Forschungsseminar: Numerik für Stochastische Modelle, Humboldt Universität zu Berlin, 6ter Mai, 2009
Vortrag: Iterative Splitting Methods: Solver for Differential Equations

Jürgen Geiser, Humboldt Universität zu Berlin, Germany

Outline of the talk

0) Introduction: Iterative splitting method
1) Analysis: Consistency and Stability
2) Extention to unbounded, timedependent, nonlinear operators
3) Numerical Experiments

Introduction

Iterative Operator splitting methods are decomposition methods, which are in the class of numerical solvers.

- Decomposition methods are used to decouple differential equations into simpler and faster solvable equation parts.
- Iterative operator splitting methods are developped to gain higher order decomposition schemes by using an iterative algorithms.

Application Fields

Decomposition methods are used in various applications, where cheap and expensive computational parts of the differential equations can be considered :

1. Transport-Reaction Processes, see [Geiser2006] [Hundsdorfer, Verwer 2003](Physical splitting);
2. Hamiltonian Systems, see [McLachlan94], [Hairer, Lubich, Wanner 02], [Chin, Geiser 2009] (Hamitonian splitting).
3. Air pollutant models, see [Zlatev95] (Vectoriell Splitting)
4. Geoscience models, see [Roger et al 1999] (Wave action model)
5. Path integral Monte Carlo models, see [Chin, Boronat 08] (Quantum statistical models)

Underlying Splitting Ideas

We deal with a differential equation, e.g. Advection-Diffusion equation (Fluid dynamics, Gas dynamics):

$$\frac{\partial c(x,t)}{\partial t} = \Delta c \; - \; \mathbf{v} \nabla c, \; (x,t) \in \Omega \times [0,T] \qquad (1)$$

$$c(x,0) = c_0(x) \; x \in \Omega \,, c(x,t) = 0 \,, (x,t) \in \partial\Omega \times [0,T],$$

A standard time decomposition method (A-B splitting, Lie-Trotter splitting) is given as:

$$\frac{\partial c_1(x,t)}{\partial t} = \Delta c_1, \ (x,t) \in \Omega \times [0,T] \tag{2}$$

$$c_1(x,0) = c_0(x) \ x \in \Omega \ , c_1(x,t) = 0 \ , (x,t) \in \partial\Omega \times [0,T]$$

$$\frac{\partial c_2(x,t)}{\partial t} = -\mathbf{v} \cdot \nabla c_2, \ (x,t) \in \Omega \times [0,T] \tag{3}$$

$$c_2(x,0) = c_1(x,t) \ x \in \Omega \ , c_2(x,t) = 0 \ , (x,t) \in \partial\Omega \times [0,T],$$

Criteria for Splitting Methods as Solvers

1.) Conservation of physics, e.g. irrevesibility, vector-fields.
2.) Accurate method: higher order splitting scheme are useful.
3.) Fast implementation and separation to simpler computable parts, that can be done with standard method.
4.) Scheme that can be applied for time-dependent and spatial dependent operators.

Benefits of Iterative Splitting Methods

The traditional splitting methods (non-iterative) have some drawbacks:

- for non-commuting operators we may have a very large constant in the local splitting error which requires the use of unrealistically small splitting time step;
- within a full splitting step in one sub-interval the inner values aren't approximate to the solution of the original problem;
- splitting the original problem into the different sub-problems with one operator (i.e. neglect the other components) is physically questionable.

Iterative Operator Splitting Methods

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \ + \ Bc_{i-1}(t), \text{ with } \ c_i(t^n) = u^n, \quad (4)$$
$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) \ + \ Bc_{i+1}(t), \text{ with } \ c_{i+1}(t^n) = u^n, \quad (5)$$

where $c_0(t)$ is any fixed function for each iteration. (Here, as before, u^n denotes the known split approximation at the time level $t = t^n$.) The split approximation at the time-level $t = t^{n+1}$ is defined as $c_{\rm sp}^{n+1} = c_{2m+1}(t^{n+1})$. (Clearly, the functions $c_k(t)$ $(k = i-1, i, i+1)$ depend on the interval $[t^n, t^{n+1}]$, too, but, for the sake of simplicity, in our notation we omit the dependence on n.)

