Meshfree Exponential Integrators

Stefan Rainer

joint work with

A. Ostermann (Innsbruck)

M. Caliari (Verona)

Leopold–Franzens Universität Innsbruck

Innovative Integrators

30 October 2010

Problem class:

- Time-dependent PDE's with dominating advection part
- Solution has a small essential support
- High dimensional problems

Goal:

- Develop an integrator with high accuracy both in space and time
- Use the special form of the solution to save memory

Example: Consider the Molenkamp–Crowley equation

$$\partial_t u = \partial_x(au) + \partial_y(bu)$$

with

$$a(x,y) = 2\pi y, \quad b(x,y) = -2\pi x.$$

The initial profile,

$$u_0(x,y) = \exp(-10(x-0.2)^2 - 10(y-0.2)^2),$$

rotates around the origin.

Given a stiff initial-value problem

$$u'(t) = F(u(t)), \qquad u(0) = u_0.$$

Linearising the problem at a state w gives

$$v'(t) = Av(t) + g(v(t)), \qquad v(0) = u_0 - w$$

with $A = DF(w)$, $v(t) = u(t) - w$ and g a nonlinear reminder.

Using the variation-of-constant formula the solution has the form

$$v(t) = e^{tA} v_0 + \int_0^t e^{(t-\tau)A} g(v(\tau)) \mathrm{d}\tau$$

Given a stiff initial-value problem

$$u'(t) = F(u(t)), \qquad u(0) = u_0.$$

Linearising the problem at a state w gives

$$v'(t) = Av(t) + g(v(t)), \qquad v(0) = u_0 - w$$

with $A = DF(w)$, $v(t) = u(t) - w$ and g a nonlinear reminder.

Using the variation-of-constant formula the solution has the form

$$v(t) = e^{tA} v_0 + \int_0^t e^{(t-\tau)A} g(v(\tau)) \mathrm{d}\tau$$

Example: *Exponential Euler Method*

$$g(v(\tau)) \approx g(v_0)$$

gives

$$v_1 = e^{hA} v_0 + h\varphi_1(hA) g(v_0).$$

Here h denotes the step size and φ_1 is the entire function $\varphi_1(z) = \frac{e^z - 1}{z}$.

Exponential and related functions will be approximated with the Leja point method (based on Newton's interpolation formula at a Leja sequence).

- The Leja point method needs a rough approximation of *largest* eigenvalues of the Operator.
- This can be done by splitting the operator in a symmetric and skew-symmetric part and computing the largest in magnitude eigenvalues of both parts.
- We get estimate of the spectrum in form of a rectangle with vertices $(-a,-\mathrm{i}b),(0,-\mathrm{i}b),(0,\mathrm{i}b),(-a,\mathrm{i}b)$, $a,b \geq 0$.
- Distinguish two cases:
  $a \geq b$, take Leja points on interval $[-a,0]$.
  $a < b$, consider conjugate pairs of Leja points on domain $\{z \in \mathbb{C} \colon \Re(z) = -a/2, \Im(z) \in [-b,b]\}$

Some properties:

- similar distribution as Chebyshev points
- defined recursively - combines well with Newton interpolation
- based on matrix-vector multiplications
- superlinear convergence
- real arithmetic

Basis function: Radial Basis Functions

Given a function $f : \mathbb{R}^d \to \mathbb{R}$ sampled at point set $X = \{x_1, \ldots, x_m\}$

Approximate f by the interpolant

$$s(\xi) = \sum_{x \in X} \lambda_x \phi\left(\|\xi - x\|\right)$$

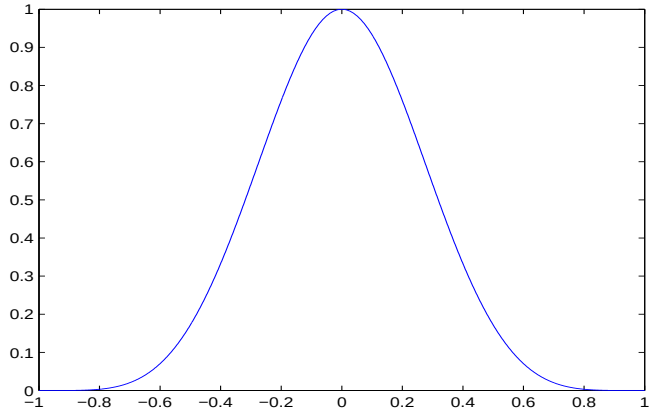
using a radial function $\phi : \mathbb{R}_+ \to \mathbb{R}$.
The coefficients $\lambda = (\lambda_x)_{x \in X}$ are chose such that

$$s|_X = f|_X \Longrightarrow A\lambda = f|_X$$

with $A = \{\phi\left(\|x_i - x_j\|\right)\}_{x_i, x_j \in X}$.

RBF interpolation:

- works in any dimension
- simple to implement
- high accuracy



Compactly supported RBFs

We use Wendland functions. Defined via

$$\phi_{d,k} = I^k \phi_{\lfloor d/2 \rfloor + k + 1}, \qquad \phi_\ell(r) = (1-r)_+^\ell$$

where $(I\phi)(r) = \int_r^\infty t\phi(t)\mathrm{d}t$ and d is the space dimension.

d	$\phi_{d,k}(r)$	smoothness
1	$\phi_{1,0}(r) = (1-r)_+$	C^0
	$\phi_{1,1}(r) = (1-r)_+^3(3r+1)$	C^2
3	$\phi_{3,0}(r) = (1-r)_+^2$	C^0
	$\phi_{3,1}(r) = (1-r)_+^4(4r+1)$	C^2
	$\phi_{3,2}(r) = (1-r)_+^6(35r^2+18r+3)$	C^4
	$\phi_{3,3}(r) = (1-r)_+^8(32r^3+25r^2+8r+1)$	C^6

Some properties of $\phi_{d,k}(r)$

- smoothness $2k$
- strictly positive definite on $\mathbb{R}^d$
- local interpolation error bounded by

$$|f(y) - s(y)| \leq c_f C h_\rho^{k+1/2}, \qquad h_\rho < h \text{ for } h \text{ small enough,}$$

  where $h_\rho(y) := \max_{y \in B_\rho(\xi)} \min_{x \in X} \|\xi - x\|_2$ for a given $\rho > 0$.
- condition number of interpolation matrix A bounded by

$$\mathrm{cond}_2(A) \leq C q_x^{-d-2k-1},$$

  where $q_X = 1/2 \min_{x_i, x_j \in X} \{\|x_i - x_j\|_2 \mid x_i \neq x_j\}$.

Combination

Consider a linear differential equation discretized at X

$$u_t(t,\cdot)|_X = Lu(t,\cdot)|_X, \qquad u(0,\cdot)|_X = u_0|_X.$$

Combination

Consider a linear differential equation discretized at X

$$s_t(t,\cdot)|_X = Ls(t,\cdot)|_X, \qquad s(\cdot,0)|_X = u_0|_X.$$

Approximate $Lu(t,\xi)$ by

$$Ls(t,\xi) = L\sum_{x\in X}\lambda_x(t)\phi(\|\xi - x\|) = \sum_{x\in X}\lambda_x(t)L\phi(\|\xi - x\|).$$

This gives

$$s_t(t,\cdot)|_X = A_L A^{-1} s(t,\cdot)|_X,$$

with $A_L = \{L\phi\left(\|x_i - x_j\|\right)\}_{x_i,x_j\in X}$.

Standard forward difference approximation in time leads to a simple integrator

$$s(t_{n+1},\cdot)|_X = s(t_n,\cdot)|_X + \triangle t A_L A^{-1} s(t_n,\cdot)|_X.$$

For the error estimate in space use two set of points:

