(NASA-TM-80934) LECTURE NOTES OF THE ICASE WORKSHOP (NASA) 228 p

N80-73399

Unclas

00/64 17267

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA 22161

ATTENTION

PORTIONS OF THIS REPORT ARE NOT LEGIBLE. HOWEVER, IT IS THE BEST REPRODUCTION AVAILABLE FROM THE COPY SENT TO NTIS.

ICASE WORKSHOP ON MULTI-GRID METHODS

1. Introduction. Basic multi-grid processes and results.

2. Nonlinear Problems. The Full Approximation Scheme (FAS).

3. Nonuniform grids and adaptation techniques.

4. Finite-element formulations.

5. Multi-grid analysis, prediction and optimization.

6. Generalized relaxation schemes.

7. Navier-Stokes equations. Distributed relaxation.

8. Transonic flow problems. (J. C. South)

9. Parabolic time-dependent problems.

10. Using composite meshes for time-dependent problems. (J. Oliger)

11. Singular perturbation multi-grid techniques.

12. Multi-grid programming.

13. The multi-grid software. (F. Gustavson)

14. Multi-grid processes on large array computers. (C. E. Grosch)

15. Multi-grid experience on the minimal-surface equations. (D. J. Jones)

16. Multi-grid experiments to elliptic problems with finite-element formulation. (C. Poling)

17. Residual weighting and non-Dirichlet boundary conditions.

18. Debugging techniques.

List of participants.

I. INTRODUCTION.

BASIC MULTI-GRID PROCESSES AND RESULTS

Multi Level Adaptive Techniques (MLAT)

PDE (elliptic, mixed type, hyp. BVP ; IVP; EVP), Integro-differential equations, functional minimization...

1. Discretization (FDE, FEM) $\Rightarrow$ n algebraic eqs.
2. Solution of the algebraic system.

MLAT: 1 & 2 intermixed.

Several levels of discretization interact, giving very efficient adapted discretization, simultaneously with fast solution.

$$\text{ERROR} = C_1 e^{-c \cdot \text{WORK}^{1/2}}$$

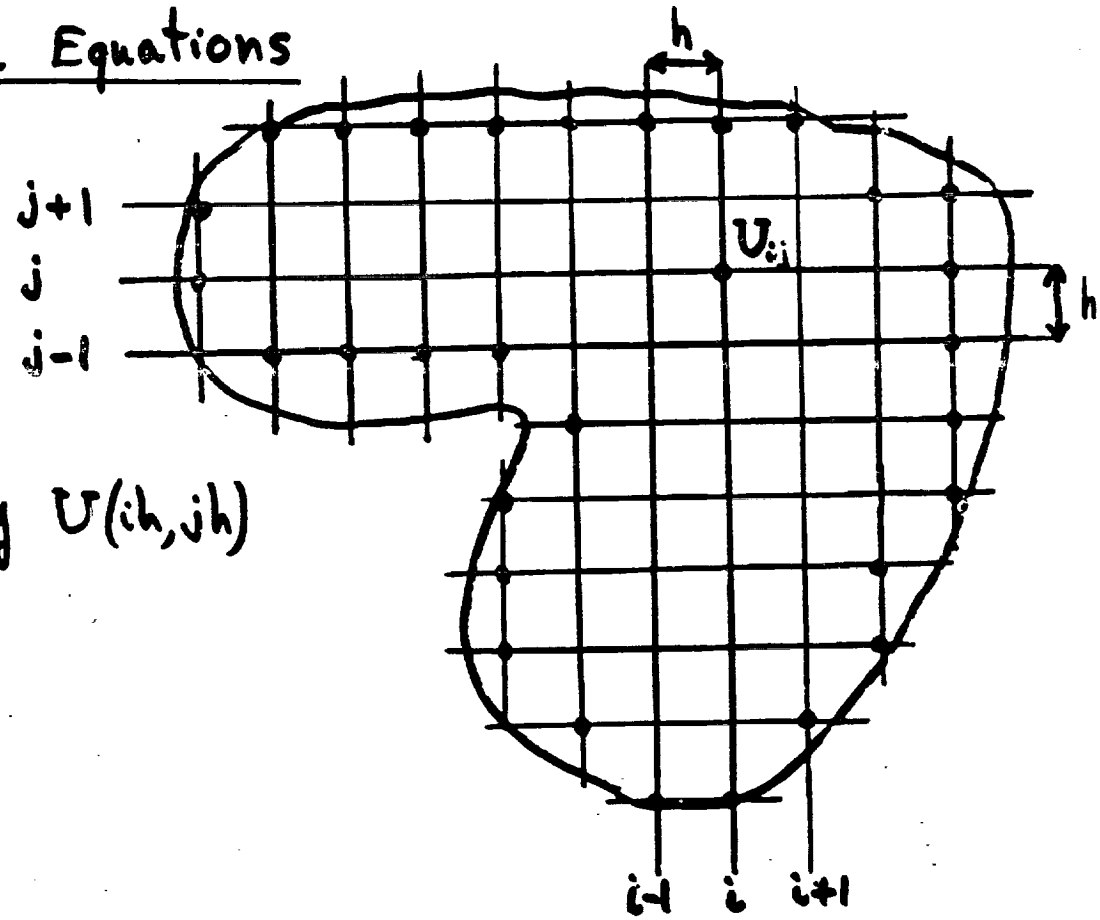
even for singularly-perturbed and discontinuous problems, nonlinear (and indefinite).

Discretization

$$\Delta U \equiv \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = F$$

e.g.,

Finite Difference Equations



U_{ij} approximating $U(ih, jh)$

$F_{ij} = F(ih, jh)$

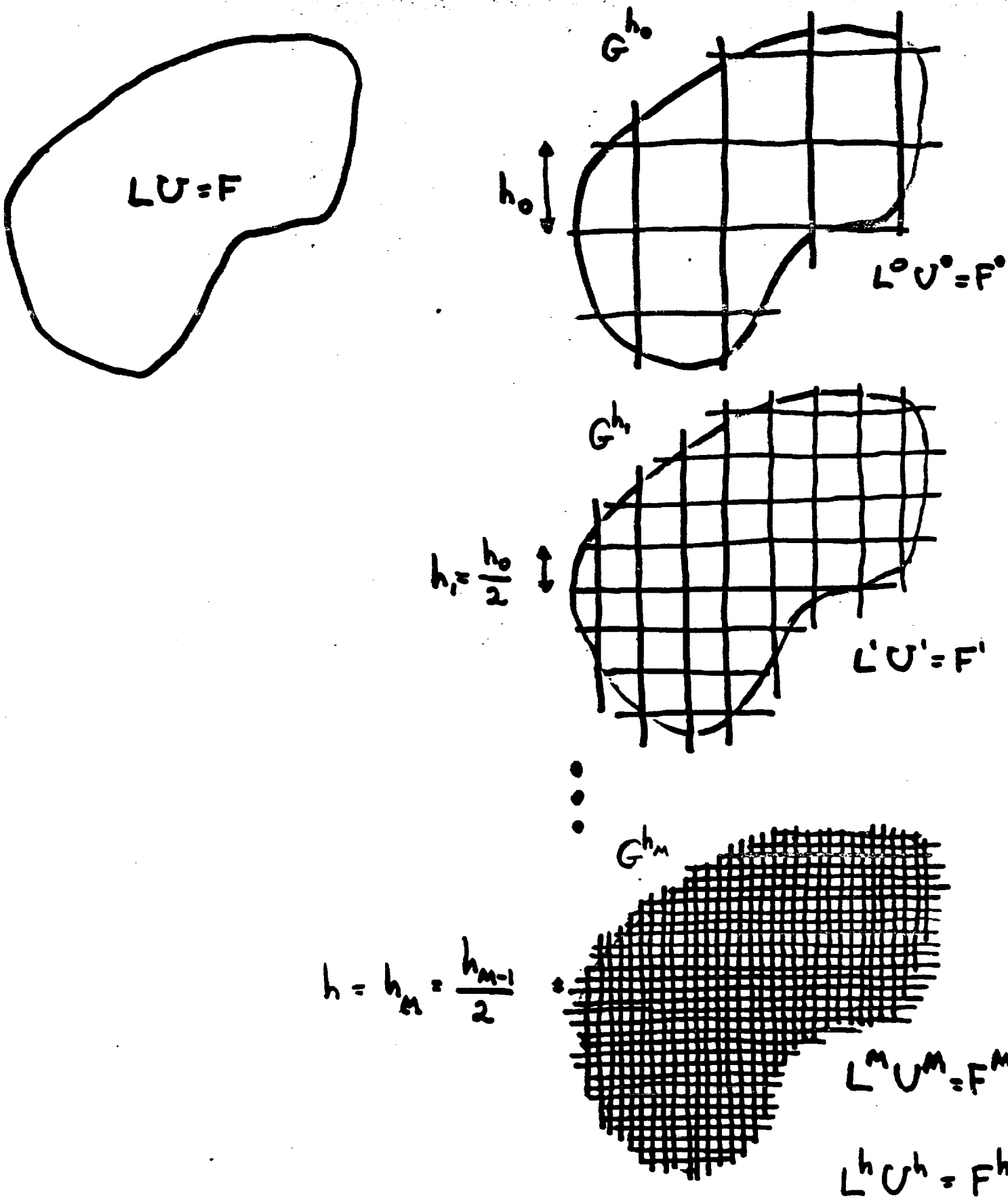
$\Delta_h U_{ij} = F_{ij}$

$$\Delta_h U_{ij} = \frac{1}{h^2}\left[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{ij}\right]$$

$$= \frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} U_{ij} = \Delta U + \frac{h^2}{12}\left[U_{xxxx}(\xi) + U_{yyyy}(\xi)\right]$$

$$\Delta_h^{(4)} U_{ij} = \frac{1}{12h^2}\begin{bmatrix} & & -1 & & \\ & & 16 & & \\ -1 & 16 & -60 & 16 & -1 \\ & & 16 & & \\ & & -1 & & \end{bmatrix} U = \Delta U + O(h^4)$$

What h and approximation-order should one use?


LU=F
G^{h_0}
h_0
$L^0U^0=F^0$
G^{h_1}
$h_1=\frac{h_0}{2}$
$L^1U^1=F^1$
G^{h_M}
$h=h_M=\frac{h_{M-1}}{2}$
$L^MU^M=F^M$
$L^hU^h=F^h$

<u>Classical use:</u> <u>first</u> approximation to U^h by interpolation from U^{2h}

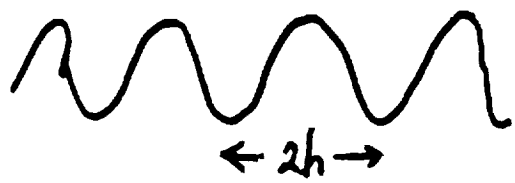
Can G^{2h} be used to <u>improve</u> the first approximation?

Approximation u^h: $L^h u^h = F^h - \underline{R^h}$ ← Residual

Correction $V^h = U^h - u^h$

Residual equation: $L^h V^h = R^h$ if linear

Can V^h be approximated by V^{2h} ?

Generally <u>NO</u>, since $V^h =$ 

But V^h and R^h can be <u>smoothed out by relaxation</u>

$$\circledast \quad U_{ij} = \frac{1}{4}\left[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - h^2 F_{ij}\right]$$

Gauss-Seidel Relaxation:

1. First approximation is given
2. Replace U_{ij} by its value in $\circledast$
3. Repeat 2 for all $(i,j) \in G^k$, in some order.
   This is a <u>relaxation sweep</u>
4. Make more such sweeps, until <u>convergence</u>
   (error norm $\leq$ tolerance)

$$\underline{\text{Convergence factor}} = \frac{\text{error norm}}{\text{error norm a sweep earlier}}$$

$$= 1 - O(h^2)$$

Relaxation

Convergence factor $\rightarrow 1 - O(h^m)$ ← Order of L

Rapid convergence slows down as the error $V^h = U^h - u^h$ and the residual $R^h = F^h - L^h u^h$ are smoothed out.

Local Fourier Analysis

	typical convergence factor
High-frequency components (wave-length $\leq 4h$)	* .5
Low-frequency components (wave-length ℓ)	$1 - O\left(\frac{h^m}{\ell^m}\right)$

As soon as convergence slows down, discontinue relaxation. Solve the (smoothed out) <u>residual</u> equation $L^h V^h = R^h$ approximately on a coarser grid G^{2h}.

$L^h U^h = F^h$ (~)
$L^h u^h = F^h - R^h$

A MULTI-GRID PROCESS

1. First approximation u^h by interpolation from a solution on G^{2h} *): $u^h = I^h_{2h} u^{2h}$

2. Smooth the error $V^h = U^h - u^h$ by a couple of relaxation sweeps.

3. Solve the residual equation $L^h V^h = R^h$ approximately on G^{2h} *), i.e.,

$$L^{2h} V^{2h} = \underline{I^{2h}_h} R^h \quad \text{(residual weighting)}$$

$$u^h_{NEW} \leftarrow u^h_{OLD} + I^h_{2h} v^{2h}$$

---

*) Solving on G^{2h} again by relaxation sweeps and approximate solutions on G^{4h}, etc.

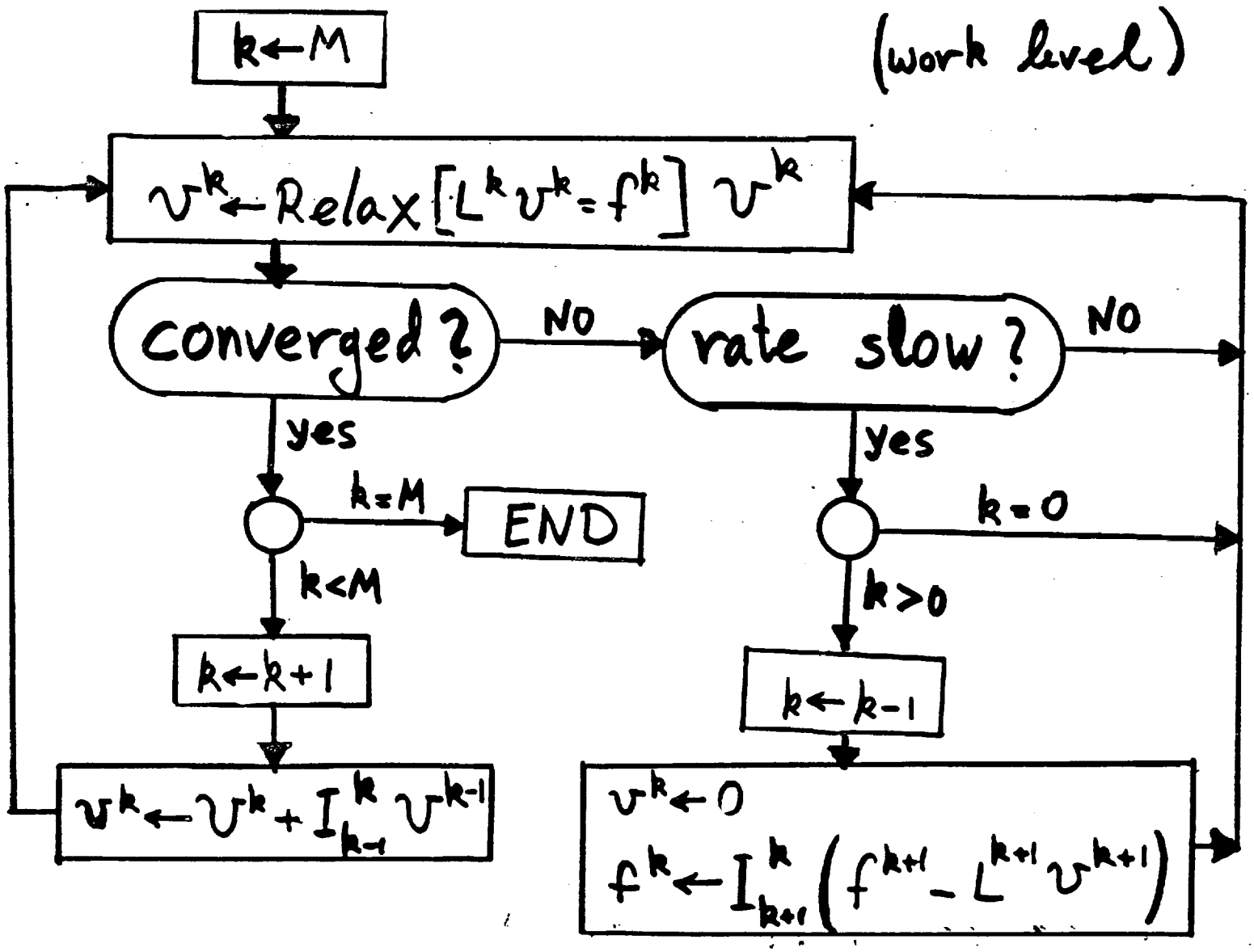
Multi-Grid Algorithm (Linear Cycle C)

${}^{k+1}_{k}v^k$ (on G^k) correction to v^{k+1}

$$\underline{L^k V^k = f^k} \equiv I^k_{k+1}\left(f^{k+1} - L^{k+1} v^{k+1}\right)$$

interpolation

$f^M = F^M$, $v^M = u^M$ given approximation on G^M



MESH SIZE	RESIDUAL NORM	ACCUM. WORK
h	28.1	1.00
h	27.6	2.00
2h	26.6	2.25
2h	25.6	2.50
4h	23.2	2.56
4h	20.9	2.62
8h	16.5	2.64
8h	12.9	2.66
16h	7.62	2.66
16h	3.84	2.66
8h	5.06	2.68
4h	8.01	2.74
4h	2.54	2.80
2h	9.74	3.05
2h	2.46	3.30
h	10.64	4.30
h	2.44	5.30
h	2.40	6.30
2h	2.35	6.55

MESH SIZE	RESIDUAL NORM	ACCUM. WORK
2h	2.30	6.80
4h	2.17	6.87
4h	2.04	6.93
8h	1.74	6.94
8h	1.45	6.96
16h	.989	6.96
16h	.618	6.97
32h	.276	6.970
32h	.052	6.971
16h	.229	6.975
8h	.546	6.99
4h	.771	7.05
4h	.116	7.11
2h	.866	7.36
2h	.106	7.61
h	.906	8.61
h	.105	9.61
h	.101	10.61
2h	.098	10.86

Factor $\frac{1}{10}$ error reduction in 4 work units
(equivalent to 4 fine-grid relaxation sweeps)

Main work: smoothing factor $\frac{1}{2}$ per sweep
Predictable computable from local operator
Independent of boundary shape, nonlinearities, ...
Similar efficiency (per sweep) for nonlinear, non-definite, non-elliptic problems with shocks.
e.g., $(K - \varphi_x)\varphi_{xx} + \varphi_{yy} = 0$

Solution of problem $\sim$ 7 work units
$42n$ operations for 5-point Poisson problems
$+27n$ ($2n$ STAR microseconds) per additional dec. digit.

Full parallel processing $\frac{50n}{\#\text{ processors}} + O(\log n)$
Storage required $\ll n$ $500 \log_2 n$
Programming per problem reduced to relaxation
Minimal round off

Suitable for adaptive process

$u = F$, Dirichlet BC, $U = \sin(3x+3y)$ in $[0,3] \times [0,2]$ | $\frac{4}{3}\tau_H^{2H}$

x96 grid h, Gauss-Seidel, injection, $\eta = .6$, $\delta = .3$ | No h Storage

l	$\|RES.\|_G$	WORK	$\|u^H - U\|_\infty$	
4h	.201	.36	.0180	.0052
4h	0		.0138	
cubic interpolation				
2h	.267	.62		
2h	.205	.87	.0168	.0047
4h	.177	.93		
4h	.153	.99	.0146	.0039
8h	.109	1.01		
8h	.075	1.02	.0089	.0024
16h	.037	1.03		
16h	.014	1.03	.0021	.0009
32h	.005	1.03		
32h	.0007	1.03	.0022	.0001
16h	.004	1.04	.0025	.0003
8h	.027	1.05	.0036	.0002
4h	.050	1.12	.0040	.0007
2h	.066	1.37	.0042	.0005
2h	0		.0034	
cubic interpolation				

h	.068	2.37		
h	.057	3.37	.0041	.0004
2h	.054	3.62		
2h	.051	3.87	.0039	.0004
4h	.045	3.93		
4h	.039	3.99	.0034	.0003
8h	.028	4.01		
8h	.019	4.02	.0021	.0002
16h	.009	4.03		
16h	.0035	4.03	.00056	.0000
32h	.0014	4.03		
32h	.0001	4.03	.00054	.0000
16h	.0011	4.04	.00062	.0000
8h	.007	4.05	.00088	.0000
4h	.013	4.12	.00098	.0000
2h	.016	$\boxed{4.37}$	.00102	.0000
h	.0182	$\boxed{5.37}$	.00103	.0000
h	0		.00086	
h	.040	5.37	.00106	.0000

2. NONLINEAR PROBLEMS.

THE FULL APPROXIMATION SCHEME (FAS)

NON-LINEAR EQUATIONS $L^h U^h = F^h$

Approximation u^h, correction sought $V^h = U^h - u^h$

<u>Residual equation:</u>

$$\hat{L}^h V^h \equiv L^h(u^h + V^h) - L^h u^h$$
$$= F^h - L^h u^h$$
$$\equiv R^h, \qquad \text{the residual.}$$

V^h, R^h smoothed by relaxation (with $\hat{L}_h$ rate).

<u>Coarse-grid correction equations:</u>

Linear case: $L^{2h} V^{2h} = \underline{I_h^{2h}} R^h$

Nonlinear : $L^{2h}(\underline{I_h^{2h}} u^h + V^{2h}) - L^{2h}(\underline{I_h^{2h}} u^h) = I_h^{2h} R^h$

<u>**FAS** form:</u> $L^{2h} U^{2h} = f^{2h}$

where: $U^{2h} = I_h^{2h} u^h + V^{2h}$

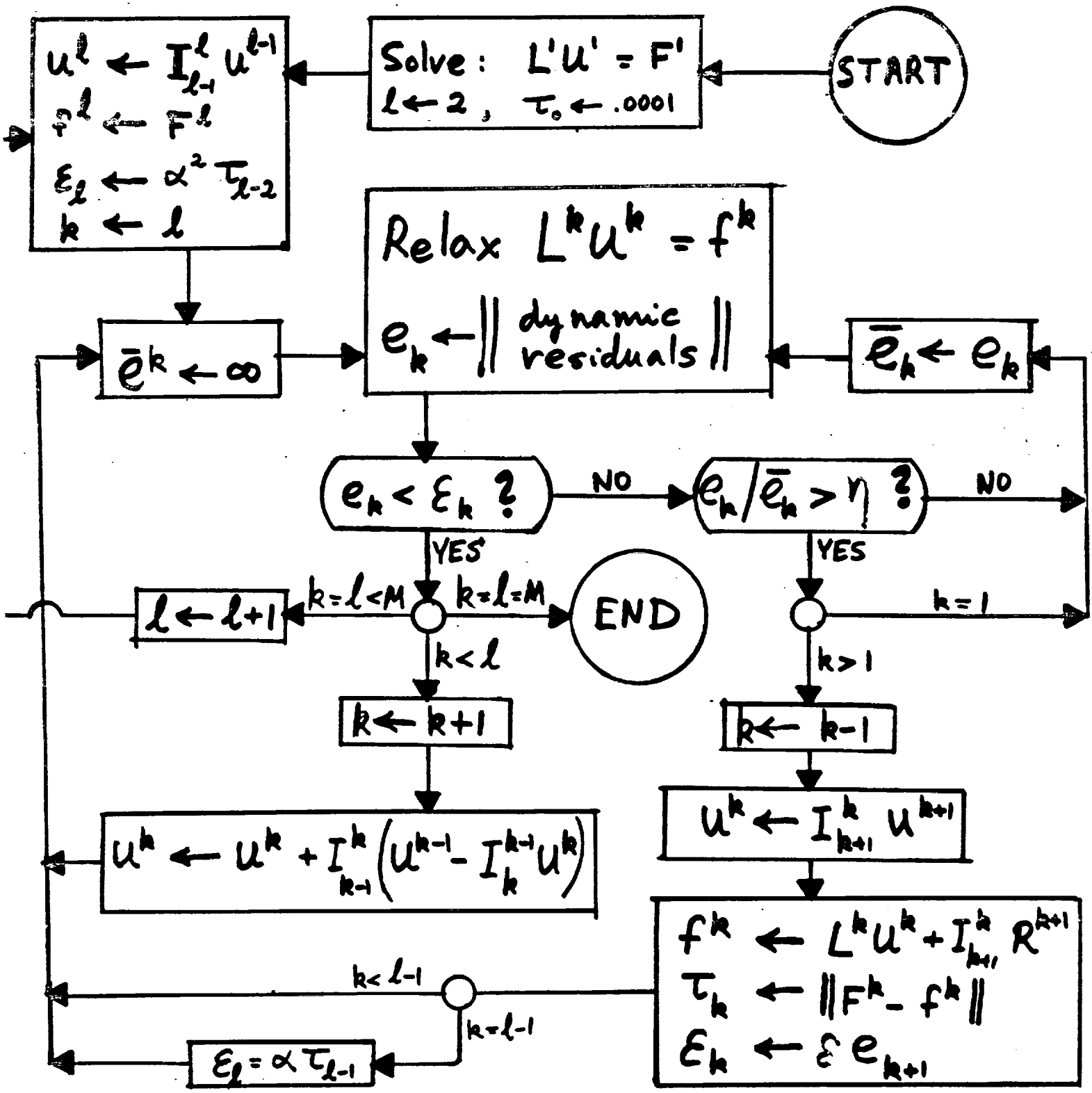
$$f^{2h} = L^{2h}(I_h^{2h} u^h) + I_h^{2h} R^h.$$

If u^{2h} approximates U^{2h}, the fine-grid correction is:

$$u^h_{NEW} \leftarrow u^h_{OLD} + I_{2h}^h \left(u^{2h} - I_h^{2h} u^h_{OLD} \right)$$

FAS Full Multi-Grid Algorithm

ℓ' - coarsest level. u^M - finest level. u^ℓ - finest so far.
ℓ^k - work level (If $k<\ell$, $u^k - I^k_{k+1} u^{k+1}$ is approx. correction to u^{k+1}).
$I^\ell_{\ell-1}$ - higher order relaxation $\qquad \alpha \approx 2^{-p}$, $\eta \approx \bar{\mu}$, $\delta \approx \bar{\mu}^r$



FULL APPROXIMATION SCHEME

- Linear problems, uniform grid: Identically the same

- Nonlinear problems: No global linearization needed. Only one relaxation Routine.

- Relative truncation errors are computed:

$$f^{2h} = L^{2h}\left(I_h^{2h} u^h\right) + I_h^{2h}\left(F^h - L^h u^h\right)$$
$$= F^{2h} - I_h^{2h} L^h u^h + L^{2h}\left(I_h^{2h} u^h\right)$$
$$= F^{2h} + \tau_h^{2h}, \qquad \tau_h^{2h} \approx \tau^{2h}$$

(truncation error

- Truncation extrapolation

- At convergence, the G^{2h} equations have U^h as their solution: $U^{2h} = U^h$

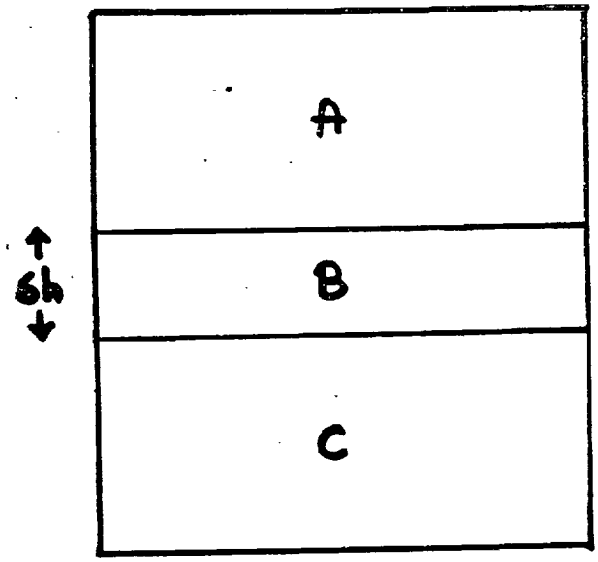
Dual multi-grid: G^{2h} basic grid, $L^{2h} U^{2h} = F + \tau_h^{2h}$

- Non uniform discretization: G^h only part of G^{2h}...

- Natural stopping criteria and grid adaptation

- Smooth, relatively small changes in U
  $\Rightarrow$ relatively small changes in τ_h^{2h}.

$O(\tau)$ changes in F on A
$\Rightarrow O(\tau^2)$ changes on C.



Hence, τ_h^{2h} can be computed separately on $A \cup B$, $B \cup C$.

- <u>Small storage algorithms</u>: f^{2h} computed in segments. Solution on n-points grid with only $20^d \log n$ storage
- <u>Optimal-control problems</u>: Control on coarse grids.
- <u>Non-well-posed problems</u>: Data fitting on coarse grids
- <u>Time-dependent problems</u>: Controlled visits to fine grids
- <u>Continuation procedures</u>: Infrequent visits to finer grids.

Fine-to-Coarse Corrections

$$LU = R \qquad \text{differential equation}$$

$$L_f U^f = R^f = I^f R \qquad \underline{\text{fine}} \text{ discrete eq.}$$

$$\Updownarrow$$

$$L_c I_f^c U^f = I_f^c R^f + \tau_f^c \qquad \text{corrected } \underline{\text{coarse}} \text{ eq.}$$

where $\tau_f^c = L_c I_f^c U^f - I_f^c L_f U^f$

= "relative truncation error"

$\tau_f^c = \tau^c \equiv L_c U - LU$ (truncation error) if $L_f = L$

τ_f^c = "deferred correction" if L_f is a higher-order approx.

τ_f^c = "fine-grid correction" if L_f is on a finer grid.

Generally, $\tau_f^c \approx \tau^c - \tau^f \approx \tau^c$

$$\|\tau_f^c\| \approx \|\tau^c\| = O\left(h^p \|U^{(m+p)}\|\right)$$

h = coarse mesh size
p = coarse app. order

$$= \begin{cases} O(h^p \|U\|) & \text{for smooth } U \\ O(h^{-m} \|U\|) & \text{for high-frequency } U \end{cases}$$

Higher-(p) Order approximations

$$L_h^p U^h = F$$ on finest grid

OR: $$L_h U^h = F + \sigma_p .$$

$$\underline{\sigma_p} = L_h^p U^h - L_h U^h$$ p-order correction

σ_p computed once in a cycle.
(Trading work for storage)

Truncation extrapolation

$$\tau^h = ch^p + O(h^q), \qquad q = p+1 \text{ or } p+2$$

$$\tau^{2h} = 2^p c h^p + O(h^q)$$

$$\tau_h^{2h} = \tau^{2h} - \tau^h = (2^p - 1) c h^p + O(h^q)$$

Instead of FAS coarse-grid RHS $f^{2h} \leftarrow F^{2h} + \tau_h^{2h}$ put

$$f^{2h} \leftarrow F^{2h} + \frac{2^p}{2^p - 1} \tau_h^{2h}$$

$$= F^{2h} + \tau^{2h} + O(h^q)$$

The new truncation error is $O(h^q)$!

- τ-ext. can be applied where Richardson ext. fails (non-uniform approximation order).
- <u>One</u> multi-grid cycle with τ-ext. gives
$$|u^h - U| < |U^h - U|$$
- Theoretically better than τ-ext. (or h-ext.) is to have q-order approx. on G^{2h}. But τ-ext. is simpler to apply, adding no effort to FAS programming

Fine-to-Coarse transfer

$$U^{2h} = I_h^{2h} U^h$$

$$U^{2h}(x,y) = \sum w_{ij} \, U^h(x+ih, y+jh)$$

$$w_{ij} = 2^{-|i|-|j|-2} = \begin{array}{ccccc} 0 & \times & 0 & \times & 0 \\ \times & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & \times \\ 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & 0 \\ \times & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & \times \\ 0 & \times & 0 & \times & 0 \end{array}$$

$U^{4h} = I_{2h}^{4h} U^{2h}$, etc.

U^H is an average of U^h with averaging scale H.

Other functions can be transferred at I_h^{2h}, such as the subgrid "energy":

$$q_{2h}^2(x,y) = \sum w_{ij} \left[U^h(x+ih, y+jh) - U^{2h}(x,y) \right]^2 + \sum w_{ij} \, q_h^2(x+ih, y+jh)$$

switch m.g. procedures when $q_H \gg u^H$

Dual multi-grid

$$L_c U = R + \tau_f^c$$

In an iterative process for L_c, whenever the error is sufficiently reduced, visit the fine-grid (i.e., FAS interpolat and make a couple of relaxation sweeps) to update τ_f^c.

This is then an iterative process for L_f, in which visits to still finer discretization can be inserted. Etc.

Wider range of possibilities

Nonlinear equations

Local refinements, grid adaptations

Natural stopping criteria

Optimal control problems

Continuation procedures. Bifurcation problems

Truncation extrapolation

Ill-posed problems

Small-storage algorithms

Switching representation (e.g., q_H).

3. NONUNIFORM GRIDS AND ADAPTATION TECHNIQUES.

A MULTI-GRID PROCESS

$L^h U^h = F^h$ 5-X

$L^h u^h = F^h - R^h$

1. First approximation u^h by interpolation from a solution on G^{2h} *): $u^h = I_{2h}^h u^{2h}$

- Decide ** where G^h is needed

2. Smooth the error $V^h = U^h - u^h$ by a couple of relaxation sweeps.

- Extend G^h? ** Change approximation order? **

3. Solve the residual equation $L^h V^h = R^h$ approximately on G^{2h} *), i.e.,

$$L^{2h} V^{2h} = \underline{I_h^{2h}} R^h \quad \text{(residual weighting)}$$

$$u^h_{NEW} \leftarrow u^h_{OLD} + I_{2h}^h v^{2h}$$

---

*) Solving on G^{2h} again by relaxation sweeps and approximate solutions on G^{4h}, etc.

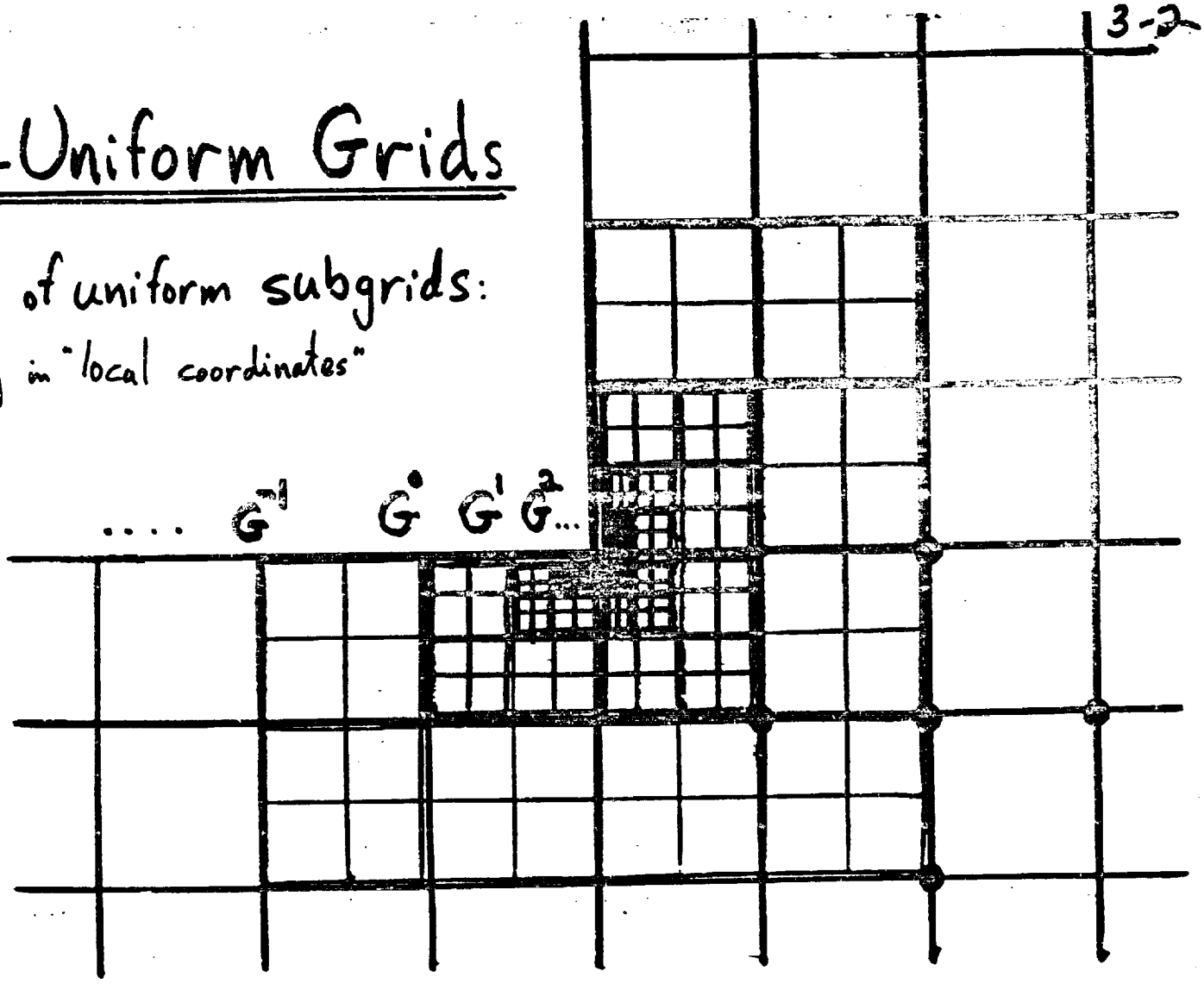
** Decision based on optimization: $\frac{-\Delta E}{\Delta W} \gtrless \lambda$

E = error, W = work, λ = Desired exchange rate

Non-Uniform Grids

Union of uniform subgrids:

possibly in "local coordinates"



Same sequence of levels $G^0, G^1, G^2, \ldots$, but not coextensiv

Increasingly finer levels near singular points or layers.

Increasingly coarser levels on unbounded domains.

<u>Multi-grid Algorithm</u>: the same! same efficiency.

