COMSOL CONFERENCE ROTTERDAM 2013

Multigrid Implementation in COMSOL Multiphysics - Comparison of Theory and Practice

Wolfgang Joppich
University of Applied Sciences Bonn-Rhein-Sieg
Grantham Allee 20
53757 Sankt Augustin

How can you easily destroy your modeling effort?

Choose an inadequate solver or a badly parameterized one!

How can you avoid this?

You need knowledge on numerical methods – at least confidence in implemented solver techniques.

How can you obtain this?

Let the results for accepted model problems convince you about properties of a particular solver. Here: **geometric MG.**

If you like this …

„COMSOL solver tuning“ (May 2014, Berlin). More general, more solver.

wolfgang.joppich@h-brs.de

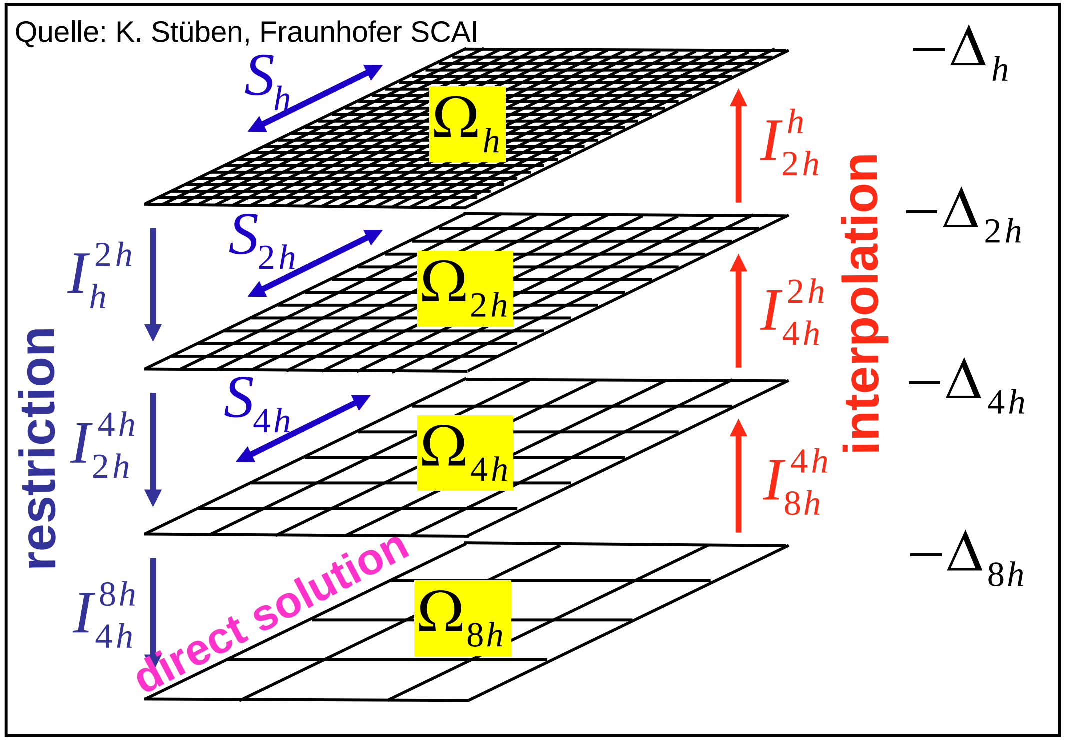
MG – a combination of **smoothing**

CGC

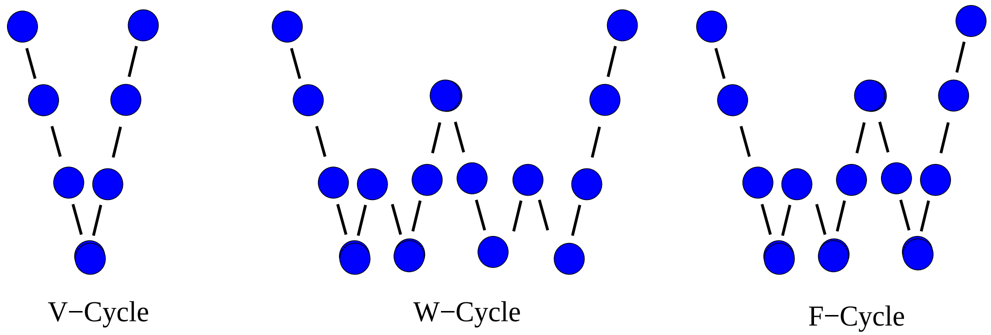
(1)	*pre-smoothing*	$\bar{w}_h^{(n)} := RELAX^{\nu_1}(w_h^{(n-1)}, L_h, f_h)$
(2)	*residual calculation*	$r_h^{(n)} := f_h - L_h \bar{w}_h^{(n)}$
(3)	*residual restriction*	$r_H^{(n)} := I_h^H r_h^{(n)}$
(4)	*exact solution of the coarse grid problem*	$L_H \tilde{e}_H^{(n)} = r_H^{(n)}$
(5)	*correction transfer*	$\tilde{e}_h^{(n)} := I_H^h \tilde{e}_H^{(n)}$
(6)	*correction*	$\tilde{w}_h^{(n)} := \bar{w}_h^{(n)} + \tilde{e}_h^{(n)}$
(7)	*post-smoothing*	$w_h^{(n)} = RELAX^{\nu_2}(\tilde{w}_h^{(n)}, L_h, f_h)$

