Experiences with the High Performance FEM Package FEAST

Ch. Becker, S. Kilian, S. Turek

Institut für Angewandte Mathematik und Numerik,
Universität Dortmund

email: featflow@featflow.de

WWW: http://www.featflow.de

Motivation

'Typical': 3D Poisson problem

- $100 \times 100 \times 100$ grid points $\longrightarrow$ problem size $N = 10^6$
- Complexity of GE: $N^{7/3} \approx 10^{14}$ FLOP
  100 sec on a 1 TFLOP/s computer
- Complexity of (opt.) multigrid: $1000N \approx 10^9$ FLOP
  100 sec on a 10 MFLOP/s computer

- $1000 \times 1000 \times 1000$ grid points $\longrightarrow$ problem size $N = 10^9$
- Complexity of GE: $N^{7/3} \approx 10^{21}$ FLOP
  1,000,000 sec on a 1 PFLOP/s computer
- Complexity of (opt.) multigrid: $1000N \approx 10^{12}$ FLOP
  1,000 sec on a 1 GFLOP/s computer

$\Rightarrow$ **Question:** Are 10 MFLOP/s realistic in modern simulation packages?

Sparse Matrix Vector Multiplication

Standard sparse matrix vector algorithm:
(DAXPY indexed)

```
      DO 10 IROW=1,N
      DO 10 ICOL=KLD(IROW),KLD(IROW+1)-1
10    Y(IROW)=DA(ICOL)*X(KCOL(ICOL))+Y(IROW)
```

Performance rates of the FEATFLOW code with different numbering schemes (Cuthill–McKee, TwoLevel, Stochastic) for matrix vector multiplication:

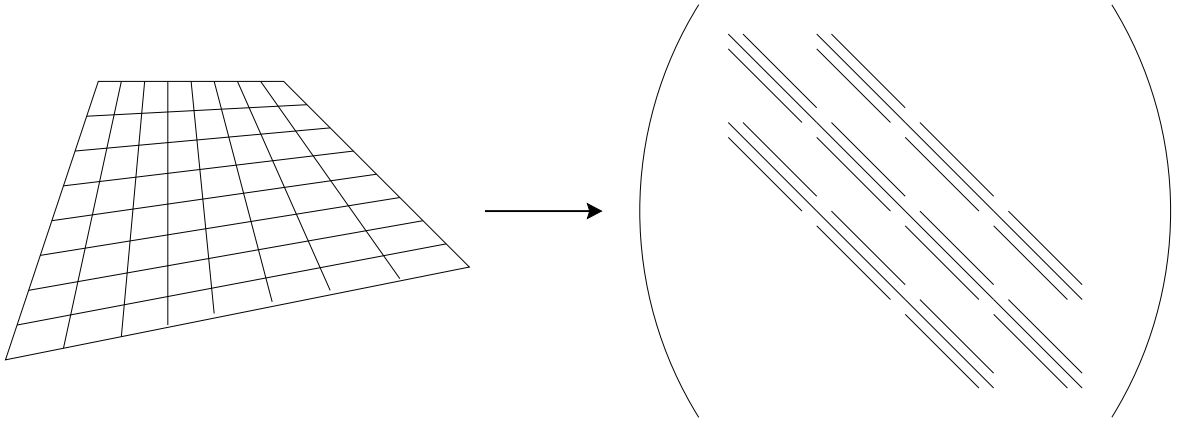
Computer	#Unknowns	**CM**	**TL**	**STO**
SUN E450 (∼ 250 MFLOP/s)	13,688	22	20	**19**
	54,256	17	15	**13**
	216,032	16	14	**6**
	862,144	16	15	**4**

- sparse techniques basis for most of the recent software packages
- different numberings can lead to identical numerical results and work (w.r.t. arith.ops and data accesses) but to huge differences in CPU time
- sparse techniques 'slow' and depend on problem size and kind of data access

Alternative: Sparse Banded Techniques

Question: How to exploit more performance?