Minimal bookkeeping. FDE all standard equi-distant

<u>Flexible structure</u>: Variable FDE order. Refinemen

(coarsening) = extension (contraction) of uniform subgrid

Self Adaptive Discretization

Mesh-size h, order of approximation p, ...
- spatial variables, automatically adapted

to nearly minimize error estimator E order of magnitude
in a given amount of computational work W.

$$E = \int G(x)\,\tau(x)\,dx. \qquad W = \omega \int \frac{\rho^2}{h^d}\,dx \qquad \text{(MG).}$$

$$\tau(x) = \tau(x,h,p) = \text{truncation error} = |LU - L_h U|$$

$$\tau(x,2h,p) - \tau(x,h,p) = \tau_h^{2h} \qquad \text{in FAS multi-grid}$$

$G(x)$ = error weighting function
reflects purpose of computations order of magnitude
e.g., $G(x) = d_x^{\,m/2-\ell}$

$2h \to h$: $\quad -\Delta E = G\left[\tau(x,2h,p) - \tau(x,h,p)\right]$ per unit volume

$$\Delta W = \omega p h^{-d}\left[1 - 2^{-d}\right]$$

$$W = \omega \int \frac{p^2}{h^d}\, dx \qquad (MG)$$

$$E = \int G\tau\, dx$$

Minimize E for fixed W (or W for fixed E)

controling $h(x), p(x), \ldots$ $\qquad p_0 \le p(x) \le p_1$

<u>Euler Equations:</u>

$$\frac{\partial E}{\partial h(x)} + \lambda \frac{\partial W}{\partial h(x)} = 0$$

$$\frac{\partial E}{\partial p(x)} + \lambda \frac{\partial W}{\partial p(x)} \begin{matrix} > \\ = \\ < \end{matrix} 0 \qquad \begin{matrix} : p = p_0 \\ \\ : p = p_1 \end{matrix}$$

λ = const. (Lagrange Multiplier) $= -\dfrac{d E_{min}}{d W}$

marginal rate of exchange

$\Rightarrow$ <u>Local Equations:</u>

$$G \frac{\partial \tau}{\partial h} - \frac{\lambda p d \omega}{h^{d+1}} = 0$$

$$G \frac{\partial \tau}{\partial p} + \frac{\lambda}{h^d} = 0$$

<u>Grid Adaptation:</u> λ - our control

τ_h, τ_p computed from the evolving solution.

In optimum all $\frac{-\Delta E}{\Delta W} = \lambda$ (constant)

λ (Lagrange multiplier) $= -\frac{dE_{min}}{dW}$ is our control

Change $h \to 2h$ if $\frac{-\Delta E}{\Delta W} < .3\lambda$

$h \to \frac{h}{2}$ if $\frac{-\Delta E}{\Delta W} > 15\lambda$

$p \to p \pm 1$ using σ_p^{p+1}

Changing computational boundaries

Omitting some FDE terms...

Multi-grid switching criteria : Direct global $\frac{\Delta E}{\Delta W}$

- Rules for stabilizing changes.
- Extrapolated tests.
- Minimizing artificial viscosity - solves shock problems.
- Parabolic Time-dependent problems : minimal "visits" to fine grids

CASE	MLAT Discretization	error $E(W)$
uniform scale $\nabla^q u(x) \le \left(\frac{2}{\eta(x)}\right)^q$	$h^d = \eta^d/e$, $p \approx cW$ d = dimension, $\frac{1}{c} = e\int \frac{dx}{\eta^d}$	$C_1 e^{-\frac{c}{d}W}$ p-arithmetic
$\eta u'' + 2u' = f$ large η	$h = \frac{\eta}{e}$, $p \approx \frac{\eta}{e} W$	$\frac{e}{2\eta} e^{-\frac{\eta}{e}W}$
small η $G \equiv 1$	in boundary layer $0 \le x \le \eta W^{\frac{1}{2}}$ $h = \frac{\eta}{e}$, $p \approx \sqrt{W}$. h grows exponentially to ext. scale	$\sqrt{W}\, e^{-\sqrt{W}}$ independent of η.
small η $G(x) = x$	No boundary layer resolution.	$C_1 e^{-cW^{\frac{1}{2}}}$ external c.
ear corner $\frac{\pi}{\alpha}$ $\tilde{g}(r) = O(r^\alpha)$	$h = a e^{\frac{-cW}{p+2}} r^{\frac{p+2-2\alpha}{p+2}}$	$C_1 e^{-cW^{\frac{1}{2}}}$
lgebraic singularity, liscontiuities Hyperbolic Eqs.	suitable local refinement	exponential decrease of artifial viscosity

Continuation Methods

Problem parameter γ. (Reynolds number, Mach, etc.)

Solution steps advances γ from γ_0 to target value γ_*.

For γ_0 - simple problem (linear, regular).

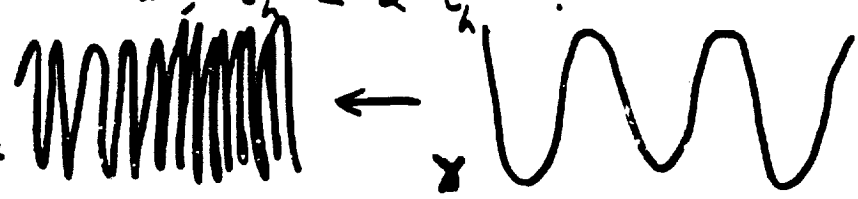
1st approx. to $U(\gamma) = u(\gamma - \delta\gamma)$.

<u>Continuation necessary</u> for non-linear problems
(where solution is only <u>defined</u> by continuation)
and for adaptive-grid formulations
(which are non-linear even if L is linear).

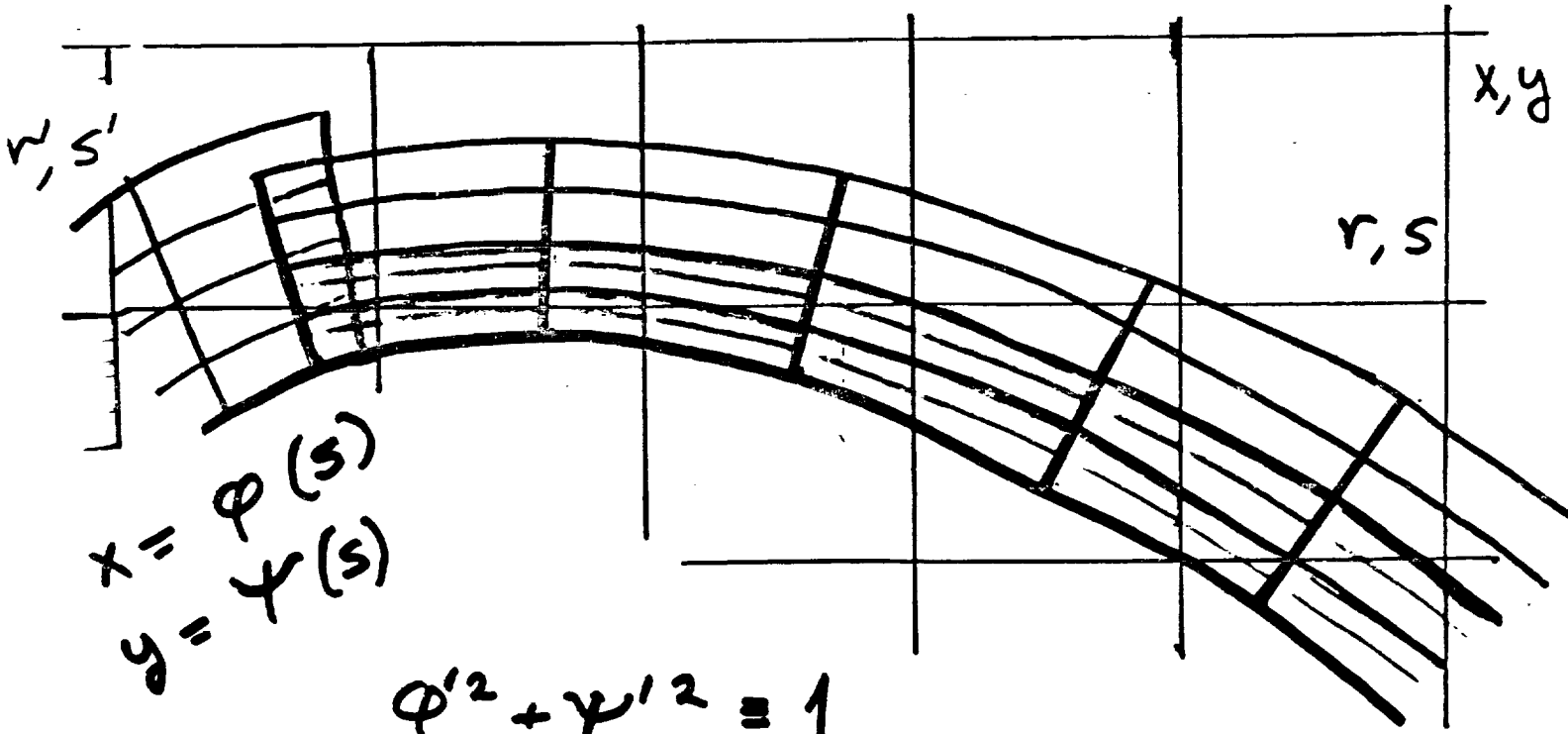
<u>Continuation inexpensive:</u> Rough accuracy -
Large $\lambda \Rightarrow$ coarser grids. (Rare visits to finer grids keeping τ_h^{2h} fixed)

<u>(γ, λ) control.</u> λ just small enough to have "refinable" approx
i.e. $\tau_h^{2h} < \alpha \tau_h^{mh}$.

<u>γ extrapolations.</u>



Local Coordinate Transformations



$$x = \varphi(s)$$
$$y = \psi(s)$$

$$\varphi'^2 + \psi'^2 \equiv 1$$

$$x(r,s) = \varphi(s) - r\psi'(s)$$
$$y(r,s) = \psi(s) + r\varphi'(s)$$

$r=0$ – the given curve

Isometric transformation

$$\frac{\partial}{\partial x} = -\psi' \frac{\partial}{\partial r} + \frac{\varphi'}{1+rq} \frac{\partial}{\partial s}$$
$$\frac{\partial}{\partial y} = \varphi' \frac{\partial}{\partial r} + \frac{\psi'}{1+rq} \frac{\partial}{\partial s}$$

$$q(s) = \frac{\varphi''(s)}{\psi(s)} = -\frac{\psi''(s)}{\varphi'(s)}$$

FDE reproducible in terms of 1-dim. arrays.

Interpolation, too.

4. FINITE-ELEMENT FORMULATIONS.

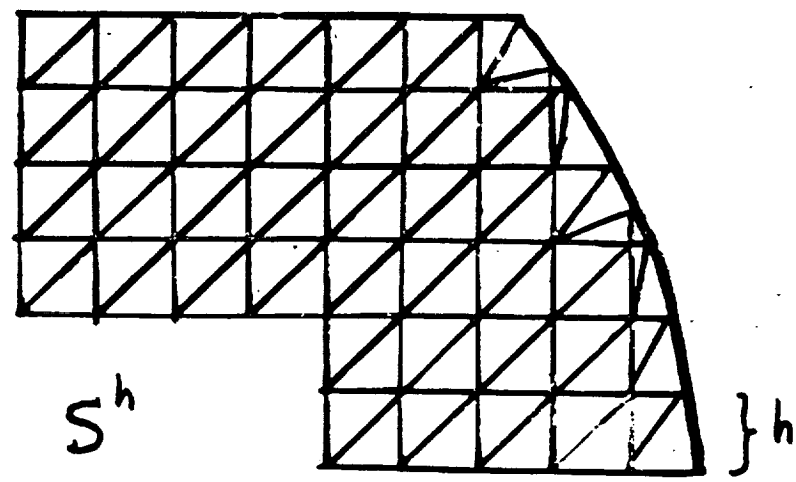
Continuous Problem

Find U, $\quad I(U) = \min\limits_{u \in S} I(u)$

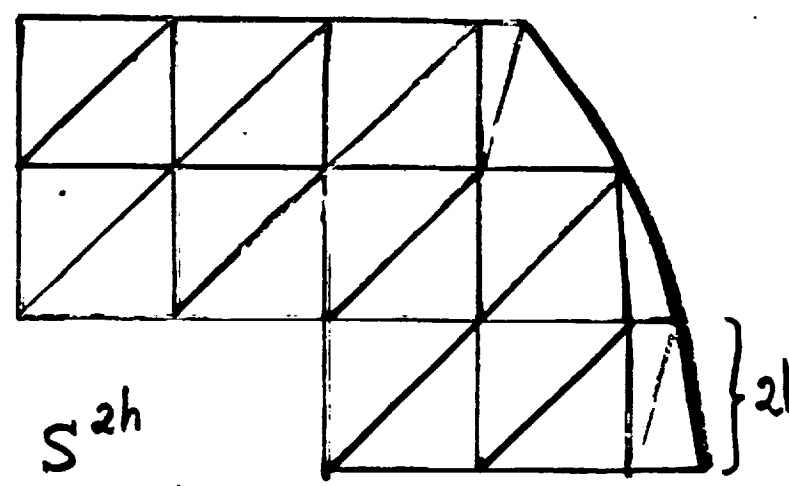
e.g., $\quad I(u) = \int_\Omega \left(u_x^2 + u_y^2 + 2Fu \right) dx\,dy$

Discrete Problem

$$I(U^h) = \min_{u^h \in S^h} I(u^h)$$



e.g., piecewise linear functions on uniform triangulation. $\Delta_h U^h = F^h$.



Coarser Approximation Spaces

$\ldots, S^{4h}, S^{2h}, S^h \subset S$

e.g., $S^{4h} \subset S^{2h} \subset S^h \subset S$ (nested)

u^h has $n = O(h^{-2})$ variables (unknowns)
u^{2h} has $\frac{1}{4}n$, u^{4h} has $\frac{1}{16}n$, ...

<u>Direct Solutions</u> (elimination, dissection)

| $20\, q\, n^{3/2}$ operations ($q=1$ for linear)
| $30\, n \log n$ storage locations.

<u>Solution by Relaxation:</u>

A given approximation is successively improved by relaxation sweeps, until $\|\text{error}\| \leq \varepsilon$.

<u>Gauss-Seidel Relaxation Sweep</u>: The variables are corrected one by one (or line by line) in some order. Each correction reduces $I(u^h)$ by local minimization.

| simple programming; no global linearization
| n storage locations
| but $O(n^2 \log n)$ operations.

A multi-grid process

1. Approximate U^h by interpolation from a coarse-grid solution u^{2h} *)

2. Smooth out the error $V^h = U^h - u^h$ by a couple of relaxation sweeps.

3. The correction V^h minimizes $I(u^h + v^h)$. Approximate V^h by $I_{2h}^h V^{2h}$ minimizing $I(u^h + I_{2h}^h v^{2h})$. *)

$$\Delta_{2h} V^{2h}(x,y) = \sum \omega_{\alpha\beta}\, r^h(x+\alpha h,\, y+\beta h)$$

$$r^h = F^h - \Delta_h u^h, \qquad \omega_{00} = \frac{2}{8},\ \omega_{-1-1} = \omega_{-10} = \omega_{0-1} = \omega_{01} = \omega_{10} = \omega_{11} = \frac{1}{8}$$

*) solving on S^{2h} again by relaxation sweeps and interpolations from S^{4h}, etc.

Finite-Elements Formulations

Minimization Problem: $\mathcal{E}(U) = \min_{u \in S} \mathcal{E}(u)$

Finite element spaces: $S^h, S^{2h}, \ldots \subset S$

$$S^h = \left\{ u^h(x) = \sum_{j=1}^{n_h} u_j^h \phi_j^h(x) \right\}$$

S^h approximation: $\mathcal{E}(U^h) = \min_{u^h \in S^h} \mathcal{E}(u^h)$

Stiffness equations: $\mathcal{E}_j^h(U^h) = 0$

$$\mathcal{E}_j^h(u^h) = \frac{\partial \mathcal{E}(u^h)}{\partial u_j^h} = \lim_{\delta \to 0} \left[\mathcal{E}(u^h + \delta \phi_j^h) - \mathcal{E}(u^h) \right] / \delta$$

Residuals of an approximation u^h: $r_j^h = -\mathcal{E}_j(u^h)$.

Error: $V^h = U^h - u^h$ smooth (by relaxation) $\rightarrow$

V^h approximable by $\underline{I_{2h}^h} V^{2h}$, $V^{2h} \in S^{2h}$

Interpolation (sometimes identity)

Coarse-grid approximation:

$$\mathcal{E}(u^h + I_{2h}^h V^{2h}) = \min_{v^{2h} \in S^{2h}} \mathcal{E}(u^h + I_{2h}^h v^{2h})$$

<u>Coarse-grid Equations</u>: $\mathcal{E}_i^{2h,h}\left(u^h + I_{2h}^h V^{2h}\right) = 0$

$$\mathcal{E}_i^{2h,h}\left(u^h\right) = \lim_{\delta \to 0} \left[\mathcal{E}\left(u^h + \delta I_{2h}^h \phi_i^{2h}\right) - \mathcal{E}\left(u^h\right)\right] / \delta$$

<u>Simplification</u> (nonlinear and/or non-nested case):

$$\mathcal{E}_i^{2h,h}\left(u^h + I_{2h}^h V^{2h}\right) - \mathcal{E}_i^{2h,h}\left(u^h\right) = \underline{-\mathcal{E}_i^{2h,h}\left(u^h\right)}$$

Since V^{2h} smooth we can approximate this by

$$\mathcal{E}_i^{2h}\left(I_h^{2h} u^h + V^{2h}\right) - \mathcal{E}_i^{2h}\left(I_h^{2h} u^h\right) = \underline{-\sum \omega_{ij} \mathcal{E}_j^h\left(u^h\right)}$$

where $I_{2h}^h \phi_i^{2h} = \sum \omega_{ij} \phi_j^h$. weighted residuals

<u>Linear</u> (CS): $\mathcal{E}_i^{2h}\left(V^{2h}\right) = \sum \omega_{ij} r_j^h$

<u>Nonlinear</u> (FAS): $\mathcal{E}_i^{2h}\left(U^{2h}\right) = \mathcal{E}_i^{2h}\left(\underline{I_h^{2h} u^h}\right) + \sum \omega_{ij} r_j^h$

where $U^{2h} = I_h^{2h} u^h + V^{2h}$. First approx: $u^{2h} \leftarrow \underline{I_h^{2h} u^h}$

$$u_{NEW}^h \leftarrow u_{OLD}^h + I_{2h}^h \left(u^{2h} - \underline{I_h^{2h} u_{OLD}^h}\right)$$

same

<u>At convergence</u> $U^{2h} = I_h^{2h} U^h$

<u>Same assembly & relaxation routines at all levels.</u>

Weak Formulation $(LU, W) = (F, W), \quad \forall\, W \in S.$
$(U \in \bar{S}).$

Test-functions space: $S^h = \{ W^h(x) = \sum W_j^h \phi_j^h(x) \} \subset S$

Galerkin approximation $(LU^h, \phi_j^h) = (F, \phi_j^h), \quad \forall j$

$$\left(L(u^h + V^h), \phi_j^h\right) = (F, \phi_j^h) \qquad (U^h \in \bar{S}^h \subset \bar{S}).$$

Residuals of an approx. u^h: $r_j^h = (F - Lu^h, \phi_j^h)$

Error $V^h = U^h - u$ smooth (after distributed relaxation) $\Rightarrow$

Coarse-grid problem:

$$\therefore \quad \left(L(u^h + I_{2h}^h V^{2h}), I_{2h}^h \phi_i^{2h}\right) = \left(F, I_{2h}^h \phi_i^{2h}\right)$$

Linear case: $\left(L I_{2h}^h V^{2h}, I_{2h}^h \phi_i^{2h}\right) = \sum \omega_{ij} r_j^h$

where $I_{2h}^h \phi_i^{2h} = \sum \omega_{ij} \phi_j^h$ (basic test functions)

Simplified: $(LV^{2h}, \phi_i^{2h}) = \sum \omega_{ij} r_j^h$

Nonlinear: $(LU^{2h}, \phi_i^{2h}) = \left(L(I_h^{2h} u^h), \phi_i^{2h}\right) + \sum \omega_{ij} r_j^h$

Using finite-elements weighting

① Easy to automate

② Depends on choice of I_{2h}^{h} of test functions. "Natural" I_{2h}^{h} does not always exist, and sometimes is not desired.

(Higher-order interpolations in first cycles, or in case of operators of order > 2).

③ Is difficult to apply to finite-difference formulations, especially near boundaries.

(Need to compute the equations in their finite-element weights).

Important: Integral-preserving weighting.

Rules for weighting near boundaries - depending on BC.

Difficulties with general triangulation

1. Data structure: expensive, non-adaptible
2. Constructing relaxation with efficient smoothing (e.g., line relaxation).
3. Defining auxiliary coarse grids
4. Writing general interpolation routines.
5. Unpredictable ?

Required only when elements correspond to physical elements

Alternatives

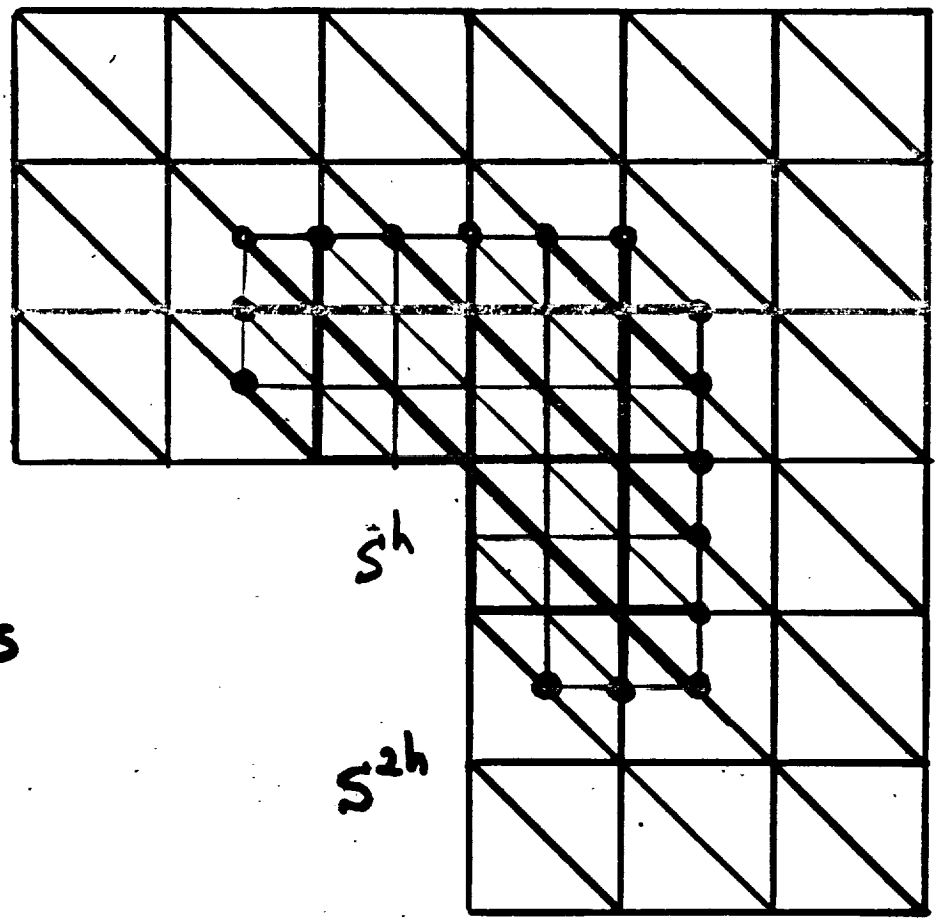
1. FEM based on uniform grids
2. FEM based on coarse triangulation of the user.

Adaptible Finite Elements

Local refinement

or

order increase



Union of uniform grids

- compatibility (interpolation)

Same Sequence of levels $\bar{S}^h, S^{2h}, \ldots$ but not coextensive: finer levels near singularities, etc

Same multi-grid algorithm: relaxing on $\bar{S}^h$ then on S^{2h} (giving also coarse-grid-correction to $\bar{S}^h$), etc.

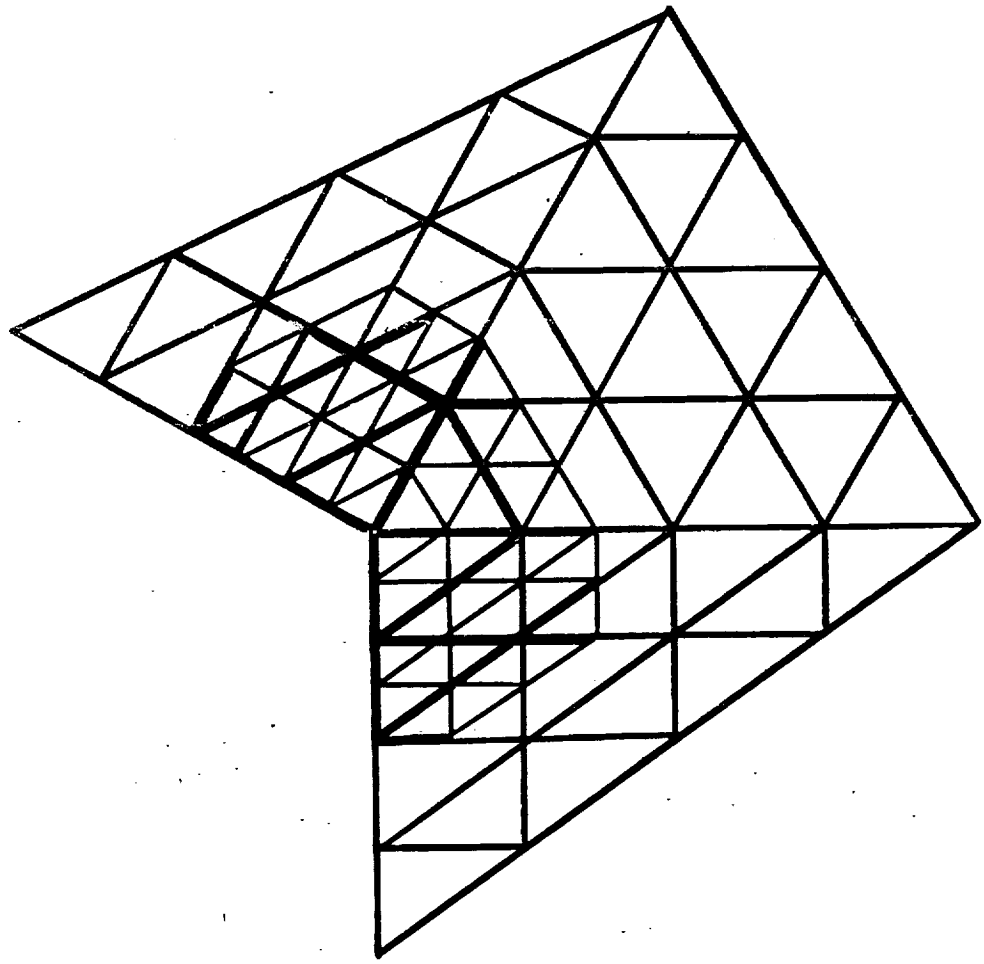
Uniform-grid relaxations with equidistant differencing.

Minimal bookkeeping $\leftrightarrow$ **Flexible structure**

Refinement = extension of a uniform fine grid.

Adaptible Uniform Subdivisions.

Coarse triangulation given.



Data structure: by levels, by super-elements

Natural adaptation error functional $E = \mathcal{E}(u^h) - \mathcal{E}(U)$

$$\Delta E = \mathcal{E}(u^{2h}) - \mathcal{E}(u^h)$$

5. MULTI-GRID ANALYSIS, PREDICTION AND OPTIMIZATION

Analysis of Relaxation - Example $a\partial_{xx} + c\partial_{yy}$

$$LU_{\alpha,\beta} \equiv a\,\frac{U_{\alpha+1,\beta} - 2U_{\alpha\beta} + U_{\alpha-1,\beta}}{h^2} + c\,\frac{U_{\alpha,\beta+1} - 2U_{\alpha\beta} + U_{\alpha,\beta-1}}{h^2} = F_{\alpha\beta}$$

$u_{\alpha\beta}$ = approx. before the sweep, $\bar{u}_{\alpha\beta}$ = approx. after

$$a\,\frac{u_{\alpha+1,\beta} - 2\bar{u}_{\alpha\beta} + \bar{u}_{\alpha-1,\beta}}{h^2} + c\,\frac{u_{\alpha,\beta+1} - 2\bar{u}_{\alpha\beta} + \bar{u}_{\alpha,\beta-1}}{h^2} = F_{\alpha\beta}$$

Errors $\quad v_{\alpha\beta} = U_{\alpha\beta} - u_{\alpha\beta}, \quad \bar{v}_{\alpha\beta} = U_{\alpha\beta} - \bar{u}_{\alpha\beta}$

$$a\,\frac{v_{\alpha+1,\beta} - 2\bar{v}_{\alpha\beta} + \bar{v}_{\alpha-1,\beta}}{h^2} + c\,\frac{v_{\alpha,\beta+1} - 2\bar{v}_{\alpha\beta} + \bar{v}_{\alpha,\beta-1}}{h^2} = 0$$

$$\bar{v}_{\alpha\beta} = \frac{a\,v_{\alpha+1,\beta} + a\,\bar{v}_{\alpha-1,\beta} + c\,v_{\alpha,\beta+1} + c\,\bar{v}_{\alpha,\beta-1}}{2a + 2c}$$

Averaging $\Longrightarrow$ fast smoothing

$\Downarrow$

Convergence

$\|\bar{v}\| \leq \|v\|$

$$\frac{\|\delta v\|}{\|v\|} \geq \alpha \Rightarrow \frac{\|\delta\bar{v}\|}{\|\delta v\|} \leq \beta < 1$$

Localness of Relaxation

$u' = u$ in some subdomain Ω'

$\bar{u} = Relax(u)$, $\bar{u}' = Relax(u')$,

$w = \bar{u}' - \bar{u}$. Then, for $(\alpha,\beta) \in \Omega'$,

$$a(-2w_{\alpha\beta} + w_{\alpha-1,\beta}) + c(-2w_{\alpha\beta} + w_{\alpha,\beta-1}) = 0$$

or

$$w_{\alpha\beta} = \frac{a w_{\alpha-1,\beta} + c\, w_{\alpha,\beta-1}}{2a + 2c}$$

$$\leq \frac{1}{2} \max\{w_{\alpha-1,\beta}, w_{\alpha,\beta-1}\}$$

1	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{27}$	...
⋮	↑	↑	↑	
1	.334	.113	.039	w on Ω'
1	.336	.117	.046	$a=c=1$
1	.344	.133	.066	
1	.375	.187	.133	
1	.500	.375	.344	
1	1	1	1	...

2h into Ω', $w_{\alpha\beta}$ decreases by an order of magnitude

[factor $\frac{1}{9}$ --- Factor $\frac{1}{25}$ if boundary discrepancy highly oscilatory 1, -1, 1, -1 ...]

$\bar{u}' = \bar{u}$ even if $u' \neq u$ just few h away

$\Rightarrow$ **Stability**, necessary condition for relaxation.

Mode (Fourier) Analysis

$$\bar{v}_{\alpha\beta} = \sum_{|\theta| \le \pi} \bar{A}_\theta \, e^{i(\alpha\theta_1 + \beta\theta_2)},$$

$\theta = (\theta_1, \theta_2)$

$|\theta| = \max |\theta_j|$

$\theta_j \cong \theta_j + 2\pi$

Ignoring boundary (∞ space):

$$\left[a e^{i\theta_1} + c e^{i\theta_2}\right] A_\theta \cancel{e^{i(\alpha\theta_1 + \beta\theta_2)}}$$

$$+ \left[-2(a+c) + a e^{-i\theta_1} + c e^{-i\theta_2}\right] \bar{A}_\theta \cancel{e^{i(\alpha\theta_1 + \beta\theta_2)}} = 0$$

Amplification factor of the θ component:

$$\mu(\theta) = \left|\frac{\bar{A}_\theta}{A_\theta}\right| = \left|\frac{a e^{i\theta_1} + c e^{i\theta_2}}{2(a+c) - a e^{-i\theta_1} - c e^{-i\theta_2}}\right|$$

$\mu(\theta) \longrightarrow 1$ as $|\theta| \to 0$

$\mu(\theta) = 1 - O(h^2)$ for $|\theta| = O(h)$

$\mu(\pi, \pi) = \frac{1}{3}$

<u>Aliasing:</u>

On coarse grid with meshsize h_c: $\theta_j \cong \theta_j + 2\pi \frac{h}{\ldots}$

Components on G^h invisible on G^{h_c}: $\rho\left(\frac{h}{h_c}\right)\pi < |\theta| \le \pi$

<u>Smoothing factor</u> $\quad \bar{\mu} = \max\limits_{\rho\pi \le |\theta| \le \pi} \mu(\theta)$

For $\rho = \frac{1}{2}$, $a = c$: $\quad \bar{\mu} = .5$

<u>Degeneracy</u>: $\quad \mu(\theta_1, 0) = \left| \dfrac{c + a e^{i\theta_1}}{c + 2a - a e^{-i\theta_1}} \right| \longrightarrow 1$

$\left(\frac{a}{c} \to 0\right)$

Indeed, for $a=0$ the system is <u>decomposed</u> to line systems $u_{yy} = F$. Any (θ_1, ε) slowly converges in any point relaxation. Including $|\theta_1| > \frac{\pi}{2}$. $\Rightarrow$ <u>slow smoothing</u>

Cure: Block relaxation.
One block for each decoupled system.

<u>y-Line Relaxation</u> $\quad \mu(\theta) = \left| \dfrac{a e^{i\theta_1}}{2a + 2c - a e^{-i\theta_1} - c e^{-i\theta_2} - c e^{i\theta_2}} \right|$

$$\bar{\mu} = \max\left\{ \mu\left(\tfrac{\pi}{2}, 0\right), \mu\left(0, \tfrac{\pi}{2}\right) \right\} = \max\left\{ .45, \frac{a}{a+2c} \right\}$$

<u>Alternating Direction Line Relaxation</u> $\quad \bar{\mu}^2 \le .45$

for any a/c

Convergence factors prediction

$\mathring{\mu}$ = convergence factor per Work-Unit (WU).

A relaxation sweep costs 1 WU on grid h, $\frac{1}{4}$ WU on grid 2h, $\frac{1}{16}$ WU on 4h, etc. Other work neglected

1. Estimating $\mathring{\mu}$ by $\bar{\mu}$.

$V = \Sigma A(\theta) e^{i\theta\cdot x/h}$, $\theta\cdot x = \theta_1 x_1 + \theta_2 x_2$

Components $\frac{\pi}{2} \le |\theta| \le \pi$ are reduced by relaxing on grid h, $\frac{\pi}{4} \le |\theta| \le \frac{\pi}{2}$ reduced on grid 2h, etc.

$\Rightarrow$ A cycle with r sweeps on each grid reduces all error components by factor $\le \bar{\mu}^r$, and costs $r + \frac{r}{4} + \frac{r}{16} + \cdots = \frac{4r}{3}$ WU.

$$\Rightarrow \quad \mathring{\mu}^{\frac{4r}{3}} \approx \bar{\mu}^r \quad \Rightarrow \quad \mathring{\mu} \approx \bar{\mu}^{3/4}$$

Neglected Effects:

(i) Boundaries, non-constant coefficients — Not significant

(ii) Initial error distribution $\mathring{\mu} \approx \left\{ \begin{matrix} \text{weighted} \\ \mu(\theta)^r \end{matrix} \right\}^{3/4r}$

(iii) Interpolations, finer-grid relaxations

errors (in predicting $\log \mathring{\mu}$) $< 20\%$

2. Full multi-grid analysis

$$V = \sum A(\theta)\, e^{i\theta\cdot x/h}$$

Relaxation: $A(\theta) \leftarrow \lambda(\theta)\, A(\theta)$, $\quad \mu(\theta) = |\lambda(\theta)|$ or $\sigma(\lambda(\theta))$

On grid 2h $\quad e^{i\theta\cdot x/h} = e^{i2\theta\cdot x/2h} \Longrightarrow$

$\theta = (\theta_1, \theta_2)$ coincides (aliases) with $(\theta_1+\pi, \theta_2)$, $(\theta_1, \theta_2+\pi)$, $(\theta_1+\pi, \theta_2+\pi)$

Coarse-grid cycle (residual transfer, solving the 2h equations, and interpolating correction back to grid h):

$$A_\theta \leftarrow \mathcal{M} A_\theta, \quad \text{where} \quad A_\theta = \begin{pmatrix} A(\theta_1, \theta_2) \\ A(\theta_1+\pi, \theta_2) \\ A(\theta_1, \theta_2+\pi) \\ A(\theta_1+\pi, \theta_2+\pi) \end{pmatrix}$$

Full multi-grid cycle (r relaxation sweeps + coarse-grid cycle):

$$A_\theta \leftarrow \mathcal{N}_r A_\theta, \quad \text{where} \quad \mathcal{N}_r = \mathcal{M} \begin{pmatrix} \lambda(\theta_1,\theta_2)^r & 0 & 0 & 0 \\ 0 & \lambda(\theta_1+\pi,\theta_2)^r & 0 & 0 \\ 0 & 0 & \lambda(\theta_1,\theta_2+\pi)^r & 0 \\ 0 & 0 & 0 & \lambda(\theta_1+\pi,\theta_2+\pi)^r \end{pmatrix}$$

$$\mathring{\mu} \approx \max_\theta \sigma(\mathcal{N}_r)^{\frac{3}{4r}} \qquad (\sigma = \text{spectral radius}).$$

eq. & relax.	$\bar{\mu}$	$\bar{\mu}^{3/4}$	$\sigma(\mathcal{N}_3)^{3/4}$	10% perturbed	$\mathring{\mu}$ (asym. experimental)
Poisson GS	.500	.595	.547	.562	.550 ± .015
Stokes DGS	.500	.595		.714	.722 ±

Multi-Grid Mode Analysis

$$v^h = \sum_{|\theta| \le \pi} A(\theta) e^{i\theta\cdot x/h}$$

$\theta^0 = (\theta_1, \theta_2, \ldots, \theta_d)$, $\quad |\theta^0| = \max |\theta_j| \le \frac{\pi}{2}$ — a 'basic' mode.