step (4) recursively solved $\Rightarrow$ **multigrid**

wolfgang.joppich@h-brs.de



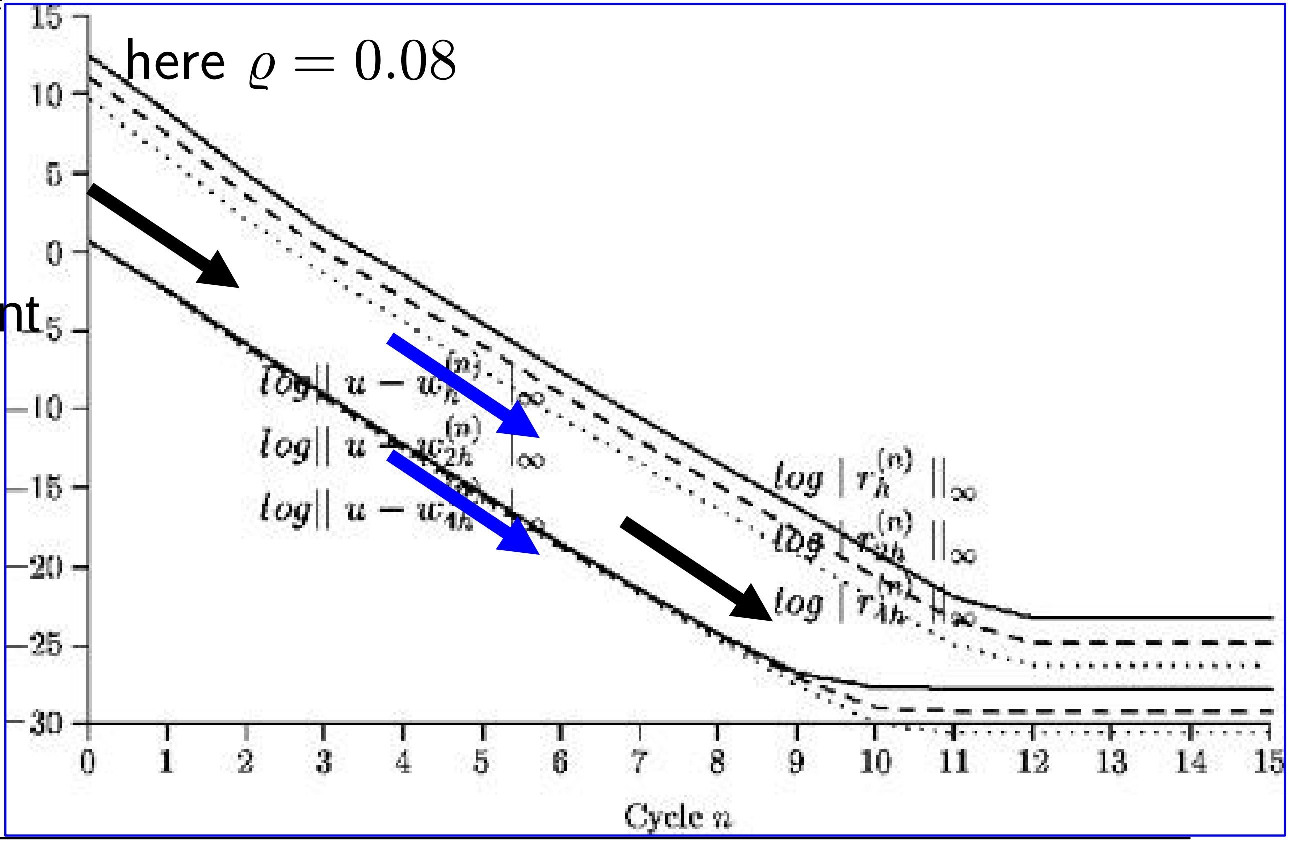
multigrid cycling



V-Cycle: **cheap,** fast
W-Cycle: **robust**, expensive, **theory**
F-Cycle: **robust, cheaper than W**

What would we like to have for a MG solver?

