POLITECNICO DI MILANO

BzzMath: an Object Oriented Numerical Project

Davide Manca and Guido Buzzi-Ferraris

Dipartimento di Chimica Materiali e Ingegneria Chimica
Politecnico di Milano – ITALY

The BzzMath project

- **BzzMath** is a numerical library developed from scratch in C++.
- Main C++ features:
  - Masking;
  - Encapsulation;
  - Polymorphism;
  - Inheritance.
- One-man-development-team, about 16 years of coding.
- At present **BzzMath** comprises:
  - Single and double precision arithmetic;
  - 234 classes;
  - 171 source files and 143 example files + tutorial + on-line user manual;
  - 2 libraries (Basic, Advanced);
  - .cpp and .hpp files sum up to 13.7 MB with 442,932 lines of code.

BzzMath structure

BzzMathBasic

» **Utilities**
- o Printing and Messages
- o Quadratic Equation
- o Matrix Coefficients Existence
- o Matrix Symmetric Coeff. Existence
- o Matrices Existence
- o Statistical Functions

» **Interpolation**

» **Cubic Interpolation**
- o Hermite
- o Smooth
- o Spline

» **BiCubic Interpolation**
- o Hermite
- o Smooth
- o Spline

» **Complex Algebra**

» **Integer Algebra**
- o Vectors & Matrices

» **Linear Algebra**
- o Vectors
- o Matrices
- o Diagonal Matrices
- o Band Matrices
- o Left Matrices
- o Right Matrices
- o Sparse Matrices
- o Diagonal Block Matrices
- o Tridiagonal Block Matrices
- o Block Matrices
- o Rectangle Band Matrices
- o Symmetric Matrices
- o Band Symmetric Matrices
- o Sparse Symmetric Matrices

» **Eigenvalues**
- o Symmetric Matrices

» **Linear Systems**
- o Factored Gauss
- o Factored Band Gauss
- o Factored Tridiagonal Gauss
- o Factored Sparse Gauss
- o Factored Diagonal Block Gauss
- o Factored Tridiagonal Block Gauss
- o Factored LQ
- o Factored Sparse LQ
- o Factored QR
- o Factored Sparse QR
- o Factored SVD

BzzMathAdvanced

» **Utilities**
- o Save & Load
- o BzzPlot.exe & BzzPlotSparse.exe

» **Numerical Differentiation**

» **Root finding**

» **Non Linear Systems**

» **Mono & Multidimensional Minimization**

» **Definite Integrals**
- o Integral
- o Integral Gauss

» **ODE**
- o Multi Value Methods Ode Non Stiff
- o Runge-Kutta Methods Ode Non Stiff
- o Multi Value Methods Ode Stiff
- o Multi Value Methods Ode Stiff Sparse

» **DAE**
- o Dae
- o Dae Sparse

» **Regressions**
- o BzzLinearRegression.exe
- o Linear Regression
- o Linear Regression Experiments Search
- o Non Linear Regression

Mixed language

- It is possible to seamlessly mix C++ and Fortran code into one executable:
  - Fortran main calls C++ source code and C++ libraries;
  - Fortran main calls C++ libraries;
  - C++ main calls Fortran subroutines and functions;
  - C++ main calls Fortran obj–files or libraries;
- Fortran 77 or higher source code required;
- Only a Fortran compiler may be sufficient when adding C++ libraries.

EXAMPLE: call the `BzzMath` ODE solver from a Fortran program:

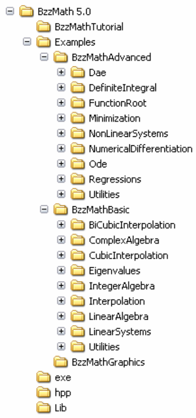
```
call BzzOde(iState,nEq,tIn,tOut,y)
```

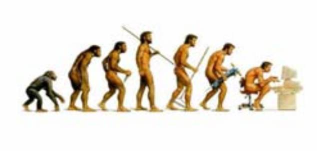
Or with more options (lower and upper bounds on the integration variables, absolute and relative tolerances, critical integration points):

```
call BzzOde(iState,nEq,tIn,tOut,y,iMinMax,yMin,yMax,
            iTol,aTol,rTol,iCrit,tCrit)
```

BzzMath availability

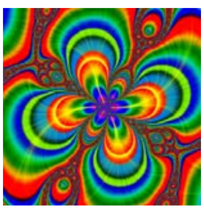
- `BzzMath` is available on the Internet:
  - www.chem.polimi.it/homes/gbuzzi/
- `BzzMath` is free for non-commercial use.
- The user can download the last version of `BzzMath 5.0`
- A zipped file of about 20 MB comprises all the libraries, examples, tutorials and on-line manuals.

