A new family of methods for global error control in ODE solvers

Anders Logg

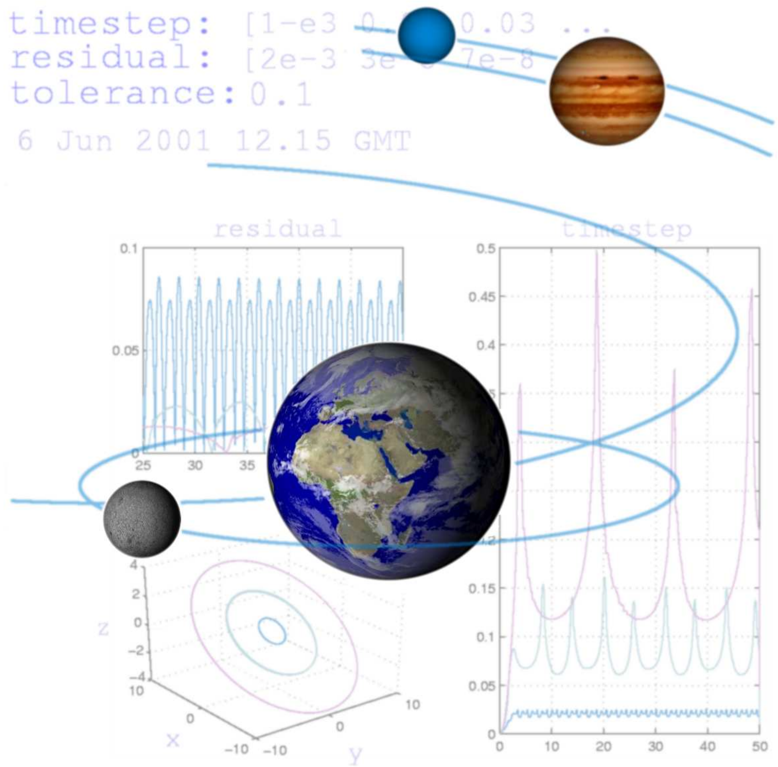
`logg@tti-c.org`

Toyota Technological Institute at Chicago

Motivation I: ordinary differential equations

A mechanical system with multiple time-scales:
The Solar System

- Moon: $T = 1/12$
- Earth: $T = 1$
- Pluto: $T = 250$
- Multiple time-scales
- Individual time steps

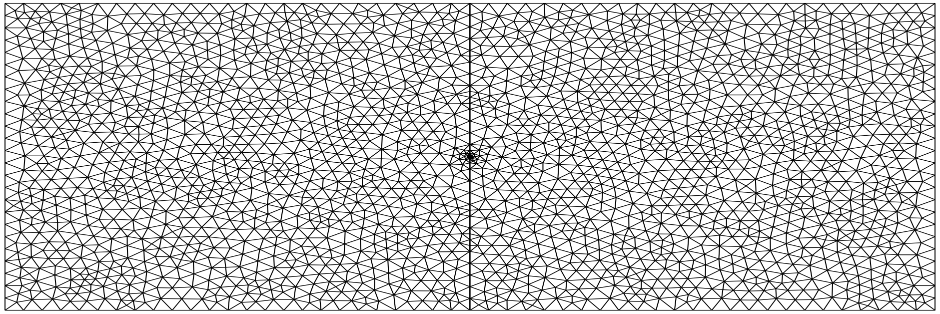


A simple system with multiple time scales

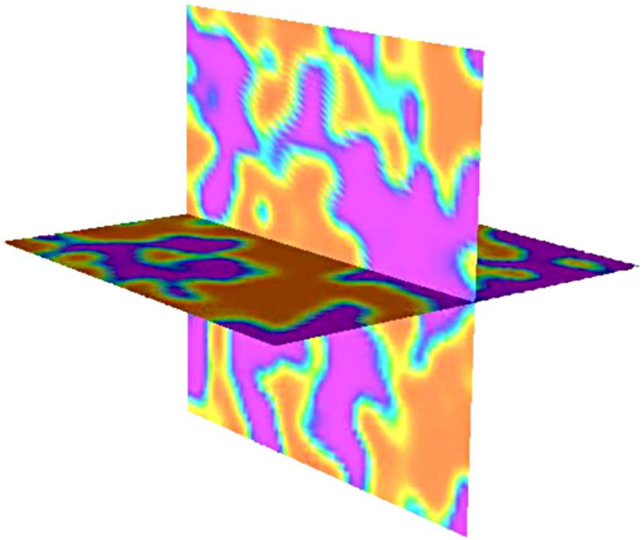
Animation contributed by Johan Jansson

Motivation II: partial differential equations

- Geometry (local refinement)



- Equation (local structures)



Outline

- Basic ideas
- Galerkin formulation
- Error estimates and adaptivity
- Implementation
- Examples and benchmarks
- Current status and future plans

Basic ideas

Objective

Solve the ODE initial value problem

$$\begin{cases} \dot{u}(t) = f(u(t),t), & t \in (0,T], \\ u(0) = u_0, \end{cases}$$

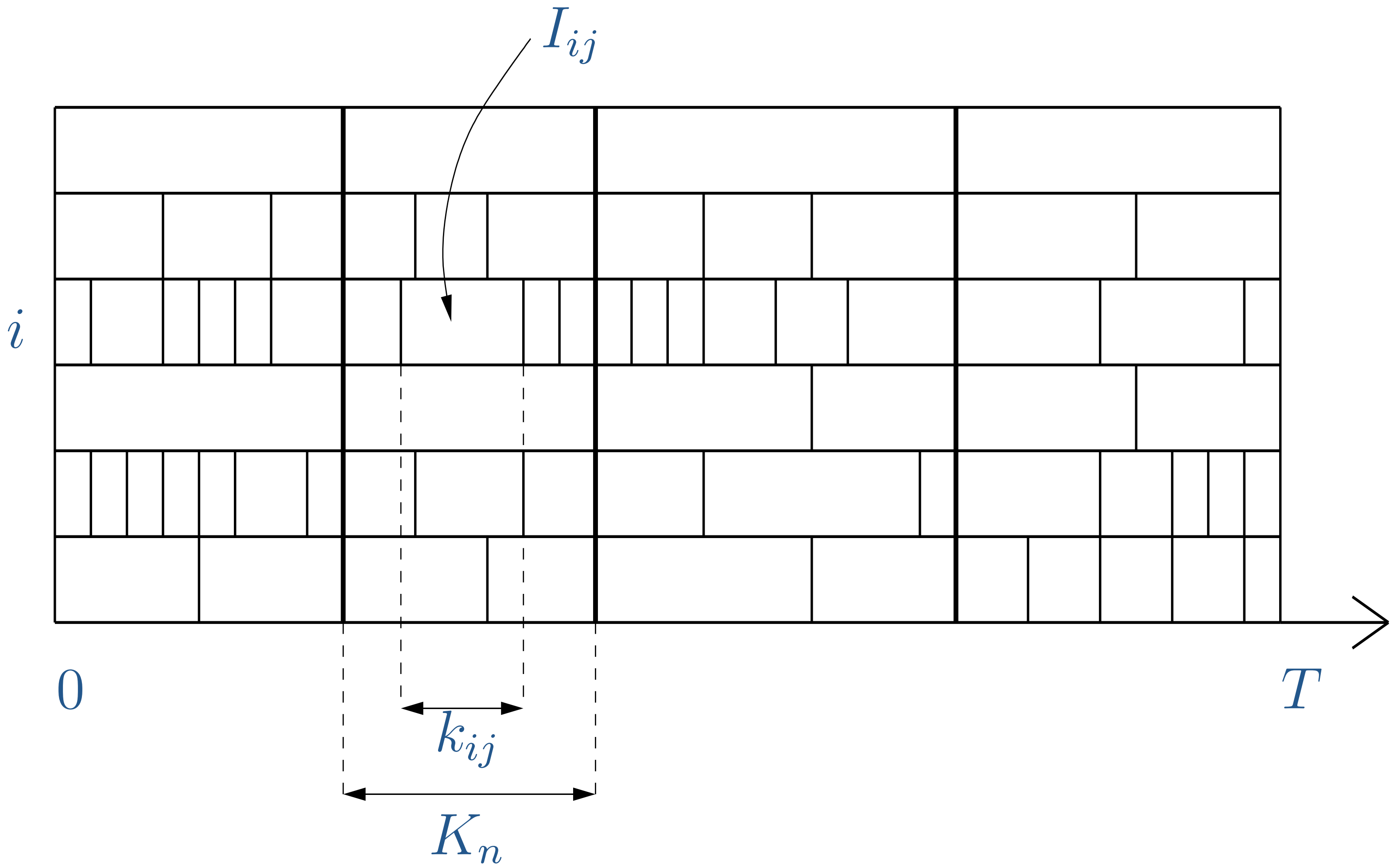
for $u : [0,T] \rightarrow \mathbb{R}^N$ with adaptive and individual time steps for the different components $u_i(t)$.