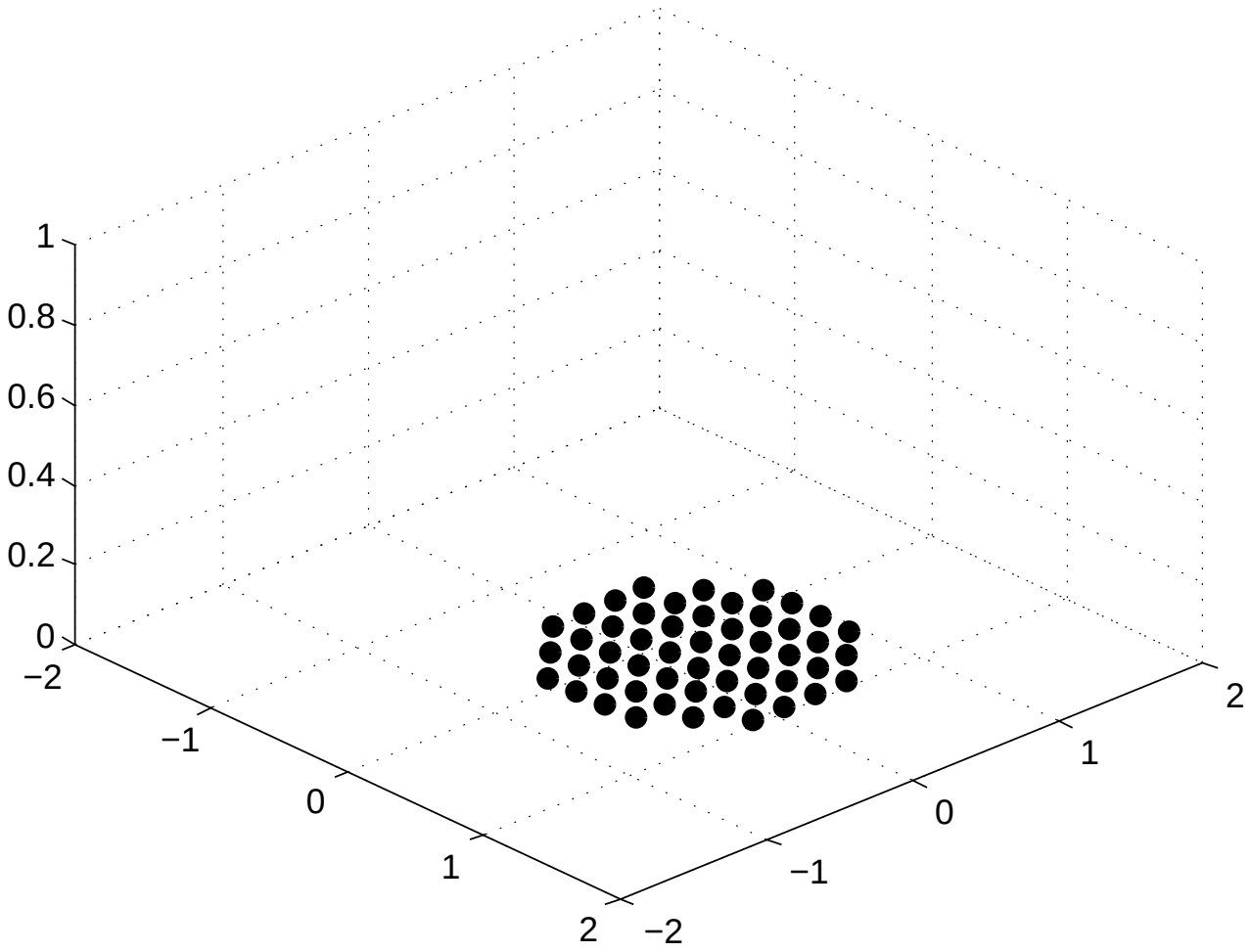
interpolation points check points

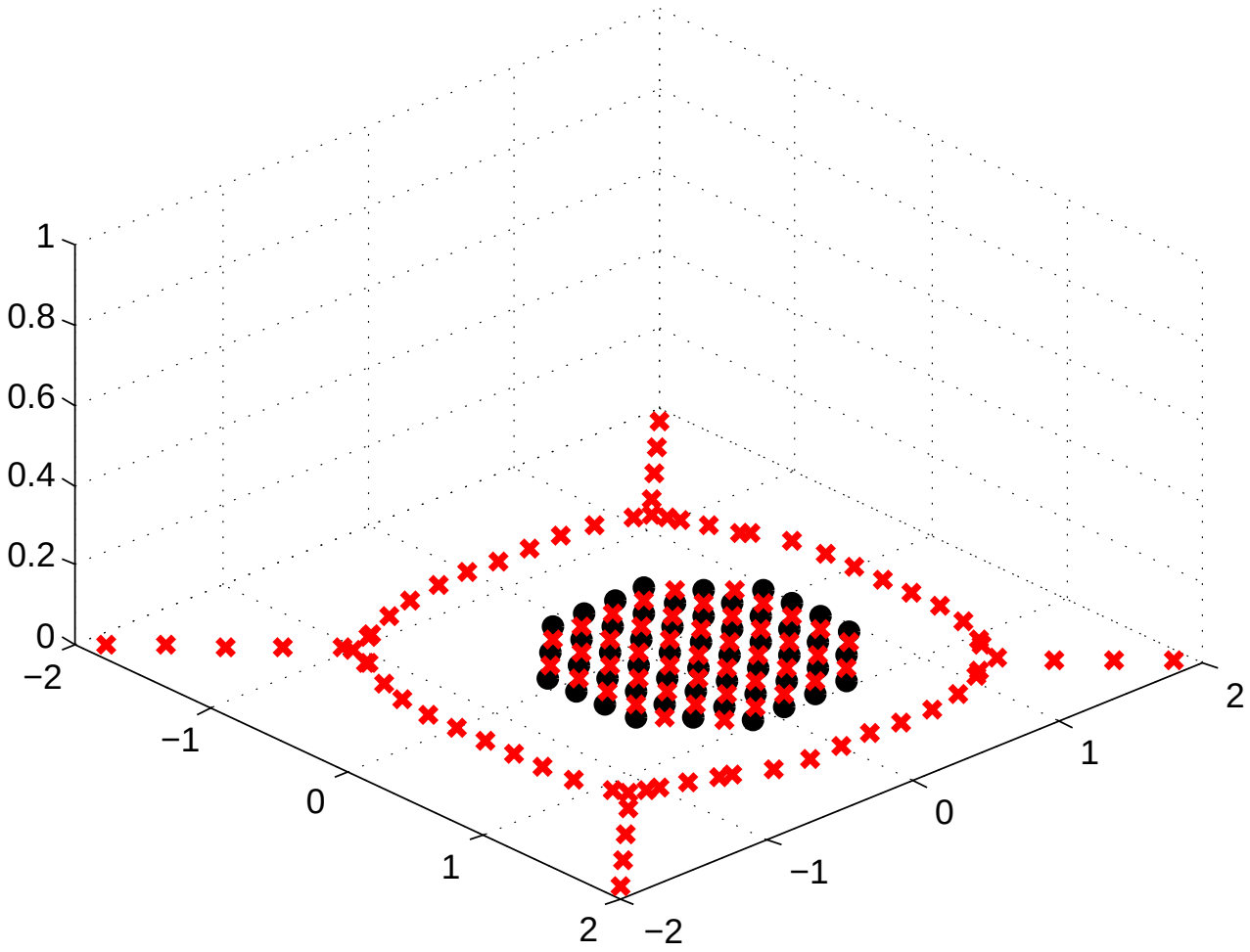
Every interpolation point has some checkpoints.
Suppose f is known everywhere.
Idea:

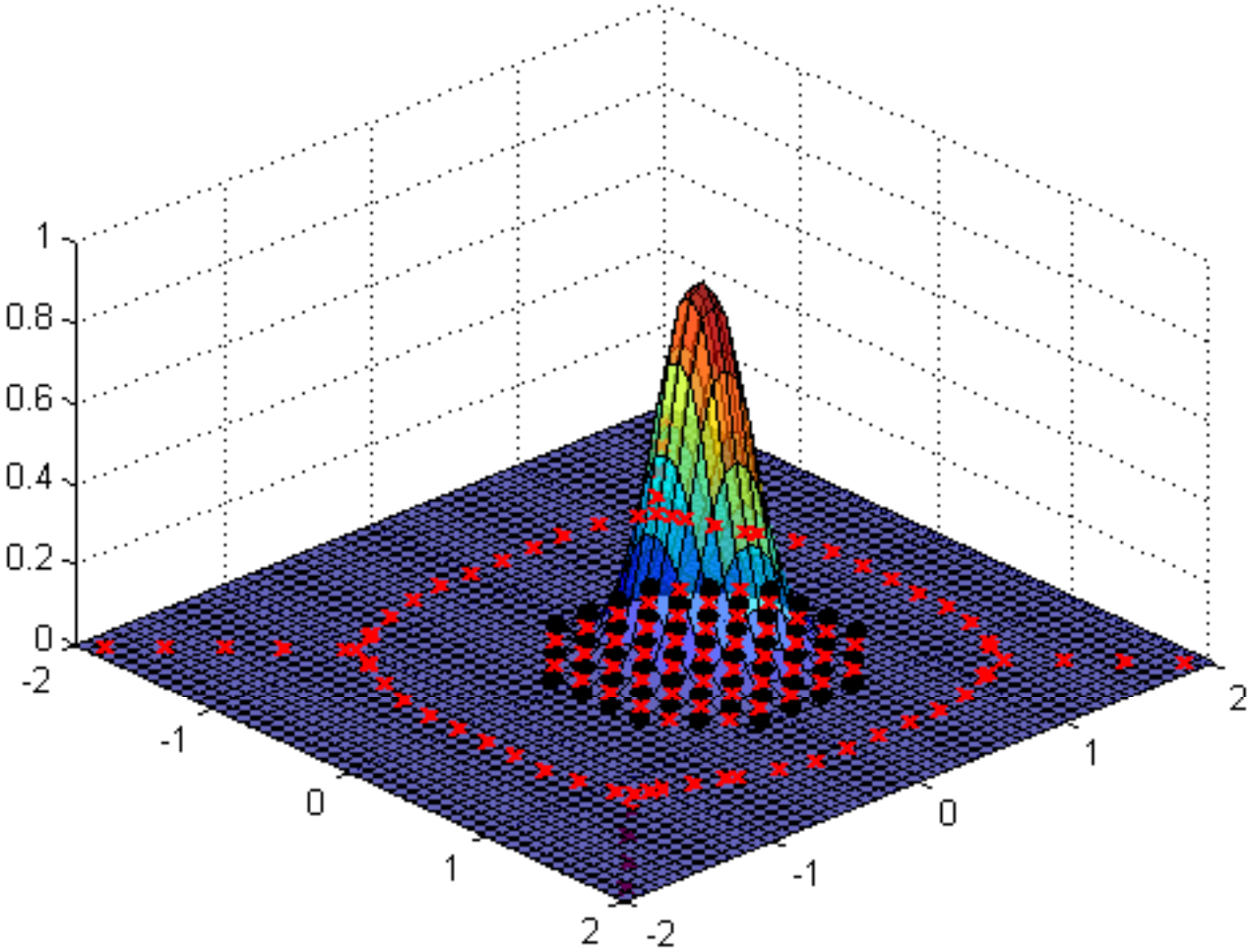
- interpolate at interpolation points
- evaluate at check points
- calculate error at check points to get error estimate in space

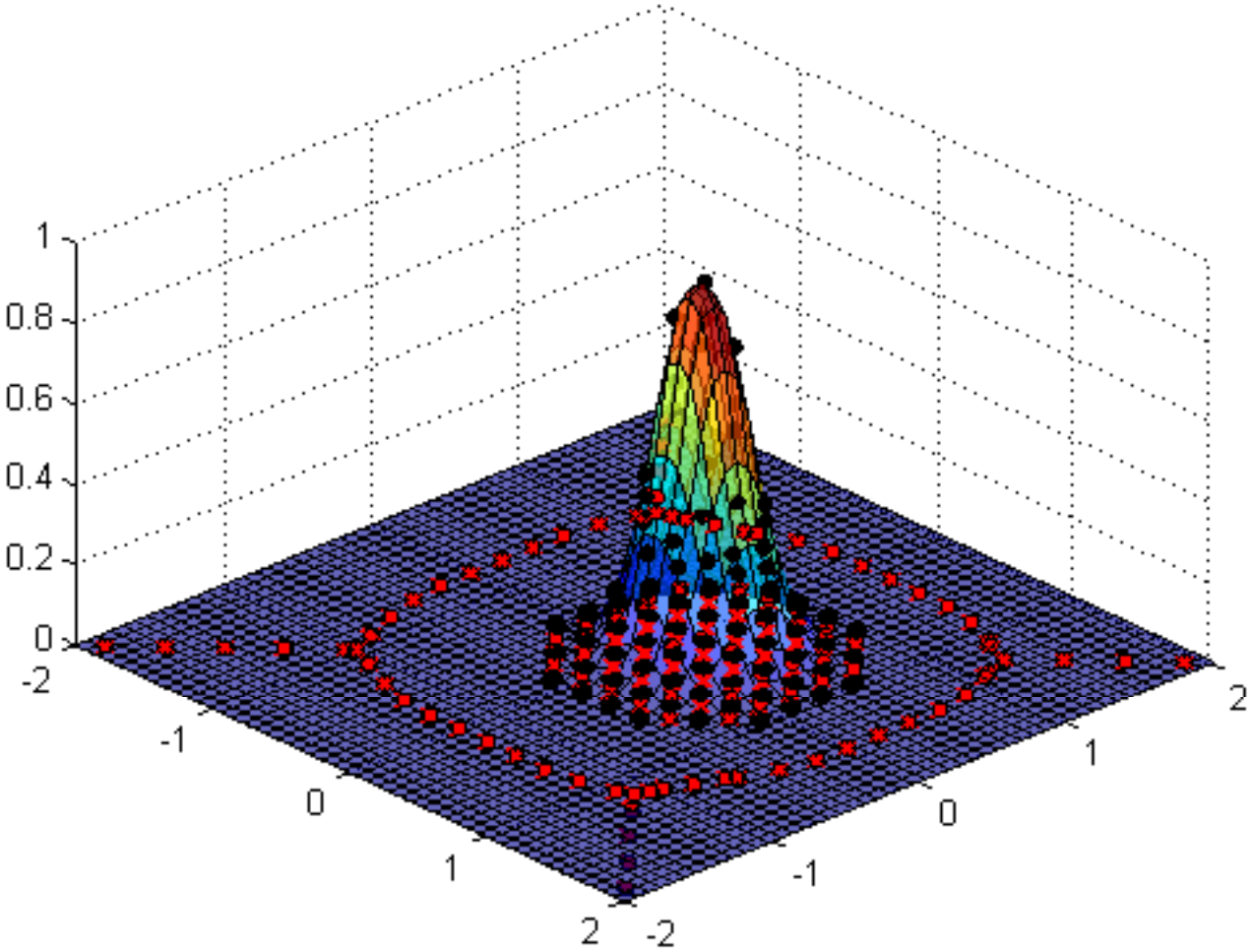
Refinement If error at a check point is too large, check point becomes interpolation point.