Analysis of the iterative method for bounded operators

For the application of ODEs, we consider bounded operators. Here the analysis is based on the Cauchy Problem:

$$\frac{dc(t)}{dt} = (A + B)c(t)\ , t \in (0, T]), \quad (6)$$

$$c(0) = c_0, \quad (7)$$

where A, B are bounded linear operators on a Banach space $\mathbf{X}$.

Practical Part:

$$\frac{dc(t)}{dt} = c(t) \, , t \in (0, T]), \tag{8}$$

$$c(0) = 1, \tag{9}$$

where the exact solution is given $u(t) = \exp(t)$.
We decompose as

$$\frac{dc_i(t)}{dt} = 0.5 c_i(t) + 0.5 c_{i-1} \, , t \in (0, T]), \tag{10}$$

$$c_i(0) = 1, \tag{11}$$

$$\frac{dc_{i+1}(t)}{dt} = 0.5 c_i(t) + 0.5 c_{i+1} \, , t \in (0, T]), \tag{12}$$

$$c_{i+1}(0) = 1, \tag{13}$$

where $c_0 = 1$ and $i = 1, 3, \ldots, 2m - 1$.

One can do the exact iterations as:

$$c_1(t) = 2\exp(\frac{t}{2}) - 1 \tag{14}$$

$$c_2(t) = t\exp(\frac{t}{2}) + 1 \tag{15}$$

$$c_3(t) = (2 + \frac{t^2}{4})\exp(\frac{t}{2}) - 1 \tag{16}$$

with local errors: $err_1 = \mathcal{O}(t^2)$, $err_2 = \mathcal{O}(t^2)$, $err_3 = \mathcal{O}(t^4)$, etc. Important are the initial estimate, otherwise the order is lowered.

Theoretical Part:

Theorem

The error $e_i(t) = c(t) - c_i(t)$ for the splitting methods is given as:

$$||e_i|| = K||B||\tau||e_{i-1}|| + O(\tau^2) \tag{17}$$

and hence

$$||e_{2m+1}|| = K_m||e_0||\tau^{2m} + O(\tau^{2m+1}), \tag{18}$$

where τ is the time-step, e_0 the initial error $e_0(t) = c(t) - c_0(t)$ and m the number of iteration-steps, K and K_m are constants, $||B||$ is the maximum norm of operator B. A and B are bounded, monotone operators, e.g. from a ODE system.

$$\partial_t e_i(t) = A e_i(t) + B e_{i-1}(t), \quad t \in (t^n, t^{n+1}], \\ e_i(t^n) = 0, \tag{19}$$

and

$$\partial_t e_{i+1}(t) = A e_i(t) + B e_{i+1}(t), \quad t \in (t^n, t^{n+1}], \\ e_{i+1}(t^n) = 0, \tag{20}$$

for $m = 0, 2, 4, \ldots$, with $e_0(0) = 0$ and $e_{-1}(t) = c(t)$.
By variation of constant formula, we have:

$$e_i(t) = \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s)) B e_{i-1}(s) \, ds, t \in (t^n, t^{n+1}], \tag{21}$$

which can be estimated as:

$$||e_i(t)|| \leq ||B|| ||e_{i-1}||_\infty \int_{t^n}^{t^{n+1}} || \exp(A(t^{n+1} - s))|| ds, \; t \in (t^n, t^{n+1}],$$

Since $(\exp(At))_{t \leq 0}$ is a C_0 semigroup, we have the growth estimate:

$$|| \exp(At)|| \leq K \exp(\omega t), \;\; t \geq 0, \tag{22}$$

and we have

$$||e_i(t)||_\infty \leq ||B|| \; \mathcal{O}(\tau)||e_{i-1}||_\infty, \;\; t \geq 0, \tag{23}$$

Same error estimates is deriven for (20) and we have:

$$||e_{i+1}(t)||_\infty \leq ||A|| \ \mathcal{O}(\tau)||e_i||_\infty, \ t \geq 0, \tag{24}$$

By induction we obtain:

$$||e_m(t)||_\infty \leq \mathcal{O}(\tau^m)||e_0||_\infty, \ t \geq 0, \tag{25}$$

where m is the number of iterations.