The individual time steps are chosen adaptively based on an a posteriori error estimate of the global error at time $t = T$.

Key features

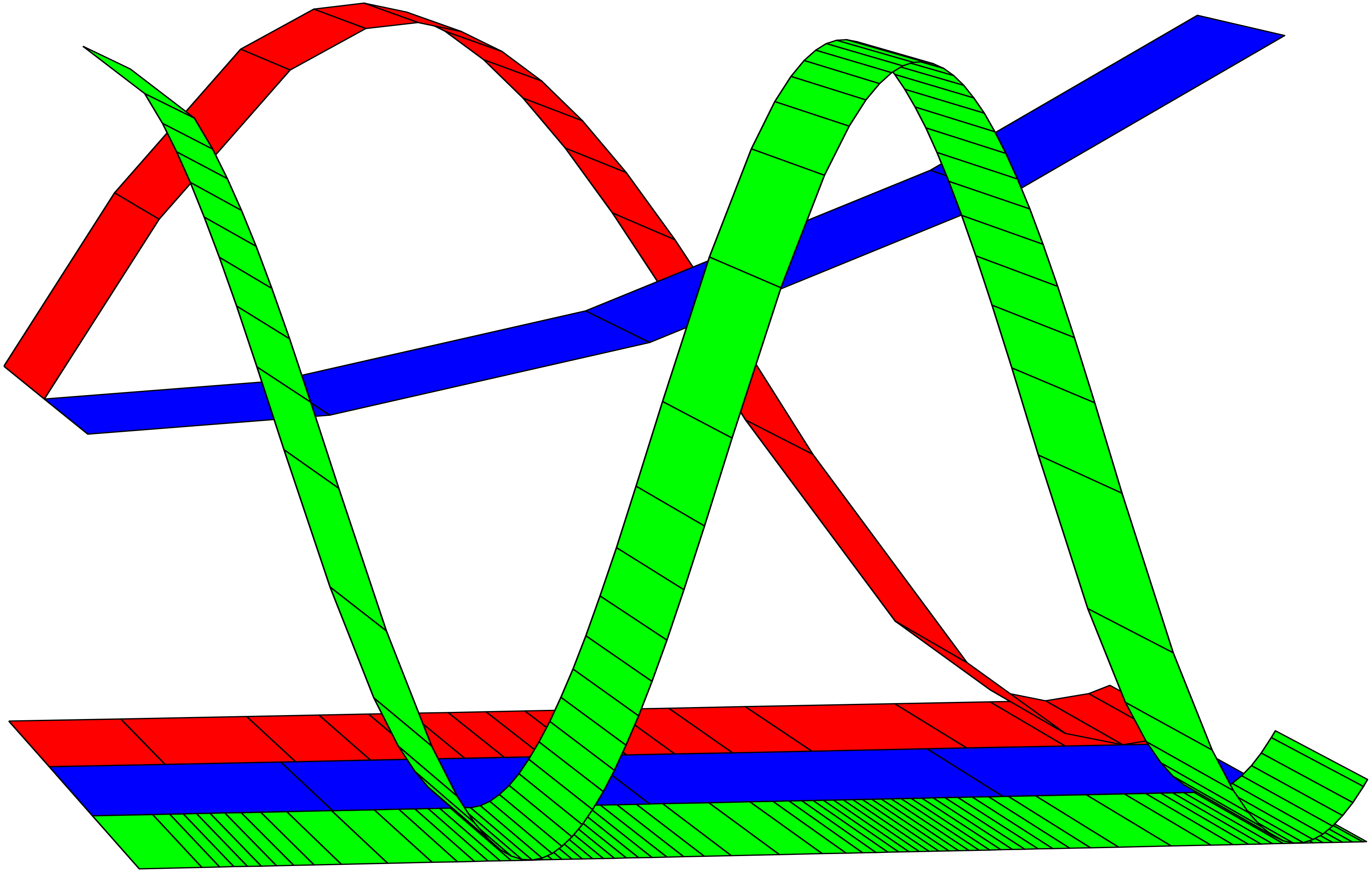
- Adaptive individual time steps
- Efficient and reliable control of the global error
- Solution of dual problems, computation of stability factors
- Efficient adaptive iterative methods
- General implementation of arbitrary order $\mathrm{mcG}(q)$ and $\mathrm{mdG}(q)$ within **DOLFIN**

Individual time steps



- Multi-adaptive solution of the bistable equation
- Multi-adaptive time steps for the bistable equation

Individual piecewise polynomials



Galerkin formulation

Standard Galerkin

Standard Galerkin, $\mathrm{cG}(q)$:

$$\int_0^T (\dot{U}, v)\,\mathrm{dt} = \int_0^T (f(U,\cdot), v)\,\mathrm{dt} \quad \forall v \in \hat{V},$$

with $U \in V$, $U(0) = u_0$, and trial and test spaces given by

$$\begin{aligned} V &= \{v \in [\mathcal{C}([0,T])]^N : v_i|_{I_j} \in \mathcal{P}^q(I_j)\}, \\ \hat{V} &= \{v : v_i|_{I_j} \in \mathcal{P}^{q-1}(I_j)\}. \end{aligned}$$

- Same time steps for all components U_i of U

Multi-adaptive Galerkin

Multi-adaptive Galerkin, $\mathrm{mcG}(q)$:

$$\int_0^T (\dot{U}, v)\,\mathrm{dt} = \int_0^T (f(U,\cdot), v)\,\mathrm{dt} \quad \forall v \in \hat{V},$$

with $U \in V$, $U(0) = u_0$, and trial and test spaces given by

$$\begin{aligned} V &= \{v \in [\mathcal{C}([0,T])]^N : v_i|_{I_{ij}} \in \mathcal{P}^{q_{ij}}(I_{ij})\}, \\ \hat{V} &= \{v : v_i|_{I_{ij}} \in \mathcal{P}^{q_{ij}-1}(I_{ij})\}. \end{aligned}$$

- Individual time steps for all components U_i of U
- Includes the standard $\mathrm{cG}(q)$ method
- Similar extension of the $\mathrm{dG}(q)$ method to $\mathrm{mdG}(q)$

The discrete equations for $\mathrm{mcG}(q)$

With the following Ansatz for U_i on I_{ij},

$$U_i(t) = \sum_{n=0}^{q_{ij}} \xi_{ijn} \lambda_n^{[q_{ij}]}(\tau_{ij}(t)),$$

we obtain

$$\xi_{ijn} = \xi_0 + \int_{I_{ij}} w_n^{[q_{ij}]}(\tau_{ij}(t)) \ f_i(U,t) \, \mathrm{dt},$$

for certain weight functions $\{w_n^{[q]}\} \subset \mathcal{P}^{q-1}(0,1)$.

The discrete equations for $\mathrm{mdG}(q)$

Similarly for the multi-adaptive discontinuous Galerkin method, $\mathrm{mdG}(q)$, we obtain

$$\xi_{ijn} = \xi_{ij0}^{-} + \int_{I_{ij}} w_n^{[q_{ij}]}(\tau_{ij}(t))\; f_i(U,t)\,\mathrm{d}t,$$

for certain weight functions $\{w_n^{[q]}\} \subset \mathcal{P}^q(0,1)$.