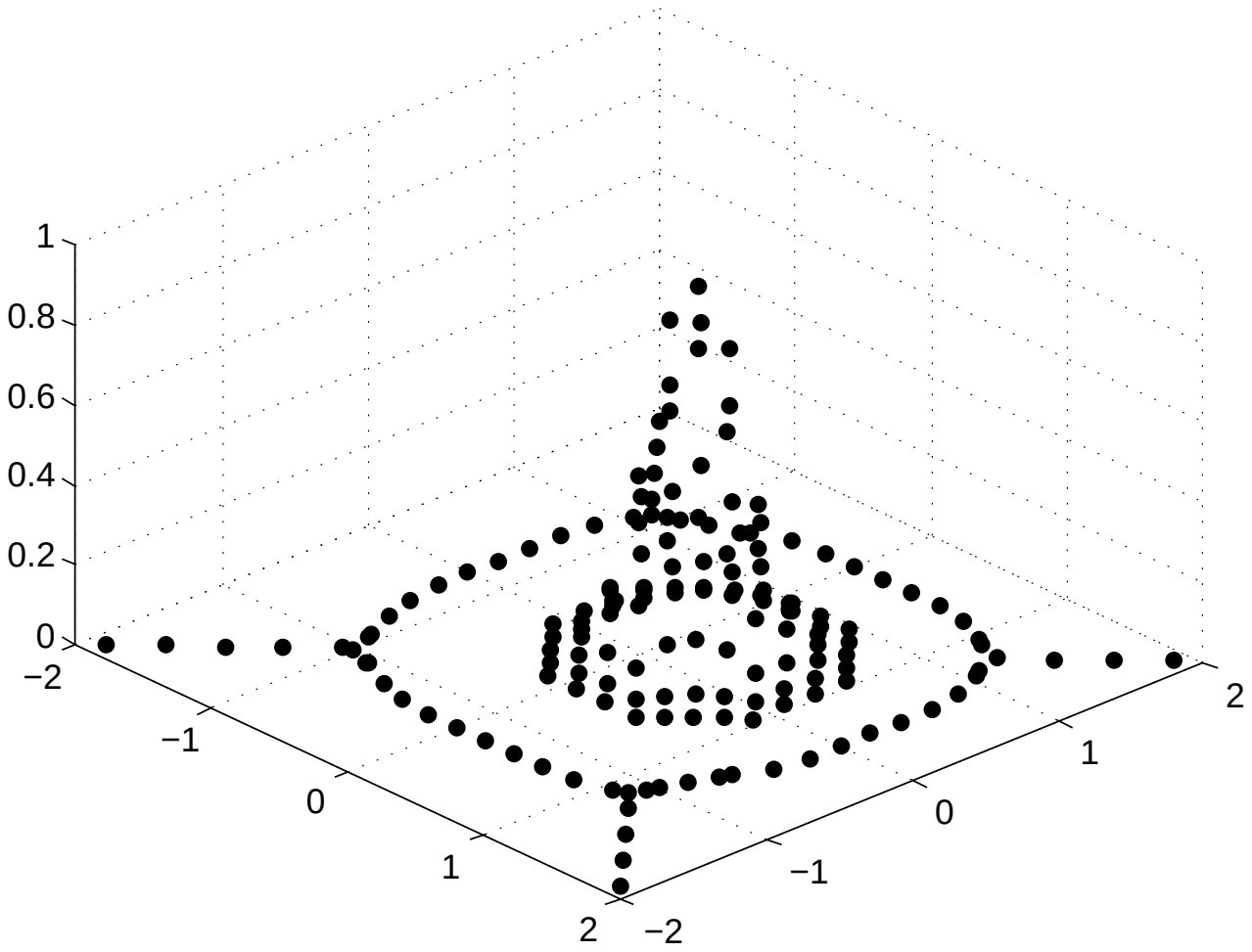
Coarsening If error at all check points corresponding to a interpolation point are small enough, interpolation point will be removed.