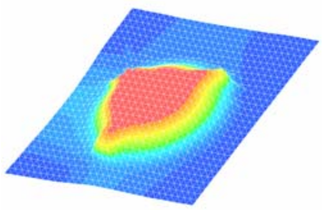


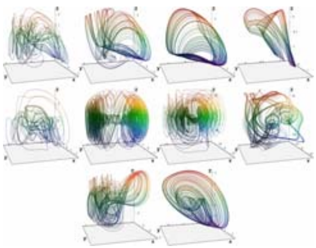


Applied numerical examples

- Some of the `BzzMath` features will be presented through applied numerical examples:
  - Robust root finding;
  - Robust minimization;
  - Robust multidimensional minimization;
  - ODE and DAE solvers;
  - Customized solver for nonlinear algebraic equations.







Robust root finding

- **PROBLEM**: Find the roots of a nonlinear algebraic equation $y(x)=0$

- **CONDITIONS**: If the numerical problem is characterized by one or more of the following features:
  - We do not know the two points x_A, x_B where $y(x_A)\,y(x_B)<0$
  - The problem has several roots;
  - The function is not defined in some intervals or points;
  - The function does not change sign within the user-defined interval.

- **SOLUTION**

  `BzzFunctionRootDoubleRobust` is the suggested class.

Robust root finding

- **EXAMPLE**: Find the roots of the following function

$$y=\left|\log\left[720+x\left(-720+(x-1)\left(360+(x-2)\left(-120+(x-3)\left(30+(x-4)(-11+x)\right)\right)\right)\right)\right]-1\right|$$

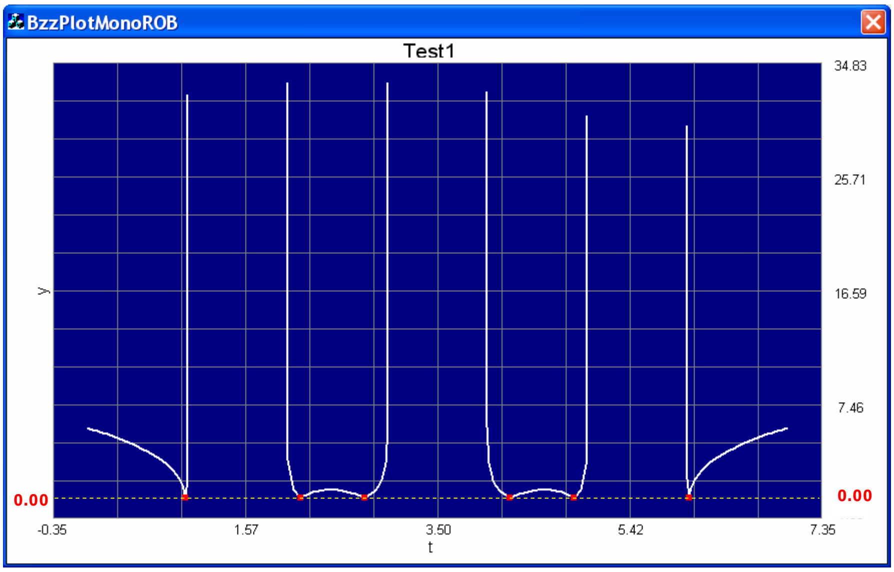
within the user-defined interval $x_{low}=0 \qquad x_{up}=7$

- **REMARK**: the function is **not defined** if

$$\left[720+x\left(-720+(x-1)\left(360+(x-2)\left(-120+(x-3)\left(30+(x-4)(-11+x)\right)\right)\right)\right)\right]\le 0$$

- **SOLUTION**: the function has **six roots** in the user-defined interval.