Properties of $\mathrm{mcG}(q)$ and $\mathrm{mdG}(q)$

- $\mathrm{mcG}(q)$ conserves energy if $k_{U_i} = k_{V_i}$
- $\mathrm{mdG}(q)$ is B-stable: if f is monotone,

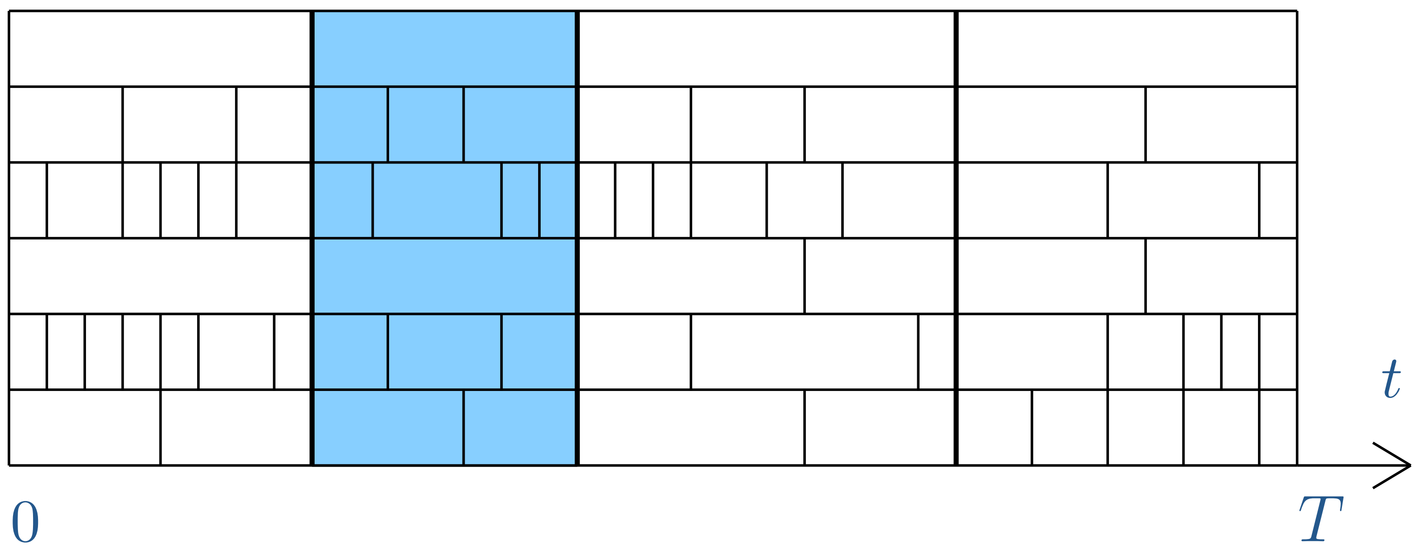
$$(f(u,\cdot) - f(v,\cdot), u - v) \leq 0 \quad \forall u, v \in \mathbb{R}^N,$$

then

$$\|U(\bar{t}^-) - V(\bar{t}^-)\| \leq \|U(0^-) - V(0^-)\|$$

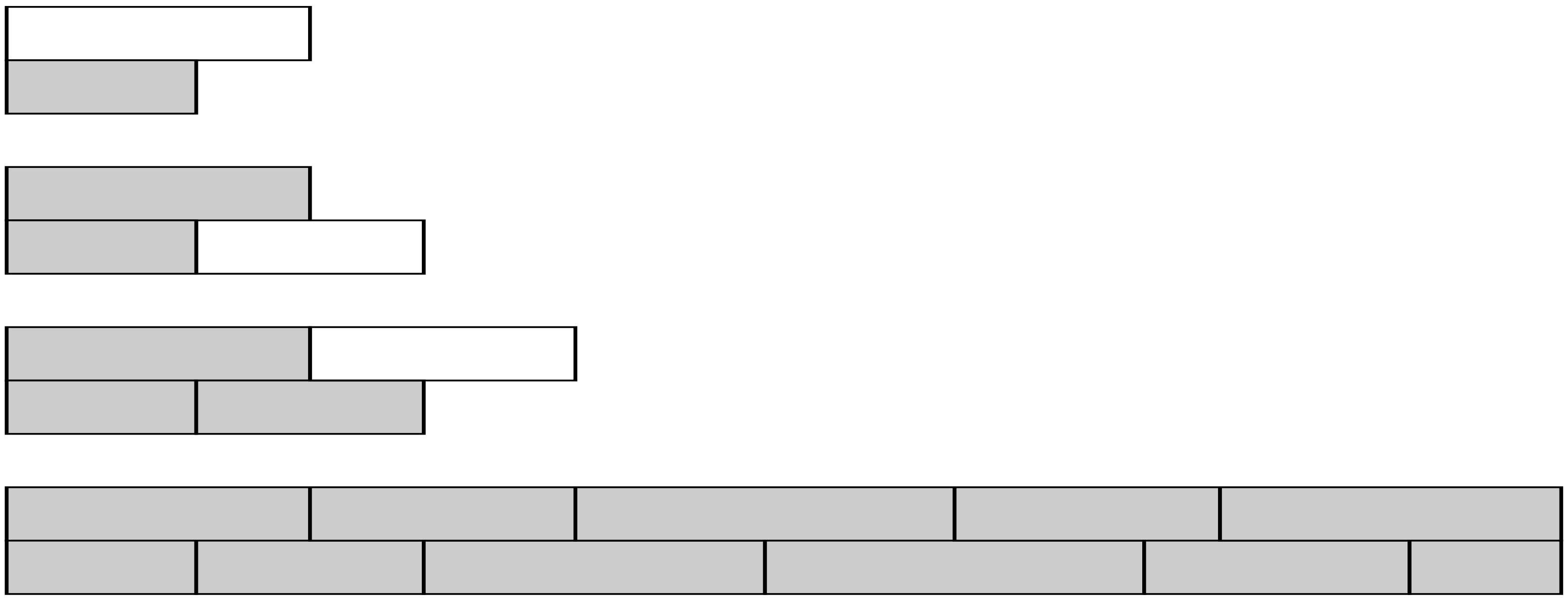
Iterative method

- Arrange elements in time slabs
- Adaptive fixed-point iteration on time slabs
- Control the computational error



Basic strategy

The last component steps first



- Recursive generation of time slabs
- Adaptive iteration on time slabs

Error estimates and adaptivity

A priori error estimates

- The order of $\mathrm{mcG}(q)$ is $2q$ (locally $2q_{ij}$):

$$\|e(T)\| \leq CS(T)\|k^{2q}u^{(2q)}\|_{L_\infty([0,T],l_1)}.$$

- The order of $\mathrm{mdG}(q)$ is $2q+1$ (locally $2q_{ij}+1$):

$$\|e(T)\| \leq CS(T)\|k^{2q+1}u^{(2q+1)}\|_{L_\infty([0,T],l_1)}.$$

$S(T)$ is a stability factor obtained from the discrete dual problem.

A posteriori error estimates

The global error at final time is controlled using an a posteriori error estimate of the form

$$|L_{\psi,g}(e)| \leq E_G + E_C + E_Q,$$

where $L_{\psi,g}(e) \equiv (e(T),\psi) + \int_0^T (e,g)\,\mathrm{dt}$ is a functional of the error $e = U - u$, and

- E_G: Galerkin error
- E_C: Computational error
- E_Q: Quadrature error

The Galerkin error: E_G

- Residual:

$$R_i(U,t) = \dot{U}_i(t) - f_i(U(t),t)$$

- Stability factor:

$$S_i^{[q]}(T) = \int_0^T |\phi_i^{(q)}| \,\mathrm{dt}$$

- Error estimate (for $\mathrm{mcG}(q)$):

$$E_G = \left| \int_0^T (R,\phi)\,\mathrm{dt} \right| = \left| \sum_{i=1}^{N} \sum_{j=1}^{M_i} \int_{I_{ij}} R_i(\phi_i - \pi_k \phi_i)\,\mathrm{dt} \right|$$

$$\leq \sum_{i=1}^{N} C S_i^{[q]}(T) \max_{[0,T]} \{ k_i^{q_i} |R_i(U)| \}$$

The Computational Error: E_C

- Computational residual:

$$R_i^C(U,t) = \frac{1}{k_{ij}} \left[U(t_{ij}) - U(t_{i,j-1}) - \int_{I_{ij}} f_i(U,\cdot)\,\mathrm{dt} \right], \quad t \in I_{ij}$$