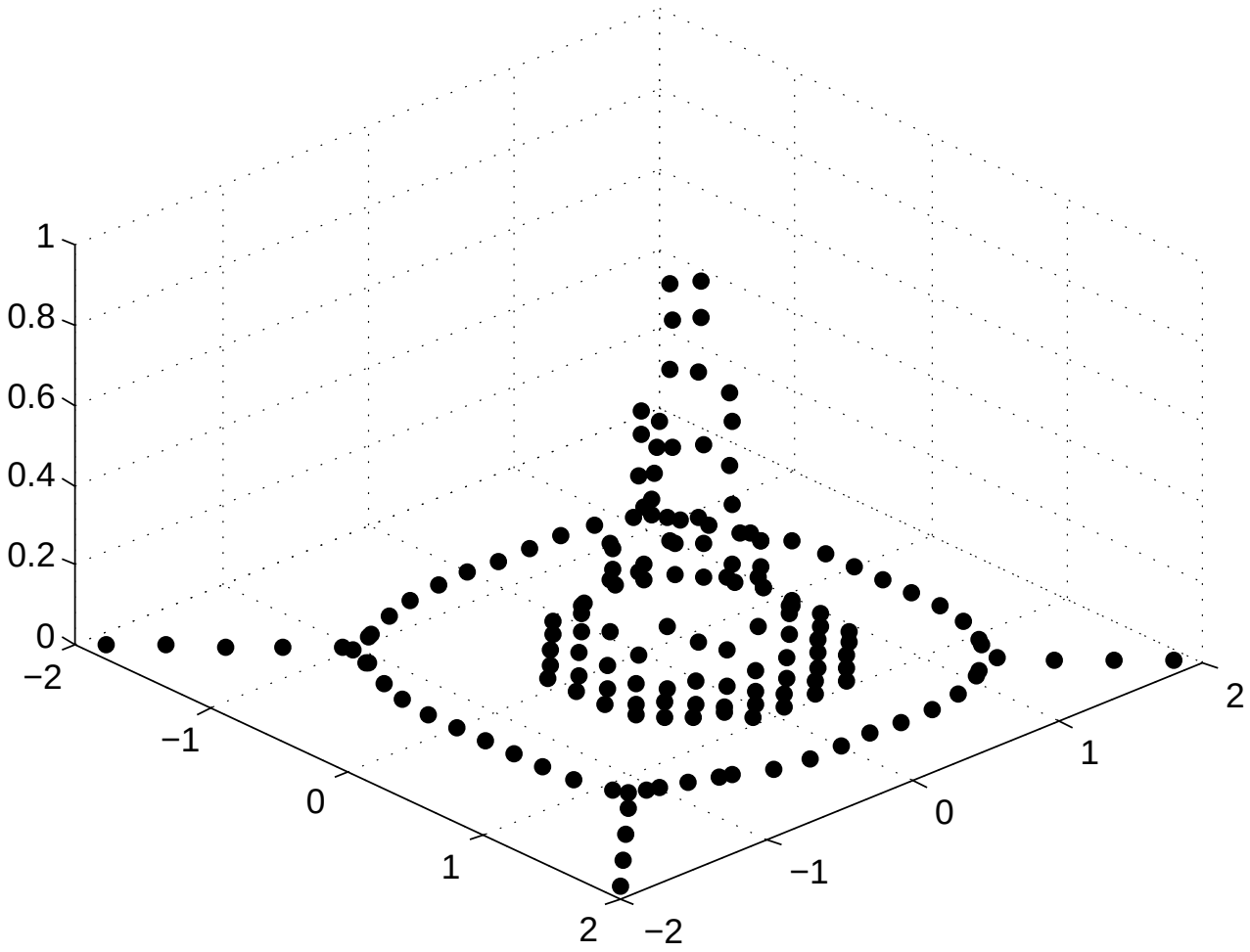

1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2

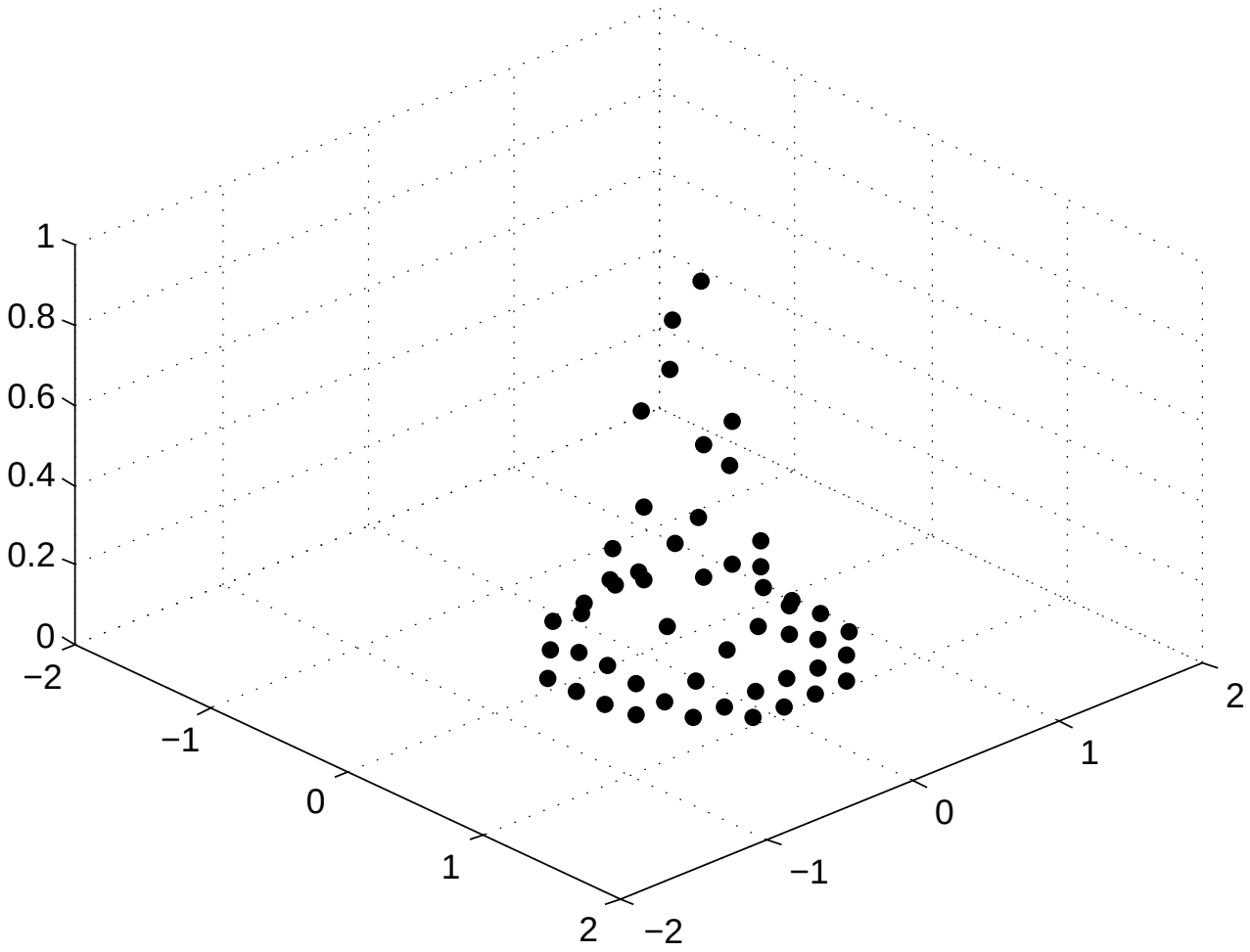

1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2

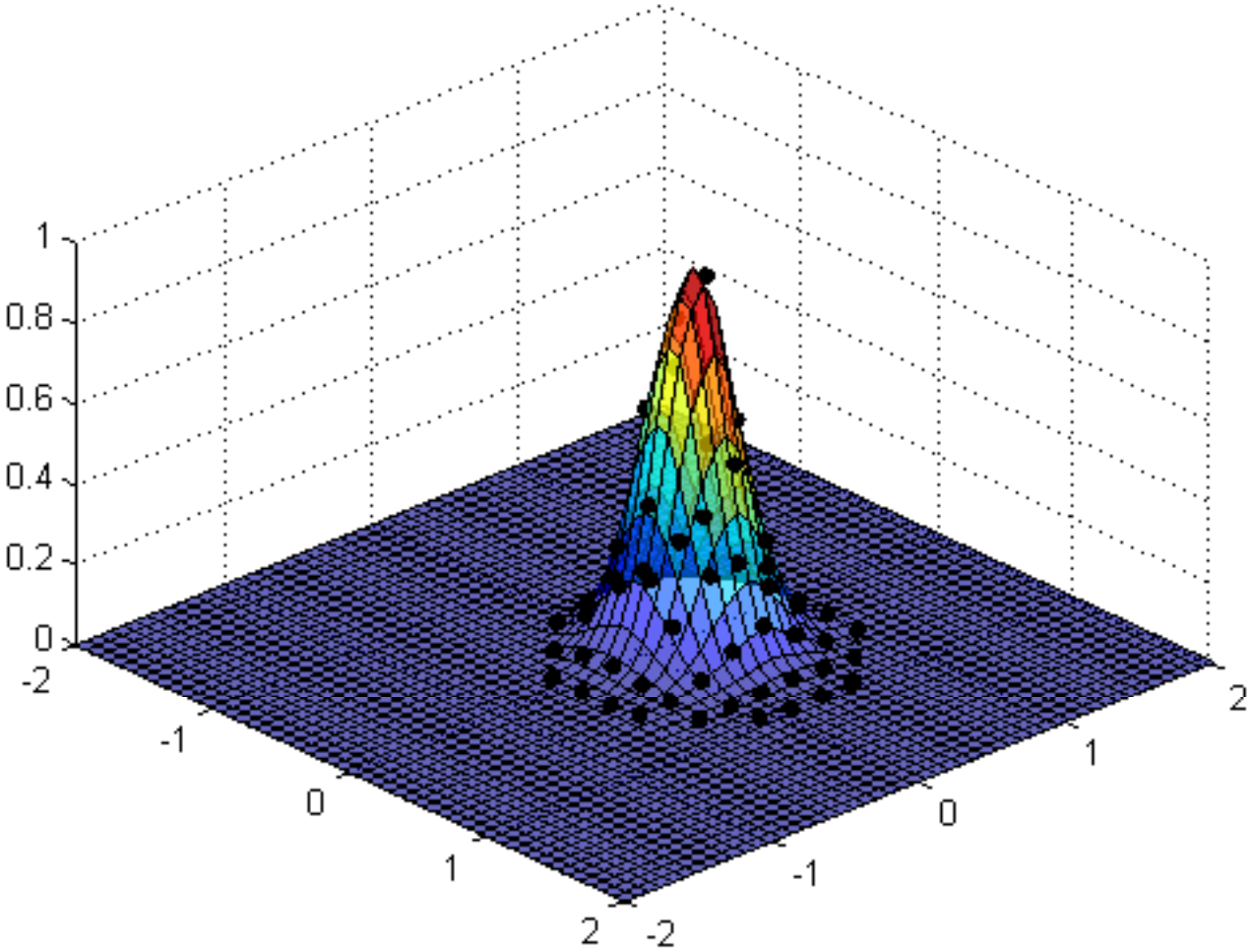

1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2
-2
-1
0
1
2

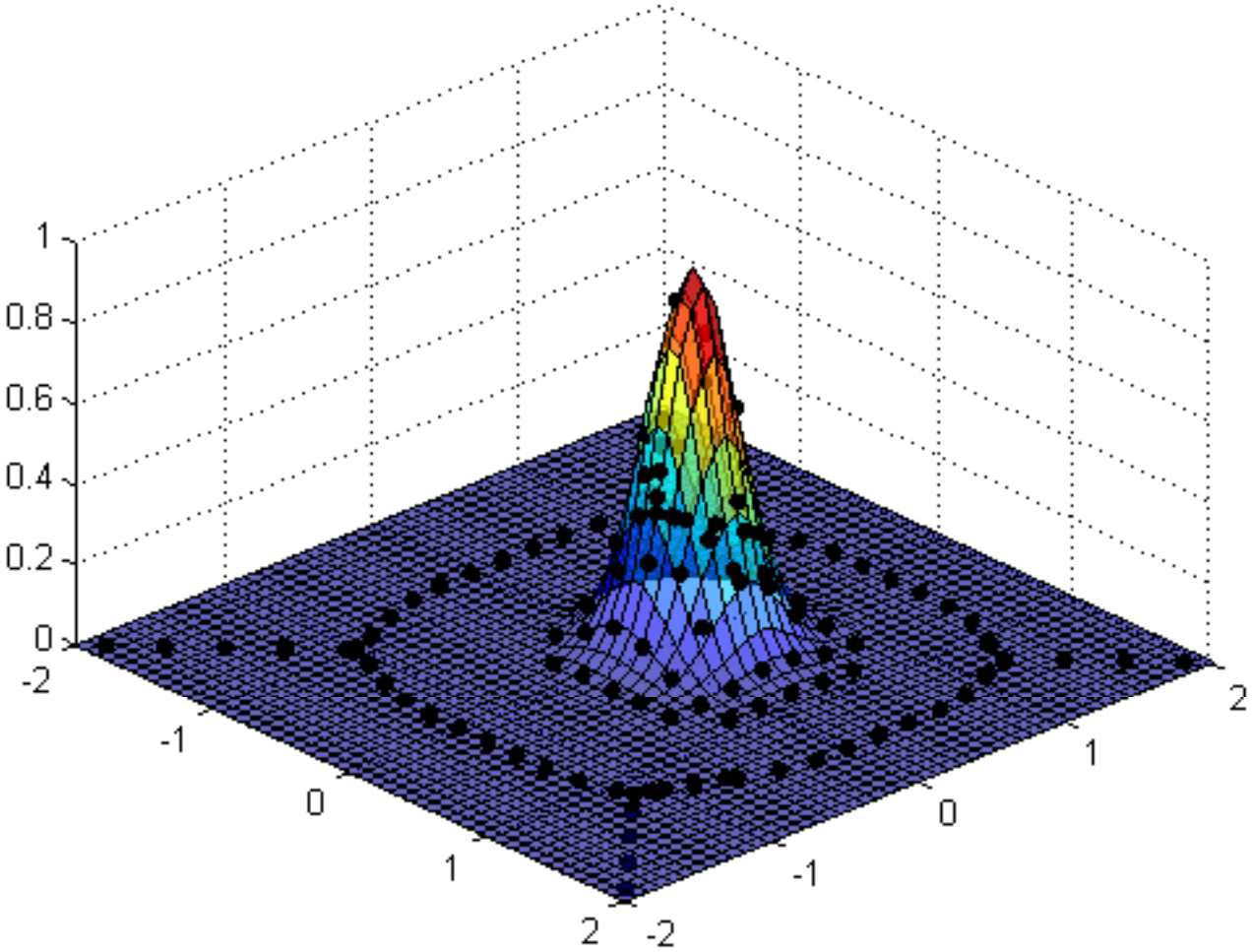

1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2
-2
-1
0
1
2

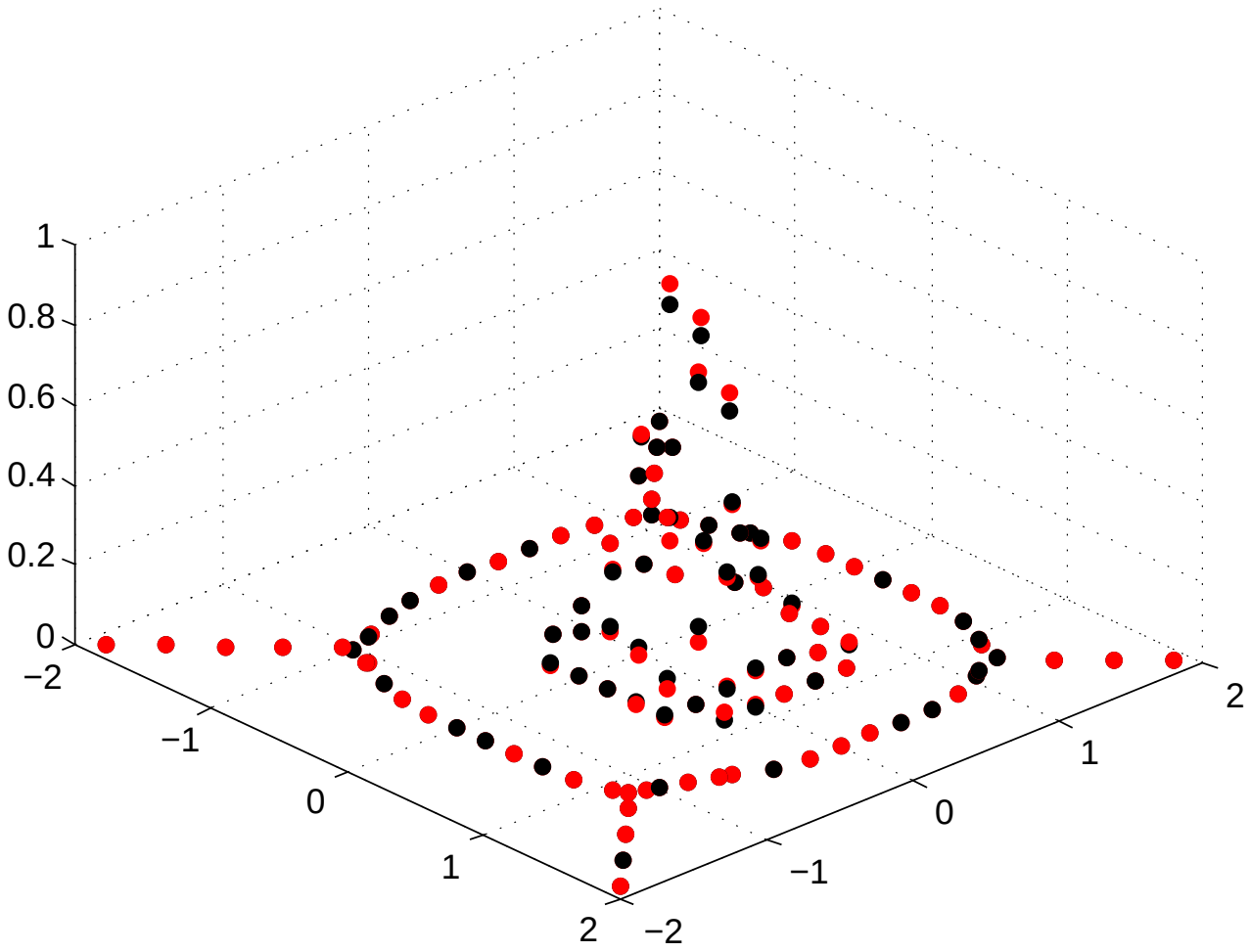

1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2

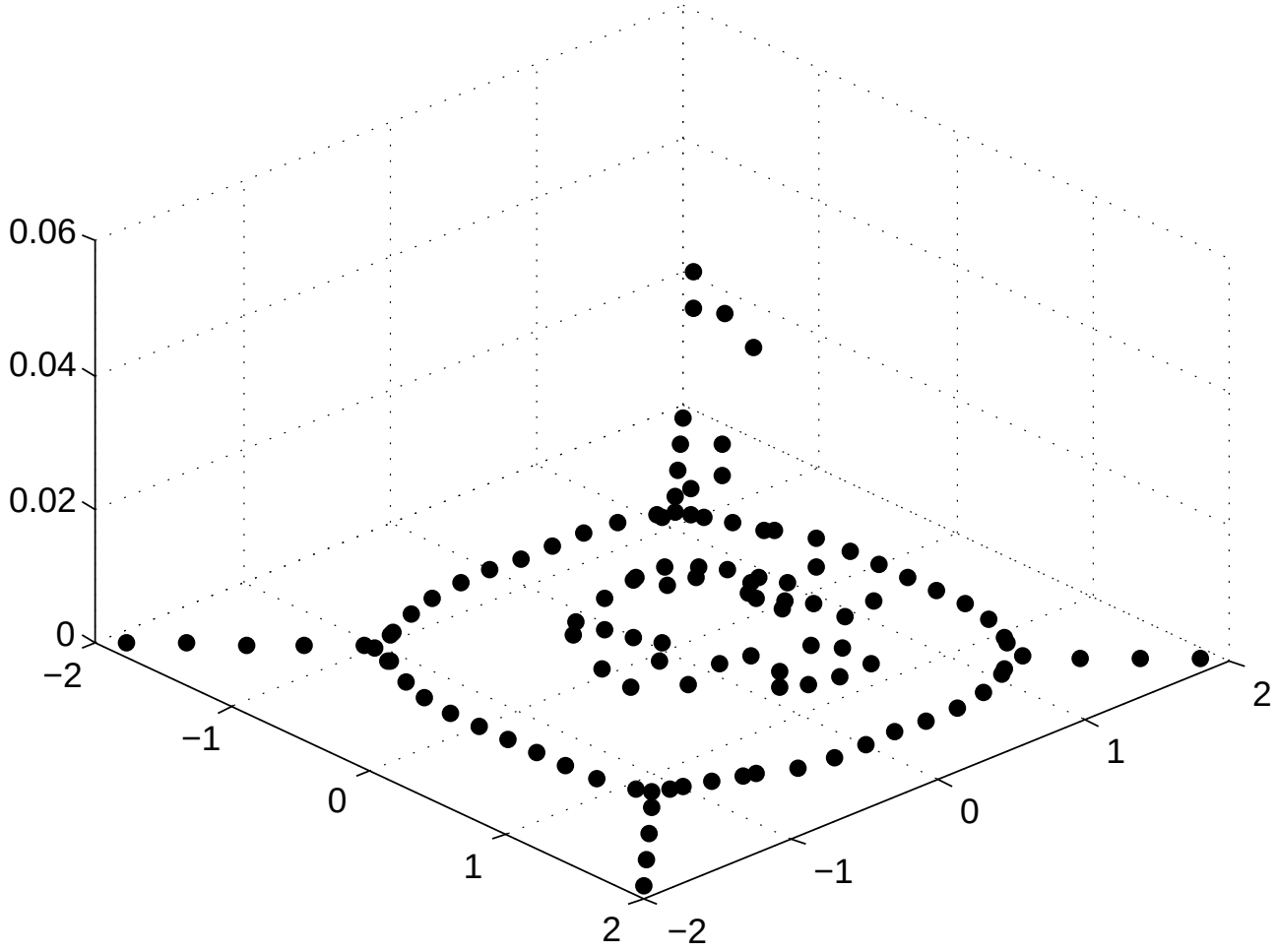

1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2


1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2


1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2
-2
-1
0
1
2


1
0.8
0.6
0.4
0.2
0
-2
-1
0
1
2
-2
-1
0
1
2


1
0.8
0.6
0.4
0.2
0
−2
−1
0
1
2
−2
−1
0
1
2


0.06
0.04
0.02
0
−2
−1
0
1
2
−2
−1
0
1
2