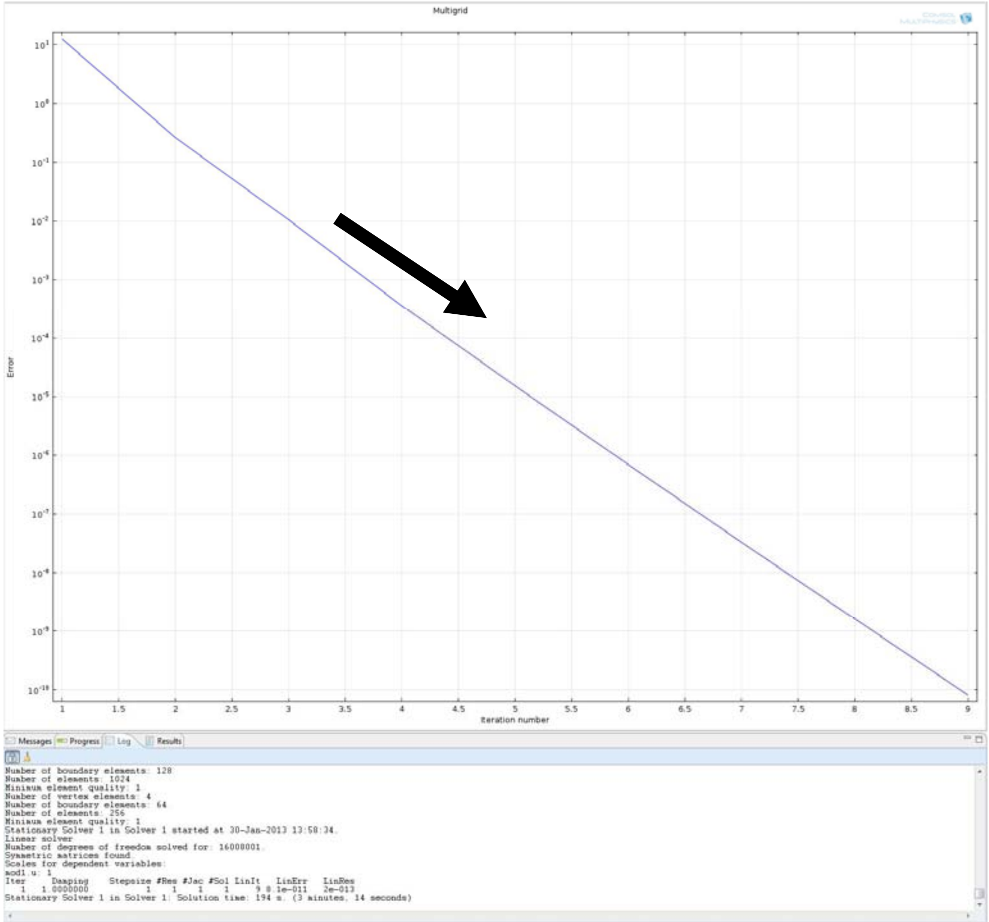
- steady convergence rate
  (both error and residual reduction)
  from the first cycle till the last one
- „h-independent“ convergence
- fast convergence
- convergence depends on the quality of smoothing
- linear complexity $O(N)$
- moderate memory requirement
- faster than other solver