Robust root finding



Robust monodim. minimization

- **PROBLEM**: minimization of a monodimensional function $\underset{x}{\text{Min}} \quad F(x)$

- **CONDITIONS**: If the numerical problem is characterized by one or more of the following features:
  - We do not know the/an interval where the function is unimodal;
  - The problem has several minima;
  - The function is not defined in some intervals or points;
  - The function and/or its derivatives are discontinuous in some intervals or points;
  - The function is not parabolic near the minimum.

- **SOLUTION**

  `BzzMinimizationDoubleRobust` is the suggested class.

Robust monodim. minimization

- **EXAMPLE**: Find the minima of the following function

$$F = 10|x-6| + \sqrt{\left(-945 + x\left(1689 + x\left(-950 + x\left(230 + x\left(-25 + x\right)\right)\right)\right)\right) + \exp(-x)}$$

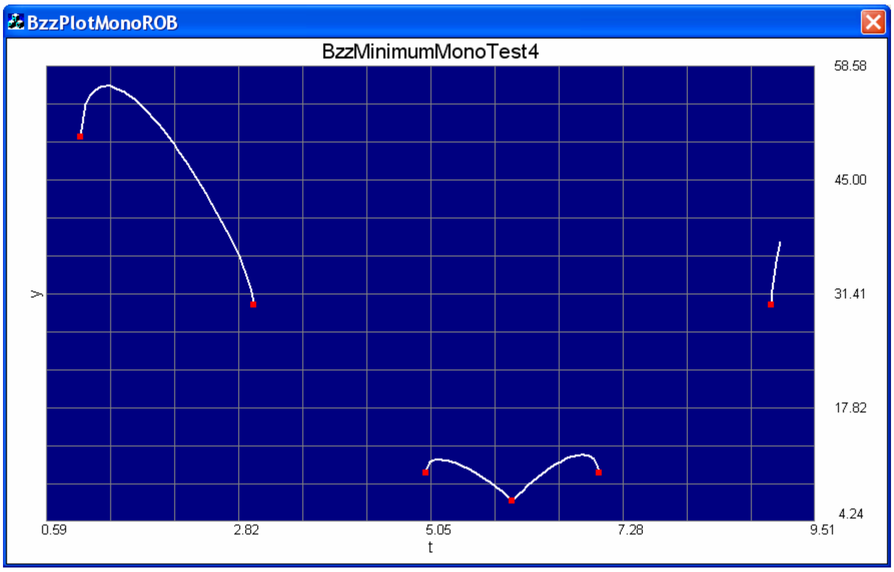
within the user-defined interval $x_{low} = 0 \quad x_{up} = 9.1$

- **REMARK**: the function is **not defined** if

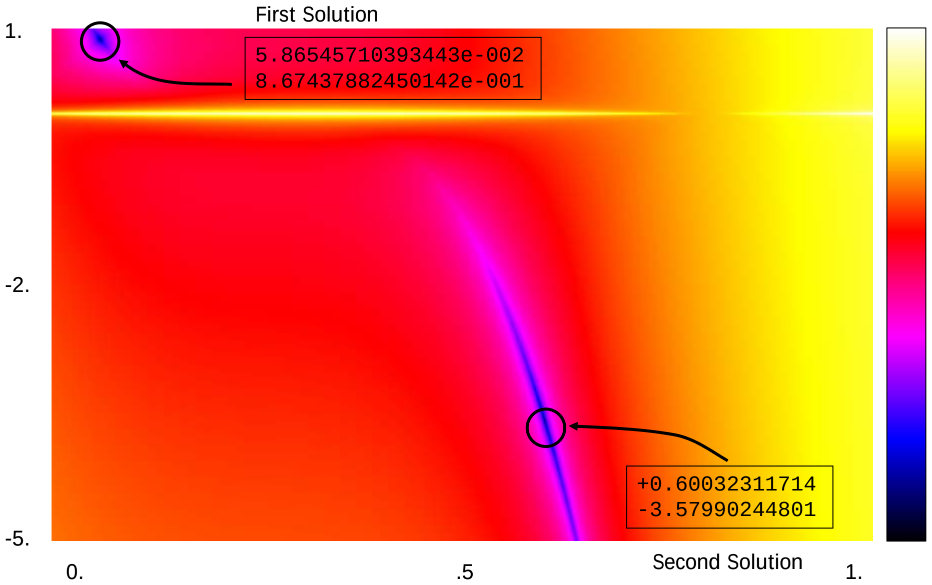
$$\left[\left(-945 + x\left(1689 + x\left(-950 + x\left(230 + x\left(-25 + x\right)\right)\right)\right)\right) + \exp(-x)\right] < 0$$

- **REMARK**: the first derivative is discontinuous in $x = 6$
- **SOLUTION**: the function has **six minima** in the user-defined interval.