check points

- For a given set of interpolation points calculate a Delaunay triangulation
- Take circumcenters of resulting triangles as check points.

Two reasons

- Check points maximize the local error bound
- If a refinement is needed, adding such a check point minimizes the growth or condition number of interpolation matrix.

Example1: Consider the Molenkamp–Crowley equation

$$\partial_t u = \partial_x(au) + \partial_y(bu)$$

with

$$a(x,y) = 2\pi y, \quad b(x,y) = -2\pi x$$

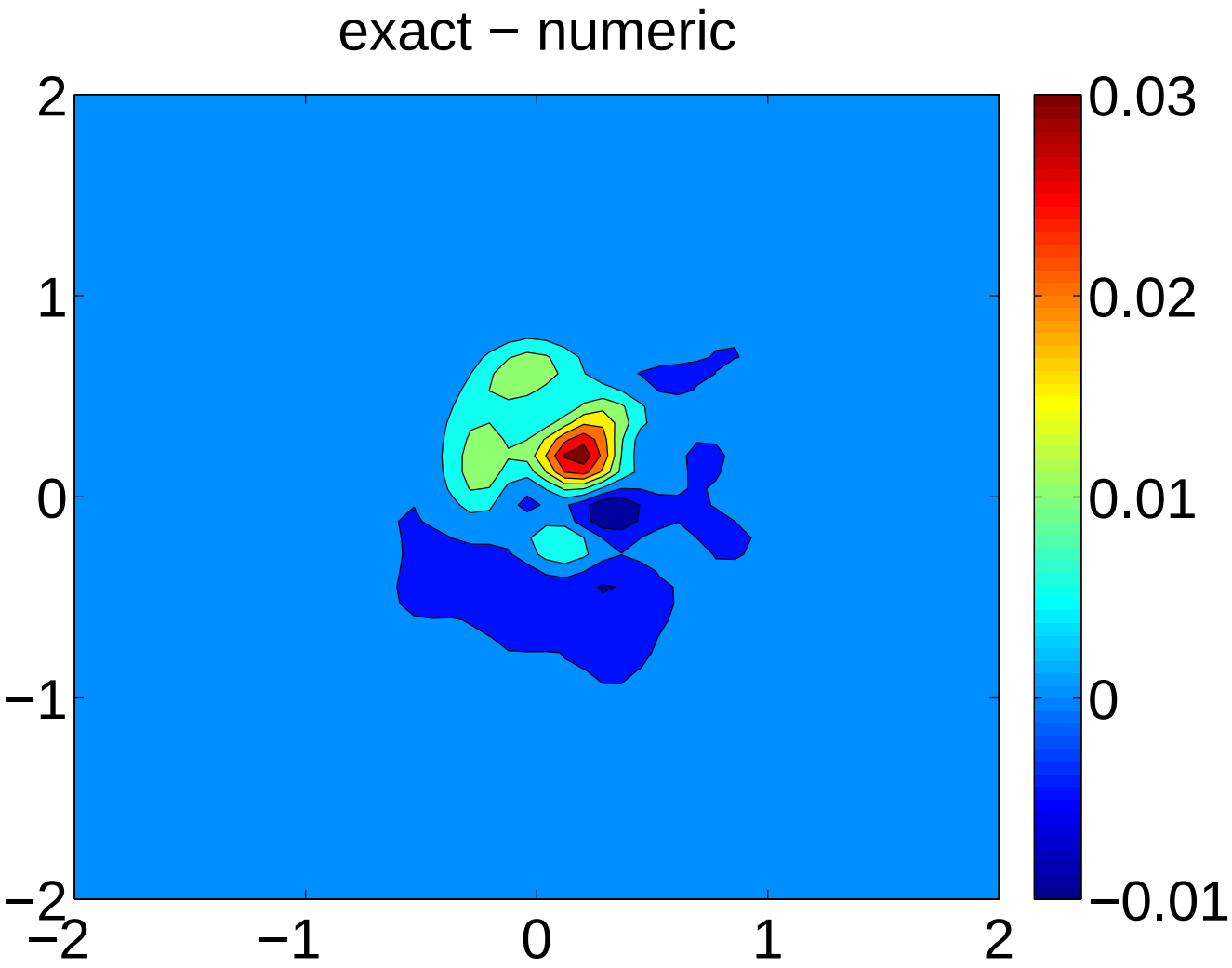
and

$$u_0(x,y) = \exp(-10(x-0.2)^2 - 10(y-0.2)^2).$$

The initial profile rotates around the origin.

Tolerances: Space: 10^{-3}; Time: 10^{-3};
constant time step size: $5 \cdot 10^{-2}$

Compute 4 turns of the pulse (80 steps)

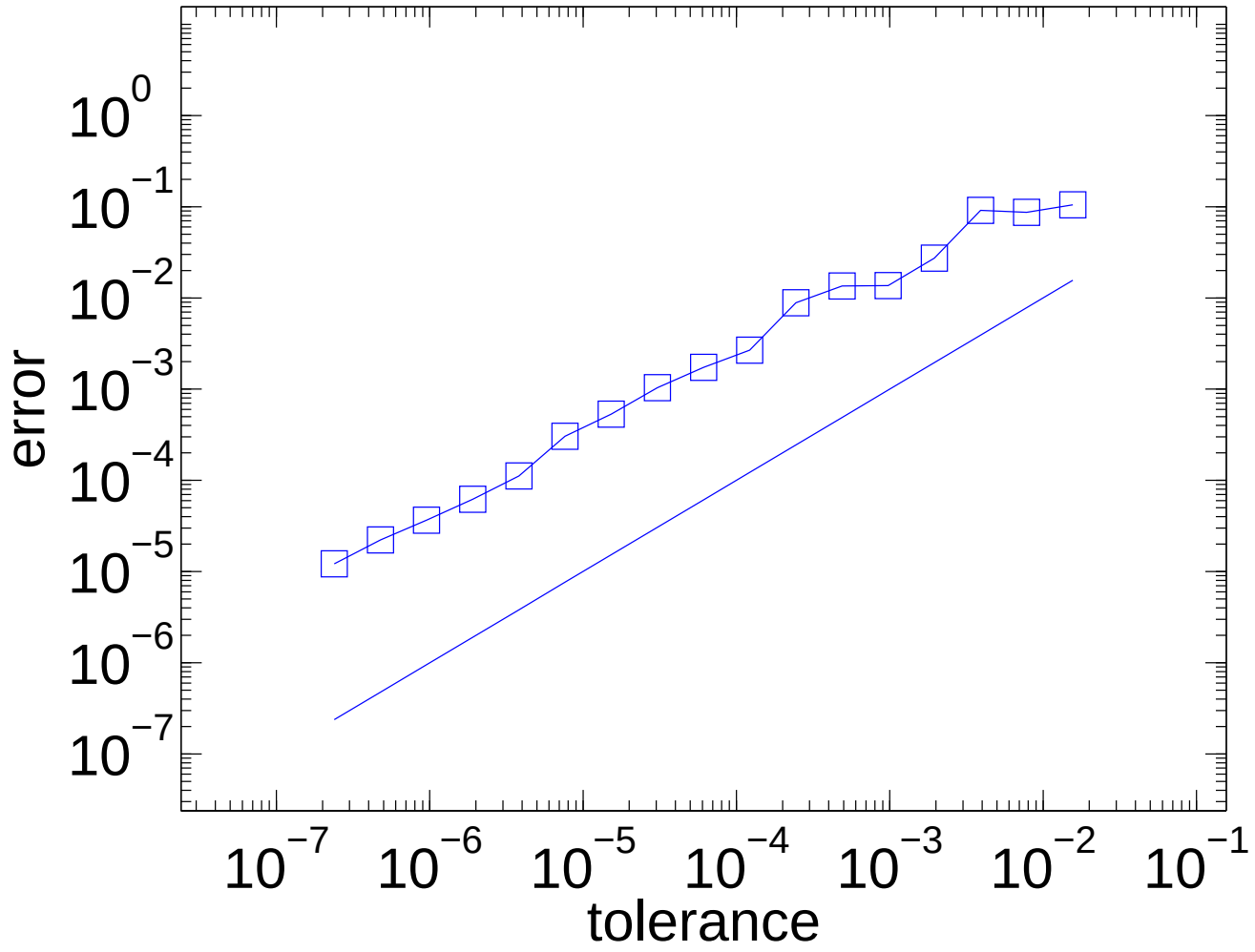

exact − numeric
2
1
0
−1
−2
−2
−1
0
1
2
0.03
0.02
0.01
0
−0.01

Two numerical experiments:

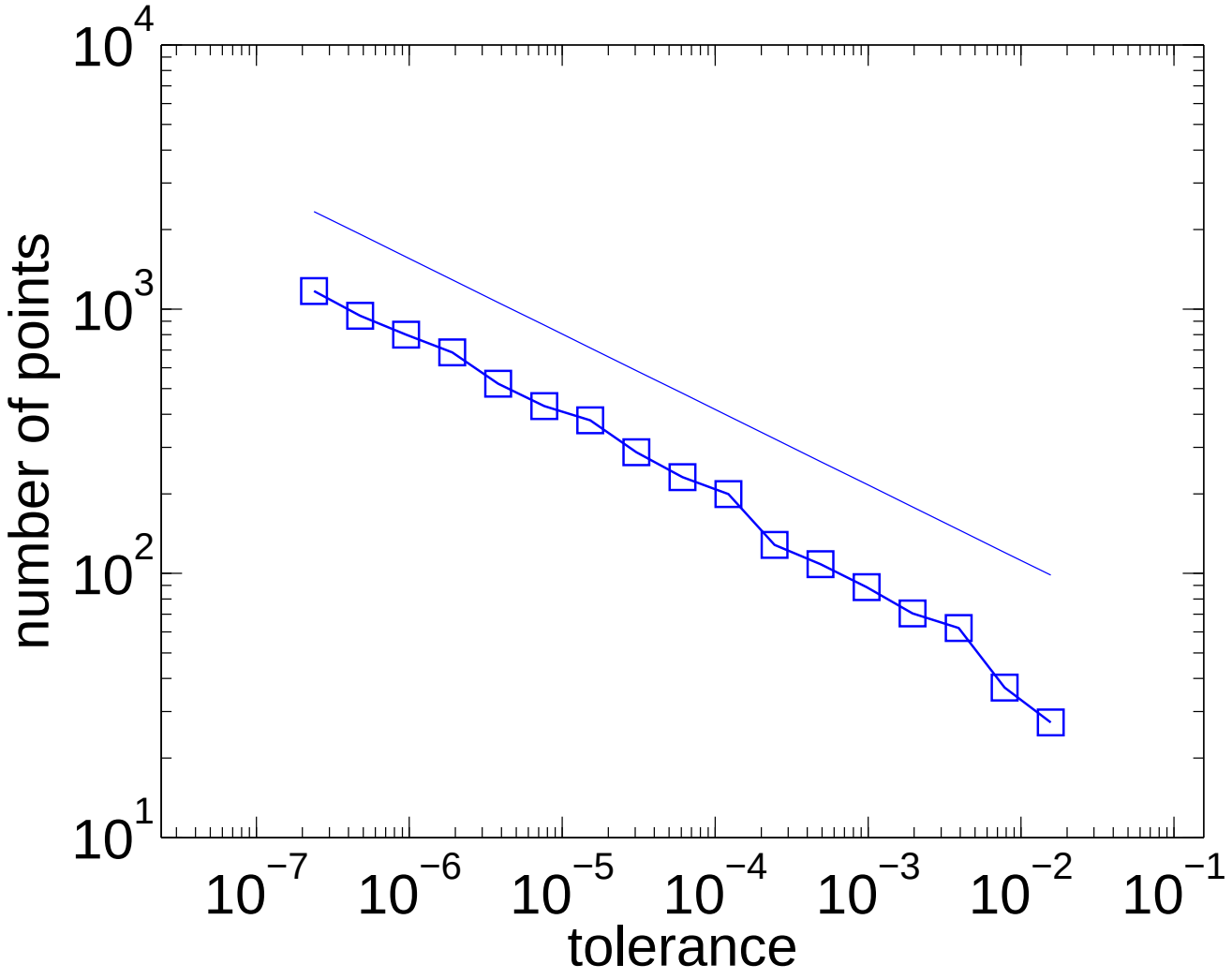
- achieved accuracy (for prescribed tolerances)
- long time computation

Step size for time evolution: 10^{-2}.