- Stability factor:

$$S_i^{[0]}(T) = \int_0^T |\phi_i| \,\mathrm{dt}$$

- Error estimate:

$$E_C \approx \sum_{i=1}^{N} S_i^{[0]} \max_{[0,T]} |R_i^C|$$

The Quadrature Error: E_Q

- Quadrature residual:

$$R_i^Q = \frac{1}{k_{ij}} \left[\tilde{\int}_{I_{ij}} f_i(U,\cdot)\,\mathrm{dt} - \int_{I_{ij}} f_i(U,\cdot)\,\mathrm{dt} \right], \quad t \in I_{ij}$$

- Stability factor:

$$S_i^{[0]}(T) = \int_0^T |\phi_i|\,\mathrm{dt}$$

- Error estimate:

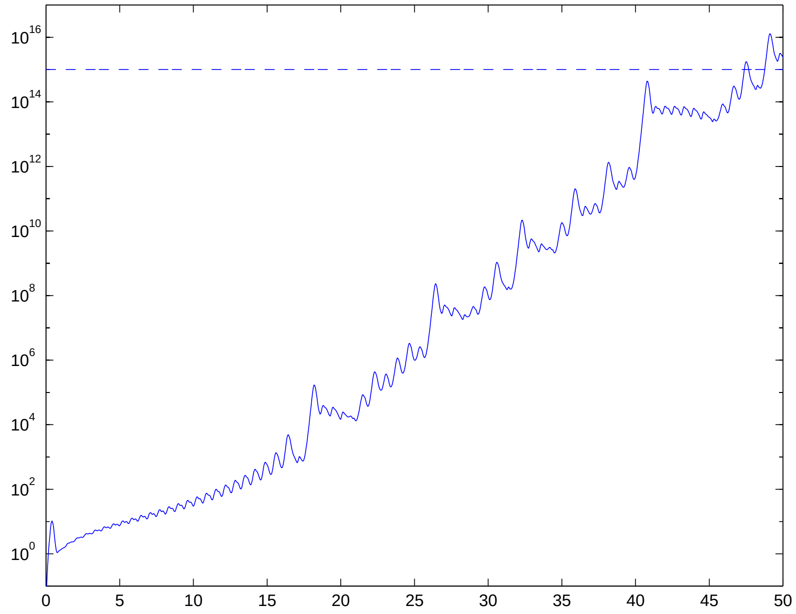
$$E_Q \approx \sum_{i=1}^{N} S_i^{[0]} \max_{[0,T]} |R_i^Q|$$

Computational cost (complexity of output)

- Computational cost given by the product $S(T)\|u^{(p)}\|$
- Determined both by the stability/sensitivity of the model and the regularity of the solution

Quantitative classification according to stability:

- Parabolic: $S(T) \sim 1$
- Hyperbolic: $S(T) \sim T$
- Exponential: $S(T) \sim \exp(T)$



The dual problem

The dual problem is given by

$$\begin{cases} -\dot{\phi}(t) &= J^\top(u, U, t)\phi(t) + g(t), \quad t \in [0, T), \\ \phi(T) &= \psi, \end{cases}$$

where

$$J(v_1, v_2, \cdot) = \int_0^1 \frac{\partial f}{\partial u}(s v_1 + (1 - s) v_2, \cdot)\, ds.$$

By choosing ψ and g, different functionals $L_{\psi,g}(e)$ can be estimated. Basic examples:

- $\psi \approx e(T)/\|e(T)\|$ and $g = 0$ gives $L_{\psi,g}(e) \approx \|e(T)\|$
- $\psi = (0, \ldots, 0, 1, 0, \ldots, 0)$ and $g = 0$ gives $L_{\psi,g}(e) = e_i(T)$
- $\psi = 0$ and $g = (1, \ldots, 1)/(NT)$ gives $L_{\psi,g}(e) = \bar{e}$

The adaptive algorithm

1. Solve the primal problem with $S_i(T) = 1$ and

$$k_{ij} = \left(\frac{\mathrm{TOL}}{C N S_i(T) \|R_i\|_{I_{ij}}} \right)^{1/q_i}$$

2. Solve the dual problem
3. Compute new stability factors $S_i(T)$
4. Compute the error estimate E
5. If $E \leq \mathrm{TOL}$ then stop, otherwise go back to 1

Implementation

Implementation

- Implemented as a C++ library (part of **DOLFIN**)
- Mono-adaptive or multi-adaptive
- Newton or fixed-point
- User implements interface specified by ODE base class:

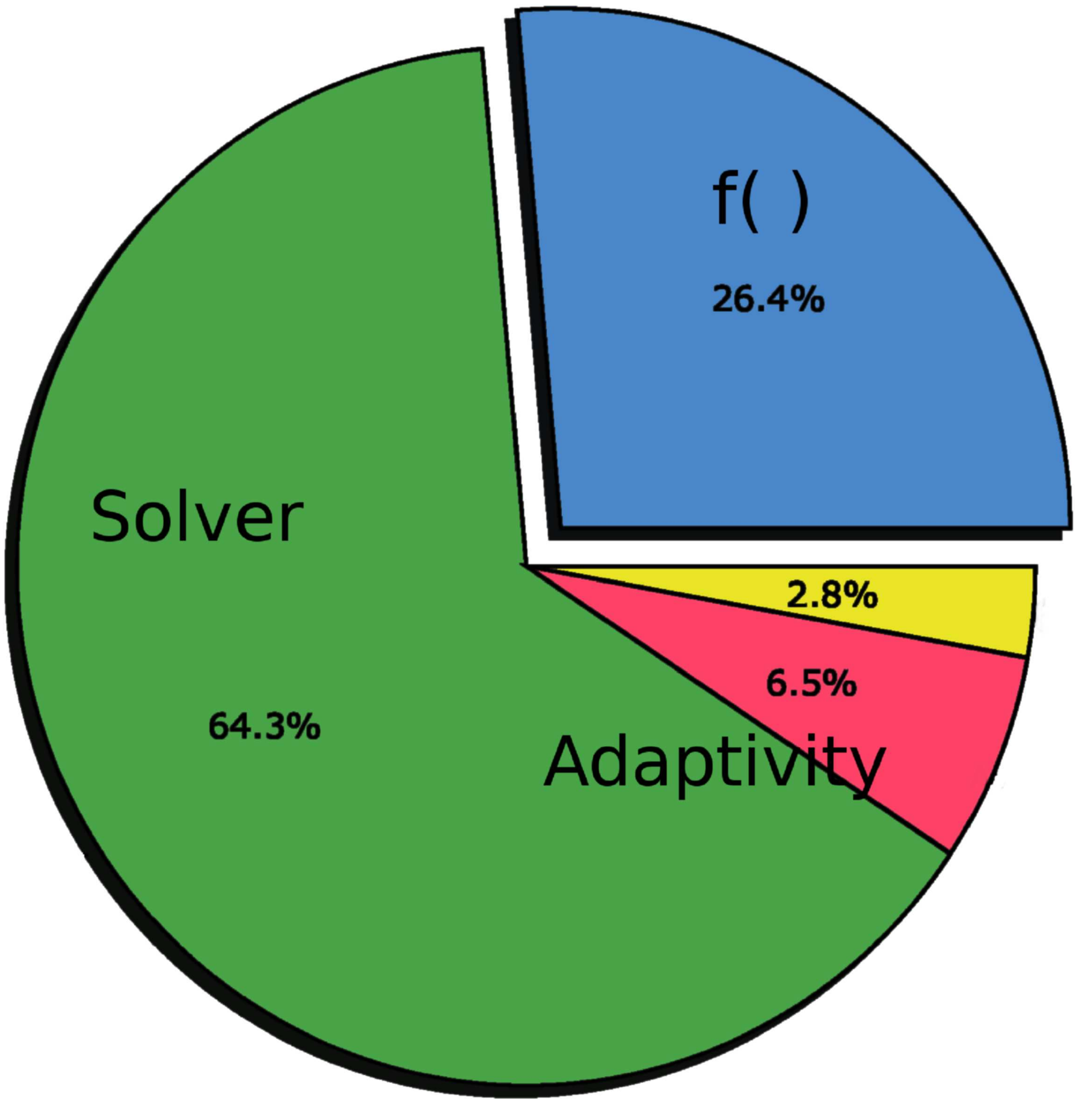
```
class ODE
{
public:

  ODE(uint N);

  virtual real f(const real u[], real t, uint i);
  virtual void f(const real u[], real t, real y[]);
  ...
```

