1. **Tools: Libraries and Software**

- In addition to standard tools like editors, compilers, or debuggers, there is a lot of (commercial or public domain) support available:

  - **Modelling:** Computer algebra programs like Mathematica, Maple, Axiom, or Reduce support derivations and proofs of theorems via symbolic means.
  
  - **Numerics:** Mathematica, Maple, or MATLAB support the development, testing, and analysis of (numerical) algorithms and allow an efficient prototyping.
  
  - **Implementation:** A zoo of (numerical) libraries provide up-to-date modules for standard tasks (numerical linear algebra etc.), tailored to specific target architectures.
  
  - **Visualization:** Packages like IDL, IRIS Explorer, or AVS/Express offer (nearly) all you want.
2. Libraries and Collections

- **GAMS**: Guide to Available Mathematical Software
  - service offered by the National Institute of Standards & Technology
  - catalogue and database of more than 100 packages and libraries with together several tens of thousands of routines
  - Topics range from number theory to statistics!
  - majority: FORTRAN programs for numerical tasks (systems of linear equations, eigenvalues, roots, differential equations, . . . )
  - includes both public domain material (at NIST or at NETLIB, see below) and commercial (/licenced) products.
  - good user guidance
3. Libraries and Collections

• Matrix market
  – see http://math.nist.gov/MatrixMarket/
  – repository of test data for use in comparative studies of algorithms for numerical linear algebra
  – features nearly 500 (sparse) matrices from various fields of applications (chemical engineering, fluid flow, power system networks, quantum physics, or structural engineering, e.g.)
  – provides also matrix generation tools
  – classification according to matrix properties:
    * number filed: real or complex
    * nonzero structure: dense, banded sparse, tridiagonal, ...
    * symmetry: none, symmetric, skewsymmetric, SPD, SSPD,...
    * shape: square, more rows than columns,...
4. Libraries and Collections

- **NETLIB**: repository of free software for numerical purposes
  - see [http://www.netlib.org/](http://www.netlib.org/)
  - offered by University of Tennessee and Oak Ridge Nat’l Lab
  - several mirrored copies all over the world
  - about 135 million requests since 1985, > 40 million in 2000
  - > 90% http, rest ftp and email
  - about 160 different libraries, among which
    - BLAS (Basic Linear Algebra Subprograms)
    - LAPACK (Linear Algebra PACKage)
    - ODEPACK (ordinary differential equations)
    - MPI (message passing interface, for parallelization)
    - PLTMG (elliptic boundary value problems)
5. **BLAS**

- collection of robust, efficient, and portable modules for elementary vector and matrix operations
- basis for LAPACK routines, for example
- allows plug-and-play for numerical subroutines
- FORTRAN, to be used from FORTRAN/C/C++
- Java BLAS available, too

**levels:**

Level 1: vector and vector-vector operations (norm, scalar product, vector addition, SAXPY, . . .)

Level 2: matrix-vector operations (rank-1-modifications, matrix-vector product, tridiagonal systems); vector processors

Level 3: matrix-matrix operations; parallel computers!
6. LAPACK

- popular collection of FORTRAN subroutines for standard problems from numerical linear algebra like linear systems, regression, eigenvalues, SVD, ...

- dense and band matrices (not general sparse ones)

- successor of EISPACK and LINPACK, tuning for modern microprocessors and supercomputer architectures (reduction of memory accesses, block operations, ...) 

- LAPACK routines use BLAS modules

- variants:
  - LAPACK90, CLAPACK, LAPACK++
  - ScaLAPACK (MIMD systems, scalability!)
7. Libraries and Collections

- Visual Numerics:
  - see http://www.vni.com
  - mathematical libraries
  - predecessor: IMSL (The Int’l Math. & Statist. Library)

- Diffpack:
  - offered by Numerical Objects, see http://www.nobjects.com
  - environment for the development of code for numerical simulation problems plus libraries of efficient routines
  - object-oriented concept, available for most UNIX platforms

- NAG (Numerical Algorithms Group):
  - see http://www.nag.co.uk/
  - non-profit software house, spin-off of Oxford University
  - FORTRAN/FORTRAN90/C/Parallel libraries;
  - AXIOM; IRIS Explorer (Visual.); Fastflo (CFD and more)
8. Libraries and Collections

• Numerical Recipes:
  – see http://www.nr.com
  – sophisticated algorithms and their implementations
  – available for FORTRAN 77, FORTRAN 90, C, Pascal, . . .
  – corresponding software is licenced and commercial
  – about 350 routines for topics like
    * solution of linear systems
    * interpolation and extrapolation
    * numerical quadrature
    * differentiation and approximation
    * roots and extrema
    * eigenvalues, differential equations, and more
9. Tools for Algorithm Development

- Libraries offer tested and efficient (w.r.t. both storage and runtime) standard modules for competitive simulation codes (do something classical cheap!)

- Another problem is algorithm development (develop something new and cheaper!):
  - design of algorithms
  - testing and rapid prototyping
  - analysis (convergence behaviour etc.)
  - not yet: production runs, memory or runtime optimization

- widespread solutions:
  - computer algebra programs like Maple or Mathematica
  - MATLAB
10. Maple

• by Waterloo Maple Inc., a spin-off of the University of Waterloo in Ontario (see http://www.maplesoft.com)

• originally a mere computer algebra program, today “interactive environment for mathematical problem solving and programming”

• focus:
  – symbolic computations, formula manipulation
  – numerical computations with arbitrary accuracy
  – 2D and 3D graphical output
  – straightforward programming for algorithm development

• structure:
  kernel + main library + mixed library + packages
11. MATLAB

• by The MathWorks (see http://www.mathworks.com/)
• originally: primarily for use in (maths) education
• today: “high-performance numerical computation and visualization software”, standard tool for scientific computing research groups:
  – development, prototyping, programming
  – computations
  – visualization
• singular success story: >500 employees, >100 countries, >2000 universities and research institutes
• structure: basic program plus a collection of specialized tool boxes