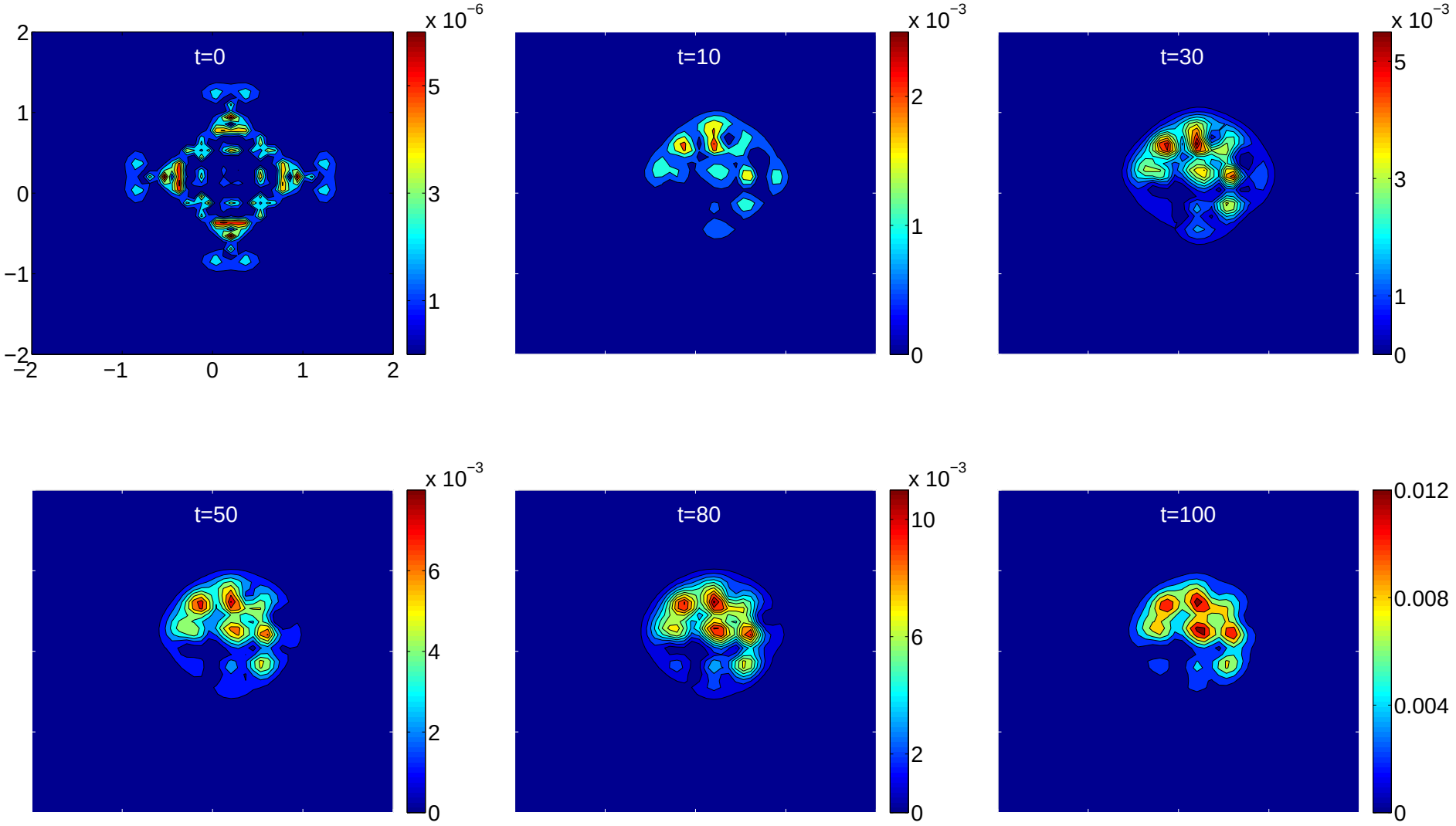
Error at $T = 1$


error
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1}
tolerance

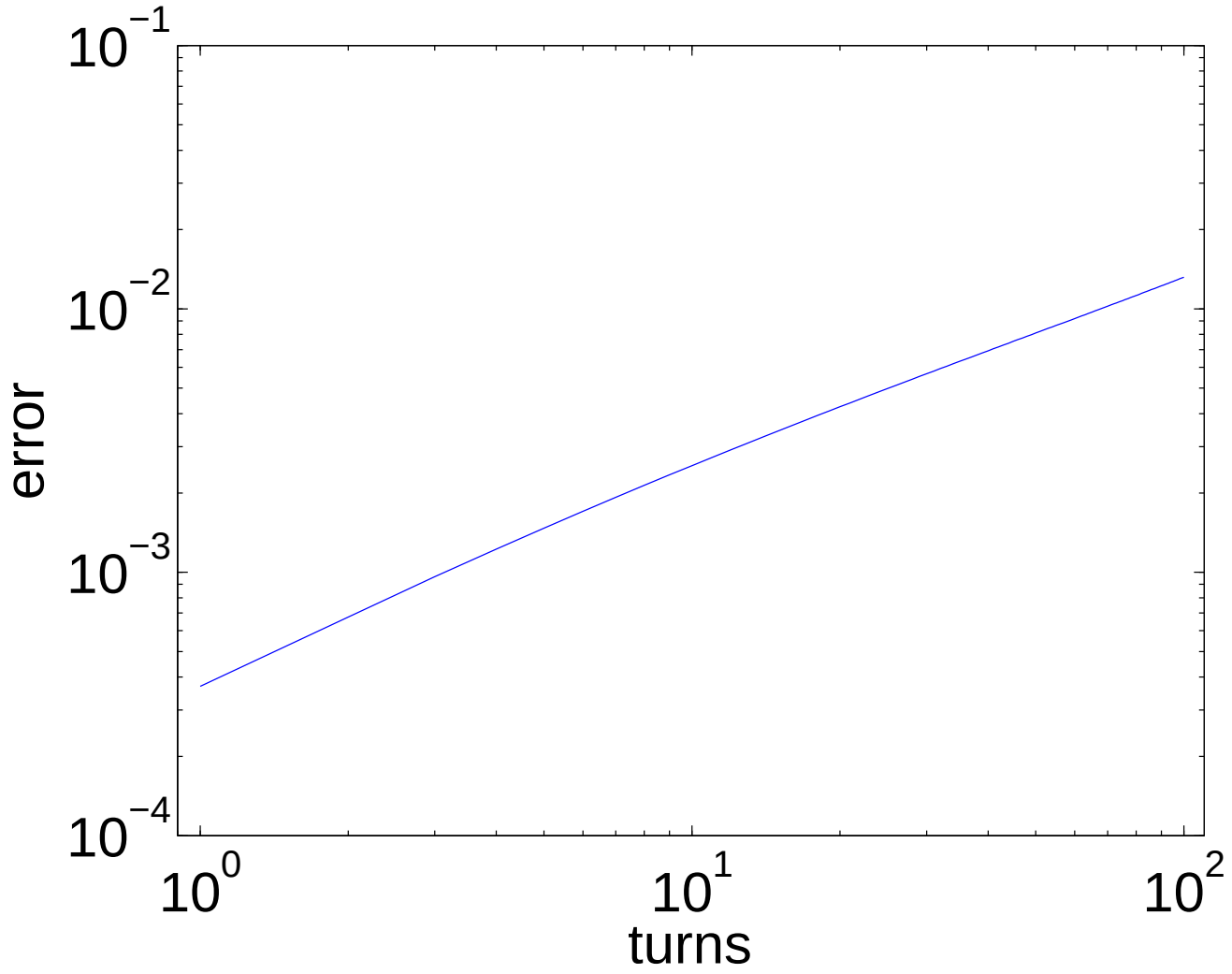
Required number of basis functions



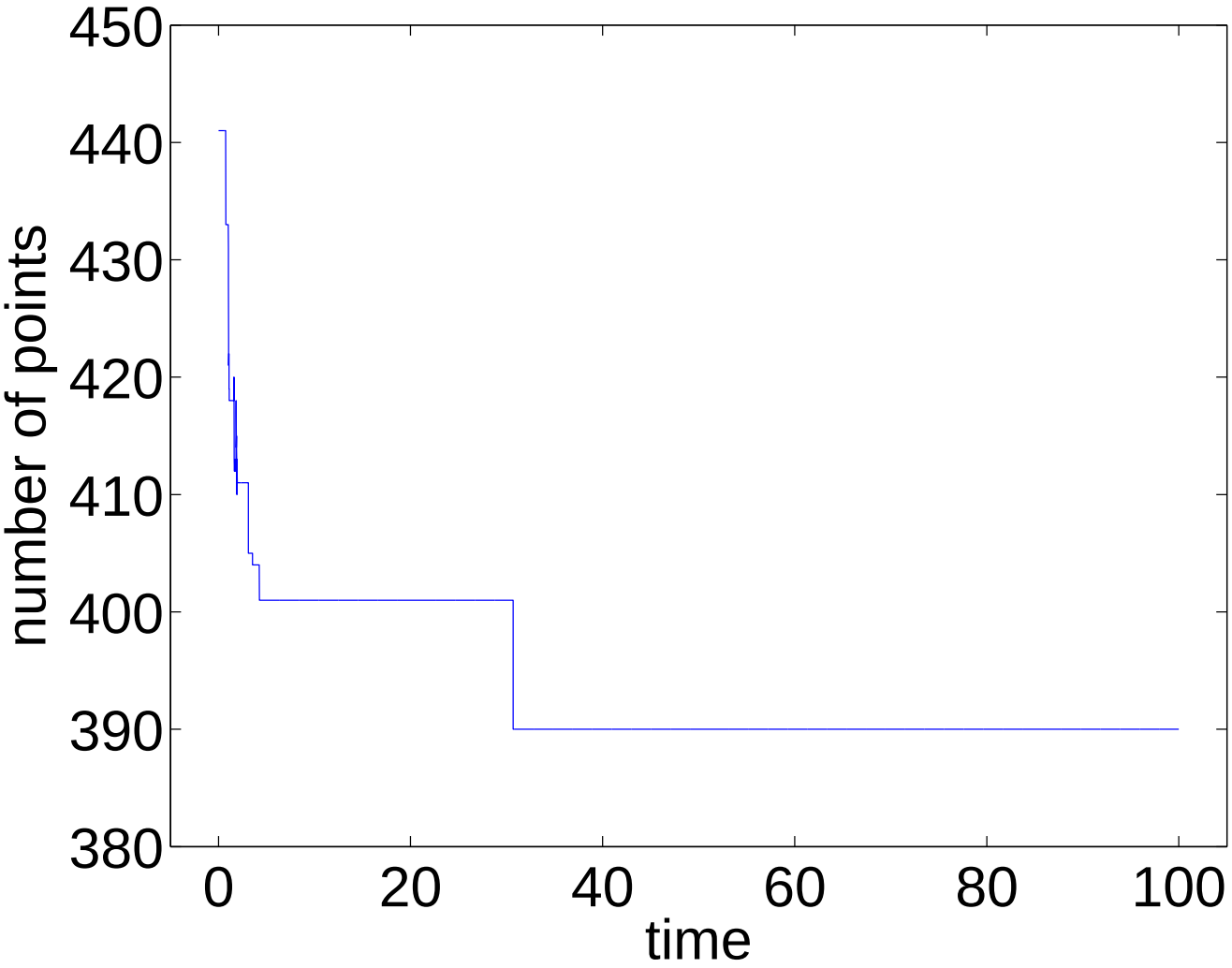
Long time computation (100 turns), tolerance = 10^{-5}



Global error



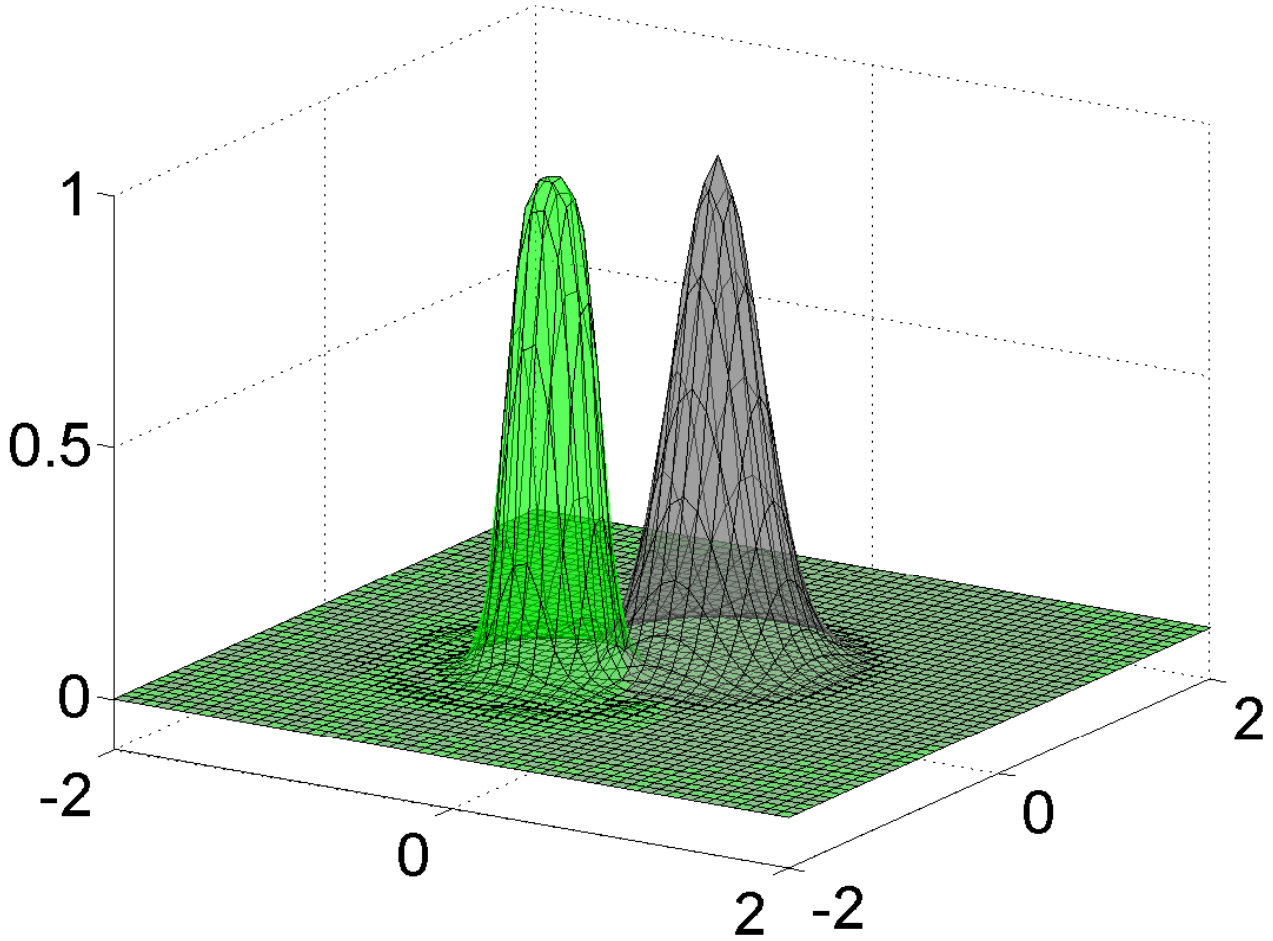
Number of basis functions



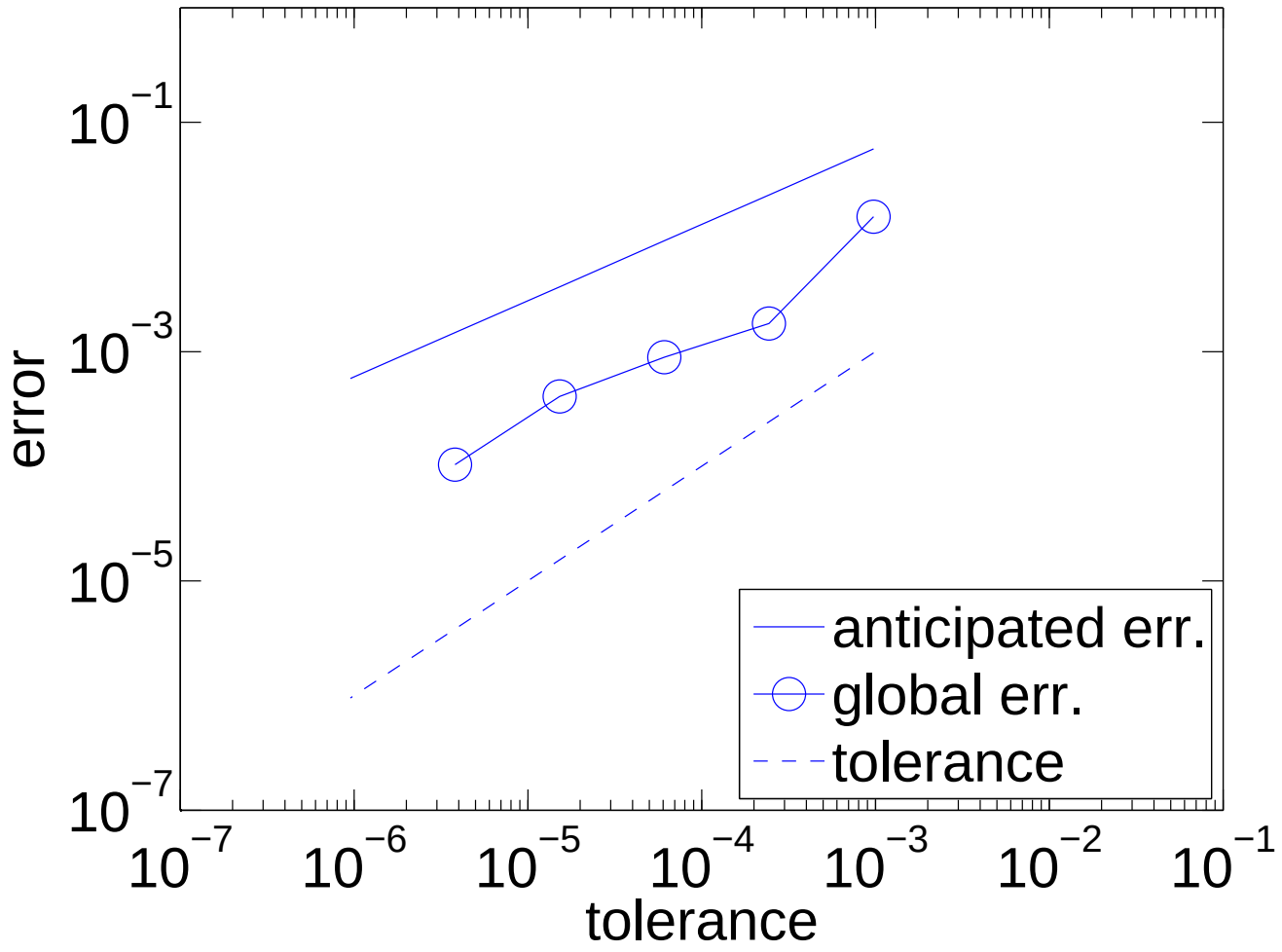
Example2: Consider the semilinear advection-diffusion-reaction problem

$$\partial_t u = 1/20\left(\partial_{xx} u + \partial_{yy} u\right) + 6\left(\partial_x u + \partial_y u\right) + 70u\left(u - 1/2\right)\left(1 - u\right)$$