Remark

The constant in the term $\mathcal{O}(\tau^m)$ can be estimated as
$||A||^{m/2}||B||^{m/2}$
for even m and
$||A||^{(m-1)/2+1}||B||^{(m-1)/2}$
for odd m.
So for stiff problems, the restriction to τ is a delicate problem, e.g. :
$\tau^m \leq \frac{1}{||A||^{m/2}||B||^{m/2}}$,
means for stiff operators $||A||, ||B|| >> 1$ the timestep τ is small.

Remark

Globally, one should have sufficient exact initial condition otherwise, iteration steps will not improve the results:

$$
\begin{aligned}
e_i(t^{n+1}) &= \exp(A\tau)e_i(t^n) \qquad (26)\\
&+ \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))Be_{i-1}(s)ds, \in (t^n, t^{n+1}],
\end{aligned}
$$

we have to obtain to have at least for all previous time-steps the same or higher order for the initial conditions.

$$||e_i(t)||_\infty \leq ||B||\mathcal{O}(\tau)||e_{i-1}(t)||_\infty + K||e_i(\tilde{t})||, \qquad (27)$$

$$t \in (t^n, t^{n+1}], \tilde{t} \in (t^{n-1}, t^n] \qquad (28)$$

$$||e_i(\tilde{t})||_\infty \leq \mathcal{O}(\tau)||e_{i-1}(\tilde{t})||_\infty \qquad (29)$$

Example:

$$\frac{du_1}{dt} = -\lambda_1 u_1 + \lambda_2 u_2, \tag{30}$$

$$\frac{du_2}{dt} = \lambda_1 u_1 - \lambda_2 u_2, \tag{31}$$

$$u_1(0) = u_{10}, \; u_2(0) = u_{20} \text{ (initial conditions)}, \tag{32}$$

where $\lambda_1 \in \mathbb{R}^+$ and $\lambda_2 \in \mathbb{R}^+$ are the decay factors and $u_{10}, u_{20} \in \mathbb{R}^+$. We consider the time interval $t \in [0, T]$.

We rewrite the equation system (30)–(32) in the operator notation, and end up with the following equations:

$$\frac{du}{dt} = Au + Bu, \quad (33)$$

$$u(0) = (u_{10}, u_{20})^T, \quad (34)$$

where $u(t) = (u_1(t), u_2(t))^T$ for $t \in [0, T]$.
Our split operators are

$$A = \begin{pmatrix} -\lambda_1 & \lambda_2 \\ 0 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 & 0 \\ \lambda_1 & -\lambda_2 \end{pmatrix}. \quad (35)$$

We chose such an example to have $AB \neq BA$, and therefore we have a splitting error of first order for the usual A-B splitting.

time part.	iter. steps	err_1 (2nd order)	err_2 (2nd order)	err_1 (4th order)	err_2 (4th order)
2	1	4.5321e-002	3.6077e-003	4.5321e-002	3.6077e-003
2	10	3.9664e-003	4.7396e-004	3.9664e-003	4.7397e-004
2	100	3.9204e-004	4.8078e-005	3.9204e-004	4.8083e-005
3	1	4.6126e-004	3.6077e-003	4.6126e-004	3.6077e-003
3	10	7.8129e-006	2.9285e-005	7.8069e-006	2.9289e-005
3	100	8.5988e-008	2.8270e-007	8.0050e-008	2.8682e-007
4	1	4.6126e-004	2.2459e-005	4.6126e-004	2.2464e-005
4	10	4.1883e-007	4.2629e-008	4.1321e-007	4.8154e-008
4	100	5.9521e-009	5.4846e-009	4.0839e-010	4.9968e-011
5	1	1.9096e-006	2.2459e-005	1.9040e-006	2.2464e-005
5	10	6.0151e-009	3.7052e-009	4.7929e-010	1.8295e-009
5	100	5.5356e-009	5.5354e-009	5.0404e-014	1.7830e-013
6	1	1.9096e-006	6.1224e-008	1.9040e-006	6.6759e-008
6	10	5.5528e-009	5.5336e-009	1.7198e-011	1.9820e-012
6	100	5.5355e-009	5.5355e-009	2.4425e-015	4.4409e-016

Table: Numerical results: $\lambda_1 = 0.25$, $\lambda_2 = 0.5$ on $t \in [0, 1]$, iterative splitting method and 2nd- and 4th-order RK method.