Poisson equation: $\varepsilon_x \frac{\partial^2 u}{\partial x^2} + \varepsilon_y \frac{\partial^2 u}{\partial y^2} = f(x,y)$ mit $\varepsilon_x = 1.0, \varepsilon_y = 1.0$

„mapped mesh": **16.008.001** d.o.f.

solver	smoother	time [s]	no. of cycles	time per cycle	ϱ_{comsol}	v. mem. (GB)
MG-V(2,1)-9L	SOR	175	10	2.8	0.046	17.0
MG-F(2,1)-9L	SOR	181	9	3.9	0.040	15.6
MG-W(2,1)-9L	SOR	184	9	4.3	0.040	15.7
MG-V(2,1)-9L	SSOR	195	10	3.7	0.053	15.9
MG-F(2,1)-9L	SSOR	197	9	5.6	0.048	15.0
MG-W(2,1)-9L	SSOR	203	9	6.3	0.048	15.0
MG-V(2,1)-9L	Vanka	196	10	4.8	0.053	18.0
MG-F(2,1)-9L	Vanka	205	9	6.1	0.048	18.0
MG-W(2,1)-9L	Vanka	210	9	6.9	0.048	18.0


Multigrid
Error
Iteration number
Messages
Progress
Log
Results
Number of boundary elements: 128
Number of elements: 1024
Minimum element quality: 1
Number of vertex elements: 4
Number of boundary elements: 64
Number of elements: 256
Minimum element quality: 1
Stationary Solver 1 in Solver 1 started at 30-Jan-2013 13:58:34.
Linear solver
Number of degrees of freedom solved for: 16008001.
Symmetric matrices found.
Scales for dependent variables:
mod1.u: 1
Iter Damping Stepsize #Res #Jac #Sol LinIt LinErr LinRes
1 1.0000000 1 1 1 1 9 8.1e-011 2e-013
Stationary Solver 1 in Solver 1: Solution time: 194 s. (3 minutes, 14 seconds)

Poisson equation: $\varepsilon_x \frac{\partial^2 u}{\partial x^2} + \varepsilon_y \frac{\partial^2 u}{\partial y^2} = f(x, y$

„free triangular mesh“: **8.786.945 (8L), and 35.1**

What predicts theory?

the weighted Jacobi-method with relaxation factor 1.0 has no good smoothing property for the Poisson equation, best with 0.8