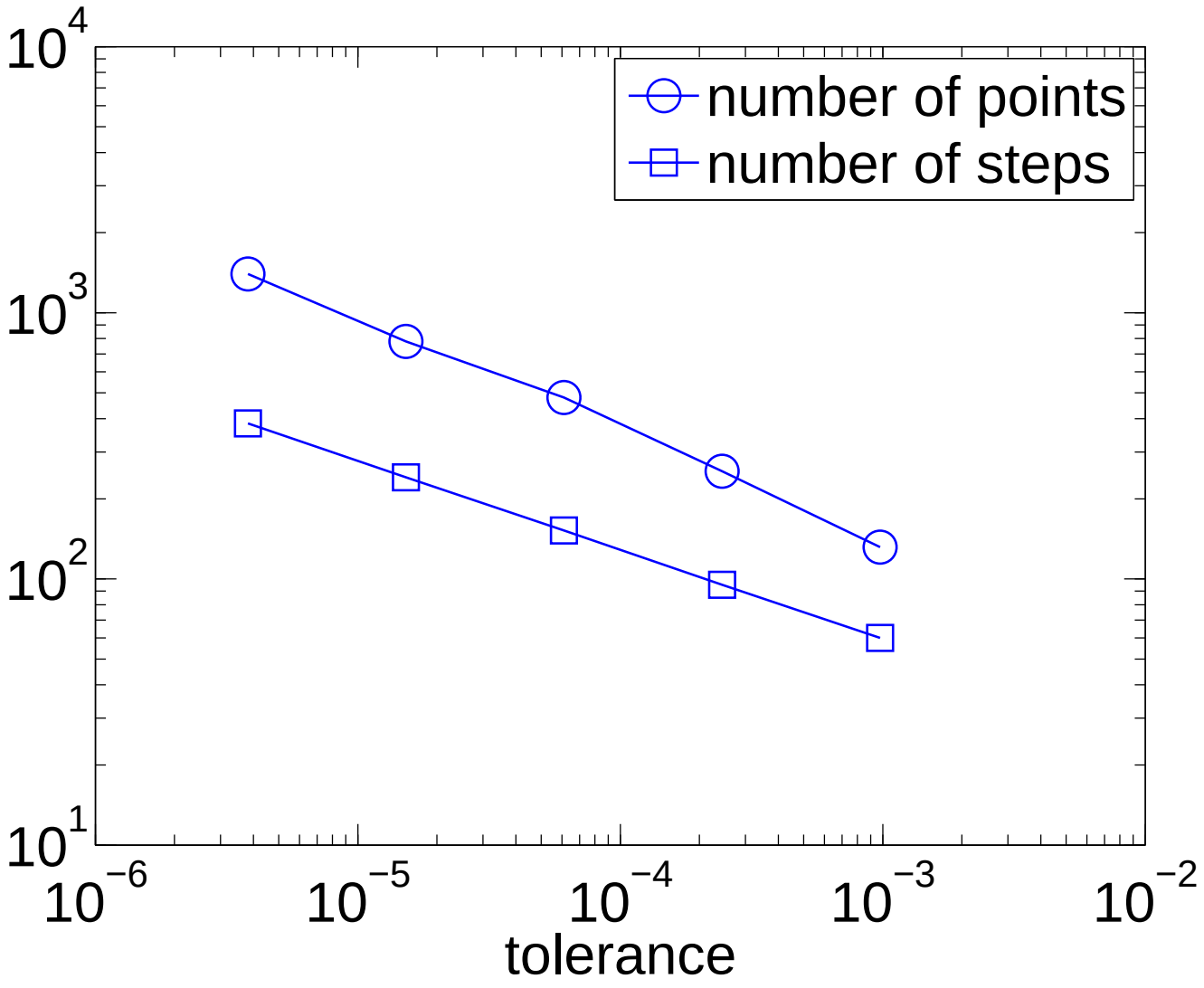
Integrated with exponential Rosenbrock 3(2) up to $T = 0.1$.


1
0.5
0
-2
0
2
-2
0
2

Global error at $T = 0.1$



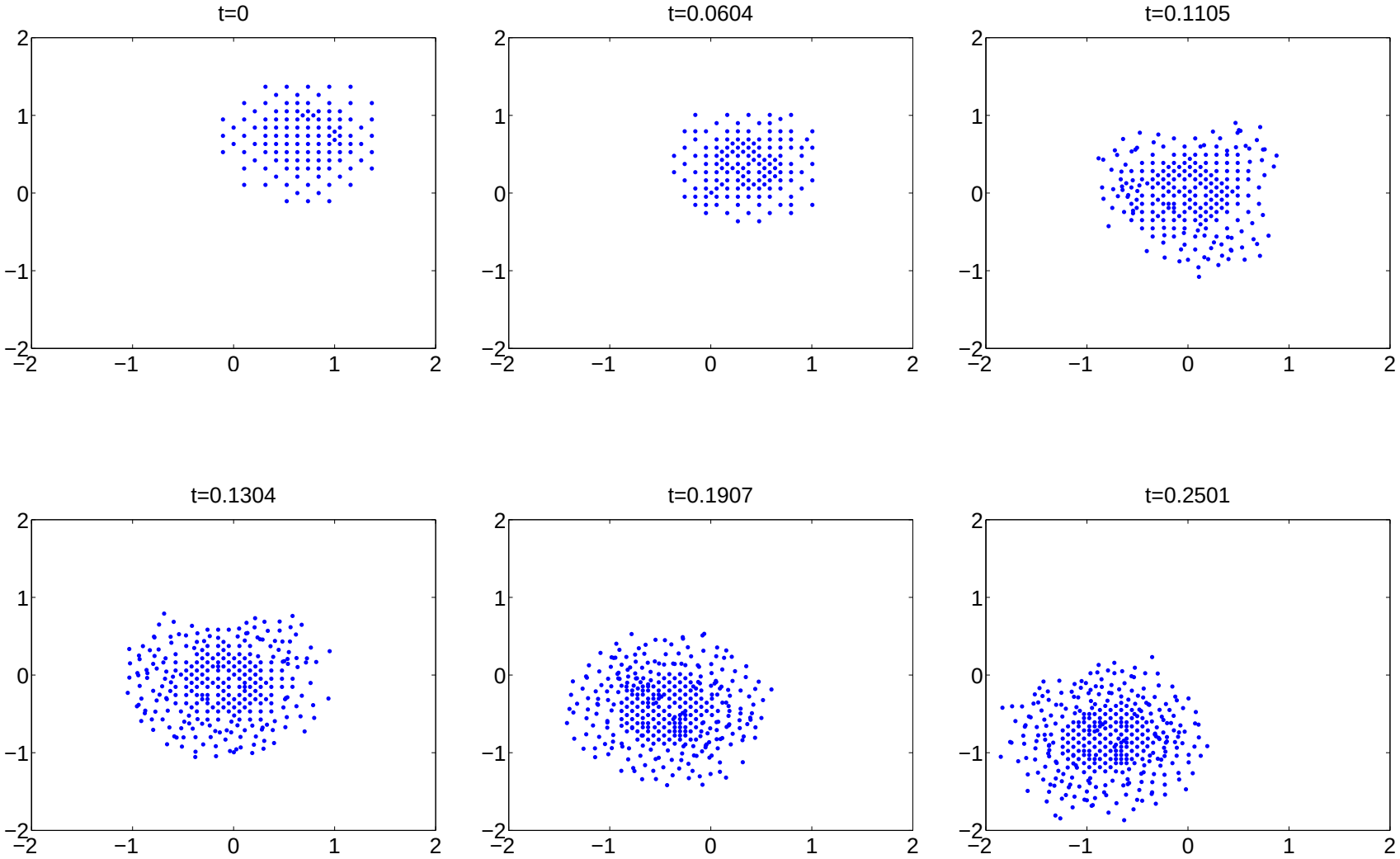
Number of time steps and number of interpolation points

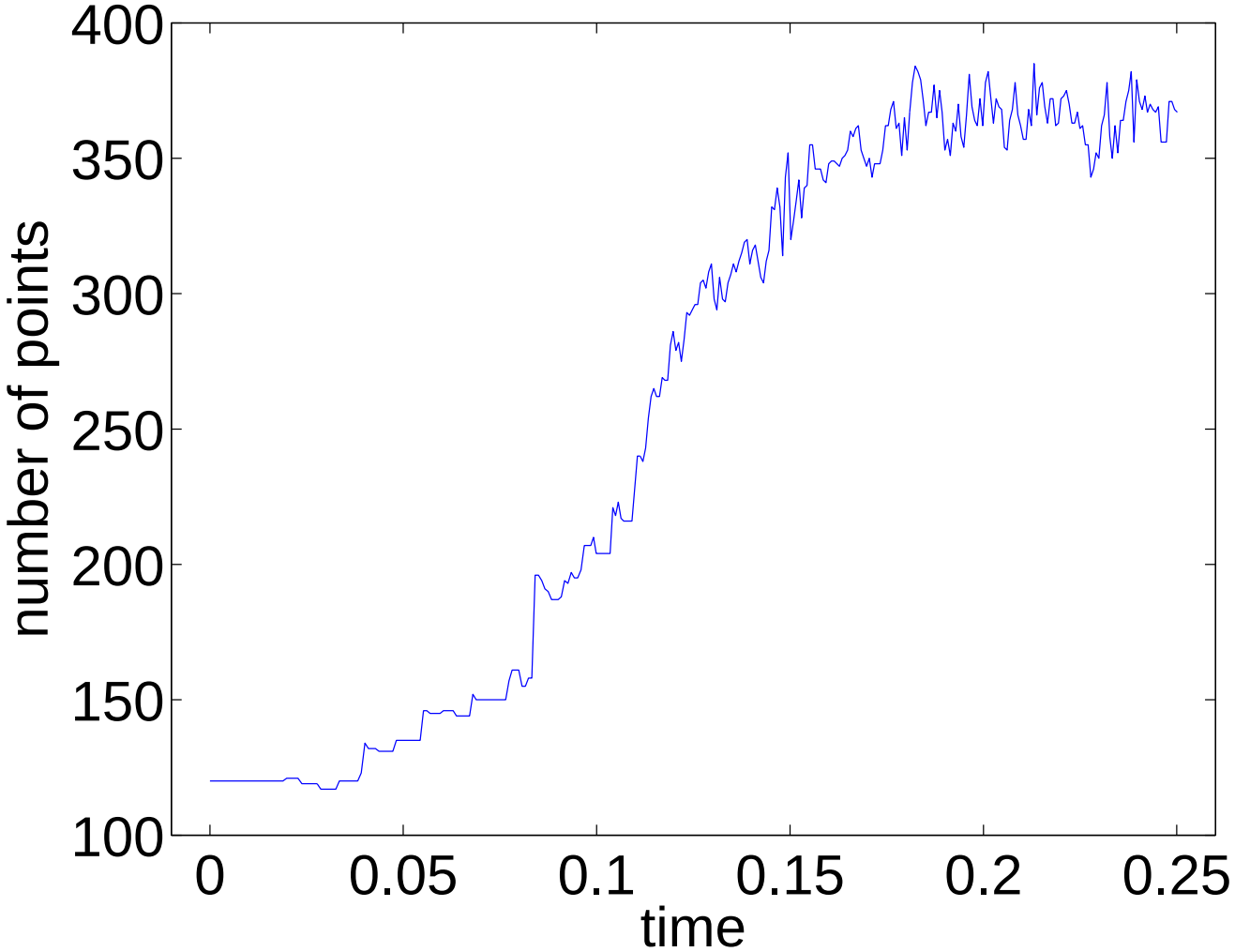


Development of interpolation point set

Integrate advection-diffusion-reaction problem up to $T = 0.3$, with tolerance $5 \cdot 10^{-4}$

Development of interpolation point set




400
350
300
250
200
150
100
number of points
0
0.05
0.1
0.15
0.2
0.25
time

Outlook

- efficient implementation
- error analysis
- solve real life problems