Extension to more complicate operators

- unbounded operators (spatial dependent operators)
- time-dependent operators
- nonlinear operators

Unbounded operators (e.g. Convection-diffusion equation)

$$\frac{\partial c}{\partial t} = \nabla D \nabla c - \mathbf{v} \cdot \nabla c, c \in C^k(H^s(\Omega) \times [0, T]), \quad (36)$$

$$c(x, 0) = f_0(x) \in C^k(H^s(\Omega)), \quad (37)$$

$$c(x, t) = f_1(x, t) \in C^k(H^s(\Omega) \times [0, T]), \quad (38)$$

Practical Part:

$$\frac{\partial c(x,t)}{\partial t} = \frac{\partial}{\partial x} c + \lambda c \, , (x,t) \in \Omega \times (0,T], \tag{39}$$
$$c(x,0) = 0,$$
$$c(x,t) = \exp(\lambda t)(\begin{array}{cc} 1 & t < x \leq t+1 \\ 0 & \mathit{else} \end{array}), \text{ on} \partial\Omega \times (0,T].$$

where the exact solutions is given

$$c(x,t) = \exp(\lambda t)(\begin{array}{cc} 1 & t < x \leq t+1 \\ 0 & \mathit{else} \end{array}). \tag{40}$$

where c is a shock function and therefore not differentiable in a strong formulation.

We decompose as

$$\frac{\partial c_i(x,t)}{\partial t} = \frac{\partial}{\partial x} c_i + \lambda c_{i-1}, \tag{41}$$

$$\frac{\partial c_{i+1}(x,t)}{\partial t} = \frac{\partial}{\partial x} c_i + \lambda c_{i+1}, \tag{42}$$

where $c_0 = 0$ and $i = 1, 3, \ldots, 2m-1$.

One can do the exact iterations as:

$$c_1(x,t) = \begin{pmatrix} 1 & t < x \leq t+1 \\ 0 & else \end{pmatrix} \tag{43}$$

but in the second iteration we have some problems:

$$\begin{aligned} c_2(x,t) \quad & = \exp(\lambda t) \\ & + \int_0^t \exp(\lambda(t-s)) \frac{\partial}{\partial x} \begin{pmatrix} 1 & s < x \leq s+1 \\ 0 & else \end{pmatrix} \, ds \end{aligned} \tag{44}$$

We have not sufficient regularity to the $c_1(x,t)$ solution.

Ideas to solve the problem:

- Smoother initial functions
- Discretization methods with implicit and explicit schemes to balance the time and spatial discretisation
- Smooth with the numerical time integration

Stability and Conservation of the differentiability

In the discrete case we can balance the loose of regulatity. We assume the two stages for the iterative method and discretised with a θ-method:

$$\begin{aligned} \overline{c}_{i+1}^{n+1} &= c_i^n + \tau(1-\theta_1)(A(c_{i+1}^n) + B(c_i^n)) \\ &+ \tau\theta_1(A(\overline{c}_{i+1}^{n+1}) + B(c_i^{n+1})) \ , \end{aligned} \tag{45}$$

$$\begin{aligned} c_{i+1}^{n+1} &= c_{i+1}^n + \tau(1-\theta_2)(A(c_{i+1}^n) + B(c_{i+1}^n)) \\ &+ \tau\theta_2(A(\overline{c}_{i+1}^{n+1}) + B(c_{i+1}^{n+1})) \ , \end{aligned} \tag{46}$$

where $c_i^n = c_{i+1}^n = c^n$ and the initialisation with $c_0^{n+1} = c^n$

For the linear system we denote $Z_1 = \tau A$ and $Z_2 = \tau B$ and we set $\theta_1 = \theta_2$.
We get the following stability equation, cf. [Hundsdorfer 2005] and for $\theta = 1/2$: We compute the first iteration with $i = 1$ and get the equation

$$\begin{aligned} c_1^{n+1} &= (I + (I - 1/2Z_2)^{-1}(I - 1/2Z_1)^{-1}(Z_1 + Z_2)c^n \,, \quad (47) \\ &= (I - 1/2Z_2)^{-1}((I - 1/2Z_2) + (I - 1/2Z_1)^{-1}(Z_1 + Z_2))c^n \\ &= (I - 1/2Z_2)^{-1}(I - 1/2Z_1)^{-1}(I + 1/2Z_1)(I + 1/2Z_2)c^n \\ c_1^{n+1} &= R_1(Z1, Z_2)c^n \end{aligned}$$

To improve this method we suggest to do a prestepping for c_0^n, which means that we define c_0^n from the known value c^n with a suitably chosen stable method. Namely, we suggest the following algorithm.

