(n^2)$)

$$v = a + \text{sign}(a_0) \cdot \|a\| \cdot e_0$$

$$a = a - 2 \cdot \frac{v^T \cdot a}{v^T \cdot v} \cdot v$$

- associativity of matrix multiplication

$$\left(\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\cdot\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\right)\cdot\begin{pmatrix}1\\3\end{pmatrix}=\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\cdot\left(\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\cdot\begin{pmatrix}1\\3\end{pmatrix}\right)$$

- distributivity of real arithmetic

$$a_1\cdot b_1\cdot c\cdot d+a_2\cdot b_2\cdot c\cdot d=(a_1\cdot b_1+a_2\cdot b_2)\cdot c\cdot d$$

- structural orthogonality of vectors

$$\left(\begin{pmatrix}1\\0\end{pmatrix}+\begin{pmatrix}0\\2\end{pmatrix}\right)\cdot 3$$

carries same information as

$$\left(\begin{pmatrix}1\\0\end{pmatrix}\cdot 3,\begin{pmatrix}0\\2\end{pmatrix}\cdot 3\right)$$

in less storage (and potentially at lower computational cost).

Outline

Motivation

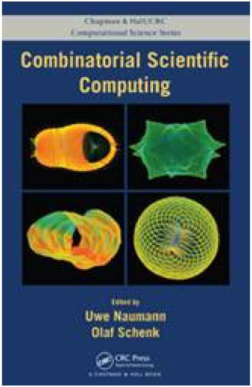
Newton's Method

SuiteSparse Matrix Collection

Admin

Further Details on RWTH Moodle

- The lecture covers general story including theory and algorithms (videos + Q&A via Zoom).
- The tutorial (via Zoom) supports the lecture by discussing both pen&paper as well as programming exercises → bonus credits
- Learning material consists of videos, slides, code, reference solutions for tutorial exercises, pointers into literature, e.g, ...



- Registration (RWTH online) "buys" you access to the RWTH Moodle page of the course.

Bonus Credits

for Tutorial Exercises

You can earn up to two bonus credits for submission of reasonable solutions to

- 60% (single credit)
- 80% (two credits)

of the tutorial exercises.

A single credit improves your final grade (≤ 4.0) by one level (e.g. from 1.7 to 1.3).

Summary

… of this lecture

- Scientific computing is an essential element of mankind's toolbox for understanding the world.
- Combinatorial problems occur in the context of many scientific computing methods.
- For example, they can be posed as ordering, coloring or elimination problems on sparse matrices and/or graphs.
- Knowledge of algorithms for the solution of these problems enables efficient and robust implementation of numerical simulation methods.
- You should attend this course … ;-)