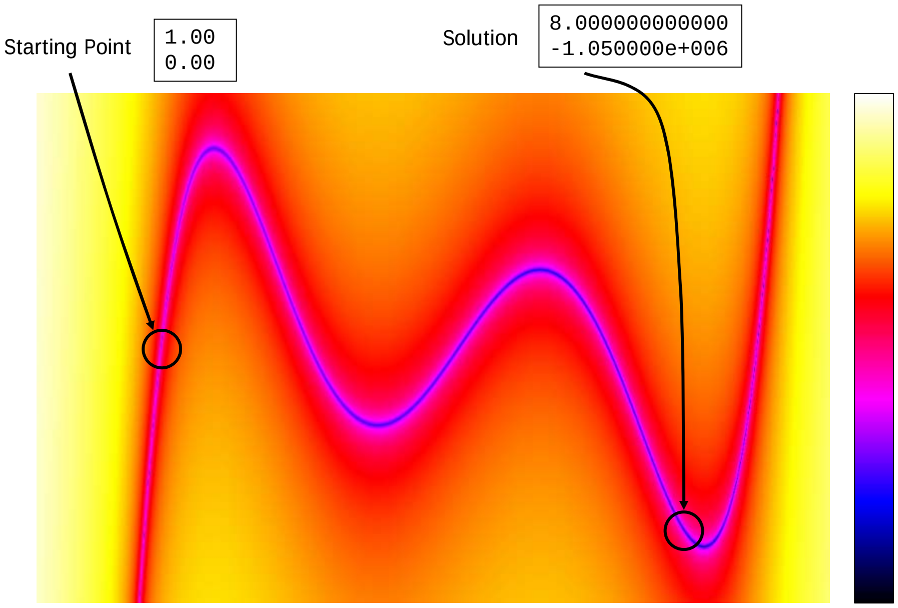
Robust monodim. minimization



Robust multidim. minimization



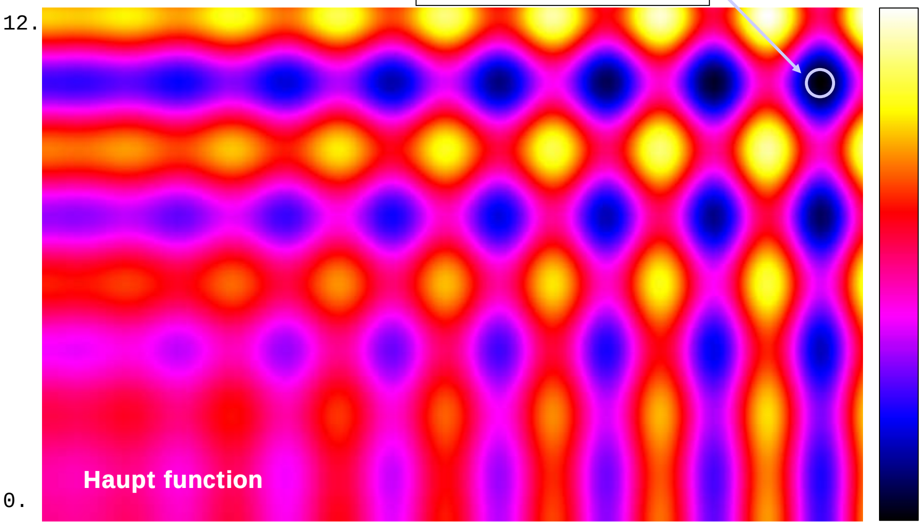
Robust multidim. minimization



Robust multidim. minimization

Global Minimum
-2.26356474410454e+001

1.13937579453077e+001
1.02345837083403e+001



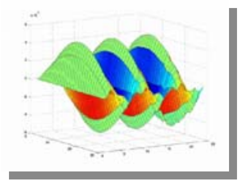
ODE and DAE systems

- **PROBLEM**: Integration of stiff/non stiff **ODE** systems with initial conditions