Implementation

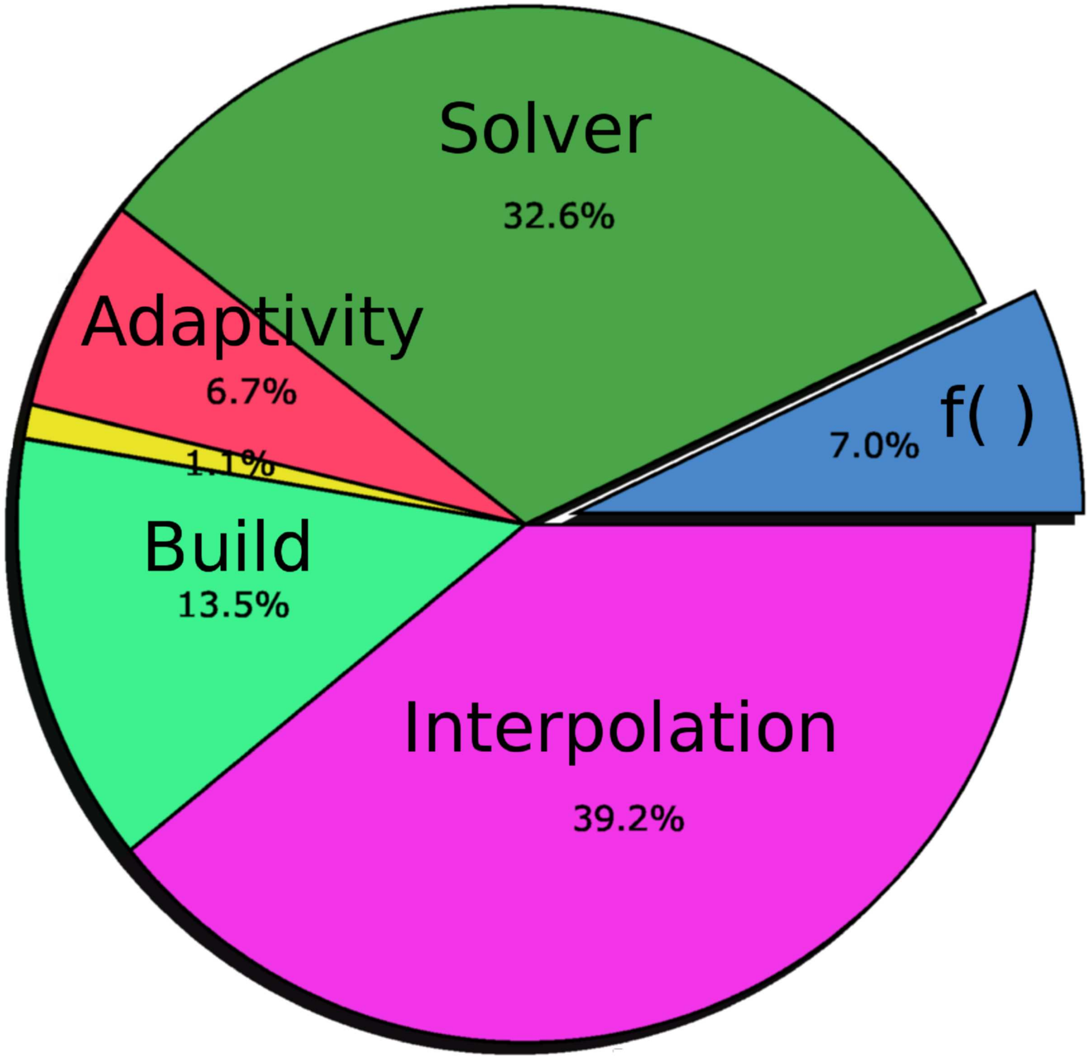
- Data stored in a "minimal" set of C arrays
- Build time slab: $\mathcal{O}(\#\text{ elements})$
- Interpolate $U_i(t)$: $\mathcal{O}(1)$

```
real* sa; // s --> start time t of sub slab s
real* sb; // s --> end time t of sub slab s
uint* ei; // e --> component index i of element e
uint* es; // e --> time slab s containing element e
uint* ee; // e --> previous element e of element e
uint* ed; // e --> first dependency d of element e
real* jx; // j --> value of dof j
int*  de; // d --> element e of dependency d
```

Mono-adaptive profi le (cG(1))



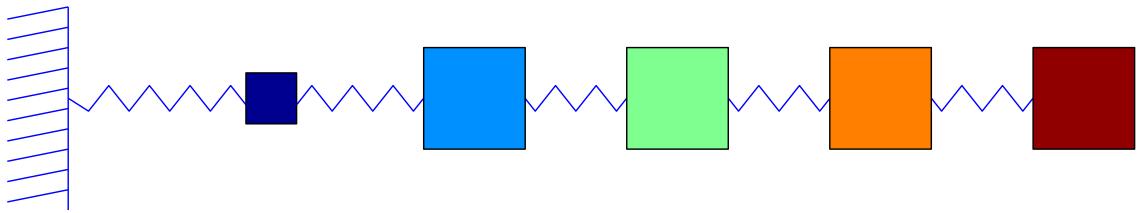
Multi-adaptive profi le ($\mathrm{mcG}(1)$)

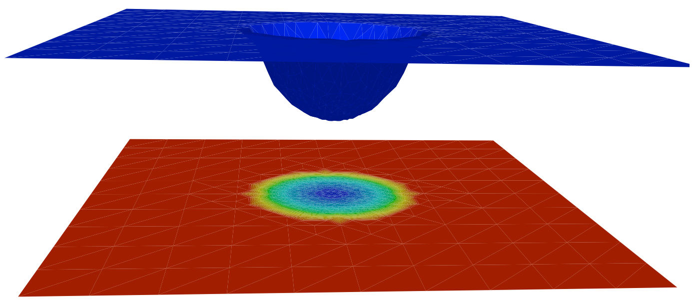


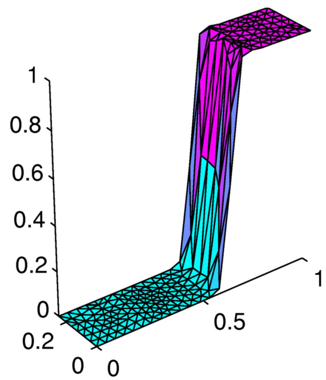
Examples and benchmarks

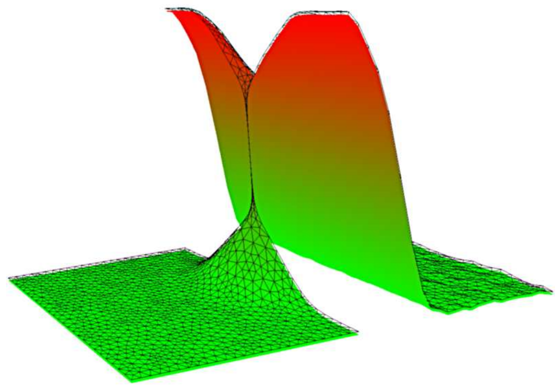
Examples

- A mechanical multi-scale system
- The heat equation
- A system of reaction–diffusion equations
- Wave propagation through a narrow slit