Hence, we will get:

$$c_1^{n+1} = R_1(Z1, Z_2)R_2(Z_2)c^{n-1/2} \quad (48)$$

$$= (I - 1/2Z_2)^{-1}(I - 1/2Z_1)^{-1}(I + 1/2Z_1)(I + 1/2Z_2) \quad (49)$$

$$(I - 1/2Z_2)^{-1}(I - 1/2Z_1)^{-1}c^{n-1/2}$$

$$= R_{impl.Euler}(1/2Z_2)R_{CN}(Z_1)R_{CN}(Z_2)R_{impl.Euler}(1/2Z_1)c^{n-1/2}$$

where $R_{impl.Euler}$ and R_{CN} are the stability function of implicit Euler and Crank-Nicolson method. So we can stabilise the scheme with a prestep $1/2\tau$ that is based on an implicit method, with the initial value $c^{n-1/2}$.
Proof is submitted to Elsevier Nov. 2007.

Extension of Splitting Methods

Methods of decomposition

- Time- and Spatial-decomposition methods:

Contribution with the decomposition

- Decoupling the time-scales, space-scales. (Reduce the stiffness in single operators).
- Decoupling the multi-physics. (Reduce the unphysical behaviour with best choise of discretization and solver methods)
- Time-adaptivity, Space-adaptivity. (Efficiency and accuracy in computational)
- Parallelization in Time and Space. (Reduction of computational time)

Results : More efficient and fast algorithms with high accuracy,

Ideas for Nonlinear Splitting Methods

The iterative operator-splitting method is used as a fixed-point scheme or with embedded Newton's method to linearize the nonlinear operators.
We concentrate again on nonlinear differential equations of the form

$$\frac{du}{dt} = A(u(t))u(t) + B(u(t))u(t), \text{ with } u(t^n) = u^n, \tag{50}$$

where $A(u), B(u)$ are matrices with nonlinear entries and densely defined, where we assume that the entries involve the spatial derivatives of c.

Iterative operator-splitting method as fixed-point scheme

We split our nonlinear differential equation (65) by applying

$$\frac{du_i(t)}{dt} = A(u_{i-1}(t))u_i(t) \; + \; B(u_{i-1}(t))u_{i-1}(t), \qquad (51)$$
$$\text{with } \; u_i(t^n) = c^n,$$
$$\frac{du_{i+1}(t)}{dt} = A(u_{i-1}(t))u_i(t) \; + \; B(u_{i-1}(t))u_{i+1}(t), \qquad (52)$$
$$\text{with } \; u_{i+1}(t^n) = c^n,$$

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \ldots, 2m+1$. $u_0(t) = c_n$ is the starting solution, where we assume that the solution c^{n+1} is near c^n, or $u_0(t) = 0$. So we have to solve the local fixed-point problem. c^n is the known split approximation at time level $t = t^n$.

Operator-splitting method with embedded Jacobian Newton iterative method

The Newton's method is used to solve the nonlinear parts of the iterative operator-splitting method, see the linearization techniques in [Kelley2003],Karlsson1997]. We apply the iterative operator-splitting method and obtain:

$$
\begin{aligned}
& F_1(u_i) = \partial_t u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0, \\
& \text{with} \quad u_i(t^n) = c^n, \\
& F_2(u_{i+1}) = \partial_t u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0, \\
& \text{with} \quad u_{i+1}(t^n) = c^n,
\end{aligned}
$$

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \ldots, 2m+1$. $c_0(t) = 0$ is the starting solution and c^n is the known split approximation at time level $t = t^n$. The results of the methods are $c(t^{n+1}) = u_{2m+2}(t^{n+1})$.