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, t) \qquad \mathbf{y}(t_0) = \mathbf{y}_0$$

Integration of **DAE** systems with initial conditions

$$\begin{cases} \mathbf{y}' = \mathbf{f}(\mathbf{y}, t) & \mathbf{y}(t_0) = \mathbf{y}_0 \\ \mathbf{g}(\mathbf{y}, t) = \mathbf{0} \end{cases}$$



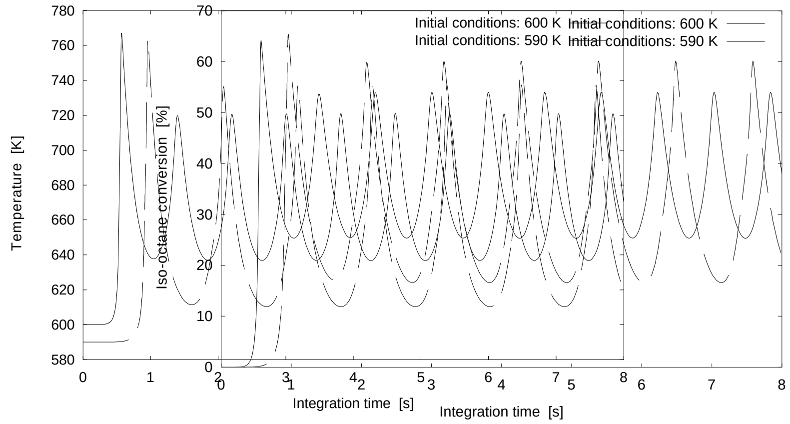
- **FEATURES**:
  - Robustness and efficiency;
  - Highly oscillating problems;
  - Problems with large discontinuities in the derivatives;
  - Lower/upper bounds for the integration variables;
  - Structured and non-structured Jacobians (banded, sparse, block, tridiagonal, …
- **SOLUTION**: **BzzOde** and **BzzDae** are the suggested classes.

ODE and DAE systems

Homogeneous combustion of iso-octane

Low temperature oxidation mechanisms and **oscillating cool flames** observed in a **Jet Stirred Reactor**

200 Ordinary differential equations



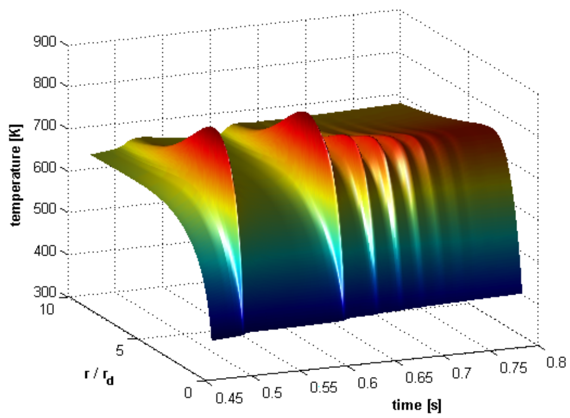
ODE and DAE systems

Evaporation and combustion of a fuel droplet

Kinetics: **200** species and over **5,000** reactions

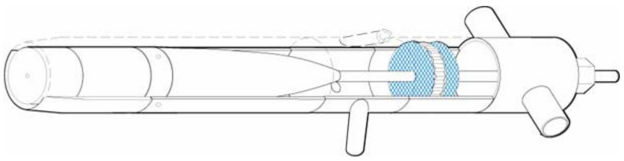
Between **25,000** and **65,000** Differential and Algebraic Equations

About **2 days** of CPU time on a Pentium IV @ 2.8 GHz



Large nonlinear algebraic systems

- **PROBLEM**: Evaluation of the macro and micropollutant concentrations produced by a nonpremixed burner.
  - Case-study: Sidney University and Sandia National Laboratories;
  - Nonpremixed Flame: CH_4/H_2 are premixed 50%-50% + air (lean).
- **TEMPERATURE PROFILE**: determined by Fluent 6.1
- **POSTPROCESSING AND DETAILED KINETICS**: `ReaNet` is a custom C++ program based on `BzzMath` Libraries.