Its 2h aliases (harmonics) are $\theta^\alpha \equiv \theta^0 \pmod{\pi}$, $\quad (\alpha = 1, 2, \ldots, 2^d - 1)$

$$I_h^{2h} e^{i\theta^\alpha\cdot x/h} = \rho(\theta^\alpha) e^{i\theta^0\cdot x/h}$$

$$I_h^{2h} v^h(x) = \sum \rho_\nu v^h(x + \nu h) \Rightarrow \rho(\theta) = \sum \rho_\nu e^{i\theta\cdot\nu}$$

$$I_{2h}^h e^{i\theta^0\cdot x/h} = \sum_\alpha R(\theta^\alpha) e^{i\theta^\alpha\cdot x/h}$$

e.g., linear $I_{2h}^h \Rightarrow R(\theta) = \frac{1}{2}(1 + \cos\theta)$

$$L_h e^{i\theta\cdot x/h} = B_h(\theta) e^{i\theta\cdot x/h}$$

$B_h(\theta)$ = the 'symbol'

<u>Relaxation sweep:</u> $\quad \overline{A(\theta)} = \lambda(\theta) A(\theta)$

<u>Multi-grid cycle</u> (r relaxation sweeps + one coarse-grid correction):

$$A_{\theta^0} = \begin{pmatrix} A(\theta^0) \\ A(\theta^1) \\ \vdots \\ A(\theta^{2^d-1}) \end{pmatrix}$$

$$\overline{A}_{\theta^0} = \mathcal{N}_r(\theta^0) A_{\theta^0}$$

$$\mathcal{N}_r(\theta^0)_{\alpha,\beta} = \left[\delta_{\alpha\beta} - \tau B_h(\theta^\alpha) R(\theta^\alpha) B_{2h}(2\theta)^{-1} \rho(\theta^\beta)\right] \lambda(\theta^\beta)^r$$

$q \times q$ matrices; $\quad \tau = 1 \pm \delta$, to simulate partial 2h solution

$$\mathring{\mu} \approx \max_{|\theta| \le \frac{\pi}{2}} \sigma(\mathcal{N}_r)^{3/4r}$$

σ = spectral radius

Rigorous Upper Bounds (Math. Comp. 1977, App. C)

In rectangular domains, estimate $\sum \overline{A(\theta)}^2 / \sum A(\theta)^2$ by

$\max_\theta \| \mathcal{N}_r(\theta) \|^2_{L_2}$ + perturbation (estimated inductively)

For 5-point Poisson and a WSD-CGC-WSD cycle (CGC = coarse-grid correction with 2 such cycles solving the 2h equatio

L_2-convergence factor $\leq .19$, costs 56 operations-per-poin

<u>In practice (GS)</u>: factor $\approx .09$, costs 27 operations-per-point

P.O. Frederickson (unpublished): similar result

R.P. Federenko (1964): number of operation to reduce error by .01 on grid with n points is $W(n) \leq 210\,000\, n + W(10^6)$.

N.S. Bakhvalov (1966): $W(n) \leq Cn$, for general 2nd order equations with continuous coefficients.

General domains (finite-elements formulations, nested spaces, trial space = test space, unspecified C):

R.A. Nicolaides (Math. Comp., 1977)

W. Hackbusch

R. Bank and T. Dupont

Mode Analysis Applications

- **Optimizations**

  Relaxation: type, directions, parameters.

  Interpolation order: $\max\{m,\ s-(i-1)p\}$

  (i-th interpolation, $\hat{u}(\theta) = O(|\theta|^s)$, m = order of L, p = order of approximation)

  Not followed by relaxation sweeps if $s - pi > m$

  Residual-weighting: Sometimes injection O.K. Usually the transfer should preserve integrals (sums).

  Switching Parameters: $\eta \approx \bar{\mu}$, $\delta \approx \bar{\mu}^r$

- **Debugging**

  Mode analysis $\longleftrightarrow$ Numerical experiments (Esp. one-mode experiments)

  Weighted $\mathring{\mu}$ $\longleftrightarrow$ Random start

  Comparisons for the separate processes.

6. GENERALIZED RELAXATION SCHEMES.

Relaxation Schemes

- Gauss-Seidel (GS). Immediate displacement.
- Simultaneous Displacement (SD). Parallel processing Jacobi
- Weighted Relaxation: $\overline{\delta u}_\nu = \sum_{\gamma \in \Gamma} \omega_\gamma \, \delta u_{\nu+\gamma}$

  e.g., relaxation parameter ω_0 $(\Gamma = \{0\})$: Successive Over-Relaxation (SOR), Jacobi under-relaxation. Weighted Simultaneous displacement (WSD).
- Block relaxation: e.g., line relaxation. Alternating Direction Line Relax (ADLR). LGS, LSD, LSOR, etc.
- Collective relaxation: for a system of equations, simultaneous solution of the equations at each point.
- Distributed relaxation: Changing several unknowns for each equation. Distributed Gauss-Seidel (DGS) — probably the best general smoother.

Optimizing relaxation: Minimize $\bar{\mu}$ (or $\hat{\mu}$) per operation, but keep it robust.

SMORATE - Smoothing Calculations

Computes:

$$\bar{\mu} = \max_{\rho\pi \le |\theta| \le \pi} \mu(\theta)$$ smoothing factor

$$\mu_r = \left\{ \frac{\int_{\rho\pi \le |\theta| \le \pi} A_\theta^{(r)\,2}\, d\theta}{\int_{\rho\pi \le |\theta| \le \pi} A_\theta^{(0)\,2}\, d\theta} \right\}^{1/2}$$ smoothing in r iterations $A_\theta^{(r)} = \bar{A}_\theta^{(r-1)}$

$$\mathring{\mu} = \bar{\mu}^{1-2^{-d}} \leftarrow \mathring{\mu}_r = \mu_r^{\frac{1}{r}(1-2^{-d})}$$ m.g. convergence factor per work u

<u>Input</u>: dependent & independent dimensions, relaxation scheme
ρ, r, A_θ ($\equiv 1$ in default)

<u>Output</u>: $\mu_r, \bar{\mu}, \mathring{\mu}, \mathring{\mu}_r$
At maximum: θ, eigenvalue, eigenvector
[Stability (error decay factor) analysis]

<u>Nonlinear Problems</u>: $\bar{\mu}$ etc. depend on local (frozen) operator. May output worst (largest $\bar{\mu}$) case only.

6-3

TABLE 1. Theoretical smoothing and MG-convergence rates.

L_h	d	Relax. Scheme	ω	$\hat{\rho}$	$\bar{\mu}$	$\overset{o}{\mu}$	$\lvert \ln \overset{o}{\mu} \rvert^{-1}$	add	mult	W_M
Δ_h	1	SOR	1	1:3	.557	.693	2.73	2	1	9.0
				1:2	.447	.668	2.49	3	2	6.9
				2:3	.378	.723	3.08	3	2	7.5
	2	SOR	1	1:3	.667	.697	2.77	4	1	6.8
			.8	1:2	.552	.640	2.24	5	2	4.1
			1		.500	.595	1.92	4	1	3.5
			1.2		.552	.640	2.24	5	2	4.1
			1	2:3	.400	.601	1.96	4	1	2.9
		LSOR	1	1:2	.447	.547	1.66	8	4	3.1
		ADLR	1		.386	.490	1.40	8	4	2.6
			.8		.456	.555	1.70	8	4	3.1
		SD	.8	1:2	.600	.682	2.61	5	2	4.8
		WSD	1.17, .195		.220	.321	0.88	9	3	1.6
			1.40, .203		.506	.600	1.96	9	3	3.6
	3	SOR	1	1:3	.738	.746	3.42	6	1	7.8
				1:2	.567	.608	2.01	6	1	3.7
				2:3	.441	.562	1.73	6	1	2.0
$\Delta_h^{(4)}$	2	SOR	.8	1:2	.581	.665	2.46	9	3	9.1
			1		.534	.625	2.13	8	2	7.9
			1.2		.582	.666	2.46	9	3	9.1
		LSOR	1		.484	.580	1.84	14	7	6.8
	3	SOR	1		.596	.636	2.21	12	2	7.0
$\partial_{xx} + 2\partial_x\partial_y + \partial_{yy}$	2	SOR	1	1:2	.62	.699	2.79	8	2	5.2
		LSOR,ADLR			.447	.547	1.66	12	5	3.1
Δ_h^2	2	SOR	1	1:2	.802	.847	6.04	11	3	11.1
			1	2:3	.666	.798	4.43	11	3	6.5
		WSD	1.552, .353	1:2	.549	.638	2.22	17	5	4.1
			1.4 , .353		1.03	div.	div.	17	5	div.
		WSDΔ	1.552, .353		.549	.638	2.22	14	4	4.1
NAVIER - STOKES		CSOR								
Rh = 0	2	downstr.	1, .5	1:2	.800	.846	5.98	18	6	11.0
any			1, .5		.800	.846	5.98	33	16	11.0
100			1.1, .5		1.73	div.	div.	33	16	div.
100			.8, .5		.93	.947	18.7	33	16	34.5
10		upstream	1, .5		.884	.912	10.8	33	16	20.0
100			1, .5		.994	.995	220.	33	16	400.
100			.8, .5		.984	.983	83.	33	16	150.
0	3	downstr.	1, .5		.845	.863	6.79	33	8	10.7
any			1, .5		.845	.863	6.79	60	25	10.7
10		upstream	1, .5		.874	.889	8.49	60	25	13.4
100			1, .5		.989	.990	100.	60	25	160.
STOKES' (Rh = 0)		SOR	1,.33		.707					
STOKES		DGS			.500	.595	1.92	24	4	.65

.54

6-4

TABLE 1. (Cont'd. Here $d=2$, $\hat{\rho}=1:2$)

L_h	Relax. Scheme	ω	$\bar{\mu}$.
$\partial_{xx} + \varepsilon\partial_{yy}$, $\varepsilon<<1$	SOR, xLSOR	any	$1-O(\varepsilon)$
$a\partial_{xx} + c\partial_{yy}$	yLSOR	1	$\max(5^{-1/2}, \frac{a}{a+2c})$
$(q = \min(\frac{a}{c}, \frac{c}{a}))$	ADLR		$5^{-1/4}(1+2q)^{-1/2}$
	SD, yLSD, ADLSD	1	1
	SD	$(2q+2)/(3q+2)$	$(q+2)/(3q+2)$
	yLSD	$(2a+2c)/(2a+3c)$	$(2a+c)/(2a+3c)$
	ADLSD	2/3, 2/3	$\leq 3^{-1/2} = .577$
$\Delta_h - \frac{\eta}{h}\partial_x$	yLSOR	1	$\max\left(\frac{1-\eta}{3-\eta}, \left[\frac{1-\eta+\eta^2/4}{5+\eta+\eta^2/4}\right]^{1/2}\right)$
$\Delta_h - \frac{\eta}{h}\partial_x^-$	yLSOR+	1	$\max\left(\frac{1}{3}, [5+6\eta+2\eta^2]^{-1/2}\right)$
$(\eta>0)$	yLSOR-		$\max\left(\frac{1}{3}, \left\lvert\frac{1+\eta}{2+\eta+i}\right\rvert\right)$
	yLSORs		$\leq 3^{-1/2} = .577$
Navier - Stokes with large Rh in 2 or 3 dimensions	SOR (pressure corrected by the continuity equation), downstream or upstream, with any relaxation parameters.		$\geq 1 - \frac{2}{Rh}$

Smoothing general elliptic equations

Ellipticity: (Thomée, 1964): $L_h e^{i\theta\cdot x/h} \neq 0$ for $h<h_*$, $0<|\theta|\leq\pi$, where $\theta\cdot x=\theta_1x_1+\dots+\theta_dx_d$, $|\theta|=\max|\theta_j|$.
[For L_h with continuous coefficients].

h-ellipticity As $\theta\not\equiv 0 \pmod{2\pi}$ changes from θ^I to $\theta^{II}\equiv\theta^I \pmod{2\pi}$, $L_h e^{i\theta\cdot x/h}$ describes a loop which does not enclose 0.
[e.g., $\mathrm{Re}\{L_h e^{i\theta\cdot x/h}\}>0$ for all $0<|\theta|\leq\pi$].

Theorem 1. L_h symmetric elliptic, and the relaxation (with parameter ω) is consistently ordered $\Rightarrow \exists h_*, \omega_*$ such that $\bar{\mu}(h,\omega)\leq\hat{\mu}(\omega)<1$, $\forall\ 0<h\leq h_*$, $0<\omega<\omega_*$

Theorem 2. If most recent values are used in the relaxation, then $\hat{\mu}(\omega)<1$ for all $0<\omega<2$.

Theorem 3. L_h is h-elliptic $\Rightarrow \exists$ relaxation method with $\bar{\mu}(h)\leq\hat{\mu}<1$ for all $0<h$

General Elliptic Problems

<u>Local mode analysis</u> (ignoring boundaries and Coefficients changes) accurately predict smoothing rates of general non-linear problems.

$$L_h U(x) \equiv \frac{1}{h^m} \sum_{\nu \in N} \overset{P\times P}{b_\nu(h)} \; \overset{P}{U(x+\nu h)} = F(x)$$

$x = (x_1, \dots, x_d)$
$\nu = (\nu_1, \dots, \nu_d)$
ν_j integer

$u, \bar{u}$ before & after relax.

$N = N_- \cup N_+ \cup N_0$ (old values, new values, simultaneou

Errors: $\bar{U}_\alpha = U_\alpha - \bar{u}_\alpha = \sum_{|\theta| \le \pi} \bar{A}_\theta e^{i\theta\cdot\alpha}$,

$\alpha = (\alpha_1, \dots, \alpha_d)$
$\theta = (\theta_1, \dots, \theta_d)$
$\theta\cdot\alpha = \sum \theta_j \alpha_j$

$$\left[B_+ + B_0\right] \bar{A}_\theta + B_- A_\theta = 0, \qquad B_+(\theta,h) = \sum_{\nu \in N_+} b_\nu(h) e^{i\theta\cdot}$$

With relaxation parameter ω

$$\left[B_+ + \tfrac{1}{\omega} B_0\right] \bar{A}_\theta + \left[B_- + \left(1 - \tfrac{1}{\omega}\right) B_0\right] A_\theta = 0$$

<u>amplification factor</u> $\mu(\theta,h,\omega) = \left| \dfrac{B_- + (1-\frac{1}{\omega})B_0}{B_+ + \frac{1}{\omega}B_0} \right|$

or largest |eigen value| of $\left[B_+ + \frac{1}{\omega}B_0\right]\lambda + \left[B_- + (1-\frac{1}{\omega})B_0\right] = s$

<u>smoothing rate</u> $\bar{\mu}(h,\omega) = \max\limits_{\delta \le |\theta| \le \pi} \mu(\theta,h,\omega) \overset{?}{<} 1-$

$\delta > 0$, independent of h.

Ellipticity

<u>Differential</u> operator $\left| \overset{p\times p}{L} \overset{p}{A} e^{i\theta\cdot x} \right| \neq 0, \quad \forall \theta \neq 0$

(No periodic solution to the homogeneous eq.)

<u>Difference</u> Operators (Thomée, 1964):

L_h elliptic if $L_h e^{i\theta\cdot x/h} \neq 0$ for sufficiently small h, $\forall \theta \neq 0$, $|\theta| \leq \pi$.

$$L_h e^{i\theta\cdot x/h} = B(\theta,h)\, e^{i\theta\cdot x/h}, \qquad B(\theta,h) = \sum_{\nu\in N} b_\nu(h)\, e^{i\theta\cdot\nu}$$
$$= B_+ + B_- + B_0$$

$$L_h \text{ elliptic} \iff B(\theta,0) \neq 0, \quad \forall\, 0 < |\theta| \leq \pi$$

<u>Example:</u> $L_h = -\Delta_h + a\partial_x = \frac{1}{h^2}$

	-1	
$-1-\frac{ah}{2}$	4	$-1+\frac{ah}{2}$
	-1	

$$B(\theta,h) = 4 - e^{i\theta_1} - e^{-i\theta_1} - e^{i\theta_2} - e^{-i\theta_2} + \frac{ah}{2}\left(e^{i\theta_1} - e^{-i\theta_1}\right)$$
$$= 2(2-\cos\theta_1 - \cos\theta_2) + ah\, i \sin\theta_1$$
$$B(\theta,0) = 2(2-\cos\theta_1 - \cos\theta_2) > 0 \quad \forall\, 0 < |\theta| < \pi$$

<u>Theorem</u> L_h, L_h^0 symmetric elliptic $\Longrightarrow$

$\exists\, h_*, \omega_*$ s.t. $\bar{\mu}(h,\omega) \le \tilde{\mu}_\omega < 1$, $\forall \begin{matrix} 0<h\le h_* \\ 0<\omega\le\omega_* \end{matrix}$

<u>Proof</u> $\lambda = 1+\lambda'\omega \Longrightarrow B + \lambda'(B_0 + \omega B_+') = 0$

$h=0 \Rightarrow B>0,\ B_0>0 \Longrightarrow \mathrm{Re}\,\lambda' < 0$ for $\omega<\omega_*$

<u>Immediate displacement</u>: Most recent values are used

$\Rightarrow B_- = \overline{B_+}$ $\Longrightarrow \mu(\theta) = \left|\dfrac{a+ib}{c-ib}\right|$

$a \ne c$ for $\omega \ne 2$; $a+c = B_+ + B_- + B_0 = B \ne 0$ for $\theta \ne$

$\Rightarrow$ For $0<\omega<2$ and $\theta\ne 0$: $\mu(\theta) < 1$

<u>Corollary</u>: L_h, L_h^0 symmetric elliptic, relaxation with immediate displacement $\Longrightarrow$

$\exists\, h_*$ s.t. $\bar{\mu}(h,\omega) \le \tilde{\mu}_\omega < 1$, $\forall \begin{matrix} 0<h\le h_* \\ 0<\omega<2 \end{matrix}$

Degenerate Cases

$L_h = a\,\partial_{xx} + c\,\partial_{yy}$, Gauss-Seidel Relax $\Rightarrow$

$$\mu\left(\tfrac{\pi}{2}, 0\right) = \left|\frac{c + ia}{2a + c - ia}\right| \to 1 \quad \text{as } a \to 0$$

(transonic flow)

y-Line Relaxation: $\bar{\mu} = \max\left(.45, \dfrac{a}{a+2c}\right)$

Alternating Direction Line Relaxation (ADLR):

$\bar{\mu} \le .45$ for any a/c.

Singular Perturbation: e.g., $-\varepsilon \Delta_h + \partial_x$

Above definition of Ellipticity - inadequate: $h_* \le$

It also allows shifts: $L'_h U(x) = L_h U(x + \nu h)$

$$\Rightarrow B' = e^{i\theta\cdot\nu} B \neq 0$$

Definition L_h is h-elliptic if the "loops"

$\{ B(\theta, h) : \theta_0 \le \theta \le \theta_1 \equiv \theta_0 \pmod{\pi} \}$ have index zero

e.g., $\operatorname{Re} B(\theta, h) > 0$, $\forall\, 0 < |\theta| \le \pi$.

<u>Theorem</u>: $B(\theta,h) \neq 0 \Rightarrow \exists$ unique translation $e^{i\theta\cdot}$ such that $e^{i\theta\cdot\nu} B(\theta,h)$ is h-elliptic.

<u>Theorem</u> L_h is h-elliptic $\Longleftrightarrow \exists$ a sequence of operators $L_h^0 = I,\ L_h^1, \ldots, L_h^m = L_h$ such that, a relaxation scheme on L_h^j where L_h^{j-1} is the implicit part, has smoothing rate $\bar{\mu} < 1$.

In the singular-perturbation case, demand also $|B(\theta,h)| \geq c(\theta)$ uniformly in ε. (Non-central FDE)

$\Rightarrow \bar{\mu} \leq \mu_* < 1$ independently of ε, provided relaxation marching direction is suitable (<u>from</u> large weights; "down-stream").

Relaxing in all directions (symmetric ADLR) $\Rightarrow$ Fast convergence at all elliptic regions, degenerate or not.

7. NAVIER-STOKES EQUATIONS.

DISTRIBUTED RELAXATION

Cauchy-Riemann Equations

① $U_y - V_x = F$

② $U_x + V_y = G$

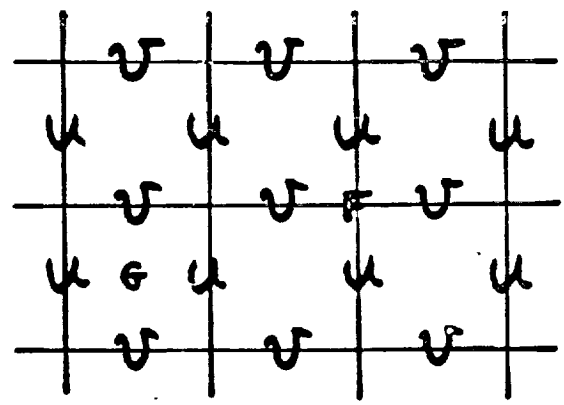
One boundary condition

$$\begin{pmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}$$

$\det \uparrow = \partial_x^2 + \partial_y^2 = \Delta$

Difference Equations:

Eq. 1 defined at grid nodes
Eq. 2 defined at cell centers

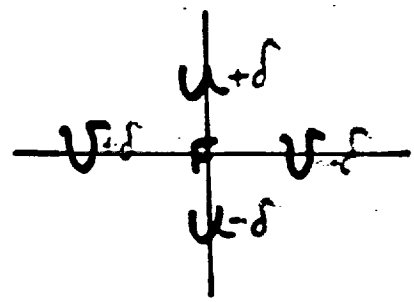


Relaxation on Equation 1:

preserves $R_2 = G - u_x - v_y$
smoothes $R_1 = F - u_y + v_x$.

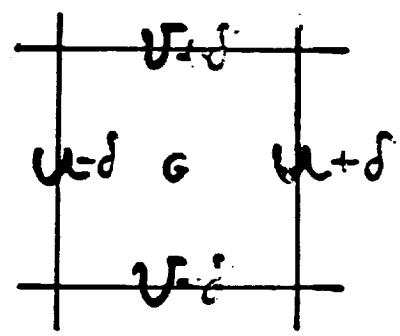
If $R_j = \sum A_j(\theta)\, e^{i\theta\cdot x/h}$ then

$\bar{A}_1(\theta) = \lambda(\theta) A_1(\theta)$, $\bar{A}_2(\theta) = A_2(\theta)$.



Relaxation on Equation 2:

$\bar{A}_1(\theta) = A_1(\theta)$, $\bar{A}_2(\theta) = \lambda(\theta)\, A_2(\theta)$



Smoothing Factor:

$$\bar{\mu} = \max_{\frac{\pi}{2} \le |\theta| \le \pi} |\lambda(\theta)| = \max \left| \frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}} \right| = .5$$

Stokes Equations

① $\Delta u - p_x = f_1$

② $\Delta v - p_y = f_2$

③ $u_x + v_y = g$

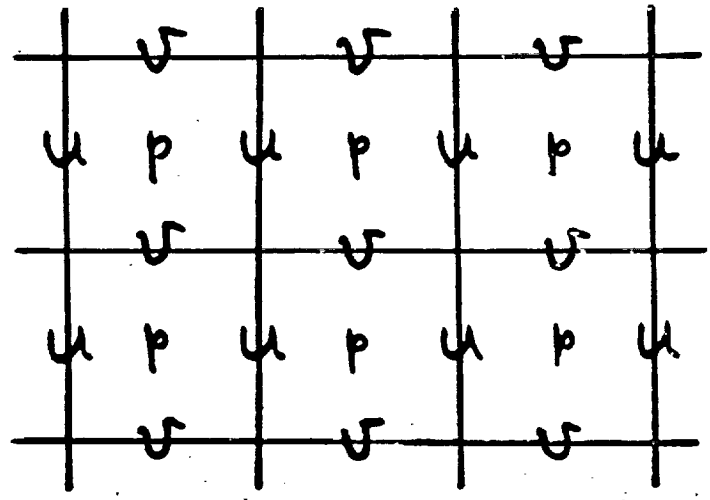
u, v given on boundaries.

$$\begin{pmatrix} \Delta & 0 & -\partial_x \\ 0 & \Delta & -\partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ g \end{pmatrix}$$

$\det \uparrow = \Delta^2$

Difference Equations:

Eq. 1	defined at	u
2		v
3		p

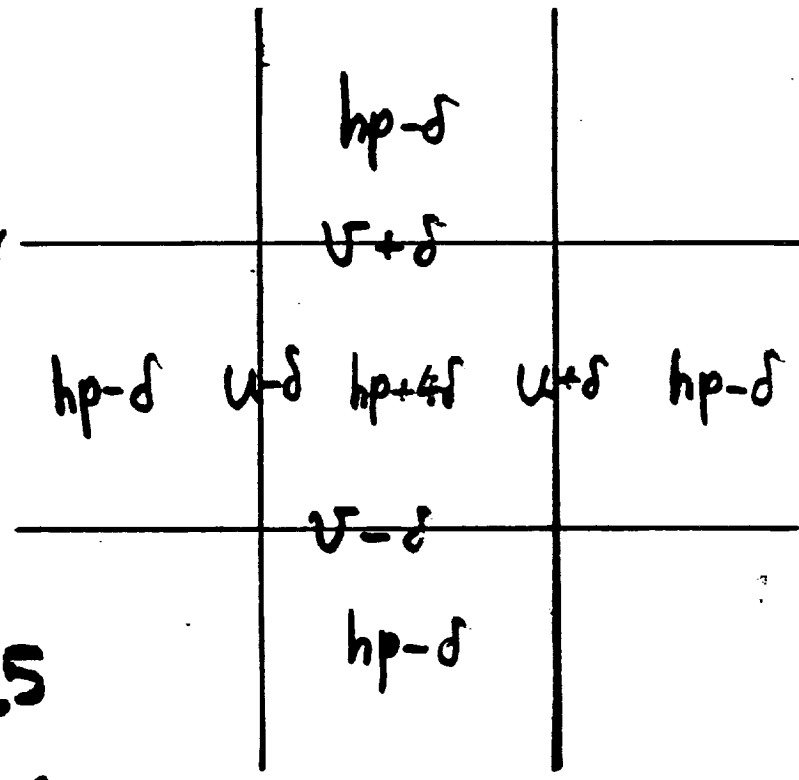


Relaxation

Eq. 1 relaxed by u

Eq. 2 relaxed by v

Eq. 3 satisfied by the simultaneous changes (preserves R_1, R_2)



$$\begin{pmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \bar{A}_3 \end{pmatrix} = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ \cdots & \cdots & \lambda \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix}$$

Smoothing factor $\max\limits_{\frac{\pi}{2} \le |\theta| \le \pi} |\lambda(\theta)| = .5$

m.g. convergence factor $\approx .65$

3D Navier-Stokes Equations

① $Qu - p_x = f_1$

② $Qv - p_y = f_2$

③ $Qw - p_z = f_3$

④ $u_x + v_y + w_z = g$

Boundary data: u, v, w

$$\begin{pmatrix} Q & 0 & 0 & -\partial_x \\ 0 & Q & 0 & -\partial_y \\ 0 & 0 & Q & -\partial_z \\ \partial_x & \partial_y & \partial_z & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \\ p \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ g \end{pmatrix}$$

$\det \uparrow = Q^2 \Delta$

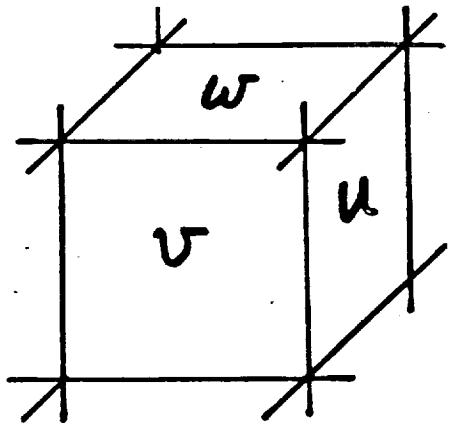
$Q = \frac{1}{R}\Delta - u\partial_x - v\partial_y - w\partial_z$

Difference Equations:

Eq. 1, 2, 3 defined at u, v, w, resp.

$$Q_h U(\underline{x}) = \frac{1}{h^2} \sum q_{\underline{\nu}} U(\underline{x} + \underline{\nu} h)$$

Eq. 4 and p – at cell centers.



Relaxation

Eq. 1, 2, 3 relaxed by u, v, w, respectively.

Eq. 4 at cell center $\underline{x} = (x, y, z)$ relaxed by

$u(x \pm \frac{h}{2}, y, z) \pm \delta$, $v(x, y \pm \frac{h}{2}, z) \pm \delta$, $w(x, y, z \pm \frac{h}{2}) \pm \delta$

$hp(\underline{x} + \underline{\nu} h) + q_{-\underline{\nu}}\, \delta$ (preserving R_1, R_2, R_3)

Smoothing factor $\approx .5$

1, 2, 3 converge faster downstream.

Numerical experiments for 2D Navier-Stokes equations:

Relaxing downstream, Cycle C algorithm, linear interpolations

$0 \leq R \leq 4096$, $\quad .65 \leq \mu \leq .75$

Further MLAT Navier-Stokes research:

- other distributed relaxation schemes ?
- Full multi-grid algorithm
- Grid adaption
- Compressible flows
- Transonic

etc.

Upstream Relaxation

$$\Delta_h - \frac{\eta}{h}\partial_x^- = \frac{1}{h^2}\begin{bmatrix} 1+\eta & -2-\eta & 1 \end{bmatrix} \qquad (\eta>0)$$

$$\approx \frac{1}{h^2}\begin{bmatrix} \eta & -\eta & 0 \end{bmatrix} \qquad (\text{large } \eta)$$

Usual $\xleftarrow{\text{GS}}$ relaxation has no smoothing

(for $\eta \to \infty$ it's a shift $v_i = v_{i+1}$)

Marching against dependence direction. $\Rightarrow$ Symmetric Relaxation

$\xleftarrow{\text{DGS}}$ Relaxation for large η: $\begin{bmatrix} u-\delta & u+\delta & u \end{bmatrix}$

Smoothing factor : $\bar{\mu} = 5^{-1/2} = .447$

Error amplification factor $.5$ (large η)

>1 $(\eta=0)$

$\xleftarrow{\text{DGS}}$ Relaxation for any η: $\begin{bmatrix} u-q\delta & u+\delta & u \end{bmatrix}$

$q = q(\eta)$. Trade-off: smoothing — error amplification

For $q = \frac{1+\eta}{2}$: error amp. $= 1$, smoothing $= \begin{cases} .354 & : \eta = 0 \\ O(\frac{1}{\eta}) & : \text{large } \eta \end{cases}$

<u>combination</u> of the Δ relaxation with the ∂_x relaxation.

Distributed Gauss-Seidel (DGS)

for asymmetric operators

$L_h = \begin{bmatrix} & d & \\ a & -a-b-c-d & c \\ & b & \end{bmatrix}$				$\bar{\mu}$	
a	b	c	d	GS	DGS *)
1	1	1	1	.500	.315
1	3	3	1	.631	.313
21	1	21	1	.913	.449
1	201	201	1	.990	.481
1	2001	2001	1	.999	.484
1	1	2001	2001	1.000	.752

*) Dinar's style:

	$u+0$	
$u+\alpha\delta$	$u+\delta$	$u+0$
	$u+\beta\delta$	

δ is chosen to satisfy the difference equation at the centr point. α, β are chosen to L_2-minimize the residual <u>changes</u> at neighboring points. For Poisson: $\alpha = \beta = \frac{4}{19}$.

Requires more operations than GS (as preferable for P...

Hyperbolic Equations

$-a\partial_{xx} + \partial_{yy}$

time-like $\downarrow$

	1	
	-2	
$-a$	$1+2a$	$-a$

Usual Line Gauss Seidel: march $\downarrow$ $\bar{\mu} \ll 1$

$\uparrow$ $\bar{\mu} \gg 1$

$\Rightarrow$ Selective Relaxation.

Distributed LGS relaxation against time-like direction

$\uparrow$

$u+p\delta$
$u-q\delta$
$u+\delta$

On the stability border: $q=1$

$p = \frac{1}{1+2a}$

$\bar{\mu} = \frac{1+2a}{1+3a}$

General System $\overset{m\times n}{A}X=b$

S. Kaczmarz
Bull. Acad. Polon
Sciences et Letter
A, 355-357
(1937)

Fully-Weighted Relaxation (FWR):

relax eq. i $\sum a_{ij} X_j = b_i$ by $\bar{x}_j = x_j + a_{ij}\delta$

(satisfy eq. i with minimum L_2 change in x)

- FWR gives all solutions, generalized inverse. Kanio Tanabe, Numer. Math. 17 (1971)
- For Δ_h, FWR has smoothing rate .80 (GS: .50).

Type of Problem	Efficiently smoothing Relaxation
Uniformly elliptic non definite, $\Delta U + CU = f$	**any** $h < h_0$
Degenerate elliptic $\varepsilon \partial_{xx} + \partial_{yy}$ or $\varepsilon \Delta + \partial_x$ or	**direction of heavy dependence** y-line relaxation $\longleftarrow$ marching (or distributed relaxation)
Hyperbolic point A depends on B	**time-like marching** A is not scanned before B (or distributed relaxation)
General System of eqs.	**Selective Symmetric Line Relaxation (SSL)** Or adaptive distributed relaxation

8. TRANSONIC FLOW PROBLEMS

APPLICATION OF MULTIGRID TO TRANSONIC FLOWS

JERRY SOUTH
ACHI BRANDT

- BRANDT & SOUTH DEVELOPED TRANSONIC MULTIGRID CODE SUMMER 1975. "FAS" METHOD DEVELOPED.

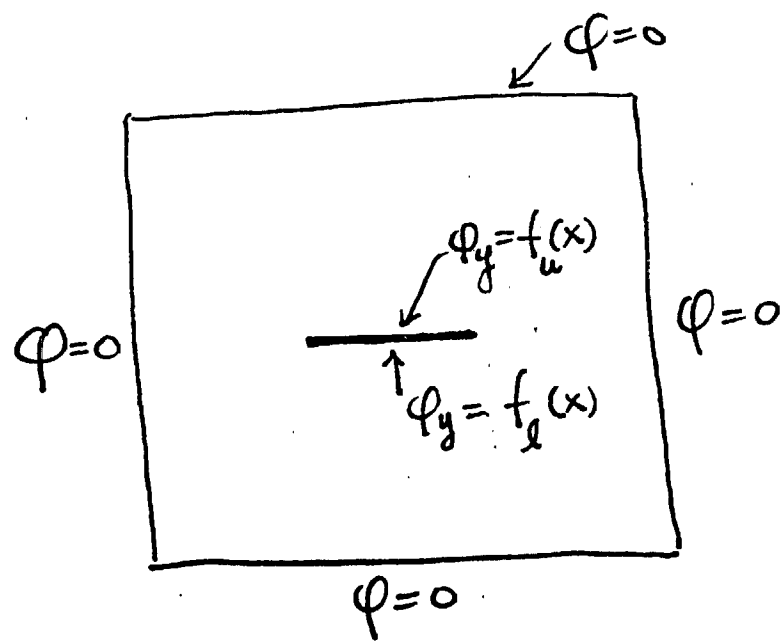
- REPORTED RESULTS FOR NONLIFTING SUPERCRITICAL FLOW UP TO $M_\infty = .95$ IN MARCH 1976. ICASE REPORT 76-8. NOTED STRETCHED GRID PROBLEM.

- REPORTED RESULTS FOR LIFTING SUPERCRITICAL FLOW UP TO $M_\infty = .85$ IN NOV. 1976 AT MEETING OF SOC. FOR. ENGR. SCI. STRETCHED GRID PROBLEM IMPROVED WITH FIRST-CUT, ALTERNATING-DIRECTION RELAXATION SWEEPS.

- LITTLE EFFORT SINCE THEN, ON TRANSONICS... KLUNKER HAS ATTEMPTED 3D STRAIGHT WINGS IN INCOMPRESSIBLE FLOW WITH LITTLE SUCCESS.

TRANSONIC SMALL-DISTURBANCE EQUATION

$$\left\{\left[K-\left(\frac{\gamma+1}{2}\right)M_\infty^2\varphi_x\right]\varphi_x\right\}_x + \varphi_{yy} = 0$$



CONSERVATIVE FINITE-DIFFERENCE ANALOG

$$P_{ij} - \left[\mu_{ij} P_{ij} - \mu_{i-1,j} P_{i-1,j} \right] + Q_{ij} = 0$$

$$P_{ij} = U_{ij} \frac{\varphi_{i+1,j} - 2\varphi_{ij} + \varphi_{i-1,j}}{\Delta x^2}$$

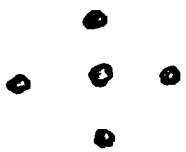
$$U_{ij} = K - (\gamma+1) M_\infty^2 \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x}$$

$$Q_{ij} = \frac{\varphi_{i,j+1} - 2\varphi_{ij} + \varphi_{i,j-1}}{\Delta y^2}$$

$$\mu_{ij} = 0 \quad \text{if } U_{ij} > 0$$
$$= 1 \quad \text{if } U_{ij} \leq 0$$

FOUR TYPES OF FLOW-FIELD POINTS

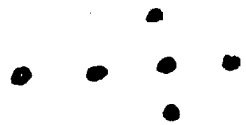
1. ELLIPTIC POINT $\mu_{ij} = \mu_{i-1,j} = 0$



2. PARABOLIC POINT $\mu_{ij} = 1$, $\mu_{i-1,j} = 0$

3. HYPERBOLIC POINT $\mu_{ij} = \mu_{i-1,j} = 1$

4. SHOCK POINT $\mu_{ij} = 0$, $\mu_{i-1,j} = 1$



VERTICAL LINE RELAXATION

$$A T_{i,j-1} + B T_{ij} + C T_{i,j-1} = R_{ij} + D T_{i-1,j} + E T_{i-2,j}$$

WHERE $T_{ij} = \phi^{+}_{ij} - \phi_{ij}$

ϕ^{+}_{ij} = "NEW" VALUE

ϕ_{ij} = "OLD" VALUE

SUMMARY OF MULTIGRID RESULTS

64 X 32 CELLS

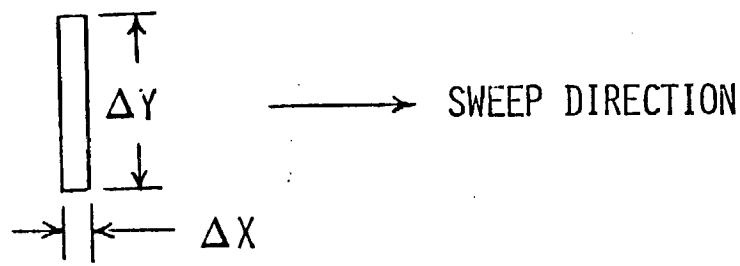
Problem Description		*Effective Spectral Radius	
		MG	SLOR
Unstretched Grid	Laplace's Eq., Smooth B.C.'s	.583 (.46 Combined Levels)	.924
	Parabolic Airfoil, $M_\infty = .70$	.549	.868
	" " " .85	.593	.855
	" " " .95	.858	.957
Stretched Grid	" " " .95	.936	.974

*Eff. Spec. Rad. = (Final Error/Initial Error) $^{1/(\text{Work Units})}$

Figure 11. - Summary of Multigrid Results.