Line- or rowwise numbering:

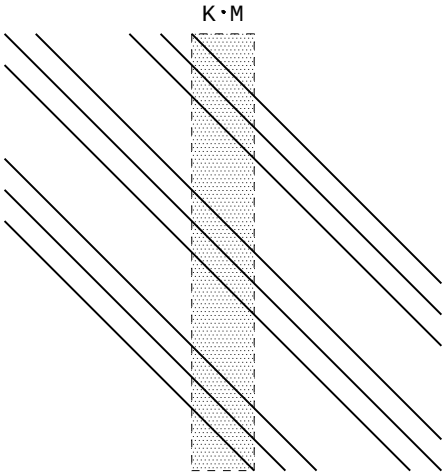


Sparse *banded* matrix vector multiplication:

- FD discretization leads to band structure on tensorproduct meshes
- storing of matrix elements in diagonals
- matrix vector multiplication 'bandwise'
- in equidistant case for certain operators diagonals are constant

Sparse Banded Techniques

Bandwise windowed multiplication (variable, constant):




```
      DO 10 IM=1,M/K
       DO 100 I=1,K*M
100     Y(I)  =Y(I)  +DD(I)*X(I)+DL(I)*X(I-1)+DU(I)*X(I+1)
       DO 200 I =1,K*M
200     Y(I-M)=Y(I-M)+LD(I)*X(I)+LL(I)*X(I-1)+LU(I)*X(I+1)
       DO 300 I=1,K*M
300     Y(I+M)=Y(I+M)+UD(I)*X(I)+UL(I)*X(I-1)+UU(I)*X(I+1)
10    CONTINUE
```

2D case	N	DAXPY-I	SBB-V	SBB-C	**MG-V**	**MG-C**
DEC 21264	65^2	205 (178)	538	795	370	452
(667 MHz)	257^2	224 (110)	358	1010	314	487
'**ES40**'	1025^2	**78 (11)**	**158**	**813**	**185**	**401**
HITACHI	65^2	173 (82)	238	391	191	266
(375 MHz)	257^2	143 (29)	243	388	198	260
'**SR8000**'	1025^2	**144 (7)**	**226**	**390**	**200**	**267**
AMD K7	65^2	203 (195)	101	556	122	355
(850 MHz)	257^2	29 (27)	78	241	72	166
'**ATHLON**'	1025^2	**31 (10)**	**64**	**236**	**58**	**126**

Question: How to use these techniques on complex domains? $\Rightarrow$ **ScaRC**

ScaRC (Scalable Recursive Clustering)