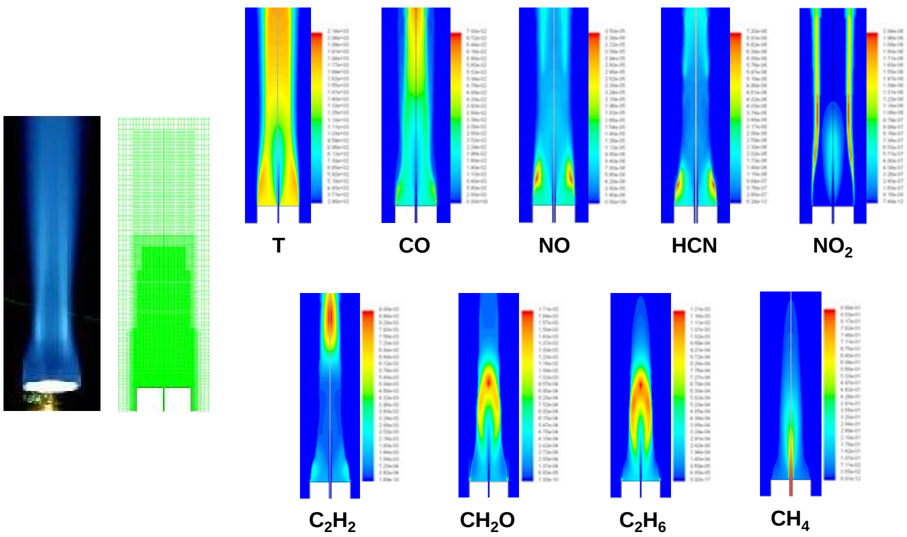
Large nonlinear algebraic systems

- **OPERATING SEQUENCE**
  - **FLUENT INPUT**: axial-symmetric geometry, non-structured grid with 34632 cells, very simplified kinetic scheme: **5 species** (CH_4, O_2, CO, CO_2, N_2) and **2-3 reactions**.
  - **FLUENT OUTPUT**: temperature, velocity and macro-species profiles.
  - **REANET INPUT**:
    - Assigned temperature and velocity profiles;
    - First guess macro-species profiles evaluated by Fluent.;
    - Detailed kinetic scheme: **105 species** and **1534 elementary reactions**.
  - **REANET OUTPUT**:
    - Detailed concentration profiles of macro and micro-pollutants;
    - Outlet concentrations of key components: NO_x, CO, HCN, C_2H_2, CH_2O, C_2H_6, ...

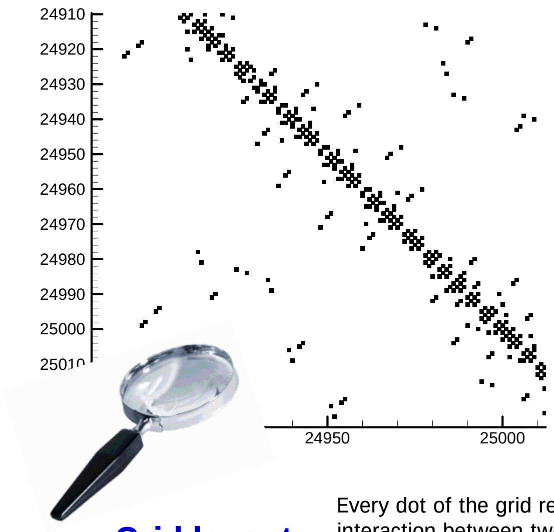
Large nonlinear algebraic systems

- **SOLUTION**
  - **NUMERICAL MODELING**: the problem is described by a large system of nonlinear equations.
  - **EQUATIONS COUNT**:
    - **34,632** cells * **105** species → **3,636,360** nonlinear equations.
  - **NUMERICAL ALGORITHM**:
    - Local Newton method (on each reactor)
      - Continuation method through integration of an ODE system;
    - Iterative substitutions;
    - Global Newton method parametric continuation
      - Parametric continuation on the whole reactors network;
  - **BZZMATH CLASS**: `BzzCSTRNetwork` derived from `BzzNLS`.
  - **CPU TIME**: 20 hours of CPU on a Pentium IV @ 2.8 GHz with 2 GB di RAM.

Large nonlinear algebraic systems



Large nonlinear algebraic systems



Grid layout

Every dot of the grid represents a physical interaction between two adjacent cells. Every dot is a 105 × 105 matrix).