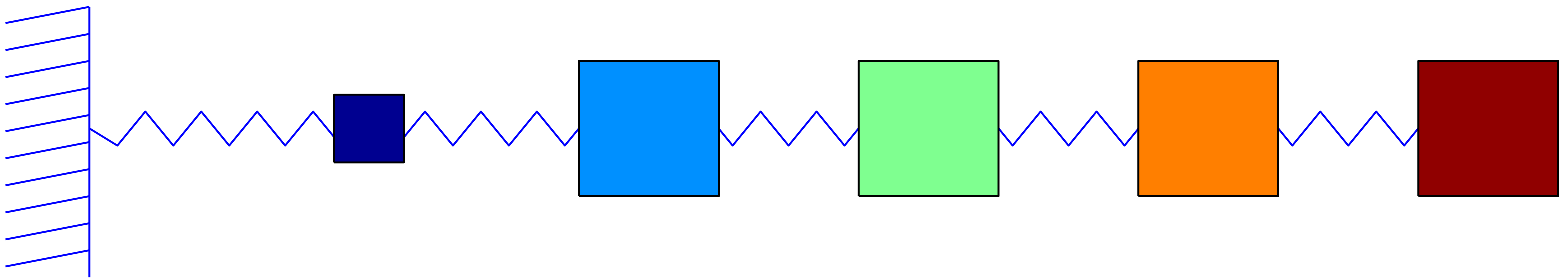




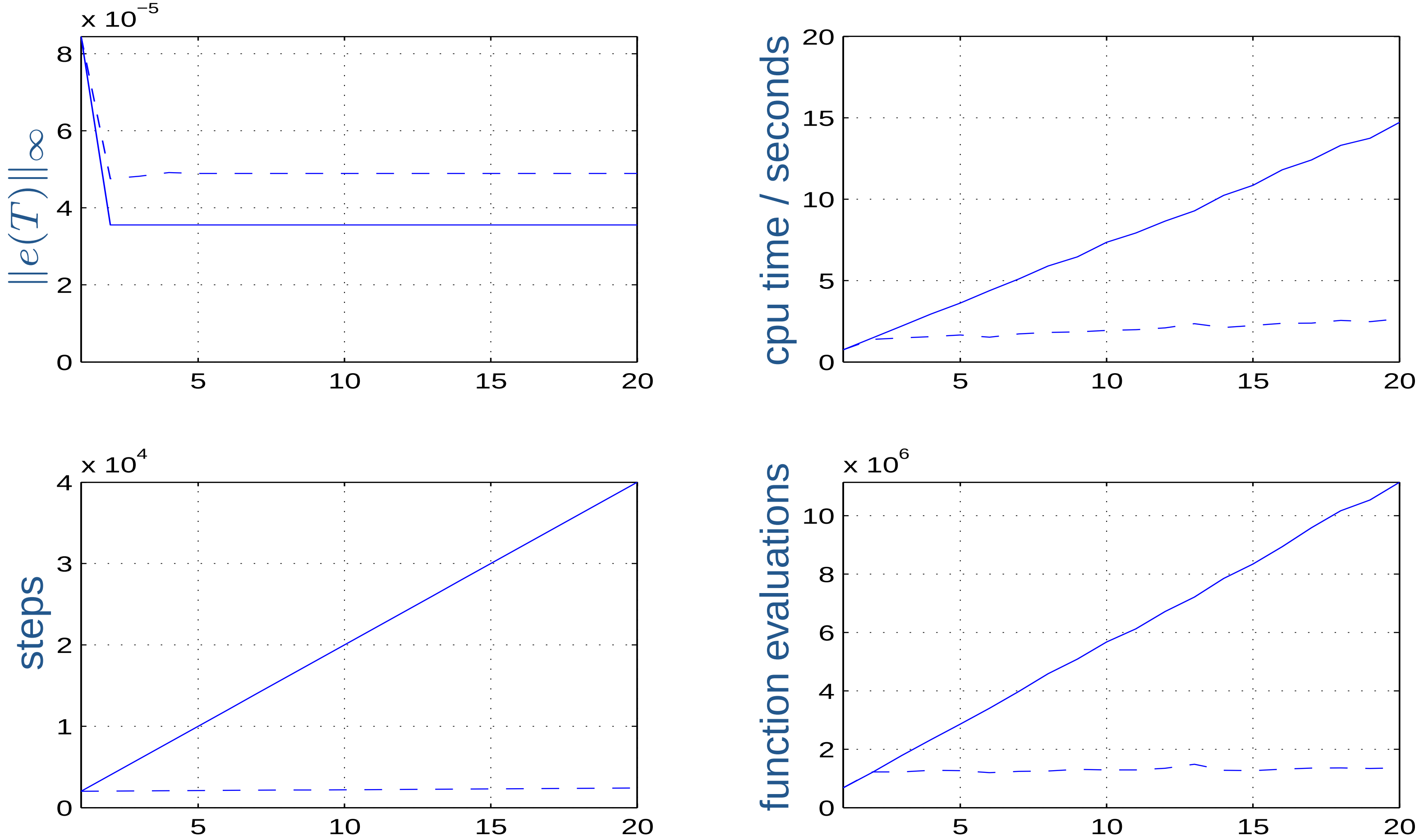


A mechanical multi-scale system

$$\begin{cases} m_i\ddot{x}_i = k(x_{i+1} - x_i) - kx_i, & i = 1, \\ m_i\ddot{x}_i = k(x_{i+1} - x_i) - k(x_i - x_{i-1}), & 1 < i < N, \\ m_i\ddot{x}_i = -k(x_i - x_{i-1}), & i = N \end{cases}$$



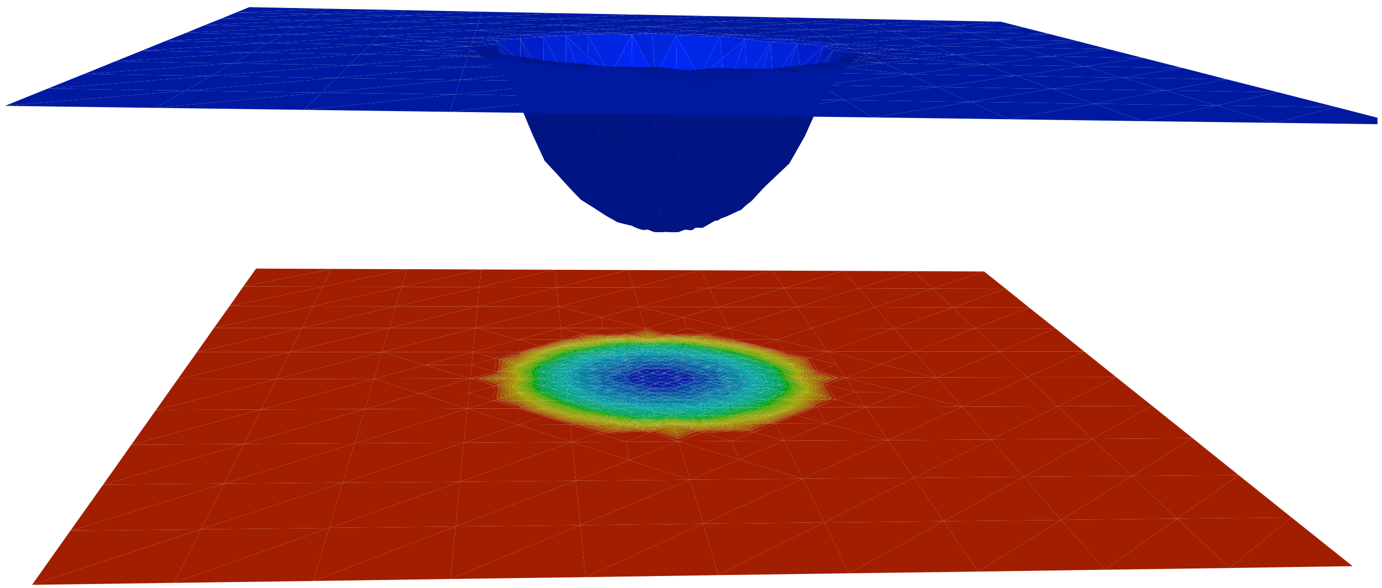
A mechanical multi-scale system



[solid: $\mathrm{cG}(q)$ dashed: $\mathrm{mcG}(q)$]