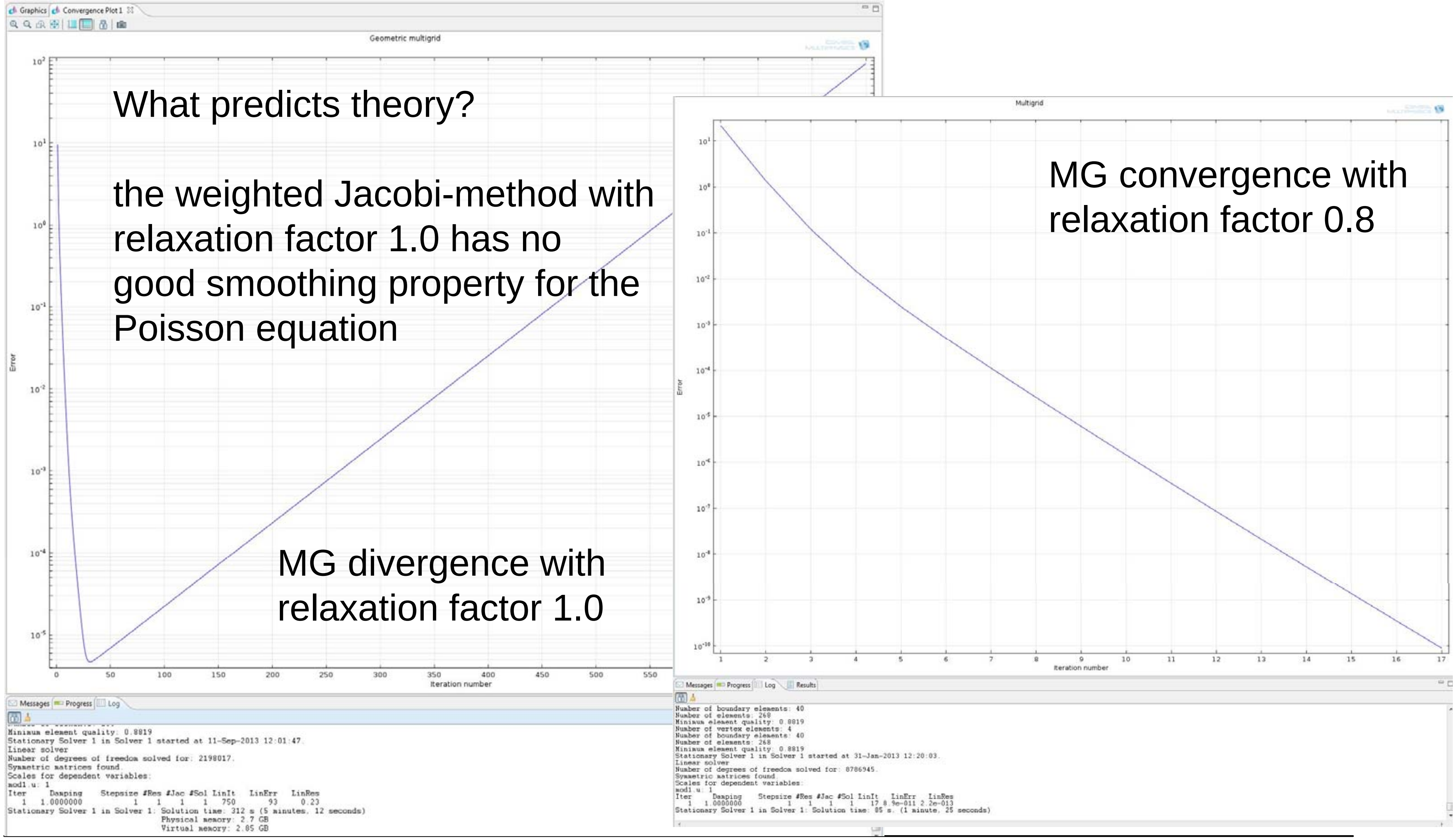
solver	smoother	total time (seconds)	no. cycl			
MG-W(2,1)-7L	Jacobi $\omega = 1.0$				divergence	
MG-V(2,1)-8L	Jacobi $\omega = 1.0$	84	29	1.1	0.402	8.7
MG-V(2,1)-8L	Jacobi $\omega = 0.8$	71	17	1.1	0.194	8.6
MG-F(2,1)-8L	Jacobi $\omega = 0.8$	77	17	1.4	0.192	8.6
MG-W(2,1)-8L	Jacobi $\omega = 0.8$	79	17	1.6	0.192	8.7
MG-V(2,1)-9L	Jacobi $\omega = 0.8$	596	17	8.4	0.193	30.7
MG-V(2,1)-9L	SOR	450	10	3.0	0.051	29.3

the 9L problem could not be solved by a direct solver

wolfgang.joppich@h-brs.de

What predicts theory?

the weighted Jacobi-method with relaxation factor 1.0 has no good smoothing property for the Poisson equation