STRETCHED GRID PROBLEM

- SLOW CONVERGENCE OF SLOR/MG WHEN $\Delta Y \gg \Delta X$



- FIX: ALTERNATE X-SWEEPS (COLUMNS) WITH PARTIAL Y-SWEEPS (ROWS)
  - Y-SWEEPS NEEDED ONLY FOR UPPER 10%-25% OF GRID
  - ZERO CORRECTION IF A POINT IS SUPERSONIC

STRETCHED GRID PROBLEM

LET $A = (\Delta y / \Delta x)^2$

THEN SLOR WITH OVERRELAXATION FACTOR ω GIVES:

$$g(\theta_x, \theta_y; A, \omega) = \frac{A\left[2(1-\omega) + \omega e^{i\theta_x}\right]}{A\left(2 - \omega e^{-i\theta_x}\right) + 2\omega(1 - \cos\theta_y)}$$

$$\left|g(0, \tfrac{\pi}{2}; A, \omega)\right| = \frac{A(2-\omega)}{A(2-\omega) + 2\omega} \longrightarrow 1 \text{ for } A \to \infty$$

$$\left|g(\tfrac{\pi}{2}, 0; A, \omega)\right| = \sqrt{\frac{4(1-\omega)^2 + \omega^2}{4 + \omega^2}} \longrightarrow 1 \text{ for } \omega \to 2$$

SUMMARY OF MULTIGRID RESULTS

(STRETCHED GRIDS, LIFTING FLOW, ALTERNATING DIRECTION SWEEPS)

64 X 64 CELLS

PROBLEM DESCRIPTION	EFF. SPECTRAL RADIUS*	
	MG	SLOR
FLAT PLATE, $\alpha = 1^0$, $M_\infty = 0$	.823 (W = 43)	.994 (W = 1392)
PARABOLIC AIRFOIL, $\alpha = 1^0$, $M_\infty = .80$	.832 (W = 68)	DIVERGED
PARABOLIC AIRFOIL, $\alpha = 1^0$, $M_\infty = .85$	DIVERGED	DIVERGED

$$^{*}\text{EFF. SPECT. RAD.} = \left(\frac{\text{FINAL ERROR}}{\text{INITIAL ERROR}}\right)^{1/(\text{WORK UNITS})}$$

0.1 10% THICK PARABOLIC ARC AIRFOIL, $\alpha = 1.0^\circ$, $M_\infty = 0.85$, 32 X 32 STRETCHED

TOTAL Q STORAGE= 2077

------X-STRETCH FUNCTIONS-----
XO=.500, AX= 8.660E-01

I	XX	SX(I)	SXH(I)
2	-2.333	4.866E-02	8.724E-02
17	0.000	1.155E+00	1.153E+00

------Y-STRETCH FUNCTIONS------
YO= .500, AY= 4.020E-01

J	YY	ZZ(J)	SY(J)
34	999.000	999.000	0.000
17	0.000	0.000	2.488

-----CONVERGENCE CRITERION FOR L1= 4 GRID IS EPS=1.008E-01

L	RMAX	RL2	DCIR	CIR	WRK	NS
4F	1.681E+02	3.492E+01	1.207E-02	.024	1.0	0
4U	1.178E-02	2.722E-03	0.	.024	1.1	0
4D	1.006E-02	2.403E-03	0.	.024	1.2	0
4F	3.581E+01	1.149E+01	1.240E-02	.030	2.2	0
4U	3.745E-02	7.426E-03	0.	.030	2.3	0
4D	1.981E-02	5.086E-03	0.	.030	2.4	0
4F	1.683E+01	8.092E+00	6.870E-03	.035	3.4	0
4U	7.395E-02	1.265E-02	0.	.035	3.5	0
4D	3.060E-02	7.626E-03	0.	.035	3.6	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

4F	6.380E+01	1.523E+01	7.008E-03	1.360	16.1	188
4U	2.410E+00	2.025E-01	0.	1.360	16.2	8
4D	4.008E-01	5.327E-02	0.	1.360	16.2	0
4F	2.598E+01	2.667E+00	1.160E-02	1.333	17.2	188
4U	2.203E-01	4.050E-02	0.	1.333	17.3	8
4D	1.756E-01	2.665E-02	0.	1.333	17.4	0
4F	4.422E+00	8.520E-01	1.197E-02	1.328	18.4	187
4U	1.460E-01	2.576E-02	0.	1.328	18.5	7
4D	9.825E-02	1.839E-02	0.	1.328	18.6	0
4F	2.721E+00	6.156E-01	5.563E-03	1.325	19.6	185
4U	1.095E-01	2.259E-02	0.	1.325	19.7	7
4D	4.747E-02	1.036E-02	0.	1.325	19.8	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

4F	1.438E+01	1.721E+00	1.418E-03	1.251	26.2	185
4U	1.477E-01	1.756E-02	0.	1.251	26.3	7
4D	3.538E-02	5.214E-03	0.	1.251	26.4	0
4F	1.330E+00	3.120E-01	6.112E-04	1.252	27.4	186
4U	4.992E-02	6.901E-03	0.	1.252	27.5	7
4D	1.851E-02	3.341E-03	0.	1.252	27.6	0
4F	4.574E-01	1.788E-01	5.173E-04	1.252	28.6	186
4U	3.489E-02	3.241E-03	0.	1.252	28.6	7
4D	1.823E-02	3.335E-03	0.	1.252	28.7	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

4F	2.471E+00	3.026E-01	1.607E-04	1.250	32.0	185
4U	1.640E-02	2.982E-03	0.	1.250	32.1	7
4D	7.020E-03	1.315E-03	0.	1.250	32.2	0
4F	2.216E-01	5.238E-02	1.534E-04	1.250	33.2	185
4U	9.278E-03	1.883E-03	0.	1.250	33.3	7
4D	2.841E-03	8.763E-04	0.	1.250	33.4	0

------EFFECTIVE SPECTRAL RADIUS= .81828------

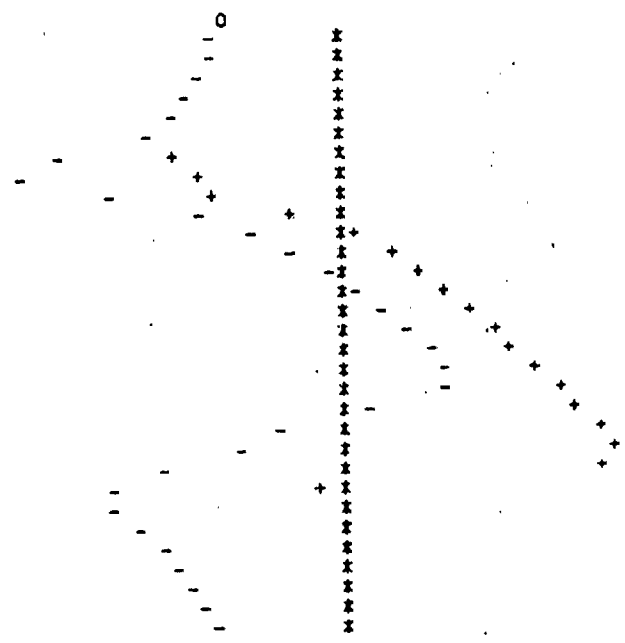
----------CONVERGENCE----------

?

PLOT OF CPBAR FOR LEVEL 4

CPSTAR=-3.201E-01

I	XX	CPL	CPU
2	-2.333	.0225	.0225
3	-1.565	.0410	.0410
4	-1.207	.0638	.0638
5	-.982	.0934	.0934
6	-.820	.1361	.1361
7	-.693	.2133	.2133
8	-.589	.4437	.1505
9	-.500	.5551	.0716
10	-.421	.3119	.0138
11	-.350	.0744	-.1965
12	-.285	-.0720	-.3458
13	-.224	-.1825	-.4637
14	-.165	-.2754	-.5405
15	-.109	-.3659	-.6102
16	-.054	-.4442	-.6741
17	0.000	-.5029	-.7342
18	.054	-.5548	-.7918
19	.109	-.5968	-.8480
20	.165	-.5991	-.9037
21	.224	-.3911	-.9598
22	.285	-.1465	-1.0173
23	.350	-.0246	-1.0771
24	.421	.1753	-1.0163
25	.500	.3245	-.2553
26	.589	.2997	.3224
27	.693	.2269	.2274
28	.820	.1696	.1698
29	.982	.1220	.1221
30	1.207	.0827	.0826
31	1.565	.0508	.0506
32	2.333	.0258	.0257



CL FROM PRESS. INTEGRATION= .4646

CL FROM PHI-JUMP AT TAIL= .5384

CHART OF MACH NOS.

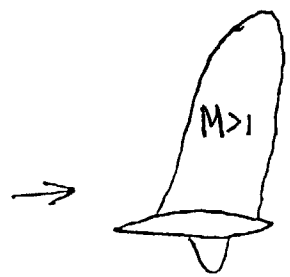
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	82	83	8														

10% Thick Parabolic Arc Airfoil, $\alpha = 1.0°$, $M_\infty = 0.85$, 32 X 32 Stretched Pressure Coefficient Distribution Along $y = 0$, upper & Lower Surfaces

39 67

11	85	84	84	84	83	83	83	82	82	81	81	80	80	79	79	79	81
12	85	84	84	84	84	84	84	84	84	84	84	84	84	84	85	86	88
13	85	84	84	85	85	85	86	86	86	86	87	87	88	89	90	92	93
14	85	84	84	85	86	86	87	88	88	89	90	91	92	93	94	96	98
15	85	84	85	85	86	87	88	89	90	91	92	94	95	96	98	100	101
16	85	84	85	86	87	88	90	91	92	94	95	97	98	100	101	103	105
17	85	84	85	86	87	89	91	92	94	96	97	99	101	102	104	105	107
18	85	84	85	86	86	89	91	93	95	97	99	101	103	104	106	108	109
19	85	84	85	86	87	89	91	93	95	97	99	102	104	106	107	109	111
20	85	84	85	86	87	88	90	91	93	95	97	99	102	104	107	109	111
21	85	84	85	85	86	87	88	89	90	91	92	93	95	97	99	101	103
22	85	84	84	85	86	86	87	87	87	88	86	88	88	86	89	90	92
23	85	84	84	85	85	85	85	85	85	85	85	84	84	84	84	84	85
24	85	84	84	84	84	84	84	83	83	82	82	81	80	79	73	77	75
25	85	84	84	84	83	83	82	82	81	80	73	78	77	75	73	70	66
26	85	84	83	83	83	82	81	81	80	79	78	77	76	74	72	70	68
27	85	83	83	83	82	81	81	80	79	79	78	77	76	75	74	73	72
28	85	83	83	82	82	81	80	80	79	79	78	78	77	77	76	76	75
29	85	83	83	82	81	81	81	80	80	79	79	79	79	78	78	78	78
30	85	83	82	82	82	81	81	81	81	81	80	80	80	80	80	80	80
31	85	83	83	82	82	82	82	82	82	82	82	82	82	82	82	82	82
32	82	82	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

Lower Region



2	83	83	83	83	83	84	84	84	84	84	84	84	84	84	85	87
3	82	82	82	83	83	83	83	83	83	83	83	83	84	84	85	85
4	81	81	81	82	82	82	82	82	82	83	83	83	84	84	85	85
5	80	80	80	80	81	81	81	82	82	82	83	83	84	85	85	86
6	77	78	78	79	80	80	81	82	82	83	83	84	85	85	86	86
7	73	75	77	78	79	80	81	82	83	83	84	85	85	86	87	87
8	76	78	79	80	81	82	83	83	84	85	85	86	86	87	87	88
9	81	81	82	83	84	84	85	25	86	86	87	87	87	88	88	88
10	84	35	86	86	87	37	87	87	88	88	88	88	89	89	89	88
11	94	92	92	91	91	90	90	90	90	90	90	90	90	90	90	89
12	101	99	97	96	95	94	94	93	93	92	92	92	91	91	90	89
13	106	104	102	101	99	98	97	96	95	94	94	93	93	92	91	90
14	109	107	106	104	103	102	100	99	98	97	96	95	94	93	92	90
15	111	110	108	107	105	104	103	102	100	99	98	97	95	94	93	91
16	114	112	111	109	108	106	105	104	103	101	100	99	97	95	94	91
17	116	114	113	111	110	109	107	106	105	103	102	100	99	97	94	91
18	118	117	115	114	112	111	109	108	106	105	103	102	100	98	95	92
19	120	119	117	116	114	113	111	110	108	107	105	103	102	99	96	92
20	122	121	119	118	116	115	113	112	110	109	107	105	103	101	97	92
21	124	123	121	120	118	117	115	114	112	110	109	107	104	102	98	92
22	126	125	123	122	120	119	117	116	114	112	110	108	106	103	99	92
23	128	127	125	124	122	121	119	118	116	114	112	110	108	104	100	92
24	126	126	126	125	124	123	121	120	118	116	114	112	109	105	98	92
25	97	102	106	110	112	114	115	116	116	115	113	111	107	101	95	91
26	67	72	77	81	85	88	90	93	94	95	96	95	95	93	92	90
27	72	71	71	71	71	72	73	74	76	78	80	82	85	87	89	89
28	75	75	75	75	75	76	76	77	78	79	80	82	84	85	87	88
29	78	78	78	78	78	78	79	79	80	80	81	82	83	84	86	87
30	80	80	80	80	80	80	81	81	81	81	82	82	83	84	85	86
31	82	82	82	82	82	82	82	82	82	83	83	83	83	84	84	85
32	83	83	83	83	83	83	83	83	84	84	84	84	84	84	84	85
	18	19	20	21	22	23	2									

Upper Region

CHART OF MACH NUMBERS IN FLOW. 10% THICK PAR. ARC

$M_\infty = .85$, $\alpha = 1°$, 32 X 32 Stretched Grid

TOTAL Q STORAGE= 7587

------X-STRETCH FUNCTIONS-----
XO=.500, AX= 8.660E-01

I	XX	SX(I)	SXH(I)
2	-3.382	1.762E-02	3.199E-02
33	0.000	1.155E+00	1.154E+00

------Y-STRETCH FUNCTIONS------
YC= .500, AY= 4.020E-01

J	YY	ZZ(J)	SY(J)
66	999.000	999.000	0.000
33	0.000	0.000	2.488

-----CONVERGENCE CRITERION FOR L1= 5 GRID IS EPS=2.480E-02

L	RMAX	RL2	DCIR	CIR	WRK	NS
5F	3.689E+02	5.130E+01	6.151E-03	.012	1.0	0
5U	2.655E-03	4.142E-04	0.	.012	1.1	0
5D	2.657E-03	4.142E-04	0.	.012	1.2	0
5F	7.948E+01	1.814E+01	6.152E-03	.015	2.2	0
5U	6.334E-03	9.946E-04	0.	.015	2.2	0
5D	6.316E-03	9.926E-04	0.	.015	2.3	0
5F	4.034E+01	1.362E+01	3.082E-03	.017	3.3	0
5U	9.709E-03	1.441E-03	0.	.017	3.4	0
5D	9.601E-03	1.425E-03	0.	.017	3.5	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

-------RL2 ON LEVEL 2= 2.670E+04-------
-------RL2 ON LEVEL 2= 5.481E-01-------

5F	1.540E+02	2.355E+01	3.297E-03	1.260	12.2	728
5U	5.091E-01	4.626E-02	0.	1.260	12.3	11
5D	2.131E-01	2.229E-02	0.	1.260	12.4	0
5F	6.344E+01	4.353E+00	4.144E-03	1.250	13.4	711
5U	1.624E-01	2.592E-02	0.	1.250	13.4	11
5D	9.955E-02	1.395E-02	0.	1.250	13.5	0
5F	7.208E+00	1.139E+00	4.177E-03	1.248	14.5	712
5U	1.281E-01	2.026E-02	0.	1.248	14.6	9
5D	6.942E-02	9.987E-03	0.	1.248	14.7	0
5F	3.696E+00	7.480E-01	1.946E-03	1.247	15.7	714
5U	9.570E-02	1.675E-02	0.	1.247	15.8	9
5D	5.099E-02	7.783E-03	0.	1.247	15.8	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

-------RL2 ON LEVEL 2= 3.566E-01-------
-------RL2 ON LEVEL 2= 3.498E-01-------
-------RL2 ON LEVEL 2= 1.404E-01-------

5F	2.528E+01	1.775E+00	5.112E-04	1.188	22.5	719
5U	4.804E-02	4.859E-03	0.	1.188	22.6	10
5D	2.008E-02	2.962E-03	0.	1.188	22.6	0
5F	4.248E+00	3.678E-01	2.623E-04	1.188	23.6	720
5U	2.143E-02	2.371E-03	0.	1.188	23.7	10
5D	7.358E-03	1.514E-03	0.	1.188	23.8	0
5F	7.007E-01	1.241E-01	1.204E-04	1.188	24.8	720
5U	1.851E-02	2.089E-03	0.	1.188	24.9	10
5D	6.225E-03	1.068E-03	0.	1.188	24.9	0
5F	4.207E-01	8.009E-02	7.007E-05	1.188	25.9	719
5U	1.504E-02	1.650E-03	0.	1.188	26.0	10
5D	5.867E-03	8.841E-04	0.	1.188	26.1	0

****
BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
****

10% Thick Parabolic Arc Air foil, $\alpha = 1°$, $M_\infty = 0.85$, 64x64 Stretched Grid
(continued next page)

```
        351E-05 1.191   29.9  717
 SU  1.212E-02   1.193E-03   0.        1.191   29.9   10
 SD  2.101E-03   2.824E-04   0.        1.191   30.0    0

 SF  3.020E+00   1.270E-01   1.130E-05 1.191   31.0  716
 SU  6.056E-03   6.681E-04   0.        1.191   31.1   10
 SD  1.154E-03   1.432E-04   0.        1.191   31.2    0

-------RL2 ON LEVEL 5= 1.270E-01-------

*********************************************************
    BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
*********************************************************

 SF  2.756E+00   2.218E-01   1.488E-05 1.187   35.2  721
 SU  9.098E-03   8.665E-04   0.        1.187   35.3   10
 SD  1.129E-03   2.164E-04   0.        1.187   35.4    0

 SF  4.426E-01   5.377E-02   9.176E-06 1.187   36.4  721
 SU  5.981E-03   4.496E-04   0.        1.187   36.5   10
 SD  6.868E-04   1.422E-04   0.        1.187   36.5    0.

 SF  2.676E-01   3.257E-02   7.000E-06 1.187   37.5  721
 SU  4.126E-03   3.137E-04   0.        1.187   37.6   10
 SD  5.503E-04   9.492E-05   0.        1.187   37.7    0

*********************************************************
    BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
*********************************************************

 SF  2.929E+00   2.046E-01   5.186E-06 1.190   43.8  717
 SU  7.288E-03   7.219E-04   0.        1.190   43.9   10
 SD  1.058E-03   2.437E-04   0.        1.190   43.9    0

 SF  2.902E+00   1.209E-01   5.885E-06 1.190   44.9  716
 SU  6.047E-03   4.047E-04   0.        1.190   45.0   10
 SD  6.828E-04   1.568E-04   0.        1.190   45.1    0

-------RL2 ON LEVEL 5= 1.209E-01-------

*********************************************************
    BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
*********************************************************

 SF  1.840E+00   8.950E-02   3.919E-06 1.190   47.2  717
 SU  9.289E-03   5.155E-04   0.        1.190   47.2   10
 SD  5.497E-04   1.155E-04   0.        1.190   47.3    0

 SF  2.715E+00   1.328E-01   1.925E-06 1.190   48.3  719
 SU  6.984E-03   4.736E-04   0.        1.190   48.4   10
 SD  4.438E-04   1.066E-04   0.        1.190   48.5    0

 SF  3.214E+00   1.510E-01   1.574E-06 1.190   49.5  717
 SU  4.235E-03   3.698E-04   0.        1.190   49.6   10
 SD  3.227E-04   8.151E-05   0.        1.190   49.6    0

*********************************************************
    BACK UP TO COARSER GRIDS TO OBTAIN CORRECTION
*********************************************************

 SF  2.231E-01   1.772E-02   1.127E-06 1.190   51.4  717
 SU  2.472E-03   1.271E-04   0.        1.190   51.5   10
 SD  1.487E-04   3.920E-05   0.        1.190   51.6    0

------EFFECTIVE SPECTRAL RADIUS= .85421------

-----------CONVERGENCE---------------

?
```

10% Thick Parabolic Arc Airfoil, $\alpha = 1°$, $M_\infty = 0.85$, 64 x 64 Stretched Grid
(Conclusion of Run)

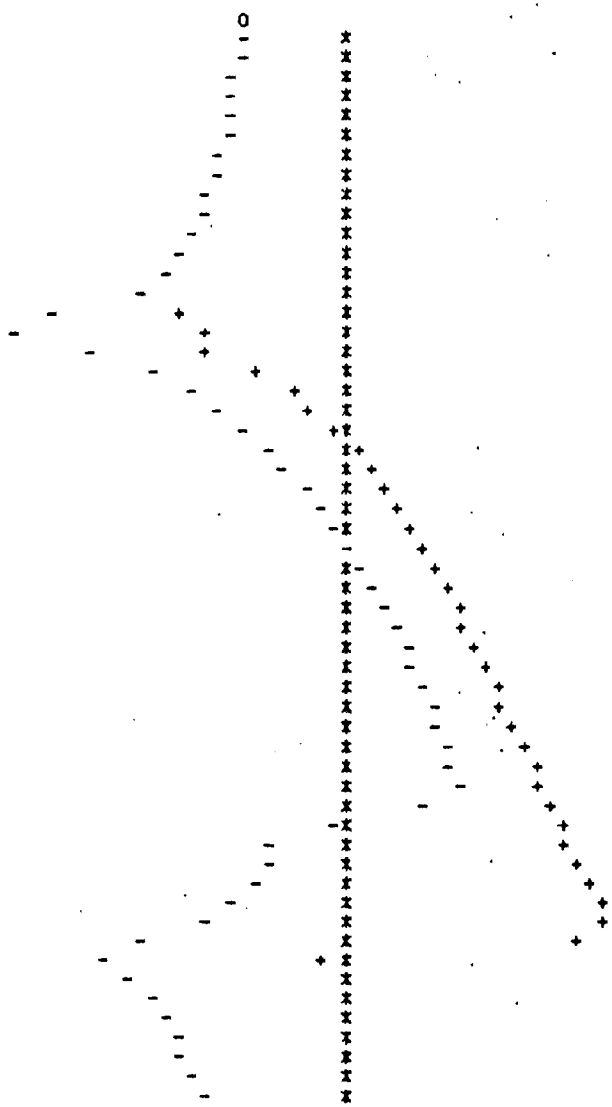
B-14

57 .982 .1174 .1174

PLOT OF CPBAR FOR LEVEL 5

CPSTAR=-3.201E-01

I	XX	CPL	CPU
2	-3.332	.0112	.0112
3	-2.333	.0193	.0193
4	-1.857	.0281	.0281
5	-1.565	.0376	.0376
6	-1.361	.0480	.0480
7	-1.207	.0597	.0597
8	-1.084	.0728	.0728
9	-.982	.0878	.0878
10	-.895	.1054	.1054
11	-.820	.1265	.1265
12	-.753	.1527	.1527
13	-.693	.1868	.1868
14	-.639	.2357	.2357
15	-.589	.3243	.3243
16	-.543	.5833	.1937
17	-.528	.7109	.1016
18	-.460	.4937	.1198
19	-.421	.2861	-.0439
20	-.385	.1605	-.1398
21	-.350	.0670	-.2136
22	-.317	-.0091	-.2763
23	-.285	-.0742	-.3393
24	-.254	-.1315	-.3964
25	-.224	-.1831	-.4408
26	-.194	-.2302	-.4817
27	-.165	-.2739	-.5199
28	-.137	-.3170	-.5560
29	-.109	-.3597	-.5903
30	-.082	-.3980	-.6232
31	-.054	-.4317	-.6550
32	-.027	-.4635	-.6857
33	0.000	-.4938	-.7157
34	.027	-.5228	-.7450
35	.054	-.5504	-.7738
36	.082	-.5768	-.8022
37	.109	-.6016	-.8303
38	.137	-.6245	-.8583
39	.165	-.6447	-.8862
40	.194	-.6571	-.9142
41	.224	-.6633	-.9423
42	.254	-.6692	-.9707
43	.285	-.6927	-.9995
44	.317	-.0756	-1.0288
45	.350	-.0338	-1.0587
46	.385	.0339	-1.0895
47	.421	.1336	-1.1212
48	.460	.3087	-1.0128
49	.500	.4181	-.2189
50	.543	.3672	.3672
51	.589	.2953	.2953
52	.639	.2520	.2520
53	.693	.2178	.2178
54	.753	.1836	.1836
55	.820	.1627	.1627
56	.895	.1391	.1391



10% Thick Parabolic Arc Airfoil, $\alpha = 1°$, $M_\infty = .85$, 64x64 Stretched Grid
Pressure Distribution along $y=0$

8-15

REMAINING PROBLEMS

- RELIABILITY FOR HIGH SUPERCRITICAL LIFTING FLOWS
- FULL POTENTIAL EQUATION
- 3-D TRANSONIC FLOW

TRANSONIC POSTSCRIPT

THE "RETARDED DENSITY", OR "ARTIFICIAL COMPRESSIBILITY" METHOD NOW UNDER DEVELOPMENT MAY BE VERY AMENABLE TO MULTIGRID RELAXATION.

$$(\tilde{\rho} u)_x + (\tilde{\rho} v)_y = 0$$

$$u = \Phi_x \qquad v = \Phi_y \qquad \rho = f(u^2, v^2)$$

$$\tilde{\rho} = \rho - \mu_{ij} \cdot \Delta s \cdot \rho_s$$

9. PARABOLIC TIME-DEPENDENT PROBLEMS.

Multi-Level Solutions to Time-Dependent Problems

(1) Multi-grid solution of <u>implicit equations</u> each time-step. $\Rightarrow$ One multi-grid cycle A per time-step, costing 3 WUs.
WU = <u>w</u>ork <u>u</u>nit = work equivalent to 1 explicit step.

(2) <u>Finer levels visited only once in many time-step</u>
(changes, relative to pure convection, are smooth)
$\Rightarrow$ Work per time-step $\ll$ 1 WU.

(3) The coarse-grid calculations (describing changes relative to fine-grid) can be made with <u>large time-steps</u>. $\Rightarrow$ No need for implicit equation

(4) <u>Adaptation techniques</u>: Visiting fine grids only <u>where</u> needed (where coarser fine-to-coarse corrections are significant).
Highly flexible Δt, Δx.

The Heat Equation

$$L_h U \equiv U_t - \Delta_h U = F(x), \qquad (x \in \Omega,\ 0<t)$$

$$U(x,t) = g(x), \qquad (x \in \partial\Omega,\ 0<t)$$

$$U(x,0) = U_0(x), \qquad (x \in \Omega).$$

$$\Rightarrow U(x,t) \to U_\infty(x) \qquad (x \in \Omega,\ t \to \infty)$$

$$\underline{V(x,t) = U(x,t) - U_\infty(x)} \to 0$$

$$LV \underset{\Omega}{=} 0, \quad V \underset{\partial\Omega}{=} 0 \qquad (0<t)$$

$$V(x,0) = U_0(x) - U_\infty(x) \qquad \text{(unknown)}$$

<u>In ∞-space:</u>

$$V(x,t) = \int A_\theta\, e^{i\theta\cdot x - \theta^2 t}\, d\theta$$

$$\theta\cdot x = \Sigma\theta_j x_j, \quad \theta^2 = \Sigma\theta_j^2$$

θ component reaches steady-state (90%) in $t = \frac{2.3}{\theta^2}$

High-frequency components on h-grid $\left(\theta = O\left(\frac{1}{h}\right)\right)$ converge to steady state in $t = O(h^2)$.

$$.23h^2 \le t_{90\%} \le .93h^2$$

Full Adaptation Outline (CS)

6 WU solution.

determined by $\|u_t\|$

$$V(t_0,t) = U(t) - u(t_0) \qquad (t > t_0 = O(h^2))$$

is smooth. Can be approximated on the 2h-grid (then the 4h-grid, etc.) with <u>high-order</u> approx

$$L_h V^h = \partial_t^h V^h - \Delta^h V^h = F(x) + \Delta^h U^h(t_0, x)$$

$$L_{2h} V^{2h} = I_h^{2h}\left(F + \Delta^h U^h(t_0, x)\right)$$

$$k_\ell = O(h_\ell^2)$$

k_ℓ - current time step
h_ℓ - current mesh-size

Near boundaries, non-fixed $F(x,t)$

$V(t_0,t)$ has algebraic singularities at (true or numerical) singularities of the boundary curve; contains residual of high-frequencies; and acquires high-frequencies from $F(x,t) - F(x,t_0)$

$\Rightarrow$ More involved fine $\leftrightarrow$ coarse interactions.

Dual (FAS) multi-grid for parabolic equations

$LU(x,t) = F(x)$ — differential eq
e.g., $L = \partial_t - \Delta$

$L_h U^h = F^h$ — fine discrete eq.

$L_{2h} U^{2h} = I_h^{2h} F^h + \tau_h^{2h}$ — correcte
coars
grid
equatio

$$\tau_h^{2h} = L_{2h} I_h^{2h} U^h - I_h^{2h} L_h U^h$$

(at convergence $I_h^{2h} U^h$ solves the corrected 2h equation)

τ_h^{2h} depends mainly on $O(h)$ wavelengths
$\Rightarrow$ It changes slowly after $t_0 = O(h^2)$.

Integrate with the coarser equation (and larger Δt), holding τ_h^{2h} fixed.
Infrequently, revisit grid h to update τ_h^{2h}.

This approach allows large saving even without employing higher-order approximations on coarser grids.
(Large Δt is not possible in low-order CS approach)

- τ_h^{2h} is held fixed for time t_h

$$u = e^{i\Theta\cdot x - \Theta^2 t}$$

$$\tau^h = L_h U \approx \left[\frac{k}{2}\Theta^4 - \frac{h^2}{2}\left(\Theta_1^4 + \Theta_2^4\right)\right] e^{i\Theta x - \Theta^2 t}$$

$$\tau^{2h} = 4\tau^h \qquad \text{(fixed order, fixed } \frac{k_\ell}{h_\ell^2}\text{)}$$

$$\tau_h^{2h} = \tau^{2h} - \tau^h = 3\tau^h \;; \quad \tau_{2h}^{4h} = 12\,\tau^h ; \ldots$$

t_h is determined by

$$\int_0^{t_h} \|\tau_h^{2h}(t) - \tau_h^{2h}(0)\|\, dt \approx \int_0^{t_h} \|\tau^h(t)\|\, dt$$

or

$$\int_0^{t_h} |3e^{-\Theta^2 t} - 3|\, dt \approx \int_0^{t_h} e^{-\Theta^2 t}\, dt$$

$$\Rightarrow \quad e^{-\Theta^2 t_h} = .56 \qquad \text{(independent of } \Theta\text{)}$$

When $\|U_t\|_2$ drops to .56 of its value, return to G^h: FAS cubic interpolation, a time step, and transfer τ_h^{2h}.

- $t_{2h} = \frac{1}{2} t_h \;, \quad t_{4h} = \frac{1}{4} t_h \;, \ldots$

$\Rightarrow$ total work – twice the work on G^h.

The heat equation: Preliminary tests

2nd order on all levels

Initial condition $\frac{\text{wavelength}}{2h}$	# grids employed	$u^h - U$		WU to 90% steady	
		h-grid	m.g.	h-grid	m.g.
4	1	.029	.029	9	9
8	2	.0068	.0044	19	34
16	2	.0017	.0006	21	124
32	3	.00041	.00020	56	509
64	3	.00010	.00013	63	2100
128	3	.000026	.000049	81	9000
256	4	.0000065	.0000050	180	36000

Self-Adaptation Methods.

Control h and p to nearly minimize error estimator E in a given amount of computational work W. (order of magnitude)

$(k = \Delta t = ch^2)$.

$$E = \int G(x,t)\,\tau(x,t)\,dx\,dt. \qquad W = n_c = \omega \int \frac{p}{h^d k}\,dx\,dt$$

$\tau = \tau(x,t;h,p)$ = truncation error $= L_h U - LU$

$$\tau(x,t;2h,p) - \tau(x,t;h,p) = \tau_h^{2h}$$

$G(x,t)$ = error weighting function (order of magnitude)

reflects purpose of computations

e.g., $G(x) = d_x^{\,m/2 - l}$

m = order of L

$u^{(l)}$ desired upto boundary

Marching on grid h (instead of $2h$)

$$-\Delta E = k \int G\,\tau_h^{2h}\,dx\,; \qquad \Delta W = \omega \int \frac{p}{h^d}\,dx - \omega \int \frac{p}{(2h)^d}\,dx$$

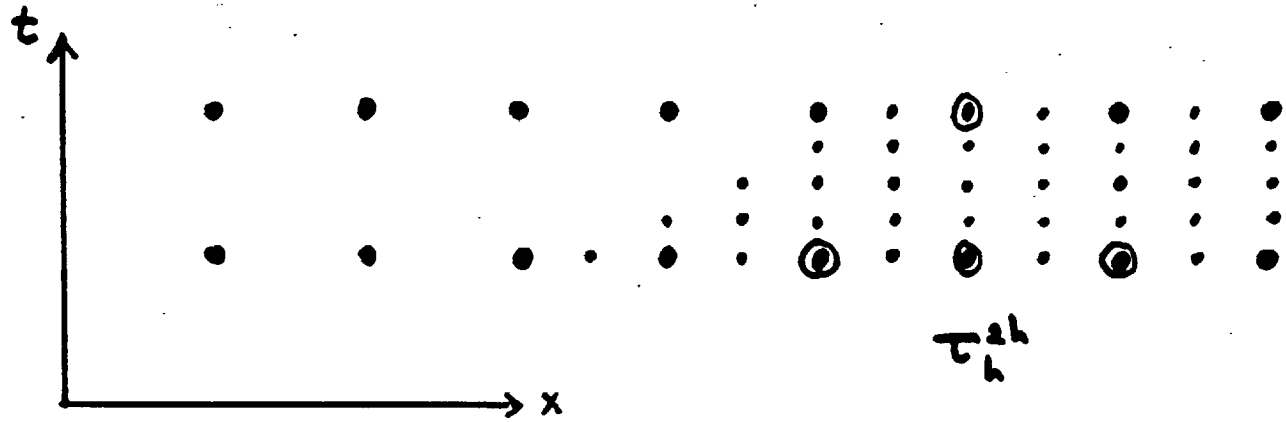
In optimum $-\Delta E/\Delta W = \lambda$ (constant)

λ (Lagrange multiplier) $= -\dfrac{dE_{min}}{dW}$ is our control:

$-\Delta E/\Delta W < .3\lambda \Rightarrow$ Next step on grid $2h$

$-\Delta E/\Delta W > 15\lambda \Rightarrow$ Next step on grid $h/2$

Local tests $\Rightarrow$ Local refinements:



Flexible non-uniform structure:

- $\Delta t, \Delta x$ easily change as functions of x, t
- All Difference equations are on equidistant points.
- High-order approximations easily implemented (Decided by similar tests with τ_f^c)
- Large computational domain treated by increasingly coarser grids.
- Singularities do not contaminate the convergence rates.
- Efficient Solution of optimal-control problems: Control on coarse grids, with fixed τ_f^c (& updating).

10. USING COMPOSITE MESHES FOR TIME-DEPENDENT PROBLEMS

Composite Meshes for Time Dependent Problems

Joseph Oliger
Stanford University

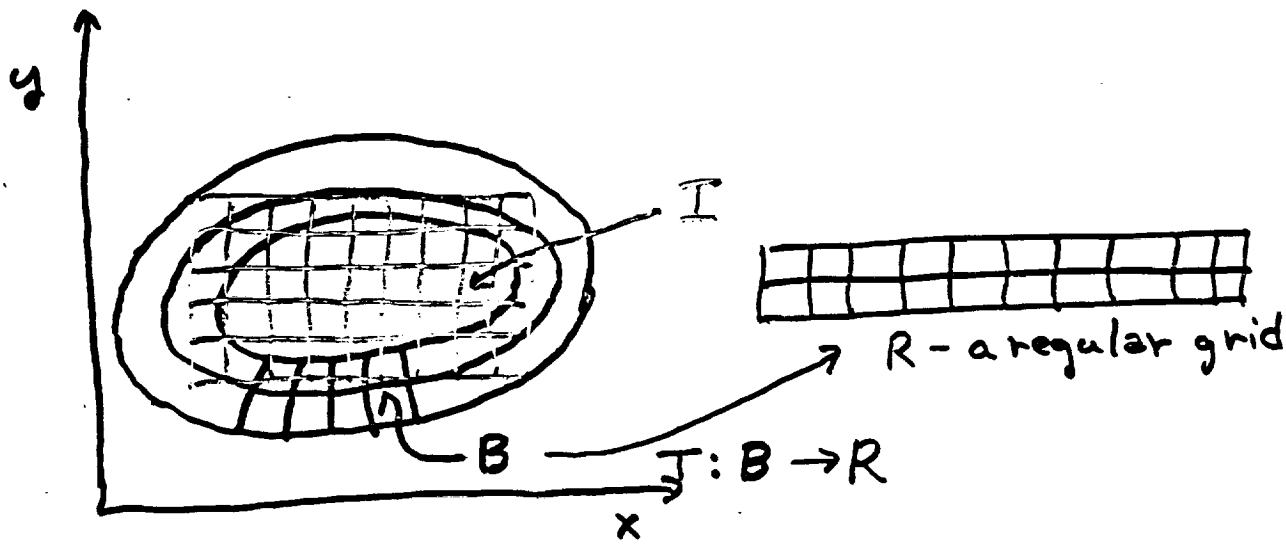
We consider some problems associated with using standard difference methods for hyperbolic and parabolic P. D. E.'s on composite meshes which are locally uniform.

We will emphasize hyperbolic and perturbed hyperbolic systems and their <u>transient solutions</u>.

The underlying basic methods might be the leap-frog or Lax-Wendroff methods or other similar methods. For simplicity we consider only explicit methods here.

Linked Boundary and Interior Meshes

(Starius, Brandt)



1) Objectives / Possibilities

a) to deal with irregular regions well

b) to approx. boundary layers well - stretching transformations can be used.

c) to approx. derivative boundary conditions well, e.g., $u_n = 0$.

2) The process

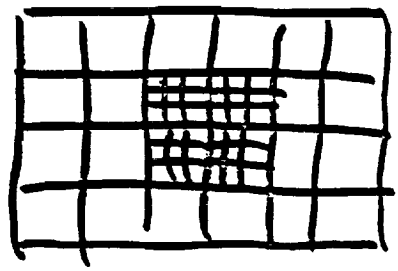
a) Use standard difference methods in interior of R and I.

b) Approx. boundary cond's for problem as transformed by T on one edge of R.

c) use periodicity conditions on both ends of R.

d) Interpolate boundary of I from R and remaining edge of R from I.

Interior Refinements



1) Objectives / Specifications

   a) locally uniform

   b) reduce data required to represent solution / work

   c) inexpensive - not too much overhead

Observation: the grid structure must generally be a function of time.