The heat equation

$$\dot{u}(x,t) - \Delta u(x,t) = f(x,t)$$



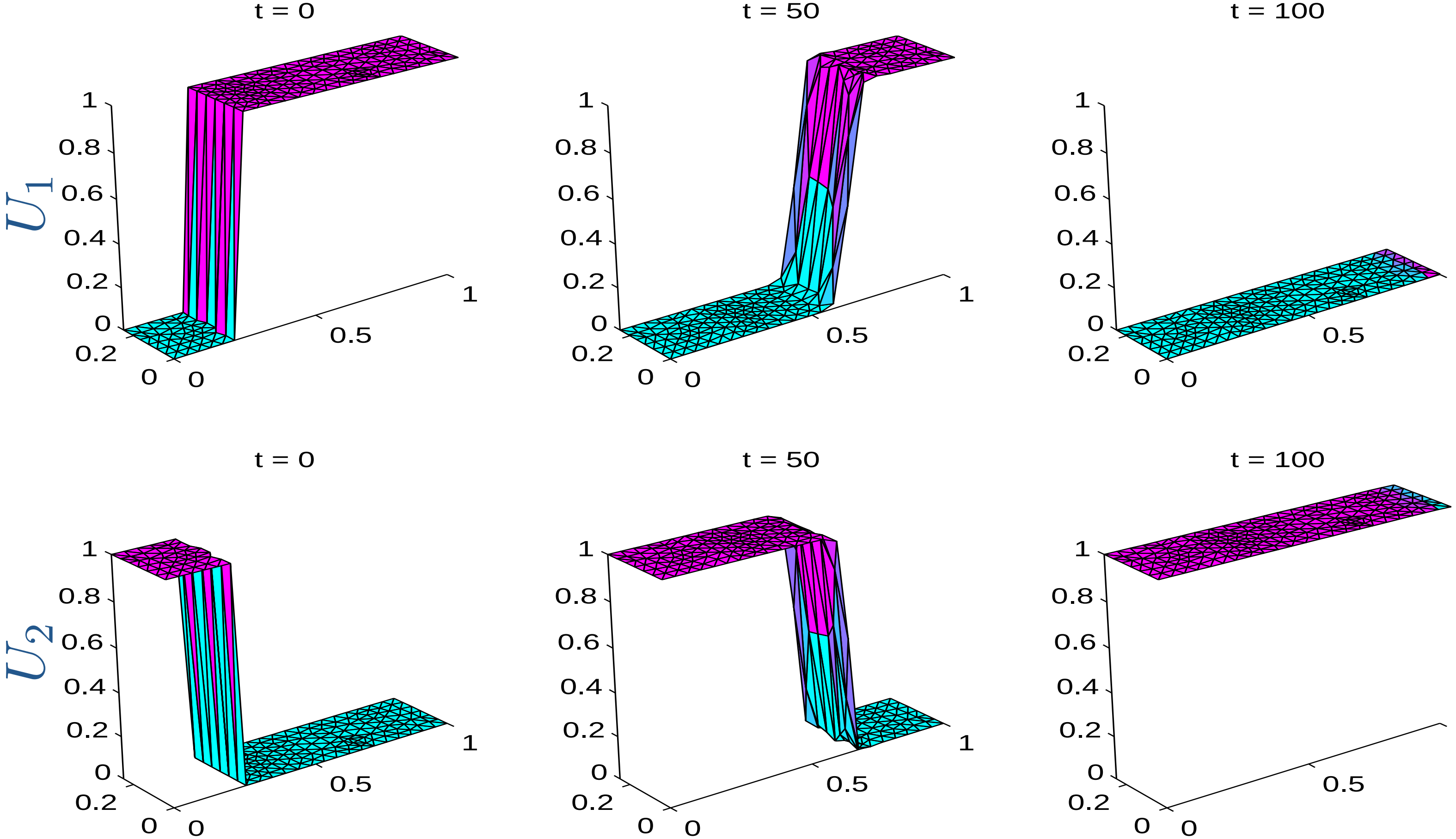
A system of reaction–diffusion equations

Two substances, A and B, distributed along $[0, 1]$ with concentrations u_1 and u_2. A reacts to form B with B working as a catalyst.

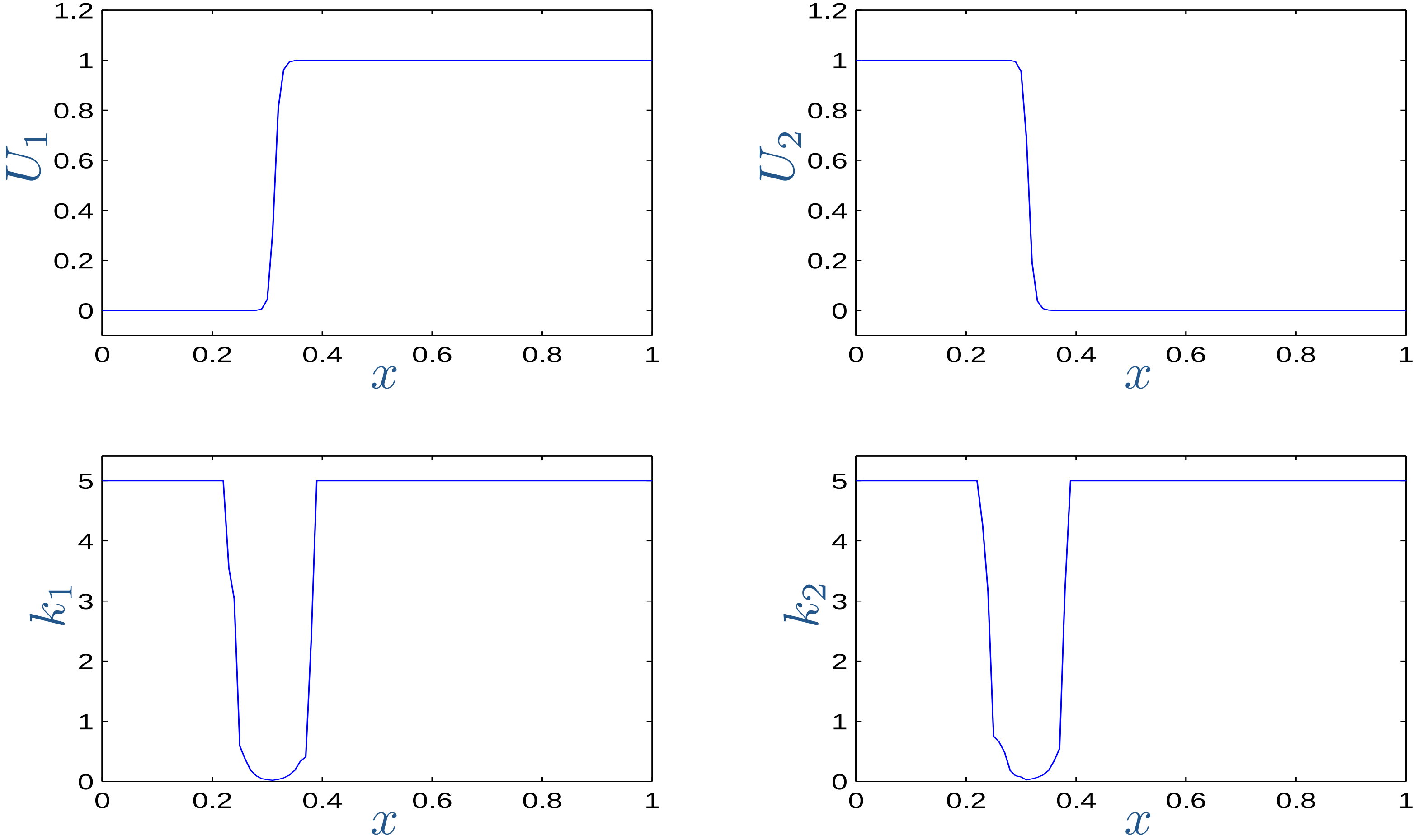
$$A + 2B \rightarrow B + 2B$$

$$\begin{cases} \dot{u}_1 - \epsilon u_1'' & = & -u_1 u_2^2 \\ \dot{u}_2 - \epsilon u_2'' & = & u_1 u_2^2 \end{cases}$$