MG divergence with relaxation factor 1.0

MG convergence with relaxation factor 0.8

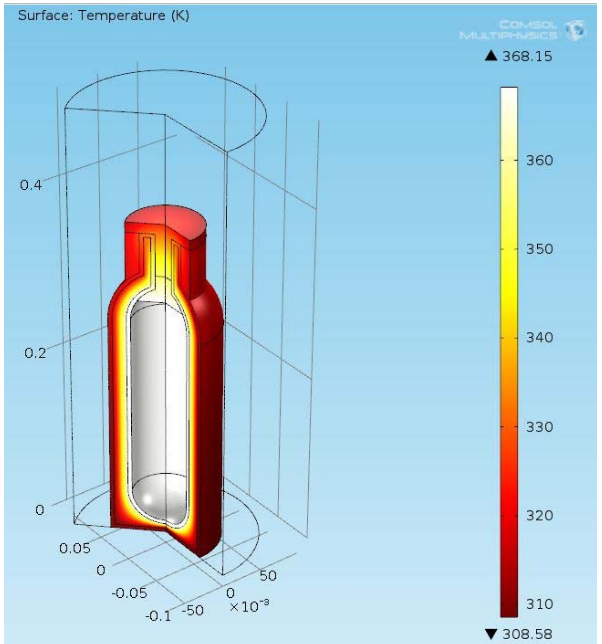
wolfgang.joppich@h-brs.de

Anisotropic Poisson equation: $\varepsilon_x \frac{\partial^2 u}{\partial x^2} + \varepsilon_y \frac{\partial^2 u}{\partial y^2} = f(x,y)$ with $\varepsilon_x = 0.01, \varepsilon_y = 1.0$

„free triangular mesh“: **8.786.945 (8L)** d.o.f.

solver	smoother	total time (seconds)	no. of cycles	time per cycle	ϱ_{comsol}	v. mem. (GB)
MG-V(2,1)-8L	SOR	391	298	1.14	0.915	8.4
MG-F(2,1)-8L	SOR	502	297	1.51	0.915	8.4
MG-W(2,1)-8L	SOR	544	297	1.65	0.915	8.4
MG-V(2,1)-8L	SSOR	407	198	1.79	0.877	8.3
MG-V(2,1)-8L	Vanka	450	198	2.00	0.877	9.7
MG-V(2,1)-8L	SORline	441	68	5.08	0.692	10.3

cooling of a thermos,
heat transfer, laminar flow,
free convection
stationary, nonlinear



d.o.f.	solver characteristics	total time (seconds)	v. mem. (GByte)
1.975.146	MG-V(2,1)-6L SOR	50	7.4
	Pardiso	83	11.1
	MUMPS	157	9.9
	Spooles	254	8.6
5.084.465	MG-V(2,1)-7L SOR	128	8.3
	Pardiso	220	19.2
	MUMPS	410	14.8
	Spooles	857	20.6
20.323.281	MG-V(2,1)-8L SOR	695	25.5
	Pardiso	cancelled	-
	MUMPS	cancelled	43.8
	Spooles	omitted	-

What has been observed?

- **numerical complexity of the cycle** is reflected well by the time per cycle
- **convergence speed** of F- and W-cycle are identical
  F- and W- converge faster than V-cycle
- MG with V(2,1)-cycle usually is the **fastest MG-solver** (time to solve)
- **steady convergence speed** for all cycles (first to last)
- convergence speed is **almost h-independent**
- **linear behavior** is (almost) given
- **moderate memory** requirements, especially when compared to direct solver
- Jacobi smoother reacts on **relaxation parameter** as known from theory
- MG without coloured relaxation pattern or block relaxation behaves as predicted
  for the **anisotropic Poisson** equation
- **MG beats all direct solver** – except for the anisotropic problem
  the direct solver could not solve the very large problems

MG implementation in COMSOL is reliable – use it

Literatur

Brandt, A., *Multigrid Techniques: 1984 Guide With Applications to Fluid Dynamics,* GMD-Studie No. 85, 1984

Hackbusch, W., *Multi-Grid Methods and Applications,* Springer, 1985

Wesseling, P., *An Introduction to Multigrid Methods,* John Wiley & Sons, Chichester, 1992

Trottenberg, U., Oosterlee, K., Schüller, A., *Multigrid,* Academic Press, 2001

Joppich, W., *Grundlagen der Mehrgittermethode*, Shaker 2011

Barrett, R., et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1994

Shewchuk, J. R., *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Carnegie Mellon University, 1994

Dahmen, W., Reusken, A., *Numerik für Ingenieure und Naturwissenschaftler,* Springer, 2008