2) Necessary conditions

   a) cannot let phenomena for which refinements are needed escape from refined region. (Browning, Kreiss, Oliger)

      i) loss of accuracy

         <u>consider wave motion from subgrid to subgrid</u>

         <u>Coarse to fine</u>

         speeds up or continues at $\approx$ correct speed

Fine to Coarse

continues at correct speed

slows down

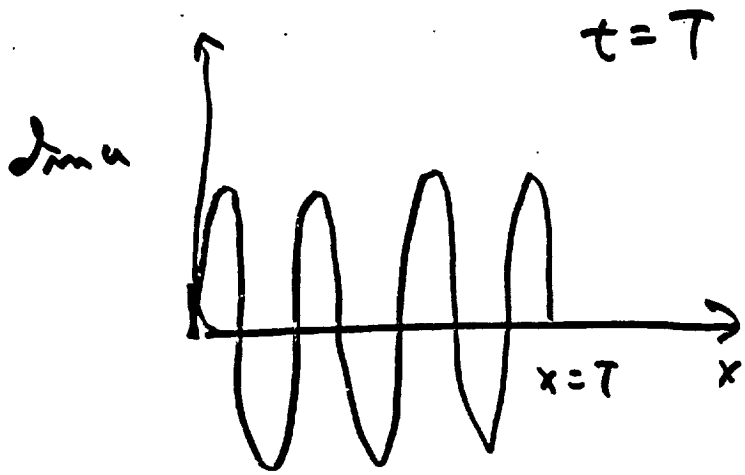
doesn't go through

Consider a leap-frog approx. of

$$u_t + u_x = 0 \qquad x \geq 0,\ t \geq 0$$

$$u(0,t) = e^{i\alpha t} \qquad t \geq 0$$

$$u(x,0) = 0 \qquad x \geq 0$$



Let $N = \frac{2\pi}{\alpha h}$, number of gridpoints per wavelength. Grid spacing $h > 0$.

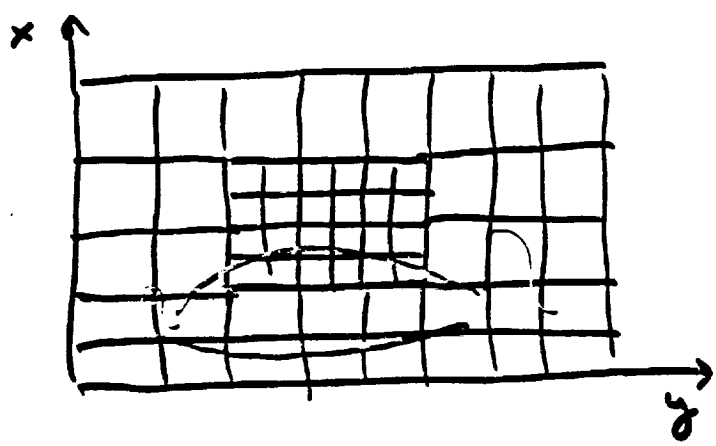
Time step small.

Wave <u>should</u> travel with speed 1.

δt travels with speed c

N	32	16	8	7	6
c	0.98	0.92	0.64	0.48	0

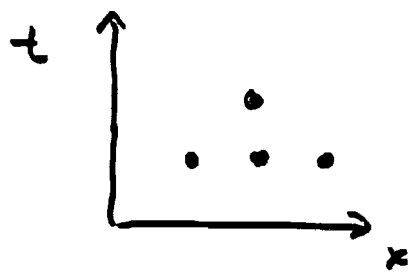
ii) <u>Weak instabilities</u> can result – polynomial growth. Regions with different mesh spacings behave like media with <u>piecewise constant transmission properties</u>.

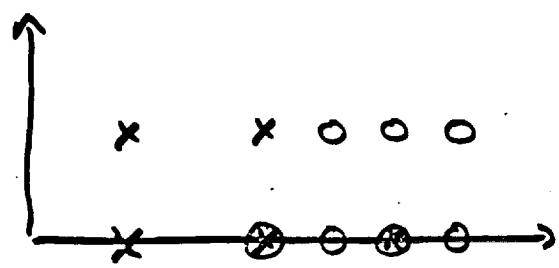


Refraction & Reflection can result.

3) Implementation as mesh refinement

Example: Lax-Wendroff approx. of $u_t = Au_x$ with stencil





values on x points = values on o points when they coincide

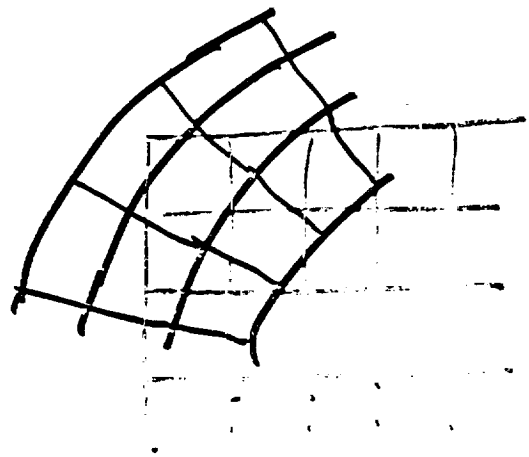
Remarks:

a) Not always stable but stable methods easily constructed and analyzed (Ciment, Oliger)

b) Can generally only be used for refinements - geometric contraints

c) Can use interpolation in time with unequal time steps

4) Implementation via Interpolation



Interior and "real boundary" points computed as usual.

Other boundary points interpolated from other grid.

Convergence Result (Engquist)

i) Max norm stable strictly parabolic difference approx. $O(h^q)$ in each part

ii) Interpolation $O(h^q)$ from points $O(h)$ away using formulae with compact support -

$$u_i = \sum_\nu a_\nu u_{i+\nu} \quad , \sum |a_\nu| \le K$$

iii) Regions overlap $O(\varepsilon)$, ε indep. of h.

$\Longrightarrow$ Convergence $O(h^q)$

Hyperbolic Case $u_t = Au_x + Bu_y$ modify as:

iv) delete i)

v) Methods stable for 2 boundary problem of regular grids with perturbed A, B

vi) Approx. is dissipative

Remarks:

i) One cannot allow growth in The overlapping zone.

ii) Cannot show anything if internal boundary introduced has a characteristic segment.

5) Advantages / Disadvantages

a) Easier to implement stable method with fewer restrictions via mesh refinement.

b) Interpolation allows one to overlap without geometrical restrictions.

Adaptive Methods

Consider a well-posed problem with smooth solution

$$Lu = f \qquad 0 \le x \le 1 \ , \quad 0 \le t \le T$$

$$u(x,0) = u_0(x)$$

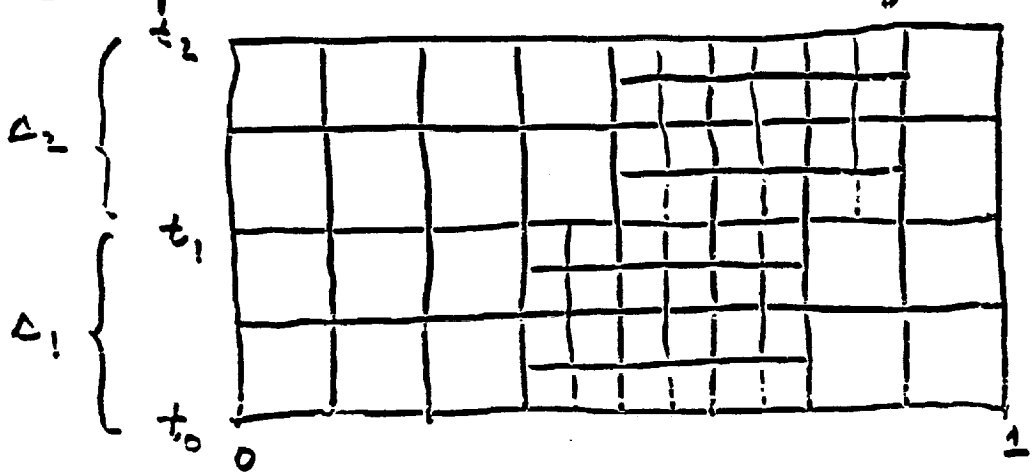
$$B_0 u(0,t) = 0 \ , \quad B_1 u(1,t) = 0$$

where

$$L = \partial_t + P(x,t,\partial_x)$$

is a linear operator.

Let $0 = t_0 < t_1 < \cdots < t_N = T$ and define a piecewise uniform grid on each strip



Let $\Delta = \cup \Delta_\ell$, assume

$$h^\ell = \max_j h_j^\ell \Big/ \min_j h_j^\ell \le K$$

On each grid we approximate by

$$L_{\Delta_\ell} v_\ell = f$$

Let L_Δ be the operator so defined over Δ.

We assume L_Δ stable, i.e.,

$$\| v(x_j^\ell, t_\ell) \|_{\ell_2} \le C_T \| v_1(x_j^1, 0) \|_{\ell_2}$$

and accurate of order q, i.e.,

$$L_\Delta u(x_j^\ell, t_j^\ell) = \tau(x_j^\ell, t_j^\ell) = O(h^q)$$

where

$$\tau(x_j^\ell, t_j^\ell) = (h_j^\ell)^q T_1(x_j^\ell, t_j^\ell) + (k_j^\ell)^r T_2(x_j^\ell, t_j^\ell) + O(h^{q+1})$$

and $T_1(x,t)$, $T_2(x,t)$ are independent of Δ.

We assume the h_j^ℓ are chosen so that the first term dominates the second.

deBoor has suggested that Δ be chosen such that

$$h_j^{\ell} \, |\tau(x_j^{\ell}, t_j^{\ell})|^2 = \text{const} = E$$

such a mesh is called <u>equidistributing</u> and has been shown optimal by Grebennikov. This approach has been followed by Pereyra and Sewell for 2-pt. boundary value problems.

Let $\quad |T_1(x,t)| \le M_1 \quad , \quad \varepsilon \ge M_1 / K^{1/\sigma}$

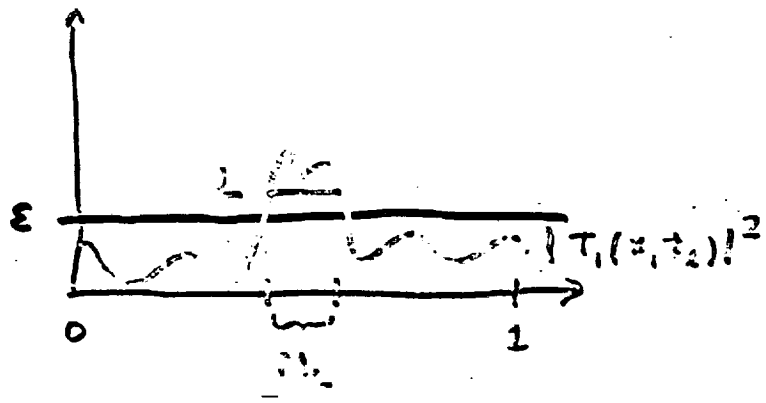
$$\sigma = 2/(2q+1)$$

Let $\quad g(x,t) = \max\left(|T_1(x,t)| , \varepsilon \right)$

A mesh which equidistributes g approximately equidistributes τ.

Let $M_L(t_\ell) = \{x : |g(x,t_\ell)| \geq L\}$ where L is chosen so that

$$\int_{M_L(t_\ell)} |g(x,t_\ell)|^2 dx = \frac{1}{2} \int_0^1 |g(x,t_\ell)|^2 dx$$



Let $\mu_{max} = \max_\ell \mu(M_L(t_\ell))$

<u>Theorem</u>. If the mesh equidistributes $\int_0^1 g^2(x)\,dx$,

Then

$$\|u(x_j^\ell, t_\ell) - v(x_j^\ell, t_\ell)\|_{\ell_2} \leq 2^{1/\sigma} C_T h^q \mu_{max}^{(2q^2+q)} \max_t \|g(x,t)\| + O(h^{q+1})$$

Differences in strategy - Transient vs. Steady-state problems

1) Refinements must be conservatively over done for transient problems. Otherwise computation must iterate over space-time domain.

2) We know what the solution looks like initially for initial value problems. Can base initial grid on initial data.

3) Hyperbolic problems - can estimate max. speed of propagation and construct refined zones large enough so that nothing can escape before regridding.

Necessary Assumption:

New phenomena requiring refinement cannot occur on time scales faster than the meshes are reconstructed and must grow smoothly

References

Linked boundary and internal meshes

1) Starius, G., "Constructing orthogonal meshes by solving initial value problems," Numer. Math. 28, 25-48, 1977.

2) Starius, G., "Composite mesh difference methods for elliptic boundary value problems," Numer. Math. 28, 242-258, 1977.

3) Starius, G., "Numerical treatment of boundary layers for perturbed hyperbolic equations," Dept. of Comp. Sci. Rpt. 69, Uppsala Univ., Uppsala, Sweden, 1978.

Stability

4) Ciment, M., "Stable matching of difference schemes," SIAM J. Numer. Anal. 9, 695-701, 1972.

5) Oliger, J., "Hybrid difference methods for the initial boundary value problem for hyperbolic equations," Math. Comp. 30, 729-738, 1976.

Grid Phenomena

6) Browning, G.; Kreiss, H.-O.; Oliger, J.; "Mesh refinement," Math. Comp. 27, 29-39, 1973.

Estimates for Adaptive Methods

7) Oliger, J., "Approximate methods for atmospheric and oceanographic problems," to appear in Proceedings of the Third International Symposium on Computing Methods in Applied Sciences and Engineering, Springer-Verlag, [illegible]

11. SINGULAR PERTURBATION MULTI-GRID TECHNIQUES

Theoretical Adaptation - 1 dim. Case

$$\varepsilon u'' + u' = 0 \quad \text{in } 0<x<1 \; ; \qquad u = e^{-\frac{x}{\varepsilon}} + u_0$$

$$\tau = \left(\frac{h}{\gamma\varepsilon}\right)^p \frac{1}{\varepsilon} e^{-x/\varepsilon} + \left(\frac{h}{\gamma}\right)^p u_0^{(p+1)}$$

For $x < (p+1)\,\varepsilon \log \frac{\eta_0}{\varepsilon}$ second term neglected.

Local adaption equations $(w=1)$ $\Rightarrow$

$$G\tau - \frac{\lambda}{h(x)} = 0 \; ; \qquad G\tau \log\left(\frac{h}{\gamma\varepsilon}\right) + \frac{\lambda}{h(x)} = 0$$

$$\Rightarrow \; h = \frac{\gamma}{e}\varepsilon, \qquad p = \log\frac{\gamma G}{\lambda} - 1 - \frac{x}{\varepsilon} \qquad (\text{if } \geq 2).$$

(a) <u>ε large enough, $G \equiv 1$:</u> $p \geq 2$

$$W = \int_0^1 \frac{p}{h}\,dx = \frac{e}{\gamma\varepsilon}\left[\log\frac{\gamma}{\lambda} - 1 - \frac{1}{2\varepsilon}\right]. \qquad p = \frac{\gamma\varepsilon}{e}W + \frac{1}{2\varepsilon} - \frac{x}{\varepsilon}$$

$$E = \int_0^1 G\,dx = \frac{\lambda e}{\gamma\varepsilon} = \frac{1}{\varepsilon} e^{-\frac{\gamma\varepsilon}{e}W - \frac{1}{2\varepsilon}}$$

holds as long as $p \geq 2$, i.e. $W \geq \dfrac{e}{2\gamma\varepsilon^2}$

(b) <u>small ε, $G \equiv 1$:</u>

$p = 2$ and $h = \lambda^{\frac{1}{3}}\gamma^{\frac{2}{3}}\varepsilon\, e^{\frac{x}{3\varepsilon}}$ for $x \geq x_0 = \varepsilon\left(\log\frac{\gamma}{\lambda} - \right.$

In practice - h grows to outer value.

$$W = \int_0^1 \frac{p}{h} \approx \frac{e}{4}\left(\log\frac{\gamma}{\lambda}\right)^2$$

$$E = \int_0^1 \tau\, dx = \frac{\lambda e}{2}\log\frac{\gamma}{\lambda} \approx \left(\frac{e}{4}W\right)^{1/2} e^{-(4W/e)^{1/2}}$$

lower power of W, but exponential rate independent of ε

(c) <u>small ε, $G(x) = x$:</u>

$$\text{"}p\text{"} = \log\frac{\gamma x}{\lambda} - 1 - \frac{x}{\varepsilon} < \log\frac{\varepsilon\gamma}{\lambda} - 2 < 0 \quad \Rightarrow p = 2$$

$$\left(\frac{h}{\gamma\varepsilon}\right)^3 = \frac{\lambda}{\gamma x} e^{x/\varepsilon} \geq \frac{\lambda e}{\gamma\varepsilon} \gg 1$$

i.e., no boundary-layer resolution.

$h \gg \varepsilon \log\frac{1}{\varepsilon}$ $\Rightarrow$ Even near 0, h & p are determined by outer solution u_0

Truncation errors in central differencing

∂_p^q = p-order central approximation to $\frac{d^q}{dx^q}$

$$\tau = \partial_p^q u - \frac{d^q}{dx^q} u$$

$$= (-1)^{\frac{p}{2}+1} \left(\frac{h}{\gamma}\right)^p u^{(q+p)}(\xi)$$

where γ dependence on p and q is given by

	$p=2$	$p=4$	$p=6$	$p=8$	$p=10$	$p\to\infty$
$q=1$	2.45	2.34	2.28	2.24	2.20	2
$q=2$	3.46	3.08	2.87	2.74	2.64	2
$q=3$	2.00	2.03	2.05	2.05		2
$q=4$	2.45	2.42	2.38			2

In theoretical adaption studies γ = const. can be taken.

Multi-Level Adaptive Solutions to Singular-Perturbation Problems

- A flexible structure + fast general solver.
- <u>High order</u> of convergence $U^h \to U$ $(E \sim e^{-W^\alpha})$ <u>uniformly</u> for all $0 < \varepsilon < \infty$.
- <u>No need for special analyses</u> (no need for defining singular perturbation, ε, reduced equation, reduced boundary conditions).
- <u>No need for matching procedures</u>.
- <u>Interior layers</u> (turning points, shocks, etc.) are automatically located and properly resolved.
- <u>Boundary-layer</u> resolution depends on purpose of the user (expressed by G).
- <u>Needed:</u> (i) Uniform discretization
  (ii) Uniformly smoothing relaxation.

Relaxation of Singular-Perturbation Equation

Required: smoothing rates uniform in ε

$\Rightarrow$ either

Gauss-Seidel relaxation with <u>alternating marching directions</u> (to conform with all convection directions of lower order terms, such as flow directions in Navier-Stokes equations with large Reynolds number R)
yields smoothing rates better than for the singular-perturbation terms (e.g., $R=0$).

or

<u>Distributed</u> Gauss-Seidel relaxation

and

<u>Block</u> relaxation for degenerate difference equations (almost reducible eqs.).

Uniformly stable Discretization

- Uniform multi-grid efficiency (# work units bounded, independent of [h] domain, data)

⇔ FDE (finite-difference eqs.) are stable, uniformly in [h], r (domain radius).

➡ FDE uniformly stable in ε.

This is also necessary for practical approximation of singular-perturbation problems.

Example of approximation not uniformly stable:

$-\varepsilon \Delta u + u_x + u_y$ discretized as

$$-\varepsilon \Delta_y u(x,y) + \frac{u(x+h,y) - u(x-h,y)}{2h} + \frac{u(x,y+h) - u(x,y-h)}{2h}.$$

This is elliptic FDE by Thomée (1964) definition.

It is stable for $h \to 0$ (fixed ε)

but not for $2\varepsilon < h \to 0$, or $\varepsilon \to 0$ (fixed h).

$$-\varepsilon \Delta_h u(x,y) + \frac{u(x,y) - u(x-h,y)}{h} + \frac{u(x,y) - u(x,y-h)}{h}$$ is uniformly stable

<u>Symbol</u> $B(\theta) = B(\theta, h, \varepsilon)$: $L_h e^{i\theta\cdot x/h} = \frac{B(\theta)}{h^m} e^{i\theta\cdot x/h}$

<u>Ellipticity</u>: $\det B(\theta) \neq 0, \quad (0 < |\theta| \leq \pi)$

<u>uniform h-ellipticity</u>: For $0 < \theta_0 \leq |\theta| \leq \pi$, the range of $\det B(\theta)$ does not close a loop around 0, [uniformly in h, ε]. Or

$$\operatorname{Re} \det B(\theta, h, \varepsilon) \geq [\delta(|\theta|)] > 0$$

- h-ellipticity is defined for finite h; entails [uniform] stability in r, ε.
- h-ellipticity (unlike ellipticity) does not require continuous coefficients.
- L_h elliptic $\Rightarrow$ for suitable translation T_h and sufficiently small h, $T_h L_h$ is h-elliptic.
- L_h h-elliptic $\Rightarrow$ Suitable under-relaxation has bounded smoothing rates, uniformly in r, ε.

<u>Construction of h-elliptic operators</u>

Construct $\operatorname{Re} B(\theta, h, \varepsilon) \geq \delta(|\theta|) > 0$ separately for each term of the PDE, or each term of $\det L$

Examples

- $L_h u \equiv -\varepsilon \Delta_h u + \frac{u(x+h,y)-u(x-h,y)}{2h} + \frac{u(x,y+h)-u(x,y}{2h}$

  $B(\theta) = 4\varepsilon\left(\sin^2\frac{\theta_1}{2} + \sin^2\frac{\theta_2}{2}\right) + ih\left(\sin\theta_1 + \sin\theta_2\right)$

  L_h is $(h-)$ elliptic.

  L_h is <u>not</u> uniformly $(h-)$ elliptic. For $\varepsilon \to 0$ solution exhibit indefinitely large oscilations.

- $\tilde{L}_h u(x,y) = L_h u(x+h,y)$

  $\tilde{B}(\theta) = e^{i\theta_1} B(\theta)$

  $\tilde{L}_h$ is elliptic, but not h-elliptic.

  A problem with L_h and $\tilde{L}_h$ can be singular

- $L_h u \equiv -\varepsilon \Delta_h u + \frac{u(x,y)-u(x-h,y)}{h} + \frac{u(x,y)-u(x,y-h)}{h}$

  $B(\theta) = 4\varepsilon\left(\sin^2\frac{\theta_1}{2} + \sin^2\frac{\theta_2}{2}\right) + h\left(2 - e^{-i\theta_1} - e^{-i\theta_2}\right)$

  L_h is uniformly $(h-)$ elliptic.

PDEs which are not uniformly stable cannot have uniformly stable approximations. Example:

<u>Helmholtz equation</u> $\varepsilon^2 \Delta u + u = 0$ with radiation BC.

<u>Direct multi-grid is inefficient</u>: Coarsest mesh-size h should satisfy $\left(\frac{h}{\varepsilon}\right)^p < \frac{r}{\varepsilon}$.

Such problems require reformulation (other variables, other norms)

$u = \sum A_\theta(x) e^{i\theta x}$ (local Fourier expansion)

Error norms: errors in ensemble averages of $|A_\theta|^2$.

- $\underline{u_t = \varepsilon \Delta u + a \cdot \nabla u}$

$h = \alpha\varepsilon$ in boundary layer and would require $\Delta t = \Delta t_h = \dfrac{h^2}{5\varepsilon + ah} \approx \alpha^2\varepsilon/5$

<u>MLAT</u>: Few Δt_h steps on Ω_h, then Δt_{2h} on Ω_{2h} (FAS), etc., until external level is reached

Every few $(\Omega_H, \Delta t_H)$ steps, go to grid $H/2$

$$u^{H/2}_{NEW} = u^{H/2}_{OLD} + I^{H/2}_{H}\left(u^H - I^H_{H/2} u^{H/2}\right)$$

and make one step $\Delta t_{H/2}$ (liquidating h-f errors of interpolation)

Every few such steps go to $H/4$. Etc.

12. MULTI-GRID PROGRAMMING.

MULTI-GRID PROCESSES:

Putting function values into a grid * †
Relaxation sweep over a grid *
Coarse-to-fine interpolation * †
Fine-to-coarse residuals transfer * (†)
Driving algorithm †
Grids creation (†)

* Once for all grids

† Independent of the problem

<u>Needed</u>: standard grid structure.
General routines

- All grid-functions are stored in one big vector Q.

- To use grid-function u_{ij}^k

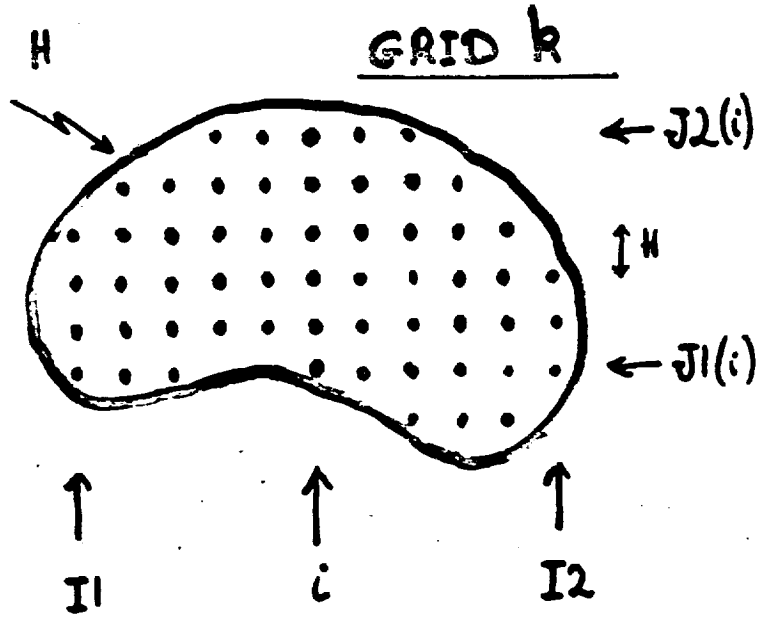
CALL KEY(k, I1, I2, J1, J2, JR, H)

will output I1, I2, J1(i), J2(i), H

JR(i) will be set so that

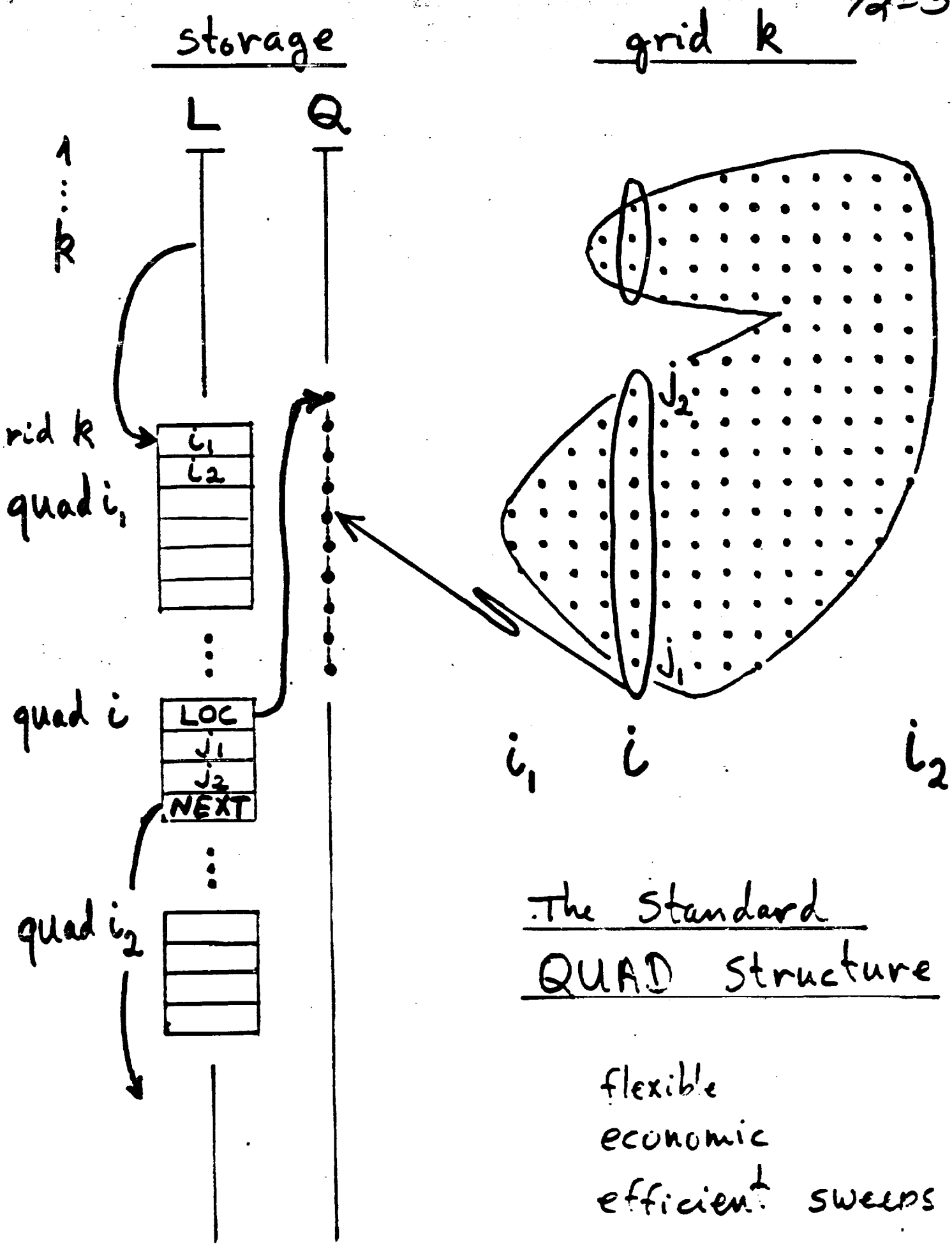
$$u_{ij}^k = Q(JR(i) + j)$$

$(I1 \le i \le I2,\ J1(i) \le j \le J2(i))$




```
SUBROUTINE PUTF(k,F)
COMMON Q(·), J1(·), J2(·), JR(·)
CALL KEY(k, I1, I2, J1, J2, JR, H)
DO 1 I=I1,I2
DO 1 J=J1(I), J2(I)
1 Q(JR(I)+J) = F(I*H, J*H)
END
```

- Once for all grids k, functions F.
- Once for all programs, problems.
- Grid number (k) is programmable
- Fully efficient sweeping
- Economic storage
- <u>Adaptability</u>: any column can be anywhere in Q.



The Standard QUAD Structure

flexible
economic
efficient sweeps

GRIDPACK Macro Statements

within FORTRAN.

Grid construction:

$k = \text{GRID}(x_0, y_0, h_x, h_y, \ldots, \text{CHAR})$

$\text{Domain} = \{(x,y) : \text{CHAR}(x,y) = 1\}$

DELETE k

k = CØARSE m

k = UNIØN m, n

k = DIFFERENCE m, n

k = INTERSECT m, n

k = TRANSPØSE m

k = INTERIØR (t) m

t = TEMPLET (0,0), (0,1), (0,-1), (1,0), (-1,0)

Coordinate Transformation

CURVE k, φ, ψ

Allocation of Q space

QSPACE k

QØFF k

PØINT k TØ m

Transfer of numerical data

TRANSFER k TØ m

PUT f TØ k $f(x,y)$

INTERPØLATE(i) k TØ m order i

- <u>KEY</u> $k, I_1, I_2, J_1, J_2, J_R$

$$u^k_{ij} = Q\big(J_R(i) + j\big), \quad \begin{array}{l} I_1 \le i \le I_2 \\ J_1(i) \le j \le J_2(i) \end{array}$$

- <u>MULTI GRID</u> $\underbrace{x_0, y_0, h^0_x, h^0_y, \ldots, \text{CHAR}}_{\text{coarse grid}}$, M, # of levels

$\underbrace{\text{Tol, Work}, \eta, \delta}_{\text{switching parameters}}$, RELAX, RESCAL

relaxation routine residual transfer routine

Examples of sweeping macro statements

DØ 5 (I,J) = GRID(k), ΔI, ΔJ, $k_1(I_1, J_1)$, ...

DØ 5 I = CØLS(k), ΔI, $k_1(I_1)$

DØ 5 J = CØL(I, k), ΔJ, $k_1(J_1)$

Examples of statements within sweeps

$k_1(I_1+1, J_1) = k(I+1, J+1) + k(I+1, J-1)$

IF((I+1, J).IN. k) GØTØ 5

IF((I,J).IN(t).k) GØTØ 5

Initialization

Compactification

Display: grids and grid-functions

APPENDIX B: SAMPLE MULTI-GRID PROGRAM AND OUTPUT.

This simple program of Cycle C (written in 1974 by the author at the Weizmann Institute) illustrates multi-grid programming techniques and exhibits the typical behavior of the solution process. For a full description of Cycle C, see Sec. 4 or the flowchart in Fig. 1.

The program solves a Dirichlet problem for Poisson equation on a rectangle. The same 5-point operator is used on all grids. The I_k^{k-1} residuals transfer is the trivial one (injection), the I_{k-1}^k interpolation is linear. The higher interpolation (A.7) and the special stopping criterion (A.16), recommended for the first $\lceil q/p \rceil$ cycles, are not implemented here.

For each grid G^k we store both v^k and f^k $(k=1,2,\ldots,M)$. For handling these arrays f^k is also called v^{k+M}. The coarsest grid has NX0 × NY0 intervals of length H0 each. Subsequent grids are defined as straight refinements, with mesh sizes H(k) = H0/2**(k-1). The function F(x,y) is the right-hand side of the Poisson equation. The function G(x,y) serves both as the Dirichlet boundary condition (Φ^M) and as the first approximation (u_o^M). The program cycles until the L_2 norm of the residuals on G^M is reduced below TOL, unless the work WU exceeds WMAX. After each relaxation sweep on any grid G^k, a line is printed out showing the level k, the L_2 norm of the ("dynamic") residuals computed in course of this relaxation, and WU, which is the accumulated relaxation work (where a sweep on the finest grid is taken as the work unit).

Note the key role of the GRDFN and KEY subroutines. The first is used to define a grid (v^k), i.e., to allocate for it space in the general vector Q (where IQ points to the next available location), and to store its parameters. To use grid v^k, CALL KEY(k,IST,M,N,H) retrieves the grid parameters (dimension M×N and mesh-size H) and sets the array IST(i) so that v_{ij}^k = Q(IST(i)+j). This makes it easy to write one routine for all grids v^k; see for example, Subroutine PUTZ(k). Or to write the same routines (RELAX, INTADD, RESCAL) for all levels.

To solve on the same domain problems other than Poisson, the only subroutines to be changed are the relaxation routine RELAX and the residual injection routine RESCAL, the latter being just a slight variation of the first.

For different domains, more general GRDFN and KEY subroutines should be written. A general GRDFN subroutine, in which the domain characteristic function is one of the parameters, has been developed, together with the corresponding KEY routine. This essentially <u>reduces the programming of any multi-grid solution to the programming of a usual relaxation routine</u>.