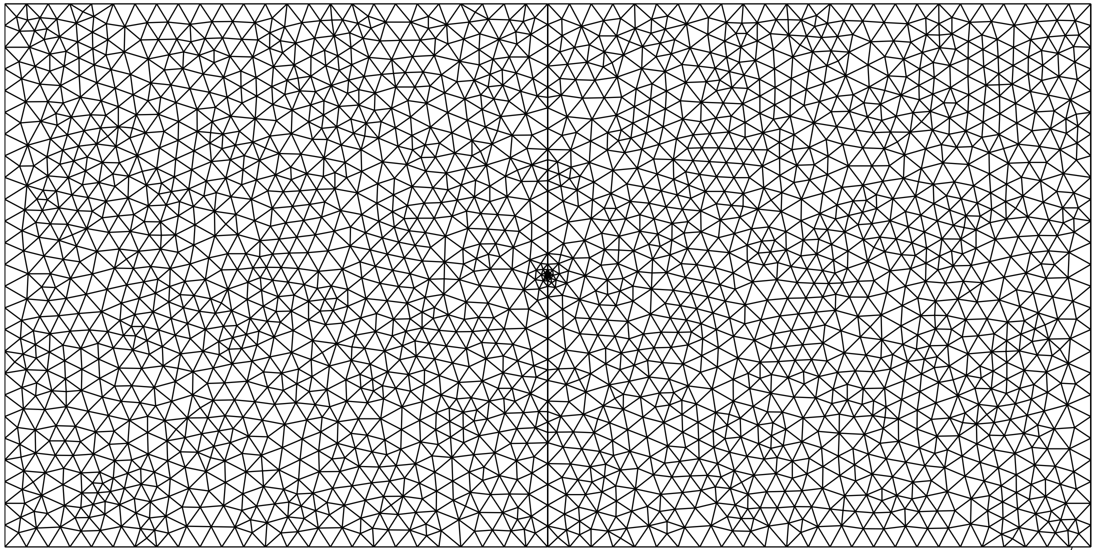
A system of reaction–diffusion equations

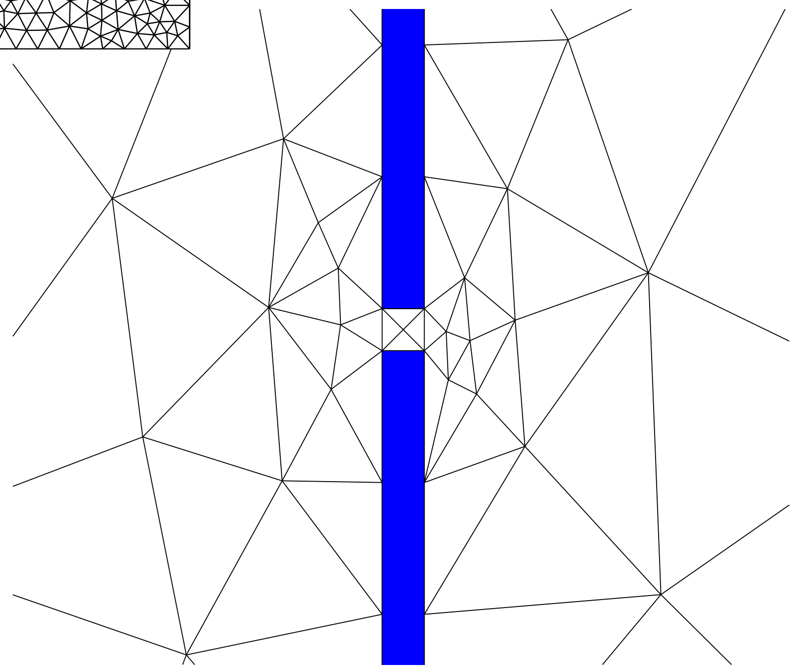


A system of reaction–diffusion equations

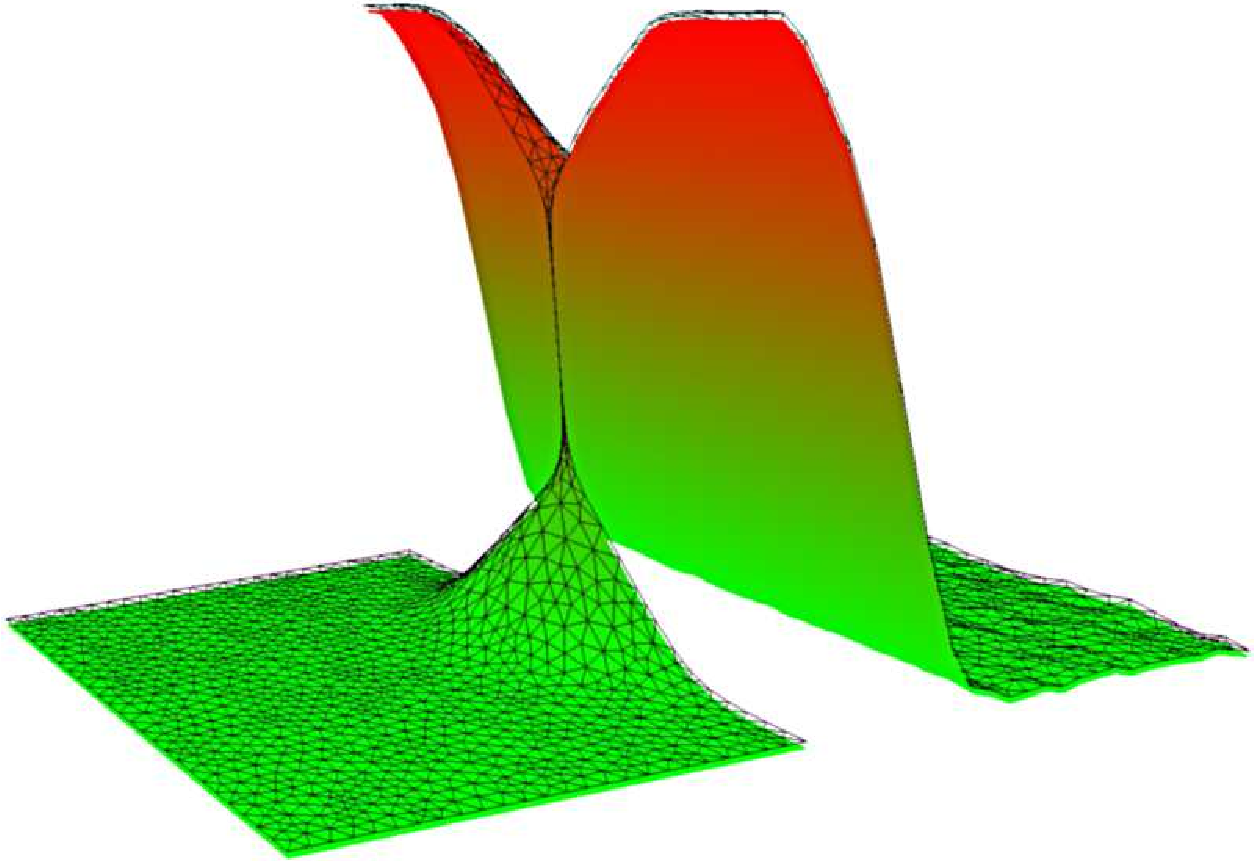


Wave propagation through a narrow slit





Wave propagation through a narrow slit



- $k \sim h$
- Multi-adaptive speedup: 3.7 (theoretical 27)

Current status and future plans

Current status and future plans

- A new improved multi-adaptive solver is currently being developed as part **DOLFIN**:

  `http://www.fenics.org/dolfin/`

- (Re-)implement dual problems and global error control
- Improve multi-adaptive preconditioners
- Integrate multi-adaptive solver with **FFC/DOLFIN**
- Testing, benchmarking, optimization

References

- *Multi-adaptive Galerkin methods for ODEs I*
  SIAM J. Sci. Comput. (2003)
- *Multi-adaptive Galerkin methods for ODEs II: Implementation and applications*
  SIAM J. Sci. Comput. (2003)
- *Multi-adaptive Galerkin methods for ODEs III: A priori error estimates*
  SIAM J. Numer. Anal. (2005)
- *Explicit time-stepping for stiff ODEs*
  SIAM J. Sci. Comput. (2003), with Eriksson/Johnson
- *Multi-adaptive time-integration*
  Appl. Numer. Math. (2004)
- *Algorithms and data structures for multi-adaptive time-stepping*
  In preparation, with Johan Jansson

`http://www.fenics.org/`