The splitting method with embedded Newton's method is given as

$$u_i^{(k+1)} = u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i-1}^{(k)})u_{i-1}^{(k)}),$$

$$\text{with } D(F_1(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}} u_i^{(k)}),$$

$$\text{and } k = 0, 1, 2, \ldots, K, \text{ with } \quad u_i(t^n) = c^n,$$

$$u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D(F_2(u_{i+1}^{(l)}))^{-1}(\partial_t u_{i+1}^{(l)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i+1}^{(k)})u_{i+1}^{(k)}),$$

$$\text{with } D(F_2(u_{i+1}^{(l)})) = -(B(u_{i+1}^{(l)}) + \frac{\partial B(u_{i+1}^{(l)})}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}),$$

$$\text{and } l = 0, 1, 2, \ldots, L, \text{ with } \quad u_{i+1}(t^n) = c^n.$$

Consistency and Stability Proofs :

1.) Bounded case : First linearisation, then application of the linear theory.

2.) Unbounded Case : Linearisation and Discretisation in time and space, then application of the discrete theory.

Transport-Reaction Models

First example : 2D Diffusion-Reaction equation

We deal with the time dependent 2-D equation:

$$\partial_t u(x,y,t) = u_{xx} + u_{yy} - 4(1+y^2)e^{-t}e^{x+y^2} \quad (53)$$

$$u(x,y,0) = e^{x+y^2} \text{ in } \Omega = [-1,1] \times [-1,1] \quad (54)$$

$$u(x,y,t) = e^{-t}e^{x+y^2} \text{ on} \partial\Omega \quad (55)$$

with exact solution

$$u(x,y,t) = e^{-t}e^{x+y^2} \quad (56)$$

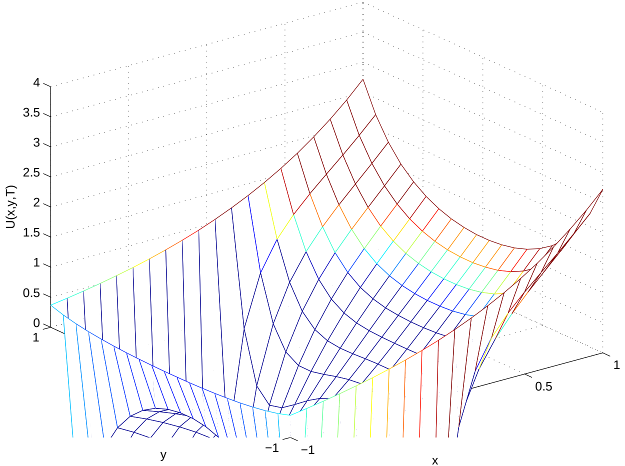
We choose the time Itervall $[0,1]$ and again use Finite Differences for the space with $\Delta x = 2/19$.

We define our operators by splitting the plane into two halfs.

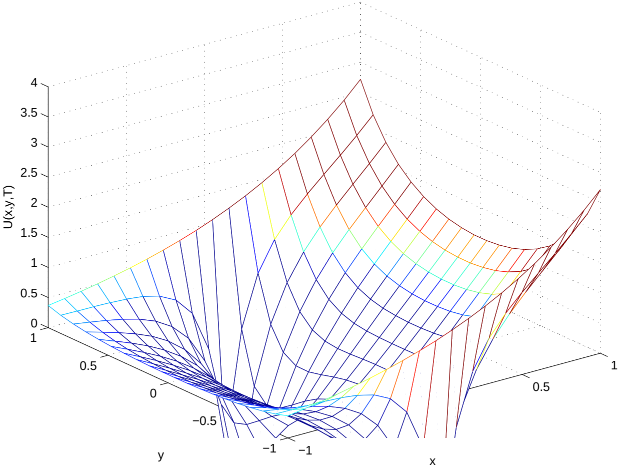
Iterative steps	Number of splitting-partitions	Max-error
1	1	2.7183e+000
2	1	8.2836e+000
3	1	3.8714e+000
4	1	2.5147e+000
5	1	1.8295e+000
10	1	6.8750e-001
15	1	2.5764e-001
20	1	8.7259e-002
25	1	2.5816e-002
30	1	5.3147e-003
35	1	2.8774e-003