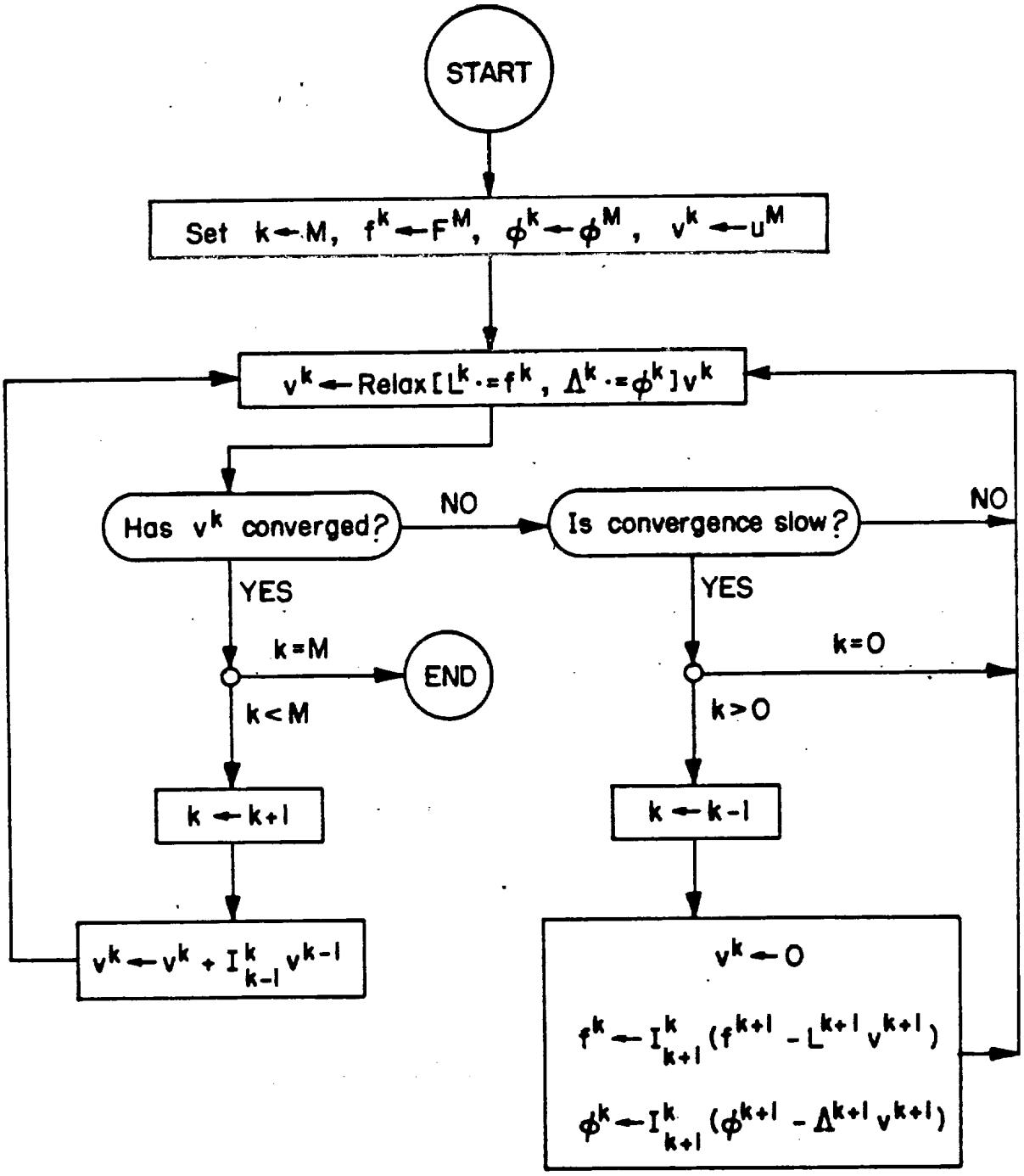


Figure 1. Cycle C, Linear Problems

CYCLE C 12-10

```
      PROGRAM CYCLE C
      EXTERNAL G,F
      CALL MULTIG(3,2,1.,6,.01,30.,G,F)
      STOP
      END

      FUNCTION F(X,Y)                                   Right-hand side of the equation
      F=SIN(3.*(X+Y))
      RETURN
      END

      FUNCTION G(X,Y)                                   Boundary values and first approximation
      G=COS(2.*(X+Y))
      RETURN
      END

      SUBROUTINE MULTIG(NXO,NYO,HO,M,TOL,WMAX,U1,F)
      EXTERNAL U1,F
      DIMENSION EPS(10)                                 Multi-grid algorithm (see Fig. 1)
      DO 1 K=1,M
      K2=2**(K-1)
      CALL GRDFN(K,NXO*K2+1,NYO*K2+1,HO/K2)
    1 CALL GRDFN(K+M,NXO*K2+1,NYO*K2+1,HO/K2)
      EPS(M)=TOL
      K=M
      WU=0
      CALL PUTF(M,U1,0)
      CALL PUTF(2*M,F,2)
    5 ERR=1.E30
    3 ERRP=ERR
      CALL RELAX(K,K+M,ERR)
      WU=WU+4.**(K-M)
      WRITE(6,4)K,ERR,WU
    4 FORMAT(' LEVEL',I2,'    RESIDUAL NORM=',1PE10.3,'    WORK=',0PF7.3)
      IF(ERR.LT.EPS(K))GOTO 2
      IF (WU.GE.WMAX)RETURN
      IF(K.EQ.1.OR. ERR/ERRP.LT. .6)GOTO 3                η=.6
      CALL RESCAL(K,K+M,K+M-1)
      EPS(K-1)=.3*ERR                                     δ=.3
      K=K-1
      CALL PUTZ(K)
      GOTO 5
    2 IF (K.EQ.M)RETURN
      CALL INTADD(K,K+1)
      K=K+1
      GOTO 5
      END

      SUBROUTINE GRDFN(N,IMAX,JMAX,HH)                  Define an IMAX × JMAX
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)              array v^N .
      DATA IQ/1/
      NST(N)=IQ
      IMX(N)=IMAX
      JMX(N)=JMAX
      H(N)=HH
      IQ=IQ+IMAX*JMAX
      RETURN
      END

      SUBROUTINE KEY(K,IST,IMAX,JMAX,HH)
```

```
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)
      DIMENSION IST(1)
      IMAX=IMX(K)
      JMAX=JMX(K)
      IS=NST(K)-JMAX-1
      DO 1 I=1,IMAX
      IS=IS + JMAX
    1 IST(I)=IS
      HH=H(K)
      RETURN
      END
```

Set IST such that

$v^k(I,J) = Q(IST(I) + J)$,

and set IMAX = IMX(K)

JMAX = JMX(K)

HH = H(K)

```
      SUBROUTINE PUTF(K,F,NH)
      COMMON Q(18000),IST(600)
      CALL KEY (K,IST,II,JJ,H)
      H2=H**NH
      DO 1 I=1,II
      DO 1 J=1,JJ
      X=(I-1)*H
      Y=(J-1)*H
    1 Q(IST(I)+J)=F(X,Y)*H2
      RETURN
      END
```

$v^K \leftarrow H(K)^{NH} \cdot F^K$

```
      SUBROUTINE PUTZ(K)
      COMMON Q(18000),IST(200)
      CALL KEY(K,IST,II,JJ,H)
      DO 1 I=1,II
      DO 1 J=1,JJ
    1 Q(IST(I)+J)=0.
      RETURN
      END
```

$v^K \leftarrow 0$

```
      SUBROUTINE RELAX(K,KRHS,ERR)
      COMMON Q(18000),IST(200),IRHS(200)
      CALL KEY(K,IST,II,JJ,H)
      CALL KEY(KRHS,IRHS,II,JJ,H)
      I1=II-1
      J1=JJ-1
      ERR=0.
      DO 1 I=2,I1
      IR=IRHS(I)
      IO=IST(I)
      IM=IST(I-1)
      IP=IST(I-1)
      DO 1 J=2,J1
      A=Q(IR+J)-Q(IO+J+1)-Q(IO+J-1)-Q(IM+J)-Q(IP+J)
      ERR=ERR+(A+4.*Q(IO+J))**2
    1 Q(IO+J)=-.25*A
      ERR=SQRT(ERR)/H
      RETURN
      END
```

A Gauss-Seidel Relaxation sweep on the equation

$$\Delta_h v^K = v^{KRHS}$$

giving

$$ERR = ||\text{residuals}||_{L_2}$$

```
      SUBROUTINE INTADD(KC,KF)
      COMMON Q(18000),ISTC(200),ISTF(200)
      CALL KEY(KC,ISTC,IIC,JJC,HC)
      CALL KEY(KF,ISTF,IIF,JJF,HF)
      DO 1 IC=2,IIC
      IF=2*IC-1
      JF=1
```

Linear interpolation and addition

$$v^{KF} \leftarrow v^{KF} + I^{KF}_{KC}\, v^{KC}$$

```
      IFO=ISTF(IF)
      IFM=ISTF(IF-1)
      ICO=ISTC(IC)
      ICM=ISTC(IC-1)
      DO 1 JC=2,JJC
      JF=JF+2
      A=.5*(Q(ICO+JC)+Q(ICO+JC-1))
      AM=.5*(Q(ICM+JC)+Q(ICM+JC-1))
      Q(IFO+JF) = Q(IFO+JF)+Q(ICO+JC)
      Q(IFM+JF) = Q(IFM+JF)+.5*(Q(ICO+JC)+Q(ICM+JC))
      Q(IFO+JF-1)=Q(IFO+JF-1)+A
    1 Q(IFM+JF-1) = Q(IFM+JF-1)+.5*(A+AM)
      RETURN
      END

      SUBROUTINE RESCAL(KF,KRF,KRC)
      COMMON Q(18000),IUF(200),IRF(200),IRC(200)
      CALL KEY(KF,IUF,IIF,JJF,HF)
      CALL KEY(KRF,IRF,IIF,JJF,HF)
      CALL KEY(KRC,IRC,IIC,JJC,HC)
      IIC1=IIC-1
      JJC1=JJC-1
      DO 1 IC=2,IIC1
      ICR=IRC(IC)
      IF=2*IC-1
      JF=1
      IFR=IRF(IF)
      IFO=IUF(IF)
      IFM=IUF(IF-1)
      IFP=IUF(IF+1)
      DO 1 JC=2,JJC1
      JF=JF+2
      S=Q(IFO+JF+1)+Q(IFO+JF-1)+Q(IFM+JF)+Q(IFP+JF)
    1 Q(ICR+JC)=4.*(Q(IFR+JF)-S+4.*Q(IFO+JF))
      RETURN
      END
```

Residuals injection

$$v^{KRC} \leftarrow I^{coarse}_{fine}\,(v^{KRF} - \Delta_h v^{KF})$$

OUTPUT 12-13

```
EL 6   RESIDUAL NORM= 2.814E+01   WORK=  1.000
EL 6   RESIDUAL NORM= 2.764E+01   WORK=  2.000
EL 5   RESIDUAL NORM= 2.659E+01   WORK=  2.250
EL 5   RESIDUAL NORM= 2.555E+01   WORK=  2.500
EL 4   RESIDUAL NORM= 2.317E+01   WORK=  2.563
EL 4   RESIDUAL NORM= 2.095E+01   WORK=  2.625
EL 3   RESIDUAL NORM= 1.649E+01   WORK=  2.641
EL 3   RESIDUAL NORM= 1.285E+01   WORK=  2.656
EL 2   RESIDUAL NORM= 7.626E+00   WORK=  2.660
EL 2   RESIDUAL NORM= 3.840E+00   WORK=  2.664
EL 3   RESIDUAL NORM= 5.058E+00   WORK=  2.680
EL 4   RESIDUAL NORM= 8.006E+00   WORK=  2.742
EL 4   RESIDUAL NORM= 2.545E+00   WORK=  2.805
EL 5   RESIDUAL NORM= 9.736E+00   WORK=  3.055
EL 5   RESIDUAL NORM= 2.464E+00   WORK=  3.305
EL 6   RESIDUAL NORM= 1.064E+01   WORK=  4.305
EL 6   RESIDUAL NORM= 2.442E+00   WORK=  5.305
EL 6   RESIDUAL NORM= 2.399E+00   WORK=  6.305
EL 5   RESIDUAL NORM= 2.351E+00   WORK=  6.555
EL 5   RESIDUAL NORM= 2.303E+00   WORK=  6.805
EL 4   RESIDUAL NORM= 2.173E+00   WORK=  6.867
EL 4   RESIDUAL NORM= 2.043E+00   WORK=  6.930
EL 3   RESIDUAL NORM= 1.739E+00   WORK=  6.945
EL 3   RESIDUAL NORM= 1.453E+00   WORK=  6.961
EL 2   RESIDUAL NORM= 9.889E-01   WORK=  6.965
EL 2   RESIDUAL NORM= 6.183E-01   WORK=  6.969
EL 1   RESIDUAL NORM= 2.760E-01   WORK=  6.970
EL 1   RESIDUAL NORM= 5.170E-02   WORK=  6.971
EL 2   RESIDUAL NORM= 2.292E-01   WORK=  6.975
EL 3   RESIDUAL NORM= 5.465E-01   WORK=  6.990
EL 4   RESIDUAL NORM= 7.710E-01   WORK=  7.053
EL 4   RESIDUAL NORM= 1.163E-01   WORK=  7.115
EL 5   RESIDUAL NORM= 8.657E-01   WORK=  7.365
EL 5   RESIDUAL NORM= 1.058E-01   WORK=  7.615
EL 6   RESIDUAL NORM= 9.059E-01   WORK=  8.615
EL 6   RESIDUAL NORM= 1.052E-01   WORK=  9.615
EL 6   RESIDUAL NORM= 1.012E-01   WORK= 10.615
EL 5   RESIDUAL NORM= 9.759E-02   WORK= 10.865
EL 5   RESIDUAL NORM= 9.452E-02   WORK= 11.115
EL 4   RESIDUAL NORM= 8.710E-02   WORK= 11.178
EL 4   RESIDUAL NORM= 7.960E-02   WORK= 11.240
EL 3   RESIDUAL NORM= 6.389E-02   WORK= 11.256
EL 3   RESIDUAL NORM= 4.931E-02   WORK= 11.271
EL 2   RESIDUAL NORM= 2.916E-02   WORK= 11.275
EL 2   RESIDUAL NORM= 1.622E-02   WORK= 11.279
EL 2   RESIDUAL NORM= 1.017E-02   WORK= 11.283
EL 3   RESIDUAL NORM= 1.949E-02   WORK= 11.299
EL 4   RESIDUAL NORM= 3.128E-02   WORK= 11.361
EL 4   RESIDUAL NORM= 8.843E-03   WORK= 11.424
EL 5   RESIDUAL NORM= 3.710E-02   WORK= 11.674
EL 5   RESIDUAL NORM= 8.486E-03   WORK= 11.924
EL 6   RESIDUAL NORM= 4.007E-02   WORK= 12.924
EL 6   RESIDUAL NORM= 9.051E-03   WORK= 13.924
```

Error reduction by a factor greater than 10 per cycle.

Each cycle costs 4.3 WU

Insensitivity: Results would be practically the same for any $.005 \leq \delta \leq .5$ or any $0 \leq \eta \leq .65$

Conversion of Correction-Scheme (CS) Cycle C Program into Full-Approximation-Scheme (FAS) Program

① Change main driving routine (MULTIG) as shown on page 12-15

② Introduce the standard routines (given on page 12-16 for the rectangular case) PUTU & SUBTRUCT. (the latter is a slight modification of the first).

③ Write COARSRES(k, k+M), a routine which adds $L_k U^k$ to the right-hand side of grid k. This routine is easily written as a modification of the relaxation routine. The present example is shown on page 12-17.

④ Check that, for your linear problems, FAS gives exactly (upto round-off errors) the same output as CS.

```
      PROGRAM CYCLE C
      EXTERNAL G,F
      CALL MULTIG(3,2,1.,6,.01,30.,G,F)
      STOP
      END

      FUNCTION F(X,Y)                                   Right-hand side of the equation
      F=SIN(3.*(X+Y))
      RETURN
      END

      FUNCTION G(X,Y)                                   Boundary values and first approximation
      G=COS(2.*(X+Y))
      RETURN
      END

      SUBROUTINE MULTIG(NXO,NYO,HO,M,TOL,WMAX,U1,F)
      EXTERNAL U1,F
      DIMENSION EPS(10)                                 Multi-grid algorithm (see Fig. 1)
      DO 1 K=1,M
      K2=2**(K-1)
      CALL GRDFN(K,NXO*K2+1,NYO*K2+1,HO/K2)
    1 CALL GRDFN(K+M,NXO*K2+1,NYO*K2+1,HO/K2)
      EPS(M)=TOL
      K=M
      WU=0
      CALL PUTF(M,U1,0)
      CALL PUTF(2*M,F,2)
    5 ERR=1.E30
    3 ERRP=ERR
      CALL RELAX(K,K+M,ERR)
      WU=WU+4.**(K-M)
      WRITE(6,4)K,ERR,WU
    4 FORMAT(' LEVEL',I2,'    RESIDUAL NORM=',1PE10.3,'    WORK=',0PF7.3)
      IF(ERR.LT.EPS(K))GOTO 2
      IF (WU.GE.WMAX)RETURN
      IF(K.EQ.1.OR. ERR/ERRP.LT. .6)GOTO 3                η=.6
      CALL RESCAL(K,K+M,K+M-1)
      EPS(K-1)=.3*ERR
      K=K-1                                             δ=.3
      CALL PUTU(K)
      GOTO 5
    2 IF (K.EQ.M)RETURN
      CALL INTADD(K,K+1)
      K=K+1
      GOTO 5
      END

      SUBROUTINE GRDFN(N,IMAX,JMAX,HH)                  Define an IMAX × JMAX
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)          array v^N .
      DATA IQ/1/
      NST(N)=IQ
      IMX(N)=IMAX
      JMX(N)=JMAX
      H(N)=HH
      IQ=IQ+IMAX*JMAX
      RETURN
      END

      SUBROUTINE KEY(K,IST,IMAX,JMAX,HH)
```

CALL PUTU(K+1,K)

CALL COARSRES(K,K+M)

CALL SUBTRCT(K+1,K)

PUTU

```
      SUBROUTINE PUTU(KF,KC)
      COMMON Q(18000),IUF(200),IUC(200)
      CALL KEY(KF,IUF,IIF,JJF,HF)
      CALL KEY(KC,IUC,IIC,JJC,HC)
      DO 1 IC=1,IIC
      IF=2*IC-1
      IFO=IUF(IF)
      ICO=IUC(IC)
      JF=-1
      DO 1 JC=1,JJC
      JF=JF+2
      Q(ICO+JC)=Q(IFO+JF)
    1 CONTINUE
      RETURN
      END
```

SUBTRCT

```
      SUBROUTINE ~~PUTU~~ SUBTRCT(KF,KC)
      COMMON Q(18000),IUF(200),IUC(200)
      CALL KEY(KF,IUF,IIF,JJF,HF)
      CALL KEY(KC,IUC,IIC,JJC,HC)
      DO 1 IC=1,IIC
      IF=2*IC-1
      IFO=IUF(IF)
      ICO=IUC(IC)
      JF=-1
      DO 1 JC=1,JJC
      JF=JF+2
      Q(ICO+JC)=~~Q(IFO+JF)~~ Q(ICØ+JC) - Q(IFØ+JF)
    1 CONTINUE
      RETURN
      END
```

```
      COMMON/GRD/NST(20),IMX(20),JMX(20),H(20)
      DIMENSION IST(1)
      IMAX=IMX(K)
      JMAX=JMX(K)
      IS=NST(K)-JMAX-1
      DO 1 I=1,IMAX
      IS=IS + JMAX
    1 IST(I)=IS
      HH=H(K)
      RETURN
      END
```

Set IST such that

$v^k(I,J) = Q(IST(I) + J)$,

and set IMAX = IMX(K)

JMAX = JMX(K)

HH = H(K)

```
      SUBROUTINE PUTF(K,F,NH)
      COMMON Q(18000),IST(600)
      CALL KEY (K,IST,II,JJ,H)
      H2=H**NH
      DO 1 I=1,II
      DO 1 J=1,JJ
      X=(I-1)*H
      Y=(J-1)*H
    1 Q(IST(I)+J)=F(X,Y)*H2
      RETURN
      END
```

$v^K \leftarrow H(K)^{NH} \cdot F^K$

```
      SUBROUTINE PUTZ(K)
      COMMON Q(18000),IST(200)
      CALL KEY(K,IST,II,JJ,H)
      DO 1 I=1,II
      DO 1 J=1,JJ
    1 Q(IST(I)+J)=0.
      RETURN
      END
```

$v^K \leftarrow 0$

COARSRES

```
      SUBROUTINE ~~RELAX~~(K,KRHS~~,ERR~~)
      COMMON Q(18000),IST(200),IRHS(200)
      CALL KEY(K,IST,II,JJ,H)
      CALL KEY(KRHS,IRHS,II,JJ,H)
      I1=II-1
      J1=JJ-1
      ERR=0.
      DO 1 I=2,I1
      IR=IRHS(I)
      IO=IST(I)
      IM=IST(I-1)
      IP=IST(I+1)
      DO 1 J=2,J1
      A=Q(IR+J)-Q(IO+J+1)-Q(IO+J-1)-Q(IM+J)-Q(IP+J)
      ~~ERR=ERR+(A/4.+Q(IO+J))**2~~
    1 ~~Q(IO+J)=.25*A~~ Q(IR+J) = -A - 4. * Q(IO+J)
      ERR=SQRT(ERR)/H
      RETURN
      END
```

(minus)

A Gauss-Seidel Relaxation sweep on the equation

$$\Delta_h v^K = v^{KRHS}$$

giving

$$ERR = ||\text{residuals}||_{L_2}$$

```
      SUBROUTINE INTADD(KC,KF)
      COMMON Q(18000),ISTC(200),ISTF(200)
      CALL KEY(KC,ISTC,IIC,JJC,HC)
      CALL KEY(KF,ISTF,IIF,JJF,HF)
      DO 1 IC=2,IIC
      IF=2*IC-1
      JF=1
```

Linear interpolation and addition

$$v^{KF} \leftarrow v^{KF} + I^{KF}_{KC}\, v^{KC}$$

13. THE MULTI-GRID SOFTWARE

IMPLEMENTATION OF THE MULTI-GRID METHOD FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

Fred G. Gustavson
Mathematical Sciences Department
IBM T. J. Watson Research Center
Yorktown Heights, New York 10598

Introduction: In the MULTI-GRID method developed by A. Brandt [1], [2] for solving partial differential equations (PDE), the boundary-value problem is discretized in several grids of widely different mesh sizes. Interaction between these levels enables one to solve the possibly nonlinear system of N discrete equations in O(N) operations and to conveniently adapt the discretization (local mesh size, local order of approximation, etc.) to the evolving solution in a nearly optimal way.

This paper presents an overview of a system of programs that allow a user to generate and manipulate arbitrary two-dimensional grids. The user would write a high-level program that would apply the MULTI-GRID theory to his PDE. In so doing he would call upon system programs that allow him to conveniently interact between various grids he must generate. In effect, the system programs convert the FORTRAN language into a language to solve PDE's via the MULTI-GRID approach.

Our overview will show how one can effectively represent and manipulate arbitrary two-dimensional grids. Generalization to three and higher dimensions will be mentioned. Section 2 of the paper will describe our data structure. Section 3 will catalog some of the system programs and the reason for their existence. Some brief details of their implementation will be mentioned.

Section 2: A MULTI-GRID DATA STRUCTURE

The basic unit of work in the MULTI-GRID method is the cost of one relaxation sweep over the finest grid. Sweeping, therefore, must be an efficient process. Assuming an x-y coordinate system* we will set up a uniform mesh given by

$$x = x_o + ih_x \qquad y = y_o + jh_y \tag{1}$$

where x_o, y_o, h_x, h_y are input parameters and grid points (x,y) are specified by integers i and j. We will assume that sweeping is done in the coordinate direction; so, for example, we may sweep vertical lines in the negative x direction. One-dimensional storage is very attractive for storing a contiguous line of data points since a minimal amount of pointer structure is necessary to access it quickly. When there are gaps in the line we shall represent the data points as a union of intervals; an interval is a contiguous line of data points. An example will clarify these ideas.

*More generally, a coordinate system described by two parameters.

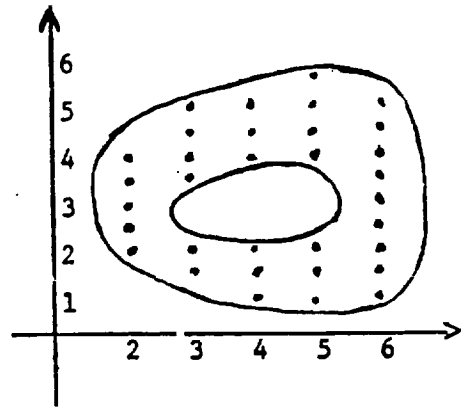


Figure 1

Let $x_o = y_o = 0$ and $h_x = 2h_y = 1$. In Figure 1 there are lines of data at x values between 2 and 6; i.e., integers in the interval [2,6]. For each i, $2 \le i \le 6$, there are one or more intervals of grid points; e.g., for $i = 5$ there are two intervals described by integer intervals [2,4] and [8,11]. (See equation (1) for the relation between numerical value y and integer value j.)

MULTI-GRID requires four different types of sweeps. They are adaptive sweeps, partial sweeps, segmental refinement, and selective sweeping. Because of this we must be able to break up and/or combine grids easily into other grids. This requires breaking up intervals and reforming them; in many cases, however, the associated data does not have to be moved. To facilitate this we propose a more flexible pointer structure, called the QUAD* structure, to describe an interval.

Our data structure for a grid is made of two parts; logical or pointer arrays to describe grid point (x,y) and data arrays for numerical values associated with grid point (x,y). Pointer and integer data will be stored in L space and numerical data in Q space. Assume that our grid is arranged in lines pointing in the y direction. A quad consisting of four integer values will describe the first x interval of lines. An x quad representing interval [a,b] has the following representation:

QUAD =	LOC	a	ℓ	NEXT
	1	2	3	4

The second and third entries describe a and b; i.e., QUAD(2) = a and QUAD(3) = b-a+1. The third entry of QUAD is the length (number of y lines) of the interval. Entries one and four of QUAD are pointers into L space. QUAD(1) contains the location in L space of the y QUAD describing line $x = x_o + ah_x$. By convention the y QUAD describing line $x = x_o + ih_x$, where $a \le i \le b$, is located at position LOC + 4(i-a) of L. QUAD(4) contains the location in L space of the <u>next</u> x quad, if any, of the

*QUAD is an abbreviation of quadruple.

given interval of lines. The structure of any y QUAD is the same except LOC points into Q space; i.e., the y intervals refer to grid points. Thus LOC points at the numerical values that are associated with the given grid point. Figure 2 contains the QUAD structure for Figure 1.

L-SPACE

ADDRESS	QUAD			
	LOC	a	ℓ	NEXT
⋮	⋮	⋮	⋮	⋮
α	β	2	5	0
⋮	⋮	⋮	⋮	⋮
β	γ	4	5	0
β+4	γ+5	3	2	β+20
β+8	γ+11	2	3	β+24
β+12	γ+17	2	3	β+28
β+16	γ+24	2	9	0
β+20	γ+7	7	4	0
β+24	γ+14	8	3	0
β+28	γ+20	8	4	0
⋮				

Q-SPACE

ADDRESS	VALUE
⋮	⋮
γ	v(2,2)
⋮	⋮
γ+4	v(2,4)
γ+5	v(3,1.5)
γ+6	v(3,2)
γ+7	v(3,3.5)
⋮	⋮
γ+11	v(4,1)
⋮	⋮
γ+14	v(4,4)
⋮	⋮
γ+17	v(5,1)
⋮	⋮
γ+20	v(5,4)
⋮	⋮
γ+24	v(6,1)
⋮	⋮
γ+32	v(6,5)
⋮	⋮

Figure 2

If we want to process the points on line $i = 5$ we access location $L(\alpha) + 4(5-2) = \beta + 12$ in L. The QUAD at location $\beta+12$ tells us that grid points (5,1), (5,1.5), and (5,2) (see equation (1) to make translation from j to y) are present; in addition, another interval is located at L location $\beta+28$. Looking at the QUAD at location $\beta+28$ tells us that grid points (5,4), (5,4.5), (5,5), and (5,5.5) are present. Also the numerical values associated with those seven points are stored, in order, at locations $\gamma+17$ to $\gamma+23$ of Q space; e.g., $v(5,1.5) = Q(\gamma+18)$.

Section 3: SOME SYSTEM PROGRAMS FOR MULTI-GRID

The System Programs for MULTI-GRID will be classified according to MULTI-GRID operations as described by the theory [1], [2]. Because of our choice of data structure, some of these operations will be subdivided into three parts. These parts are logical creation of pointer structure, allocation of Q space for numerical values, and putting numerical values on a grid. Table 1 below details our routines: they are listed as Routines 0 to Routines 4.

Routines 0 - INITIALIZATION, COMPACTIFICATION

Name	FUNCTION/DESCRIPTION
INIT	Get parameters from user; set up L and Q space
CMPCTL(h)	Compactify L space via garbage collection
CMPCTQ(k)	Compactify Q space via garbage collection

Routines 1 - LOGICAL GRID OPERATIONS

Name	FUNCTION/DESCRIPTION
LGRDCR	Create logical grid structure given a characteristic function CR(x,y); i.e., $CR(x,y) = \begin{cases} 0 & (x,y) \notin \text{grid} \\ 1 & (x,y) \in \text{grid} \end{cases}$
LGRD2F	Create logical grid structure given 2 functions, i.e., Inequalities $a \le x \le b$ and $f(x) \le y \le g(x)$ describe the boundary curve of the grid
LCOARS(k,ℓ)	Create logical coarsening of grid k; call it grid ℓ
LUNION(k,ℓ,m)	Create logical union of two grids; $m = k \cup \ell$
LSECT(k,ℓ,m)	Create logical intersection of two grids, $m = k \cap \ell$
LTPOSE(k,ℓ)	Create logical transpose of two grids; $\ell = k^T$: i.e., Convert a grid stored by vertical lines into one stored by horizontal lines.
LMINUS(k,ℓ,m)	Create logical difference of two grids; $m = k-\ell$
LINNER(k,ℓ)	Create set of inner points from grid k; ℓ = INNER(k)
DELETE(k)	Delete logical structure of grid k

Routines 2 - ALLOCATION OF Q SPACE AND INSERTION OF NUMERICAL VALUES

Name	FUNCTION/DESCRIPTION
QSPACE(k)	Allocate storage in Q and supply pointers in L for grid k
QOFF(k)	Delete storage in Q previously allocated to grid k.
POINTO(k,ℓ)	Point the logical structure of grid k into the numerical storage of grid ℓ
TFER(k,ℓ)	Transfer numerical values from grid k to grid ℓ
TFERTP(k,ℓ)	Transfer numerical values in transposed order from grid k to grid ℓ
PUTF(k,F)	Put function values F(x,y) into the Q space of grid k. Each grid point (x,y) can have several values associated with it. A parameter NF standing for number of function values is associated with each grid. If NF > 1 then PUTF inserts the first function value into Q space
PUTVF(k,VF)	Put vector function VF(x,y) into the Q space of grid k.
PUTSF(k,j,F)	Put scalar function F(x,y) into the jth function of grid k; $1 \le j \le NF$
PUTC(k,C) PUTVC(k,VC) PUTSC(k,j,C)	Replace function by constant in the above descriptions of PUTF, PUTVF, and PUTSC.

Routines 3 - INTERPOLATIONS

Name	FUNCTION/DESCRIPTION
CTF(k,ℓ)	Transfer coarse values from grid k to fine values on grid ℓ
FTC(k,ℓ)	Transfer fine values from grid k to coarse values on grid ℓ
LINT(k,ℓ)	Perform linear interpolation on grid k and put interpolants on grid ℓ

Routines 4 - EXPOSURE ROUTINES

Name	FUNCTION/DESCRIPTION
KEYS	Supply user with array JR so that v(I,J) = Q(JR(IR+I)+J). JR is the KEY array and S stands for single. Single means each line consists of exactly one interval; i.e., no continuation strings.
KEYG	Supply user with several arrays so that user can sweep easily an arbitrary grid.

Table 1

Several of the above routines are easy to implement. The more difficult ones are in Routines 1 and 3. Since our data structure is predicated on the interval many of the programs in Routines 1 become simpler if one considers the corresponding operation for interval. In LUNION, for example, we consider the problem of finding a union of intervals; i.e., given $[a_1,b_1],\ldots,[a_I,b_I]$ and $[c_1,d_1],\ldots,[c_J,d_J]$ we want to form $[e_1,f_1],\ldots,[e_K,f_K]$ as the union of these two lines. An efficient algorithm can be implemented to solve this subproblem; the algorithm can then be used as a subroutine within LUNION to find the union of two grids.

A similar algorithm exists for the intersection of two sets of intervals. However a complication arises which necessitates a type of recursive programming. To see this, suppose $[a,b]$ and $[c,d]$ have a nonempty intersection $[e,f]$. If $[e,f]$ represents an interval of lines that are common to both grid k and grid ℓ then $[e,f]$ may <u>not</u> be $m = k \cap \ell$. For this to be true each i, $e \le i \le f$, must have the property that two lines in grids k and ℓ associated with i have a nonempty intersection. In general $[e,f]$ is itself an intersection of intervals. To find this out we must call the intersection of intervals program for each i; thus we have recursion of depth d equal to the number of dimensions. This phenomena occurs whenever grid points are deleted from a region; a change occurs in the quad structure when a part of the region becomes empty. Another example is the logical coarsening routine in which a fine mesh is replaced with a coarse one; e.g., $H_x = 2h_x$ and $H_y = 2h_y$. There are less points in the coarse grid and possibly the quad structure changes because of the coarsening.

Probably the most difficult routine to implement is the logical transpose routine. The problem here is to convert vertical lines into horizontal lines; i.e., interchange the order of sweeping. We will only mention that our solution is $O(q \log q)$ where q is the number of quads. The $\log q$ comes in because it appears crucial to sort separately the end points of the entire set of intervals.

The Routines 3 are difficult because we accept essentially arbitrary grid configurations. Let K and k represent a coarse and fine grid and assume that K contains values which we want to interpolate to k. We use bilinear interpolation and everything is okay when there are enough coarse points near a fine point to interpolate. It turns out, even under reasonable assumptions, that there are a large number of abnormal cases to consider. Proper programming requires code to handle these cases no matter how infrequently they might occur. It is the cataloging and programming of these cases that makes the Routines 3 difficult. Also, we want to program so that code goes over to higher order interpolation; e.g., cubic interpolation.

Acknowledgement

The implementation described in this report has benefitted from many fruitful discussions with Achi Brandt, Allan Goodman and Donald Quarles. In addition, Allan Goodman has implemented the programs in Routines 2.

References

Brandt, A. - "Multi-Level Adaptive Techniques (MLAT) I. The Multi-Grid Method, RC 6026, June 3, 1976.

Brandt, A. - "Multi-Level Adaptive Solutions to Boundary-Value Problems, RC 6159, August 20, 1976.

MULTI-GRID SOFTWARE

The User writes a high level program to apply MULTI-GRID theory to his PDE

User calls on system pgms. to allow

a) Creation of various grids

b) Interaction between grids

c) Efficient use of storage

d) Fast execution

proportional to the operation counts predicted in the theory

Implementation is for 2D

Can be generalized to 3 and higher dimensions

Processes NEEDED

1) Place function values at a grid point
2) Relaxation Sweep over a grid
3) Coarse-to-fine interpolation
4) Fine-to-coarse residual xfer
5) Driver algorithm

Flexible 1-D Structure needed to represent Uniform Grid

Requirement

Efficient for sweeping

4 types

a) adaptive
b) partial
c) segmental refinement
d) selective refinement

$\Rightarrow$ Grid must be easy to change and modify

BASIC OPERATIONS

LOGICAL

$n \leftarrow$ UNION(k, l)

$m \leftarrow$ INTERSECTION(k, l)

$l \leftarrow$ TRANSPOSE(k)

$m \leftarrow$ DIFFERENCE(k, l)

$l \leftarrow$ COARSE(k)

$l \leftarrow$ INTERIOR$(t)(k)$ $\quad t = \begin{smallmatrix} \cdot & & \cdot \\ & \circ & \\ \cdot & & \cdot \end{smallmatrix}$

INTERPOLATION with GRID of next size

CTF(k, l) - xfer Coarse values from k to fine values on l

FTC(k, l) - residual transfer

LINT(k, l) - perform linear interpolation on k and put interpolants on grid l

GRID CREATION

LGRDCR - characteristic function $CR(x,y)$

$$CR(x,y) = \begin{cases} 0 & (x,y) \notin \text{grid} \\ 1 & (x,y) \in \text{grid} \end{cases}$$

LGRD2F - grid defined by

$a \le x \le b$ and $f(x) \le y \le g(x)$

i.e., by two functions

GRID EXPOSURE

KEYS

$V(I,J) = Q(JR(IR+I) + J)$

supply user with JR( ) and IR to access 1D store

KEYG - Supply several arrays so user can sweep arbitrary grid.