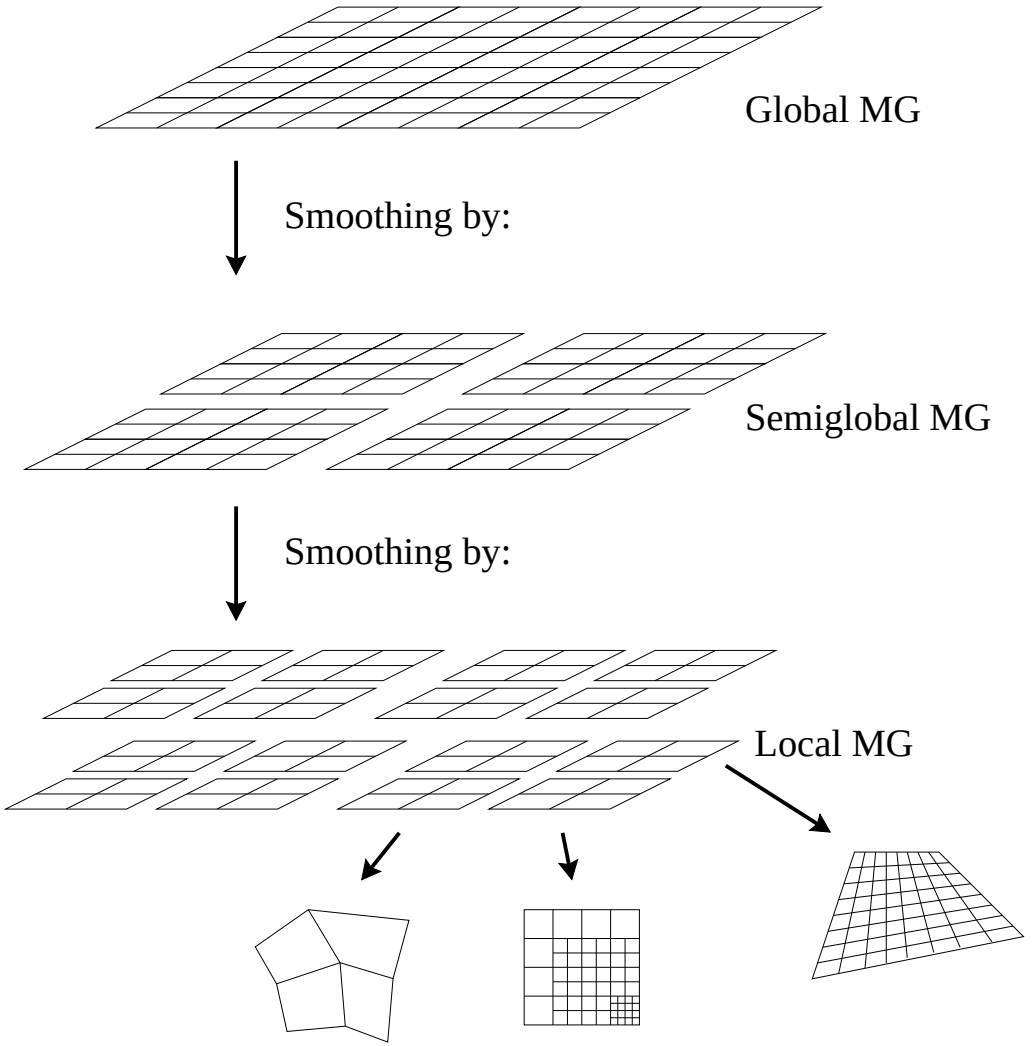
Standard multigrid with
(recursively defined)
block smoothers

plus

Standard Domain Decomposition
with minimal overlap,
sequence of coarse grid
problems via multigrid

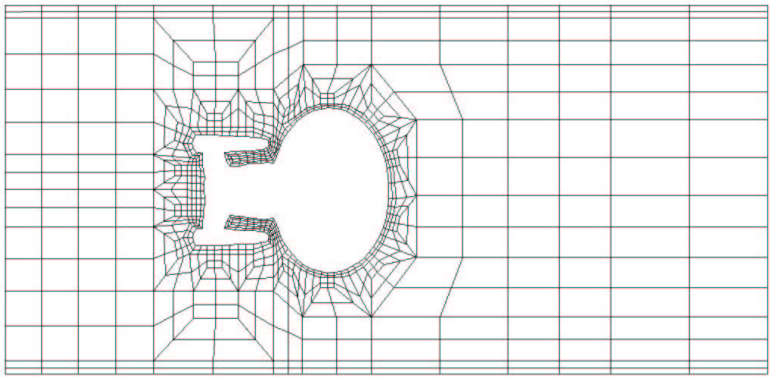
plus

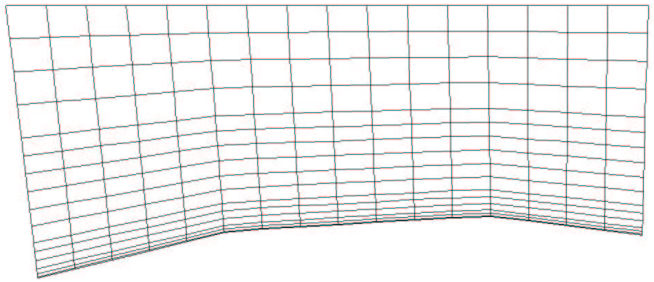
Embedded into
standard CG–method



Example: Realization of ScaRC in FEAST

2D decomposition and zoomed (macro) element (LEVEL 3) with locally anisotropic refinement towards the wall