Table: Numerical results for the first example with the Iterative Operator Splitting method and BDF3 with $h = 10^{-1}$

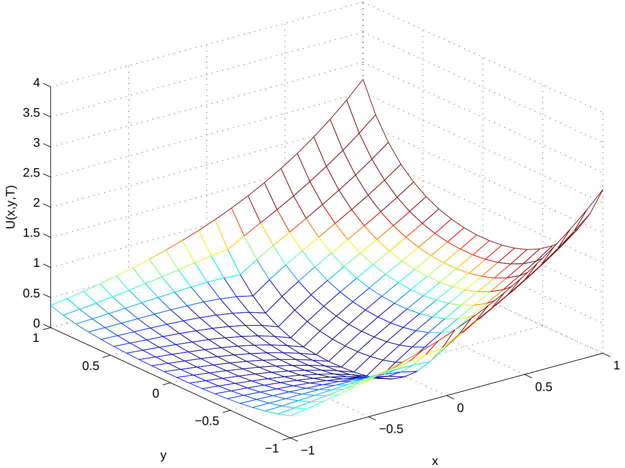
Relaxation of the model



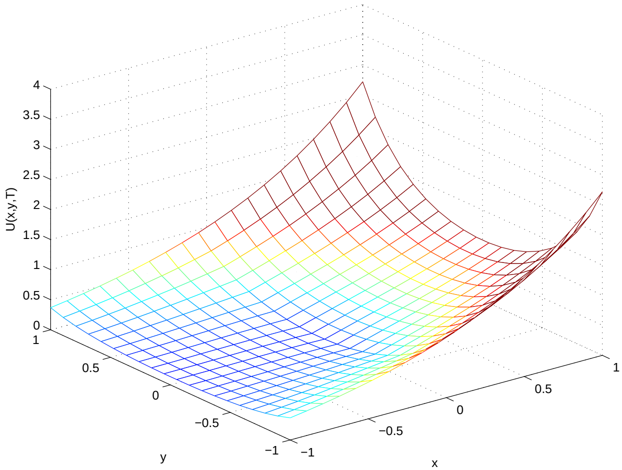
Relaxation of the model



Relaxation of the model



Relaxation of the model



Test example 2: Burgers equation

We deal with a 2D example where we can derive an analytical solution.

$$\partial_t u = -u\partial_x u - u\partial_y u + \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t), \quad (57)$$

$$(x, y, t) \in \Omega \times [0, T]$$

$$u(x, y, 0) = u_{\mathrm{ana}}(x, y, 0), \ (x, y) \in \Omega \quad (58)$$

$$\text{with } u(x, y, t) = u_{\mathrm{ana}}(x, y, t) \text{ on} \partial\Omega \times [0, T], \quad (59)$$

where $\Omega = [0, 1] \times [0, 1]$, $T = 1.25$, and μ is the viscosity.
The analytical solution is given as

$$u_{\mathrm{ana}}(x, y, t) = (1 + \exp(\frac{x + y - t}{2\mu}))^{-1}, \quad (60)$$

where $f(x, y, t) = 0$.

The operators are given as:
$A(u)u = -u\partial_x u - u\partial_y u$, hence $A(u) = -u\partial_x - u\partial_y$ (the nonlinear operator),
$Bu = \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t)$ (the linear operator).
We apply the nonlinear Algorithm 66 to the first equation and obtain
$A(u_{i-1})u_i = -u_{i-i}\partial_x u_i - u_{i-i}\partial_y u_i$ and
$Bu_{i-1} = \mu(\partial_{xx} + \partial_{yy})u_{i-1} + f$,
and we obtain linear operators, because u_{i-1} is known from the previous time step.
In the second equation we obtain by using Algorithm 67:
$A(u_{i-1})u_i = -u_{i-i}\partial_x u_i - u_{i-i}\partial_y u_i$ and
$Bu_{i+1} = \mu(\partial_{xx} + \partial_{yy})u_{i+1} + f$,
and we have also linear operators.

We have the following results, see Tables 3, for different steps in time and space and different viscosities.