Allocation of Q space and Insertion of Numerical values

QSPACE(h) Allocate storage for h

QOFF(h) Release storage for h

POINTO(h, l) Point logical structure of h into the numerical storage of l

TFER(h, l) Xfer numerical values from h to l

TFERTP(h, l) Xfer numerical values in transposed order from h to l

PUTF(h, F) <u>Put</u> function value F(x,y) into the Q space of grid h

BASIC OPERATIONS

PUTVF(h, VF) Put vector function VF(x,y) into the Q space of h

PUTSF(h, j, F) Put scalar function F(x,y) into the jth function of grid h; $1 \le j \le nf$

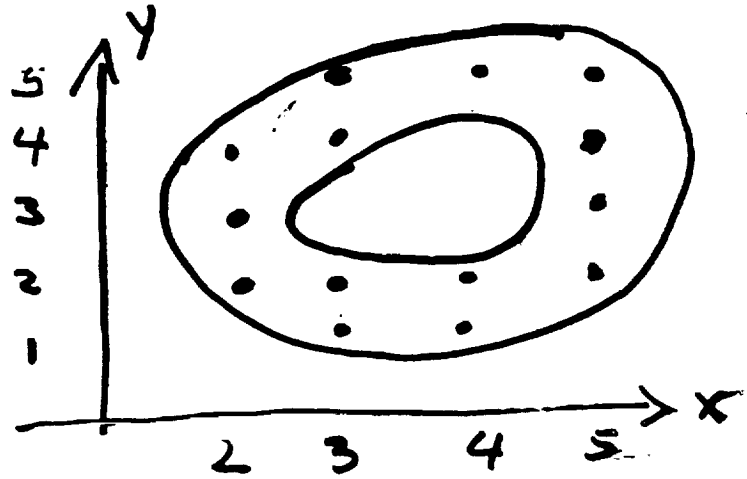
PUTC(h, C)
PUTVC(h, VC)
PUTSC(h, j, C)
} Replace function by constant in the above descriptions of PUTF, PUTVF, and PUTSF

INITIALIZATION, COMPACTIFICATION

INIT Get parameter from user; set up L and Q space

CMPCTL Compactify L space via garbage collection

CMPCTQ Compactify Q space via garbage collection



x=2	[2, 4]	
x=3	[1, 2]	[4, 5]
x=4	[1 2]	[5, 5]
x=5	[2, 5]	

Address space LSPACE

γ	B	2	4	0

β	α	2	3	0
β+4	α+3	1	2	β+16
β+8	α+7	1	2	β+20
β+12	α+10	2	4	0
β+16	α+5	4	2	0
β+	α+9	5	1	0

Q Space

α	v(2,2)
	v(2,3)
	v(2,4)
α+3	v(3,1)
	v(3,2)
α+5	v(3,4)
	v(3,5)
α+7	v(4,1)
	v(4,2)
α+9	v(4,5)
α+10	v(5,2)
	v(5,3)
	v(5,4)
	v(5,5)

DATA Structure

QUAD : | LOC | a | ℓ | NEXT |

REPRESENTS $[a, b]$

$$\ell = b - a + 1$$

LOC and NEXT are pointers

There are two types of quads
- distinguished by LOC

a) Logical quad - LOC points at other quad - LOC points to L space

b) Numerical quad - LOC points at numerical data - LOC points at Q space

NEXT always points at L space
- Represents the connective tissue linking together a union of intervals

WITH STRUCTURE

a) Easy to sweep

b) Good for any number of dimensions

c) Low storage overhead

a) Chain way through QUADS to sweep grid

b) Logical QUAD's represent first d-1 dimension - QUAD's of last dimension point to Q space

c) OVERHEAD/DATA RATIO $\approx$ surface area/volume $\rightarrow 0$

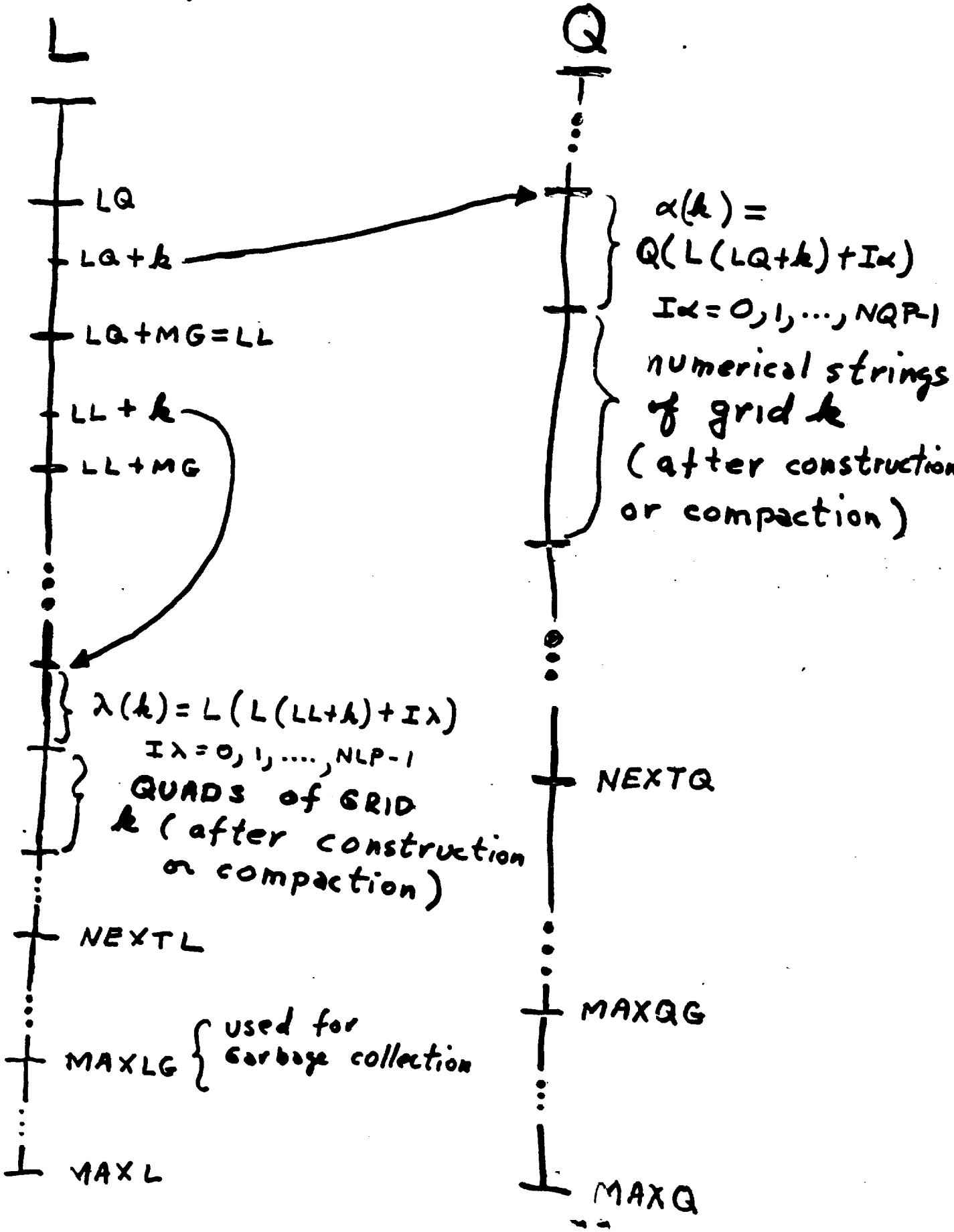
as $h = (h_1, \ldots, h_d) \rightarrow 0$

SOFTWARE DESIGN DECISIONS

Separate Grid Processing into 3 parts

1) LOGICAL STRUCTURE CREATION
2) Allocation of Numerical Space - called Q space
3) Placement, Access, and Transfer of Numerical values

Layout of L and Q



GRID REPRESENTATION

1) Fixed parameters
   a) logical
   b) numerical

a) NF - # of functions at each grid point
ICOL - 1 for COL order, 0 for ROW order
ITYPE - 0 if GENERAL
IQS - the grid # whose numerical storage is used; 0 if no storage allocated
LAG - the next grid # in a cycle of storage-associated grids; 0 if none
LOC - address of 1st quad
NSETS - # of outer quads
NCOL - # of strings
NST - # of inner quads
MST - maximum # of quads in a string
NLL - amount of L space
NLQ - amount of Q space
IMN - minimum outer dimension
IMX - maximum outer "
JMN - minimum inner "
JMX - maximum inner "

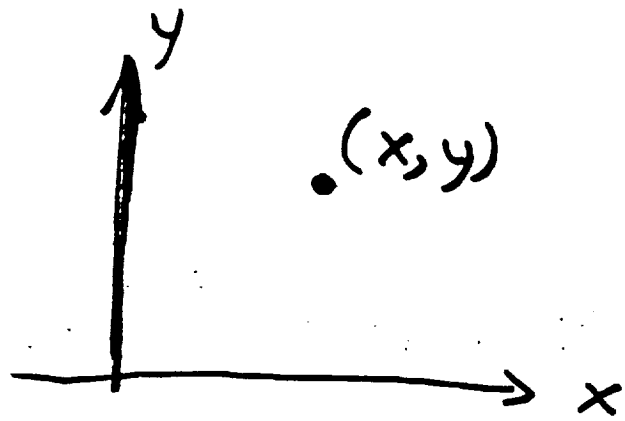
b)

h_x

h_y

x_0

y_0



$$x = x_0 + i h_x$$
$$y = y_0 + j h_y$$

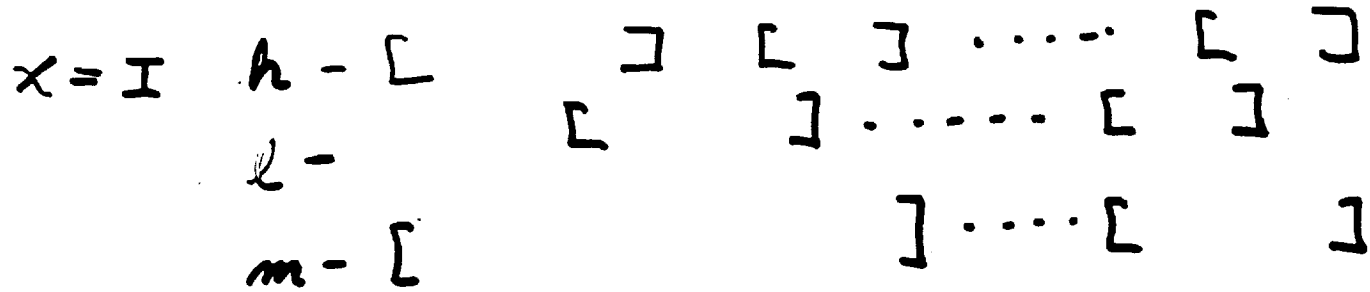
1-1 relation between Grid points and (i, j)

---

2) QUAD STRUCTURE

Operations on Grids

LUNION $\quad m = h \cup l$



Problem

given $A = \bigcup [a, b]$

$B = \bigcup [c, d]$

form $C = A \cup B = \bigcup [e, f]$

An efficient algorithm can be implemented using the QUAD data structure to solve the above Problem

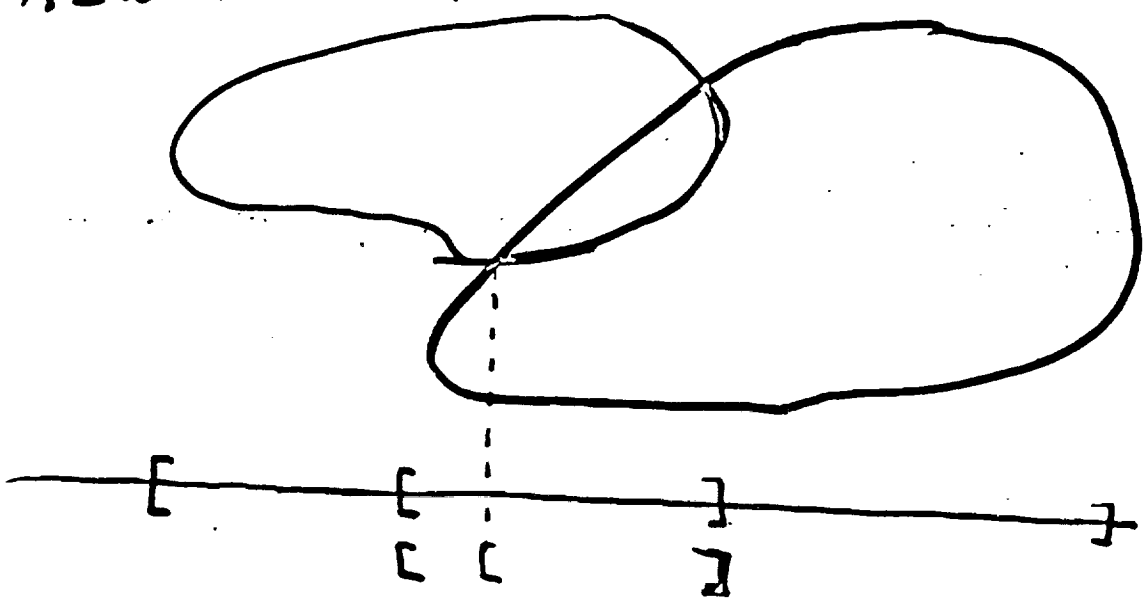
This algorithm is then used as a subroutine in LUNION

LSECT $m = h \cap l$

Problem : GIVEN $A = \cup[a,b]$, $B = \cup[c,d]$
Find $C = A \cap B = \cup[e,f]$

Use efficient solution to problem as subroutine in LSECT

NEW PROBLEM



Actual intersection $\subset$ computed intersection

General case $[e,f] \in l \cap h$ IFF for each $i \in [e,f]$ the 2 strings in h and l have a non-empty intersection

In general $[e,f] = \cup [e_i, f_i]$

REMEDY OF .....

MUST RECURSE TO LOWEST DIMENSION TO FIND OUT STRUCTURE OF [e, f]

This phenomina occurs whenever GRID POINT ARE DELETED FROM A REGION; a change occurs in the logical quad structure when part of a region becomes empty

---

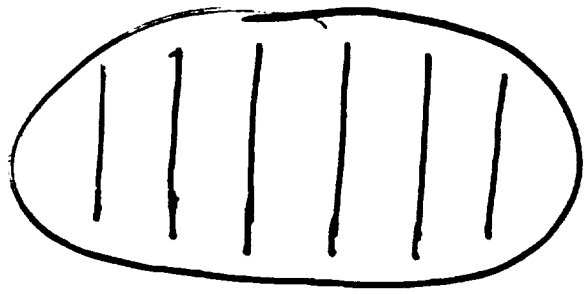
COARSEN ← LCOARS(h)

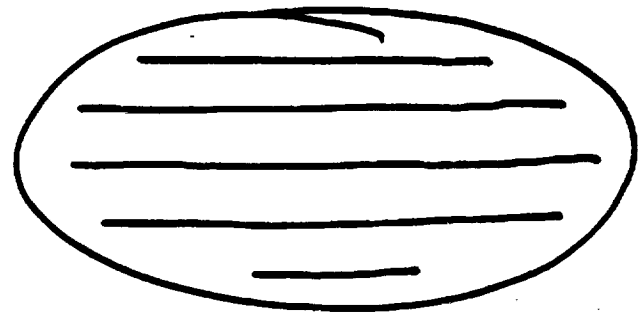
is an example of the above phenomina e.g. $H_x = 2h_x$, $H_y = 2h_y$

There are less points in the coarse grid and we must recurse to the lowest dimension to determine the quad stucture at the outer dimension.

This program is quite similar to LSECT

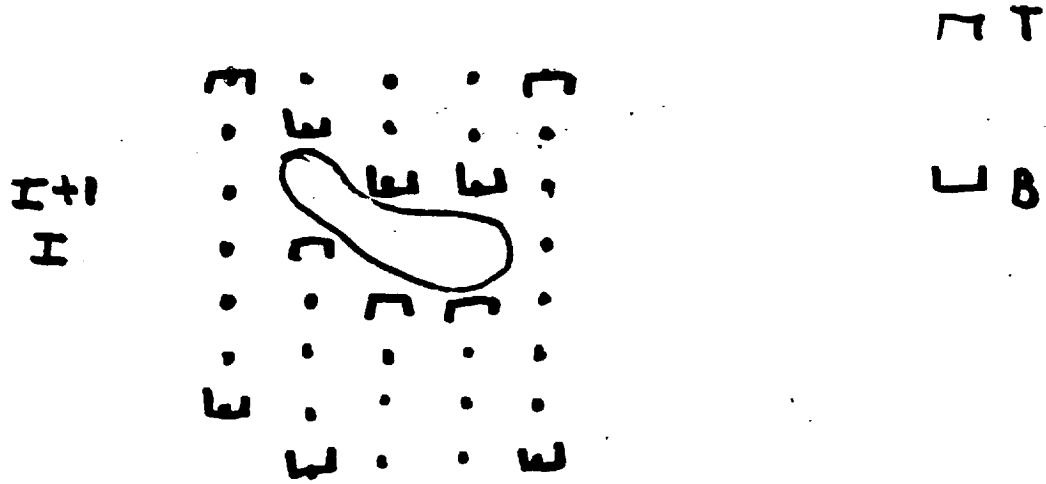
TRANSPOSE





FUNDAMENTAL RELATION

$$S_{I+1} = S_I - T_I + B_{I+1}$$



ALGORITHM FOR LTPOSE

① Merge SORT bottom parts of intervals
" " top " " "

② Perform 2 pass algorithm

a) Pass 1 - Determine the outer quad structure in order to layout the logical storage. Use

$$|S_{I+1}| = |S_I - T_I| + |B_{I+1}|$$

b) Pass 2 - Fill out the logical quad structure that was defined in Pass 1. Use

$$S_{I+1} = S_I - T_I + B_{I+1}$$

Modular Design

Subroutine SI (A, S, B)

$A = \cup [a, b]$, $S = \{s_1, \ldots, s_k\}$

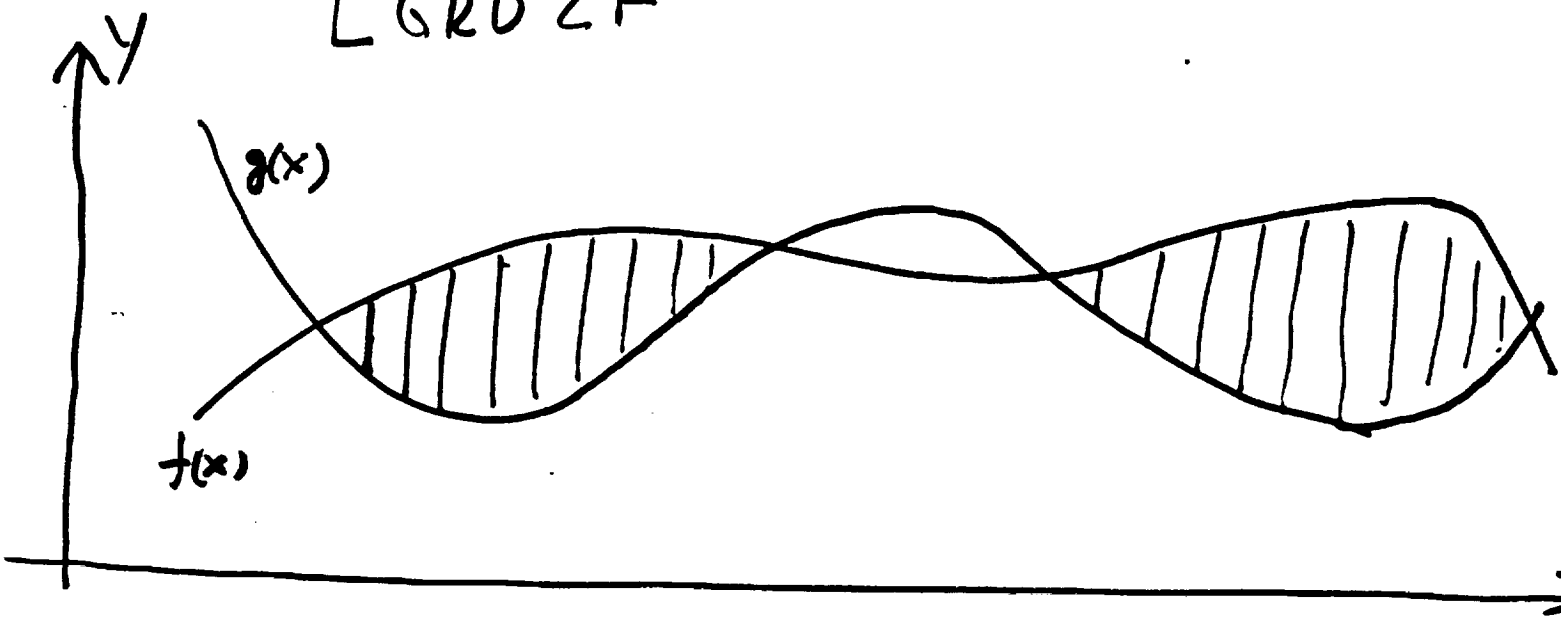
$B = \cup [c, d]$

Subroutine AI (A, S, B)

Cost

$O(q \log q)$ where q is # of quads

LGRD2F



NEEDED efficient CEIL and FLOOR functions as FORTRAN interge divide and MOD functions truncate towards zero.

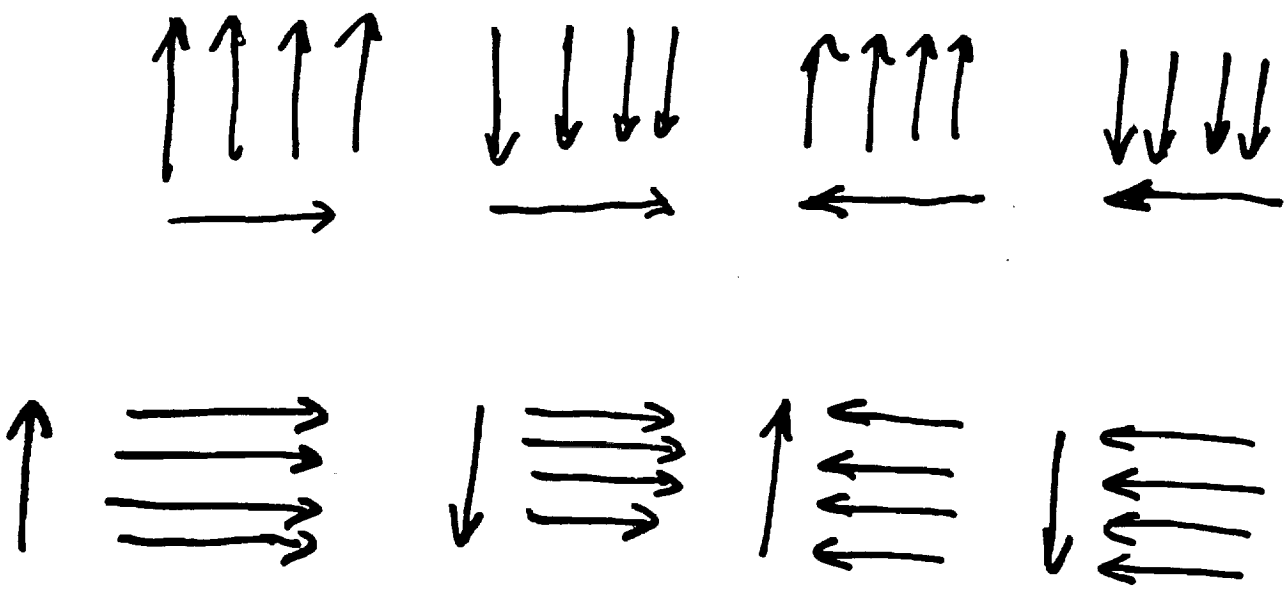
Exposure Routines

KEYS - one quad per string

KEYG - multiply quads per string

ways to sweep a grid (assuming direction along coordinate lines) is

$2^d d!$ (8 ways in 2D)

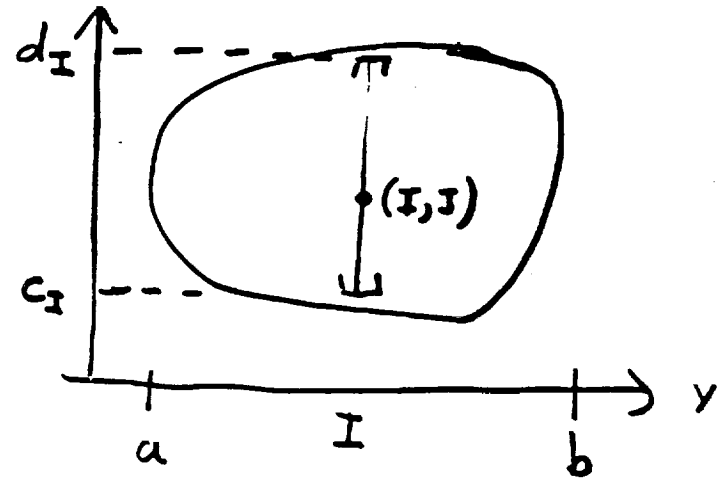


DESCRIPTION of KEYS

KEYS:

User inputs

ΔI and ΔJ



if $\Delta I = 1$, $\Delta J = 1$, and ICOL = 1 then

KEYS returns

IBEG = a

IEND = b

IR = -a and arrays JBEG, JEND, and JR of size ncol = b-a+1

JBEG(I) = c_I

JEND(I) = d_I

JR(I) = LOC(I)

Now $v(I,J) = Q(JR(IR+I)+J)$

Modular Programming vs Efficiency

a) Subroutine calls and linkage costly

b) Embed subroutines in main algorithm

c) Use computed go to's and minimize internal linkage

d) Hopefully get best of both worlds

FORTRAN PITFALLS

a) I did <u>not</u> write correct program the first time

b) Bugs were easier to find because the program was modular – I could isolate the error to a small section of code

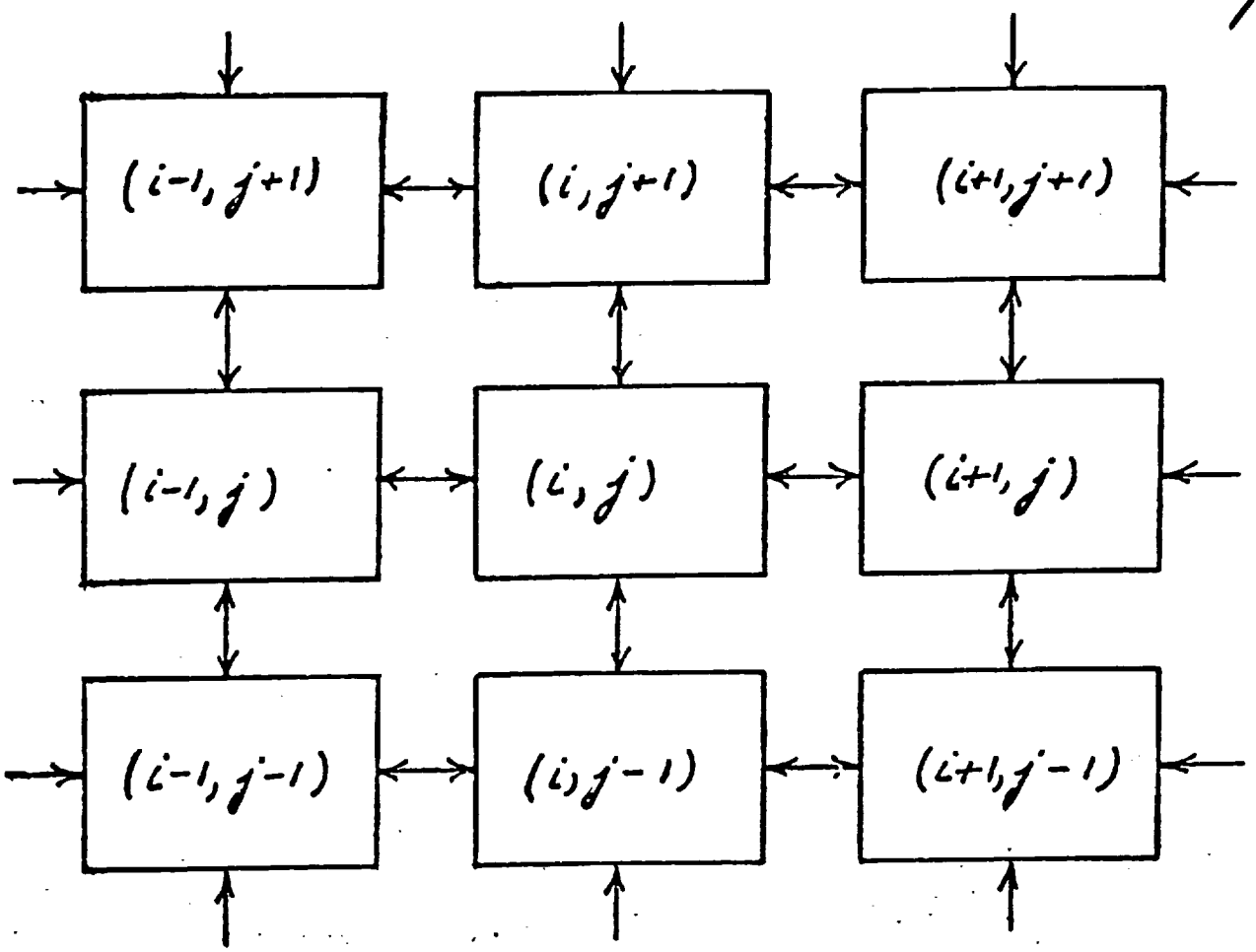
c) FORTRAN is too detailed a language → Assembly

d) Errors occurred because I used many variable to control looping and iterations – these got out of sync in exceptional cases.

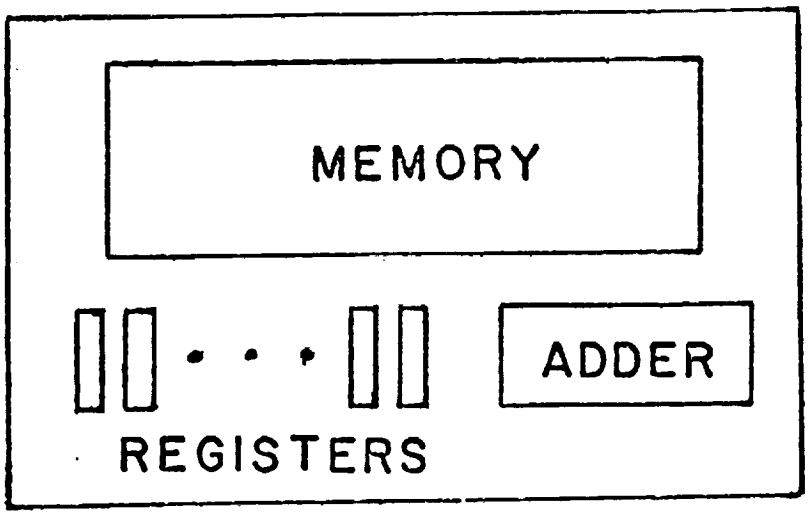
14. MULTI-GRID PROCESSES ON LARGE ARRAY COMPUTERS

Fig. 3

14-1



CELL ARRAY



ONE CELL

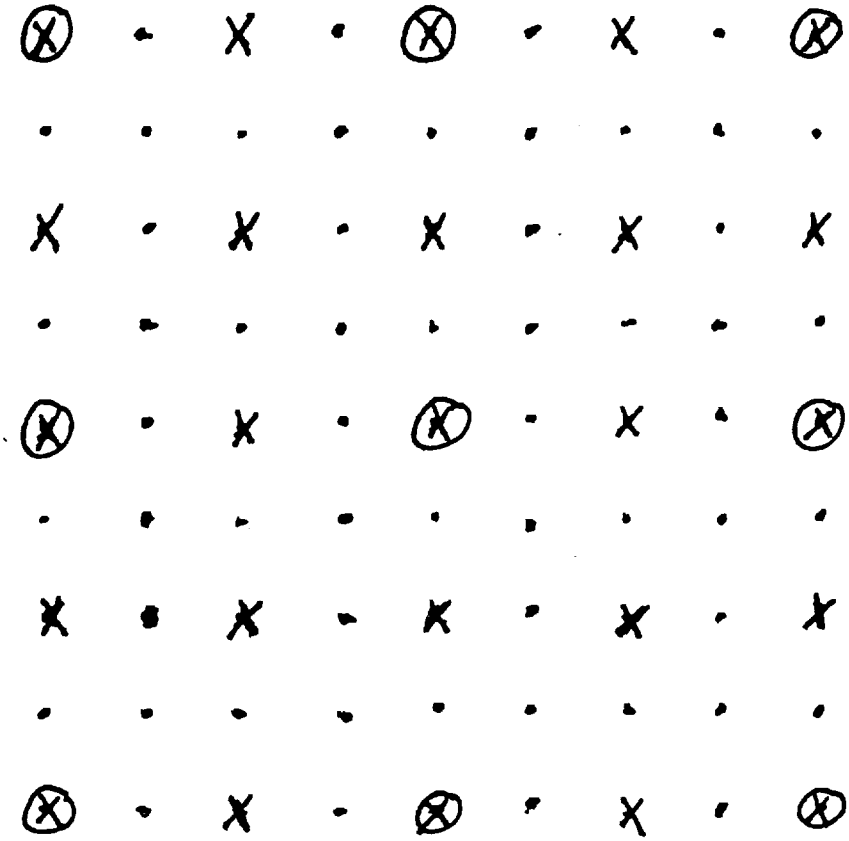
14-2

Table 5

TIME AND MEMORY ESTIMATES

Problem	Calculation Time for Interior Points per Time Step (sec)	Calculation Time for Boundary Points per Time Step (sec)	Calculation Time for the Pressure Field per Time Step (sec)	Calculation Time for the Velocity Field per Time Step (sec)	Total Calculation Time per Time Step (sec)	Memory per Cell (words/bits)	Total Calculation Time for One Time Step on a Fast Conventional Computer (sec)
#1 1/3 million point problem	0.060	0.027	0.076	0.011	0.087	432/~14K*	20.0
#2 1 million point problem	0.242	0.112	0.309	0.045	0.354	1584/~50K	81.9
#3 10 million point problem	0.242	0.112	0.309	0.045	0.354	1587/~50K	839

*1K = 1024



S_{CE} = # transfers to do C&E from level ($M \to 2$) & return ($2 \to M$).

$$= 2\sum_{j=2}^{M-1}(2^j - 2) = 4\left[2^{M-1} - (M-1)\right]$$

S_{IP} = # transfers factor to do <u>one</u> iteration at each level from M to 2.

$$= 2\sum_{j=0}^{M-2} 2^j = 2\left(2^{M-1} - 1\right)$$

K = # iterations at each level (Avg.)

N_0 = # transfers per iteration in a compressed grid

$N_{CE} = K \cdot N_0 + S_{CE} = K N_0 + 4\left[2^{M-1} - (M-1)\right]$

$N_{IP} = K \cdot N_0 \cdot S_{IP} = 2K N_0 (2^{M-1} - 1)$

$N_{CE}/N_{IP} = \left\{K N_0 + 4\left[2^{M-1} - (M-1)\right]\right\} / 2K N_0 (2^{M-1} - 1) < 1$

$\approx \frac{2}{K N_0}$ for $M \gg 1$; $N_0 = O(10)$; $K \geq 2$

$N_{CE}/N_{IP} \approx \frac{1}{4} - \frac{1}{25}$ ∴ Always Comp. & Expand.

C.E.G.

$t \Rightarrow$ Internal Transfer ; $T \Rightarrow$ Cell to cell transfer

$a \Rightarrow$ Addition ; $m \Rightarrow$ Multipiclation

	a_0	m_0	t_0	T_0	Cost**
Linear Interpolation :	4	1	4	6	5.1
Injection :	0	0	1	0	0.0
Single Weight Jacobi + Residuals :	5	1	0	4	8.8
Five Weight Jacobi + Residuals :	11	3	0	12	11.3
Jacobi + Residuals :	9	2	0	8	7.7
Red-Black SOR :	5	1	0	4	8.8
Compression :	0	0	0	$2^m - 2^*$	—
Expansion :	0	0	0	$2^m - 2^*$	—

* Moving Data from 2^m+1 to 2^{m+1} grid or
from $2^{m-1}+1$ to 2^m+1 grid

** $a_0 = 500 \times 10^{-9}$ sec $\rightarrow$ 0.10
$m_0 = 5 \times 10^{-6}$ " $\rightarrow$ 1.00
$t_0 = 100 \times 10^{-9}$ " $\rightarrow$ 0.02
$T_0 = 3 \times 10^{-6}$ " $\rightarrow$ 0.6

C.E.G

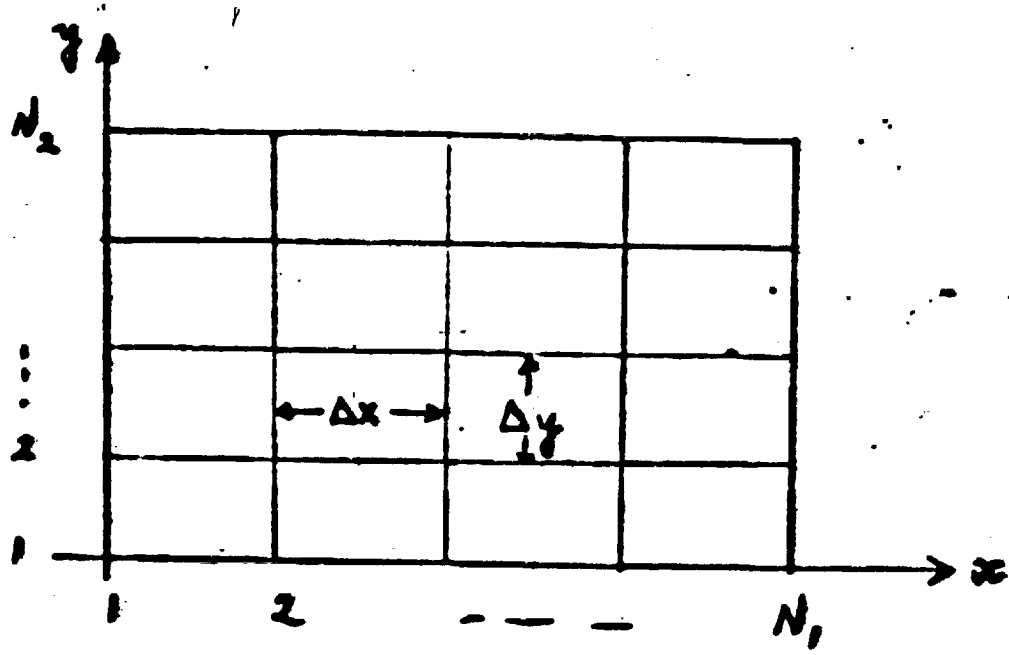
Test Problem

$\nabla^2 P = Q$; Dirchlet Boundary Conditions

$M = 6$ (65×65)

Method	Number Relaxations	Work
Jacobi	4944	44,990
1 Point Jacobi (M.G.)	278	2,684
5 Point Jacobi (M.G.)	201	1,950
Red-Black SOR ($\omega = 1.85$)	264	2,313

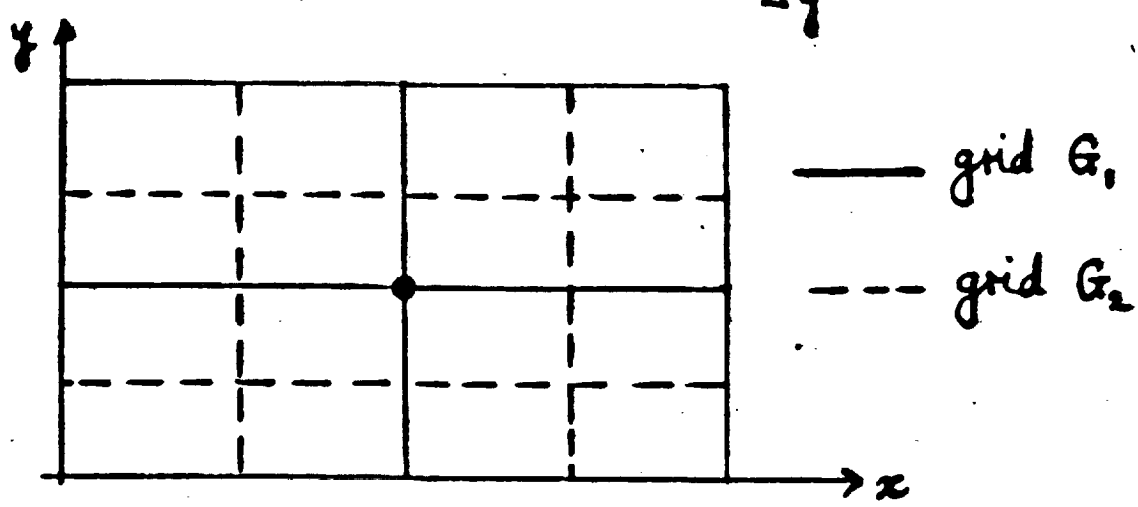
15. MULTI-GRID EXPERIENCE ON THE MINIMAL-SURFACE EQUATIONS



<u>Equation</u> $L\,u(x) = f(x)$ e.g. $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x,y)$

In <u>difference</u> form $L^m U^m = f^m$ on grid G^m

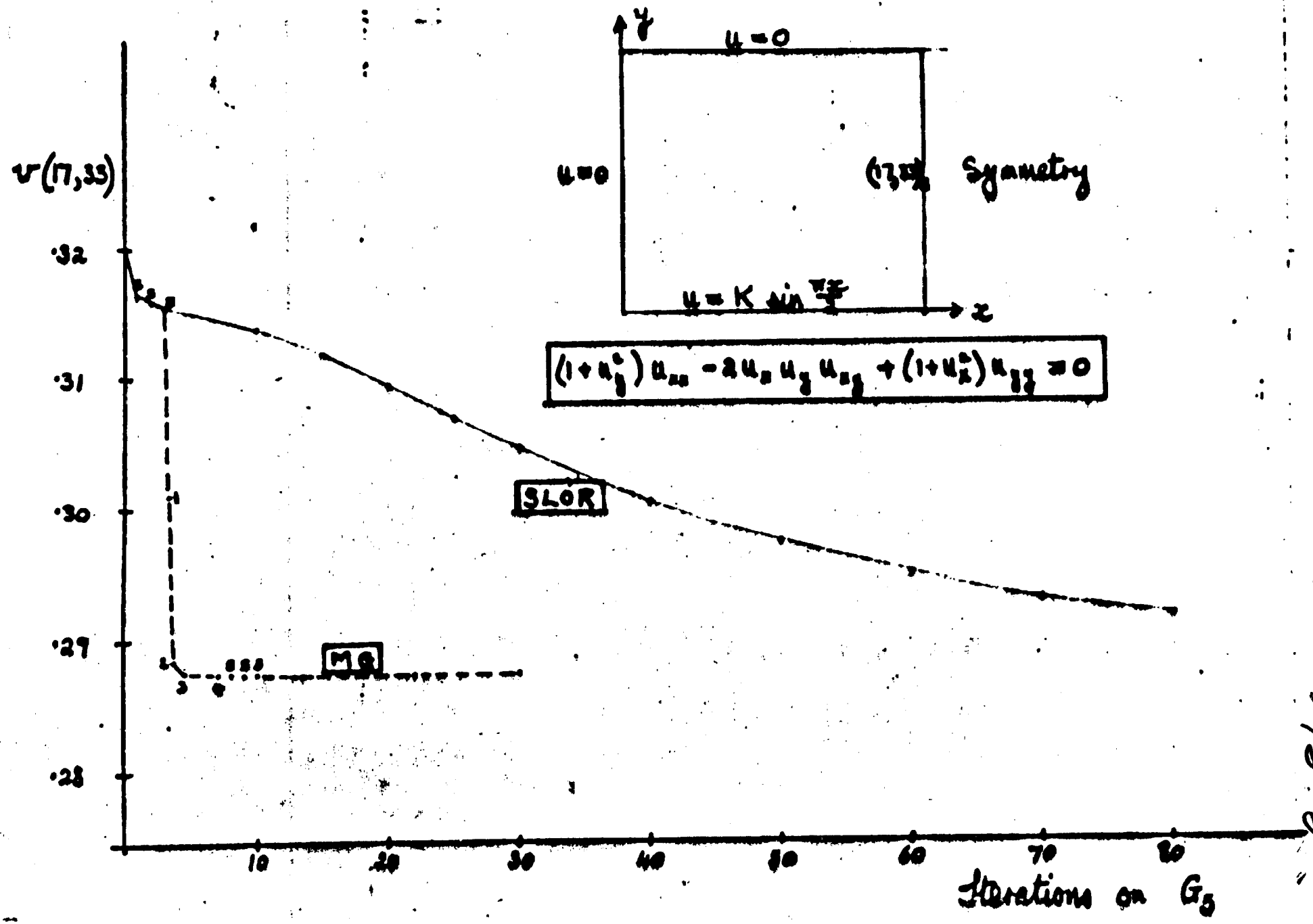
e.g. $\dfrac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{\Delta x^2} + \dfrac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{\Delta y^2} = f(x_i, y_j)$



<u>Interpolate</u> U^m from grid G_m to grid G_k: $I_m^k U^m$

$k < m$: quadratic interpolation for example

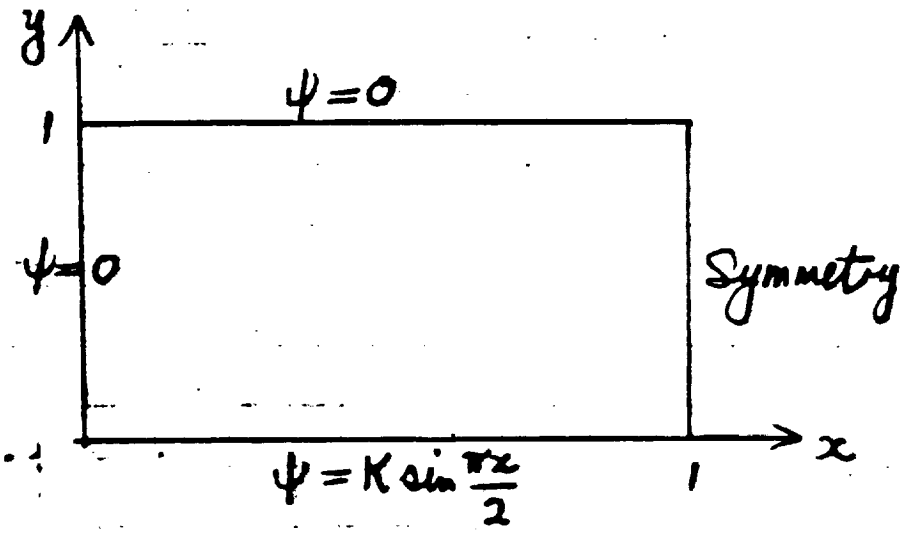
Multigrid Compared to SLOR. Minimal Surface Problem, K = 1.



$$(1+u_y^2)\,u_{xx} - 2u_x u_y u_{xy} + (1+u_x^2)\,u_{yy} = 0$$

15-3

Minimal Surface Problem.



$$(1+\psi_y^2)\,\psi_{xx} - 2\psi_x\psi_y\,\psi_{xy} + (1+\psi_x^2)\,\psi_{yy} = 0$$

Iteration scheme

$$(1+\psi_y^2)\,\frac{\psi_{i+1,j} + \psi^{+}_{i-1,j} - \frac{2}{\omega}\psi^{+}_{ij} - 2\left(1-\frac{1}{\omega}\right)\psi_{ij}}{\Delta x^2}$$

$$-2\psi_x\psi_y\,\frac{\psi_{i+1,j+1} - \psi_{i+1,j-1} + \psi^{+}_{i-1,j-1} - \psi^{+}_{i-1,j+1}}{4\Delta x\,\Delta y}$$

$$+(1+\psi_x^2)\,\frac{\psi^{+}_{i,j+1} + \psi^{+}_{i,j-1} - 2\psi^{+}_{ij}}{\Delta y^2} = f_{ij}$$

where $\psi_x = \dfrac{\psi_{i+1,j} - \psi^{+}_{i-1,j}}{2\Delta x}$

For M.G. use $\omega = 1$.

$$\psi_y = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta y}$$

For nonlinear systems $L^k(I_m^k u^m + V^k)$ may involve complicated linearisations. To avoid this let

$$U^k = I_m^k u^m + V^k$$

then

$$L^k U^k = I_m^k r^m + L^k(I_m^k u^m)$$

$$= F_m^k \quad \text{say} \qquad (4)$$

giving the <u>Full Approximation Scheme</u> (FAS).