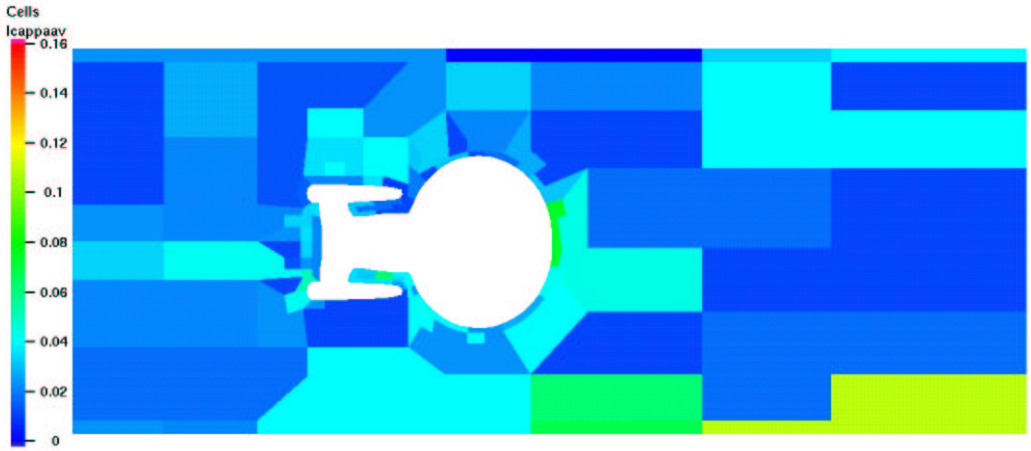


ScaRC-CG solver (smoothing steps: 1 global ScaRC ; 1 local 'MG-TriGS') for locally (an)isotropic refinement

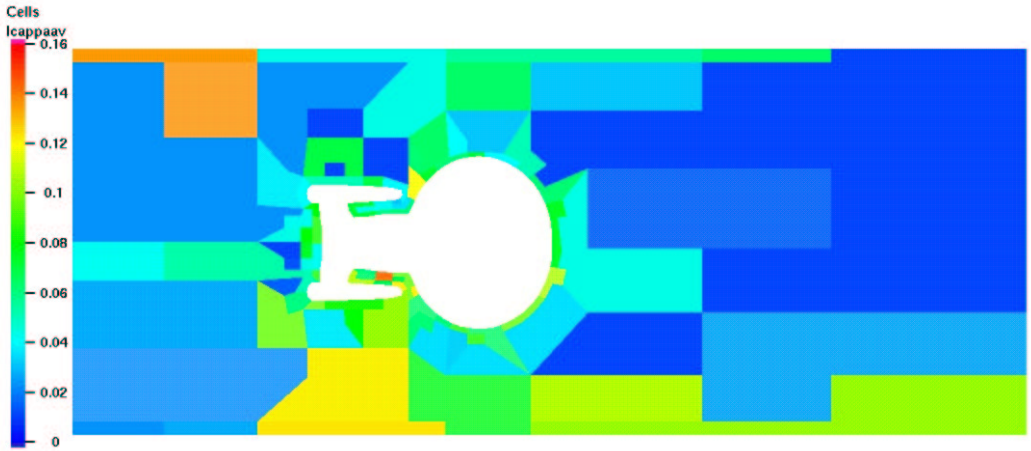
Global (parallel) convergence rates

$\#NEQ$	Dirichlet 'Velocity'		Neumann 'Pressure'	
	$AR \approx 10$	$AR \approx 10^6$	$AR \approx 10$	$AR \approx 10^6$
210,944	0.17 (**8**)	0.18 (**8**)	0.21 (**9**)	0.15 (**8**)
843,776	0.17 (**8**)	0.17 (**8**)	0.20 (**9**)	0.17 (**8**)
3,375,104	0.18 (**9**)	0.19 (**9**)	0.22 (**10**)	0.22 (**10**)
13,500,416	0.19 (**9**)	0.18 (**9**)	0.23 (**10**)	0.23 (**10**)

Local convergence rates (for $AR \approx 1$)



Local convergence rates (for $AR \approx 10^6$)



Parallel efficency

CPU times for 843776 unknowns (CRAY T3E without optimized SparseBandedBLAS and multigrid driver)

#P	CPU(s)
4	158.2
8	65.13
16	36.32
32	26.79
64	17.97

Outlook

Further work has to be done for

- optimal SparseBandedBLAS for different architectures
- optimized multigrid driver
- loadbalancing