$\Delta x = \Delta y$	Δt	err_{L_1}	$\mathrm{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/10	1/10	0.0549	0.1867		
1/20	1/10	0.0468	0.1599	0.2303	0.2234
1/40	1/10	0.0418	0.1431	0.1630	0.1608
1/10	1/20	0.0447	0.1626		
1/20	1/20	0.0331	0.1215	0.4353	0.4210
1/40	1/20	0.0262	0.0943	0.3352	0.3645
1/10	1/40	0.0405	0.1551		
1/20	1/40	0.0265	0.1040	0.6108	0.5768
1/40	1/40	0.0181	0.0695	0.5517	0.5804

Table: Numerical results for the Burgers equation with viscosity $\mu = 0.05$, initial condition $u_0(t) = c_n$, and two iterations per time step.

Figure 3 presents the profile of the 2D nonlinear Burgers equation.

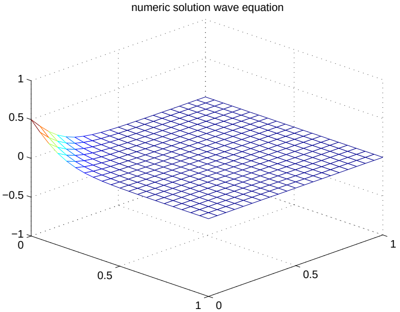


Figure: Burgers equation at initial time $t = 0.0$ for viscosity $\mu = 0.05$.

Figure 3 presents the profile of the 2D nonlinear Burgers equation.

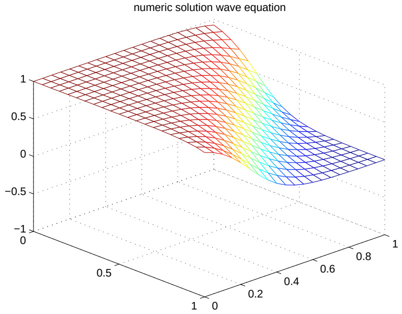


Figure: Burgers equation at end time $t = 1.25$ for viscosity $\mu = 0.05$.

Transport-Reaction Models

Second example : CVD-Modell (2D Convection-Diffusion-Reaction equation)

We deal with a gas-tranport model

For this model we can assume a continuum flow, and the fluid equations can be treated with a Navier-Stokes or especially with a reaction-diffusion equation.

$$\frac{\partial}{\partial t} c + \nabla \cdot F - R_g = 0, \text{ in } \Omega \times [0, T] \quad (61)$$

$$F = \mathbf{v} c - D \nabla c,$$

$$c(x, t) = c_0(x), \text{ on } \Omega, \quad (62)$$

$$c(x, t) = c_1(x, t), \text{ on } \partial\Omega \times [0, T], \quad (63)$$

where c is the molar concentration and F the flux of the species. v is the velocity, D is the diffusivity matrix and R_g is

Transport-Reaction Models

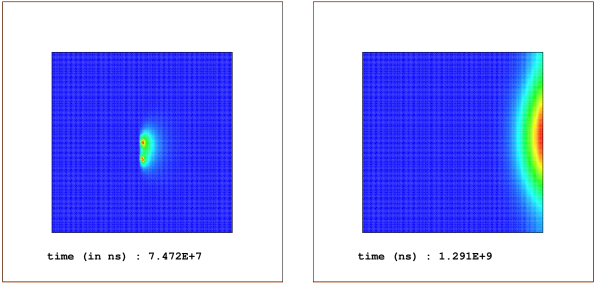


Figure: Source is moving in y direction with step 5 .

Transport-Reaction Models

In Figure 4 we present the deposition rates with a single source.

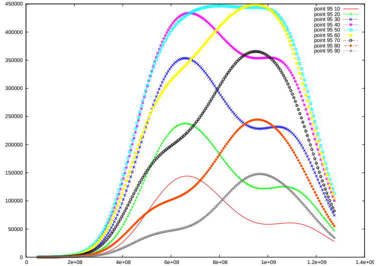


Figure: Source is moving in y direction with step 5 .

FutureWorks

Outview

1) Theory for time-dependent and nonlinear problems.
2) Applications in quantum statistics, e.g. pathway integrals.
3) Improved decomposition for deposition process, e.g., kinetic (exact) and flux processes (numerical)