Introduction to Scientific Computing

Steve Marschner

Cornell CS 322

Outline

Ariane 5 explosion

Ariane 5 explosion

First launch of Ariane 5 expendable launcher lost control and exploded at $t + 40$ sec.

Problem: number format conversion error

- Convert 64-bit FP (Matlab `double`) 16-bit signed integer (Matlab `int16`)
- Value is $\geq 2^15$
- Uncaught exception $\implies$ crash of guidance system

Loss: 5×10^8 dollars (that's $25,856 as an `int16`)

Patriot anti-missile system failure

Patriot anti-missile system failure

Patriot had trouble hitting fast-moving targets, sometimes.

In 1991 a Scud missile hit a barracks, killing 28, triggering a US government investigation. . .

- System makes time calculations to sync to radar
- Starts to fail after it's been up for a day or so

Patriot anti-missile system failure

Clock reads in tenths of a second since boot time

Software multiplies by 0.1 to get seconds

- Number format: fixed point with 24 bits of fraction
  - precision better than a millionth of a second
  - does not represent 1/10 exactly

What is the error in representing 0.1 and what is its effect?

Sleipner A offshore platform failure

Norwegian offshore drilling platform collapses and sinks

Concrete structure designed using finite element structural analysis

- What this means, roughly
- Approximation not precise enough

An example of a higher-level problem
$\longrightarrow$ maybe not a numerical one, really.

These stories borrowed from Douglas Arnold

read more: http://www.ima.umn.edu/∼arnold/disasters/disasters.html

Standard types of problems

- Systems of equations
  - linear geometric transformation circuit analysis
  - nonlinear orbital calculations
- Overconstrained systems
  - linear linear data modeling circuit analysis
  - nonlinear nonlinear data modeling
- Approximation
  - linear fitting polynomials
  - nonlinear fitting gamma curves

Standard types of problems

- Optimization
  - nonlinear practically everything
  - constrained spacetime constraints
  - linear programming economics
- Integration
  - definite integrals (quadrature) lighting
  - differential equations
    - initial value problem mechanical systems
    - boundary value problem ordinary: hanging chain cloth, fluids

Types of methods

- Direct (fixed sequence of operations leads to an answer—number of steps is predetermined)
- Iterative (repeatedly refine an approximation until some accuracy goal is reached—number of steps variable)
- Probabilistic (approximate the answer by some computation based on a number of stochastic “experiments”)

Properties of methods

- Accuracy
- Stability
- Rate of convergence
- Execution time (time complexity)
- Memory requirements (space complexity)

Themes of CS322

- continuous mathematics
- finite precision arithmetic
  - precision, accuracy
  - intuitive sense for precision
- high level thinking
- stability
  - of problems
  - of algorithms

Themes of CS322

- iterative approximation
  - convergence (guarantees)
  - convergence (rate)
- efficiency
  - computational complexity
  - effective use of linear algebra libraries
- applications
- visualization
- direct vs. iterative methods
- linear vs. nonlinear problems

Deconstructing a matrix-vector product

There are a number of ways to think about the meaning of a matrix-vector product. Some examples follow, but first a more systematic nomenclature:

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

This 3×3 example can serve to define this notation for any size matrices and vectors. Note that the row index is first, then the column index (contrary to C, Java, etc.).

Deconstructing a matrix-vector product

scalar level

A matrix-vector product is shorthand for a sum of products of entries:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$b_i = \sum_{j=1}^{3} a_{ij} x_j \quad \text{for } i = 1, 2, 3$$

Deconstructing a matrix-vector product

scalar level

A matrix-vector product is shorthand for a sum of products of entries:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$b_i = \sum_{j=1}^{n} a_{ij} x_j \quad \text{for } i = 1, \ldots, n$$

Deconstructing a matrix-vector product

dot product

A matrix-vector product is shorthand for dot products with the rows:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\begin{bmatrix} — \mathbf{r}_1 — \\ — \mathbf{r}_2 — \\ — \mathbf{r}_3 — \end{bmatrix} \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \mathbf{r}_2 \cdot \mathbf{x} \\ \mathbf{r}_3 \cdot \mathbf{x} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$b_i = \mathbf{r}_i \cdot \mathbf{x} \quad \text{for } i = 1, 2, 3$$

Deconstructing a matrix-vector product

dot product

A matrix-vector product is shorthand for dot products with the rows:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \text{---} \ \mathbf{r}_1 \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{r}_n \ \text{---} \end{bmatrix} \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \vdots \\ \mathbf{r}_n \cdot \mathbf{x} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$b_i = \mathbf{r}_i \cdot \mathbf{x} \quad \text{for } i = 1, \ldots, n$$

Deconstructing a matrix-vector product

linear combination

A matrix-vector product is shorthand for a linear combination of columns:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\begin{bmatrix} | & | & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} | \\ \mathbf{b} \\ | \end{bmatrix}$$

$$\mathbf{b} = x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + x_3\mathbf{c}_3$$

Deconstructing a matrix-vector product

linear combination

A matrix-vector product is shorthand for a linear combination of columns:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} | & & | \\ \mathbf{c}_1 & \cdots & \mathbf{c}_n \\ | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} | \\ \mathbf{b} \\ | \end{bmatrix}$$

$$b = \sum_{j=1}^{n} x_j \mathbf{c}_j$$