On grid m we had

$$L^m U^m = F^m \quad \ldots \quad (1)$$

thus the FAS gives exactly the <u>same form</u> on each grid.

Find the approx solution to (4), u^k say, then

$$V^k = u^k - I_m^k u^m$$

and $$u^m \leftarrow u^m + I_k^m V^k$$

giving the next approximation to u^m.

FLOW DIAGRAM FOR MULTIGRID.

NCYC = 5 VCRIT = ·9

'Solve' system on G_1 : $L' U' = F'$ (iterate NCYC times)

Interpolate (quadratically) U' up to grid M

i.e. $u^M = I_1^M U'$

G_M : iterate until CRIT > VCRIT or NCYC times

where $\text{CRIT} \equiv \sqrt{\dfrac{\left(\sum \Delta u_{ij}^2\right)^n}{\left(\sum \Delta u_{ij}^2\right)^{n-1}}}$

SOLUTION: u^M

G_1 : Let $u^1 = I_M^1 u^M$

Iterate NCYC times : $L' u' = I_M^1 r^M + L'(I_M^1 u^M)$

SOLUTION: u^1

FORM : $V^1 = u^1 - I_M^1 u^M$

G_2 : Let $u^2 = I_M^2 u^M + I_1^2 V^1$

Iterate NCYC times

SOLUTION : u^2 FORM : $V^2 = u^2 - I_M^2 u^M$

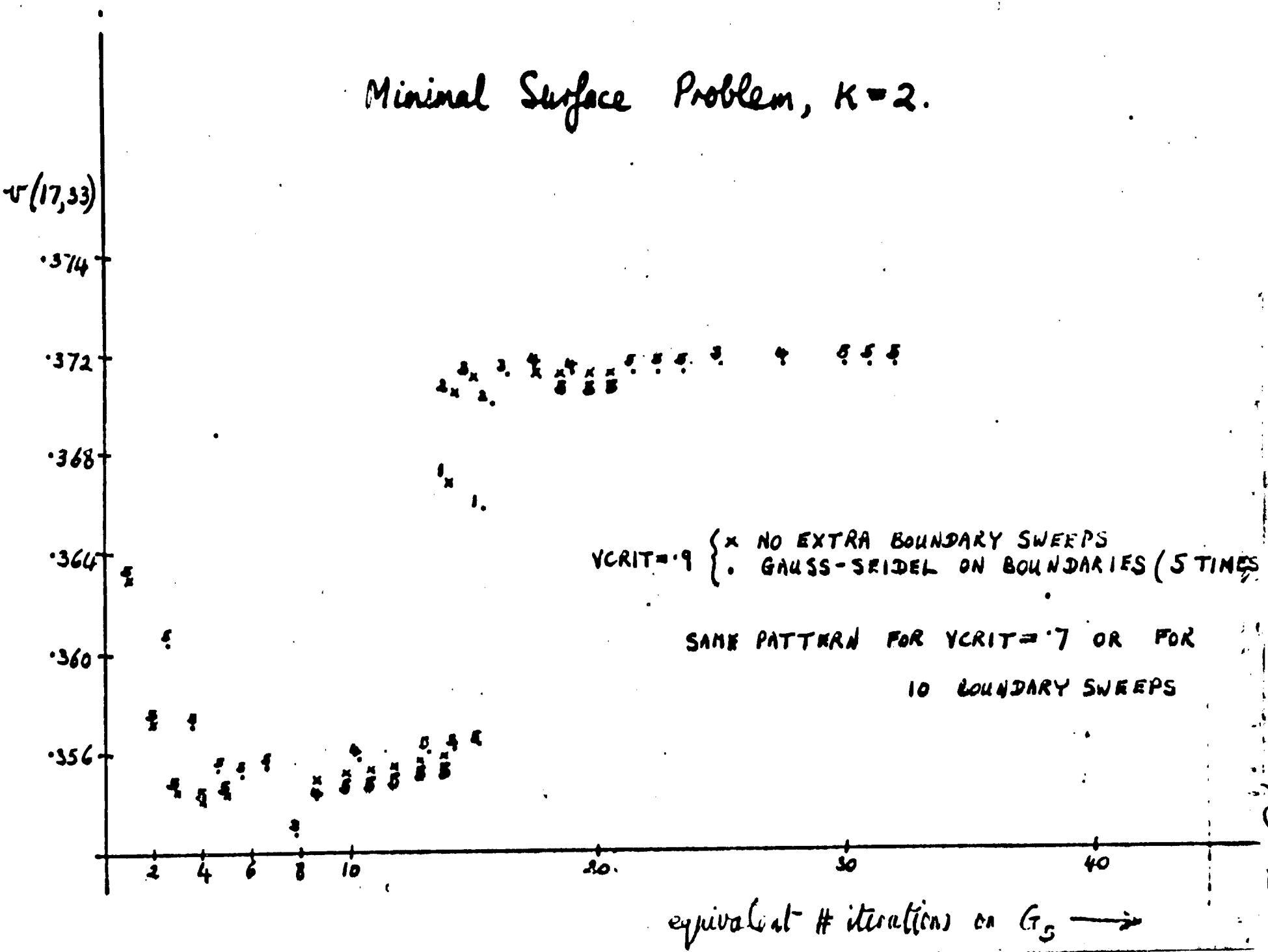
..... ETC.

G_{M-1} : $u^{M-1} = I_M^{M-1} u^M + I_{M-2}^{M-1} V^{M-2}$

Iterate NCYC times

Let $u^M = \cancel{I_M^{M-1}} u^M + I_{M-1}^M V^{M-1}$

Minimal Surface Problem, K = 2.



15-6

WORK UNITS ON GRID G_k. $k < M(=5)$.

	RESID TIME	SWEEP TIME (5)	INTERP I_k^{k+1}	TOTAL	COMPLETE GRID TIME
G_1	·016	·044	·005	·1	·1
G_2	·034	·196	·010	·2	·4
G_3	·102	·329	·029	·5	·8
G_4	·410	1·25	·102	1·8	2·3
TOTAL	·6	1·8	·2	2·6	3·6

Complete grid time includes $I_M^k u^n$, transferring 2-D arrays, adding corrections.

16. MULTI-GRID EXPERIMENTS TO ELLIPTIC PROBLEMS WITH FINITE-ELEMENT FORMULATION

$$J(u) = \iint_D A U_x^2 + B U_y^2 - C U^2 - 2FU - \int_{\Gamma_1} 2 G_1 U$$

$$(A U_x)_x + (B U_y)_y + C U = -F$$

$$0 \le x \le 1, \quad 0 \le y \le 1$$

B.C. $(A U_x, B U_y) \cdot \eta = G_1$ on Γ_1

$U = G_2$ on $\Gamma_2 = \Gamma - \Gamma_1$

1. Poisson Equations - used as standard by which to judge the efficiency of more difficult problems
2. Problems with variable but well behaved coefficients
3. Problems with discontinuous or singular coefficients

ICASE Report Number 77-19 by R.A. Nicolaides

Fixed multi-grid strategies (p, q)

p - number of relaxation sweeps on a given grid before residuals are transferred

q - number of relaxation sweeps on a grid after it has received interpolated corrections

A multi-grid cycle begins on the finest grid G^m and ends with an interpolation from G^{m-1} to G^m. *

<u>Notation</u>:

Represent the error between an initial approximation and the solution to the difference equations at mesh point (α, β) by:

$$E_i(\alpha, \beta) \simeq \sum_{\frac{\pi}{2} \lesssim |\theta| \leq \pi} A_\theta e^{i(\alpha\theta_1 + \beta\theta_2)}$$

And the error after K multi-grid cycles by:

$$E_K(\alpha, \beta) \simeq \sum_{\frac{\pi}{2} \leq |\theta| \leq \pi} \mu^{KR}(\theta) A_\theta e^{i(\alpha\theta_1 + \beta\theta_2)}$$

R - number of relaxation sweeps on each grid

$\mu(\theta)$ - the smoothing rate of the relaxation scheme

$$\rho_{KR} = \left\| \frac{\epsilon_K}{\epsilon_i} \right\|$$

$$= \left[\frac{\sum_\theta \mu^{2KR}(\theta) A_\theta^2}{\sum_\theta A_\theta^2} \right]^{1/2} \quad \frac{\pi}{2} \le |\theta| \le \pi$$

ρ_{KR} - reduction in the initial error at (α, β) after K multigrid cycles

$$\sigma_{KR} = (\rho_{KR})^{1/K}$$

σ_{KR} - reduction in the error per cycle

$$\tau_{KR} = \left(-\mathrm{Log}_{10}(\sigma_{KR})\right)^{-1}$$

τ_{KR} - number of multigrid cycles required to reduce the error by a factor of 10.

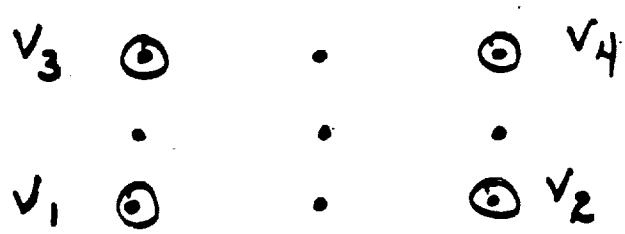
Table 1:

Poisson Equation with Dirichlet boundary data

Discretization, bilinear elements $\varphi_j = a_j x + b_j xy + c_j y + d_j$

Relaxation, 9 point SOR $\omega = 1$

Interpolation and residual weighting where the natural ones arising from the use of bilinear elements



$$EV = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} V$$

$$r_{i,j} + \frac{1}{4}\left(r_{i-1,j-1} + r_{i+1,j-1} + r_{i-1,j+1} + r_{i+1,j+1}\right) + \frac{1}{2}\left(r_{i,j-1} + r_{i-1,j} + r_{i+1,j} + r_{i,j+1}\right)$$

Bilinear Trial Functions
Residual Weighting

	1 Cycle					2 Cycles					3 Cycles				
	Best Strategy	Cost in WU per cycle	$\tilde{\tau}_1$ cycles per 10^{-1} reduc.	Cost in WU per 10^{-1} reduc.	ERROR	Best Strategy	Cost in WU per cycle	$\tilde{\tau}_2$ cycles per 10^{-1} reduc.	Cost in WU per 10^{-1} reduc.	Error	Best Strategy	Cost in WU per cycle	$\tilde{\tau}_3$ cycles per 10^{-1} reduc.	Cost in WU per 10^{-1} reduc.	ERROR
17x17 (225)	(2,0)	3.6	.76	2.78	R	(2,0)	3.7	.80	2.98	R	(2,0)	3.7	.78	2.89	R
	(1,1)	2.8	.83	2.87	C	(1,1)	2.8	.89	2.47	C	(1,1)	2.8	1.01	2.86	C
33x33 (961)	(2,0)	3.6	.78	2.79	R	(2,0)	3.6	.79	2.84	R	(2,0)	3.6	.80	2.90	R
	(1,1)	2.7	.79	2.16	C	(1,1)	2.8	.84	2.33	C	(1,1)	2.9	.97	2.76	C
65x65 (3969)	(2,0)	3.7	.77	2.87	R	(2,0)	3.8	.79	3.01	R	(2,0)	3.8	.82	3.09	R
	(1,1)	3.0	.75	2.26	C	(1,1)	2.9	.80	2.36	C	(1,1)	3.0	.93	2.76	C

ALL costs in workunits: 1 workunit = time for 1 SOR sweep on finest grid

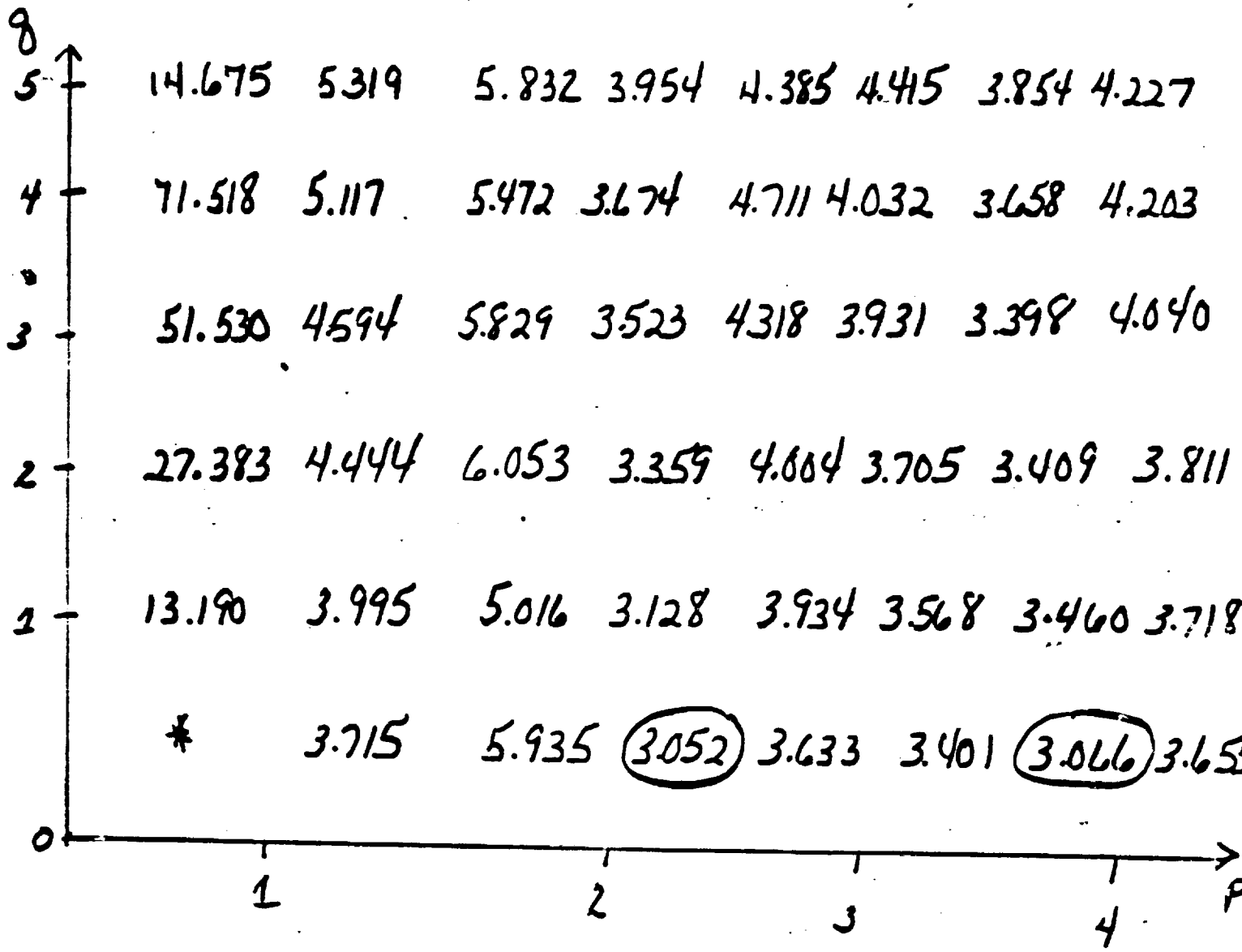
Times used for workunit:

17x17	31×10^{-3} seconds	6400 times derived from relax portion of code
33x33	135×10^{-3} seconds	
65x65	540×10^{-3} seconds	

R = Random initial approximation
C = Constant initial approximation

Random Initial Error

Injected Residuals WU/10^{-1} Error Reduction				Weighted Residuals WU/10^{-1} Error Reduction			
14.675	5.319	5.832	3.954	4.385	4.415	3.854	4.227
71.518	5.117	5.472	3.674	4.711	4.032	3.658	4.203
51.530	4.594	5.829	3.523	4.318	3.931	3.398	4.040
27.383	4.444	6.053	3.359	4.604	3.705	3.409	3.811
13.190	3.995	5.016	3.128	3.934	3.568	3.460	3.718
*	3.715	5.935	3.052	3.633	3.401	3.066	3.65



33 x 33

3 Cycles

Bilinear Elements

Coefficients	Cost/10^{-1} Reduction	Grid Size
$A=B=[1+.5(x^4-y^4)]^2,\ c=0$	3.3	65x65
$A=B=[1+\sin(.5\pi(x+y))]^2,\ c=0$	3.3	65x65
$A=B=[2+\tanh(4(x+y-1))]^2,\ c=0$	3.3	65x65
$A=B=[1+4\lvert x-.5\rvert]^2,\ c=0$	3.3	65x65
$A=B=e^{xy}\sin(\sqrt{x}+y^2),\ c=0$	3.0	33x33
$A=1,\ B=e^{xy}\sin(\sqrt{x}+y^2),\ c=0$	3.4	33x33

$K = 3$ 33 x 33 Grid

ε	Predicted τ_{KR}	Experim. $\tilde{\tau}_{KR}$	Deviation % $\tilde{\tau}_{KR} - \tilde{\tau}_{KR} / \tilde{\tau}_{KR} \times 10^2$
1.0	.933	.807	+ 15.7
.75	.949	.876	+ 8.3
.50	1.020	.991	+ 2.8
.25	1.264	1.273	− .7
10^{-1}	2.030	1.967	+3.0
10^{-2}	4.443	4.024	+10.4
10^{-3}	5.210	4.729	+10.2
10^{-4}	5.304	4.817	+10.8
10^{-5}	5.314	4.758	+11.7

$$\varepsilon U_{xx} + U_{yy} = 0$$

$K = 1$

ε	Predicted τ_{KR}	Experim. $\tilde{\tau}_{KR}$	Deviat. % $\tau_{KR}\,\tilde{\tau}_{KR}/\tilde{\tau}_{KR} \times 10^2$
1.	.732	.778	-5.8
.75	.758	.798	-5.0
.50	.843	.876	-3.7
.25	1.064	1.089	-2.3
10^{-1}	1.503	1.498	+.3
10^{-2}	2.340	2.300	+1.7
10^{-3}	2.503	2.467	+1.4
10^{-4}	2.520	2.486	+1.4
10^{-5}	2.522	2.487	+1.4

$K = 3$

ε	Predicted τ_{KR}	Experim. $\tilde{\tau}_{KR}$	Deviat. % $\tau_{KR}\,\tilde{\tau}_{KR}/\tilde{\tau}_{KR} \times 10^2$
1.	.933	.813	+14.7
.75	.949	.865	+9.8
.50	1.020	.987	+3.4
.25	1.264	1.289	-1.9
10^{-1}	2.020	2.063	-1.1
10^{-2}	4.452	4.299	+3.6
10^{-3}	5.218	5.045	+3.5
10^{-4}	5.312	5.139	+3.4
10^{-5}	5.322	5.149	+3.4

$K = 5$

ε	Predicted τ_{KR}	Experim. $\tilde{\tau}_{KR}$	Deviational % $\tau_{KR}\,\tilde{\tau}_{KR}/\tilde{\tau}_{KR} \times 10^2$
1.	1.033	.856	+20.8
.75	1.043	.909	+14.8
.50	1.104	1.033	+6.8
.25	1.380	1.407	-1.9
10^{-1}	2.419	2.444	-1.0
10^{-2}	6.176	6.920	+4.3
10^{-3}	7.771	7.417	+4.8
10^{-4}	7.984	7.627	+4.7
10^{-5}	8.006	7.649	+4.7

65×65

$$\varepsilon U_{xx} + U_{yy} = 0$$

$K = 3$ $\qquad$ 33×33

$$\varepsilon U_{xx} + U_{yy} = 0$$

ε	Cost / 10^{-1} reduction
1.0	2.5
.75	2.9
.50	4.0
.25	7.1
10^{-1}	16.3
10^{-2}	103.8
10^{-3}	213.0
10^{-4}	237.5
10^{-5}	250.5

$$\mu\left(\tfrac{\pi}{2}, 0\right) = \left[\frac{5\varepsilon^2 - 8\varepsilon + 5}{29\varepsilon^2 + 16\varepsilon + 5}\right]^{1/2} \quad (\text{for } \omega = 1)$$

Discontinuous Coefficients:

$$A = B = \begin{array}{|c|c|} \hline \varepsilon & 1 \\ \hline 1 & \varepsilon \\ \hline \end{array} \qquad \begin{array}{l} 0 \le x \le 1 \\ 0 \le y \le 1 \end{array}$$

$$C = 0$$

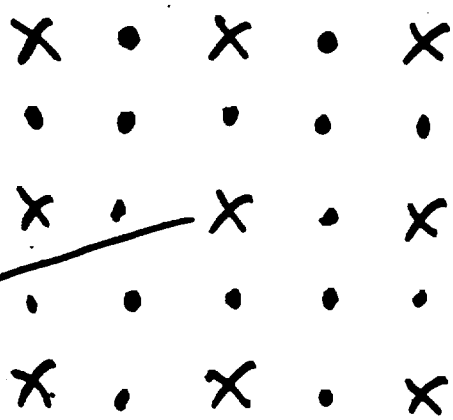
More precisely divide the unit square into $2^K \times 2^K$ smaller but equal squares and set A and B equal to 1 or ε in alternate squares.

K/ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
2	4.7	5.9	6.1	6.0	6.2	5.9
3	4.0	4.9	5.1	5.1	5.1	5.1

K/ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
1	3.6	4.2	4.1	4.1	4.1	4.2

17. RESIDUAL WEIGHTING AND NON-DIRICHLET BOUNDARY CONDITIONS.

Weighting Residuals: $R^h = F^h - L_h u^h$



$$I_h^{2h} R_{ij}^h = \omega_0 R_{ij} + \omega_1 \left(R_{i+1,j} + R_{i,j+1} + R_{i-1,j} + R_{i,j-1} \right) + \omega_2 \left(R_{i+1,j+1} + R_{i+1,j-1} + R_{i-1,j+1} + R_{i-1,j-1} \right)$$

Injection: $\omega_0 = 1$, $\omega_1 = \omega_2 = 0$

Full weighting: $\omega_0 = \frac{1}{4}$, $\omega_1 = \frac{1}{8}$, $\omega_2 = \frac{1}{16}$

Accompanying High-Frequencies

e.g., in

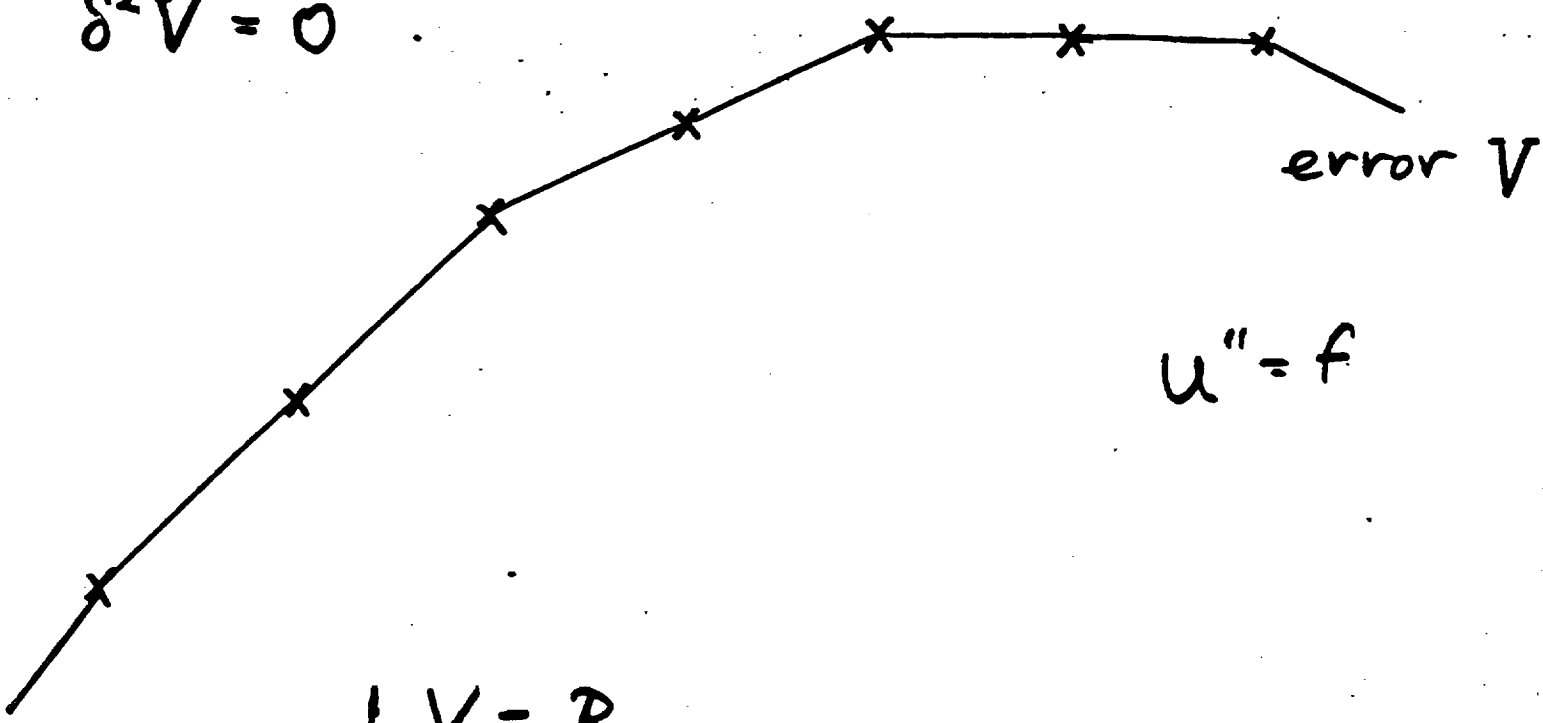
① Highly oscilating difference operators

② Highly Oscilating relaxation schemes (red-black ordering)

③ Near non-Dirichlet boundaries.

at odd points

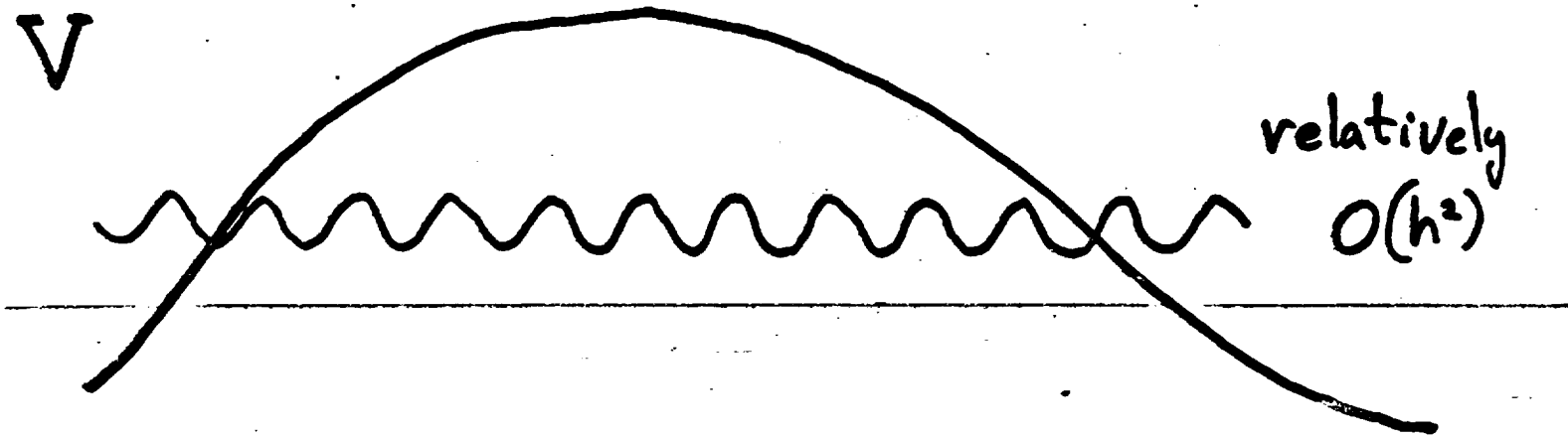
$\delta^2 V = 0$

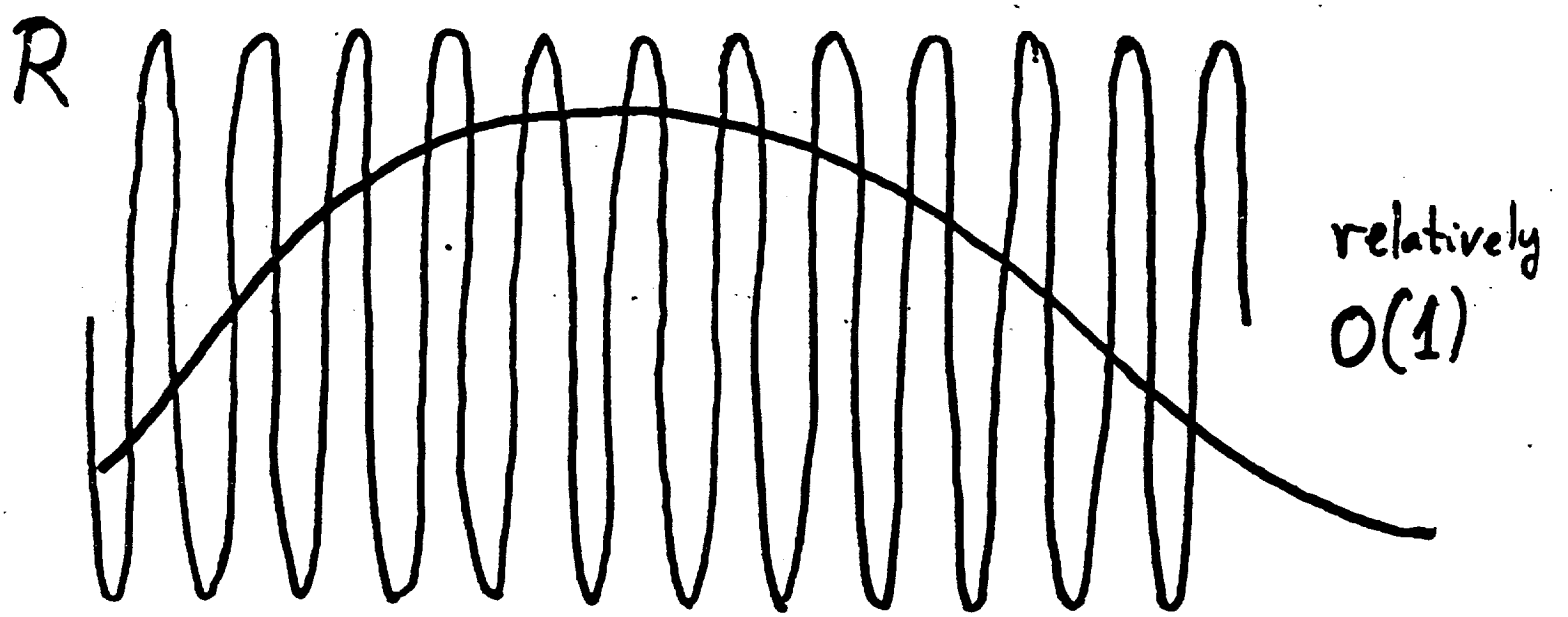


$u'' = f$

$LV = R$

	high-fr.	low-fr.
$V:R$	$O(h^m):1$	$O(1):1$

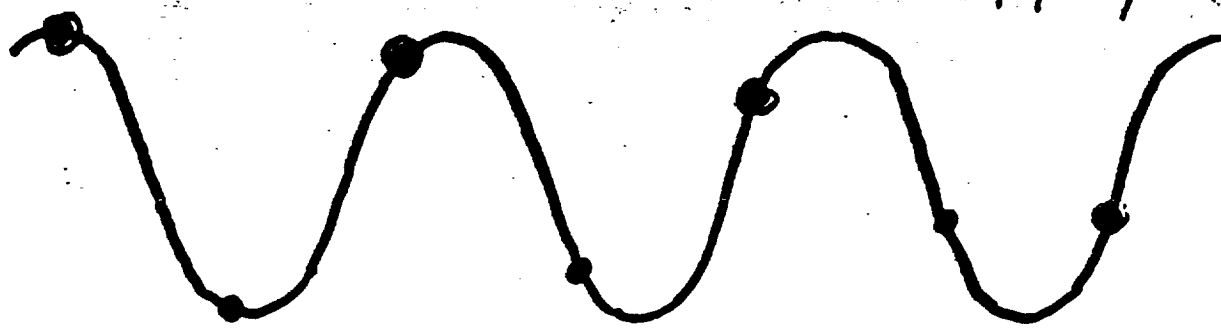




If the residual transfer is $h-f \to h-f$
- the noise does little harm

$$e^{i\theta x/h} \longrightarrow e^{i2\theta x/2h}$$

$\theta \to 2\theta \pmod{2\pi}$. $\qquad \pi \longrightarrow 0$



$\theta \approx \pi \longrightarrow$ low frequency.

These components should be (nearly) eliminated in the residual transfer

$$I_{fine}^{coarse} R_j = \rho_{-1} R_{j-1} + \rho_0 R_j + \rho_1 R_{j+1}$$

$$R_j = e^{i\theta j} \longrightarrow \left(\rho_{-1} e^{-i\theta} + \rho_0 + \rho_1 e^{i\theta}\right) e^{i\theta j}$$

$\longrightarrow 0$ for $\theta = \pi$ if $\rho_{-1} = \frac{1}{4}$, $\rho_0 = \frac{1}{2}$, $\rho_1 = \frac{1}{4}$

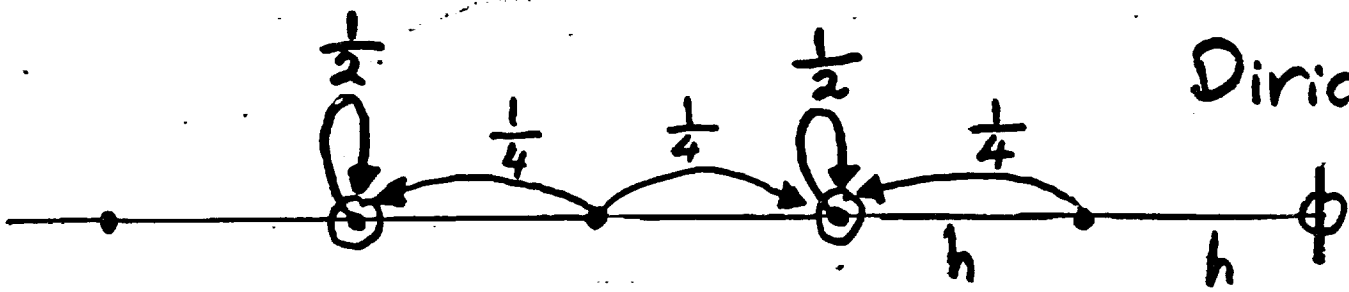
$\longrightarrow \dfrac{1+\cos\theta}{2}$ for general θ, — " —

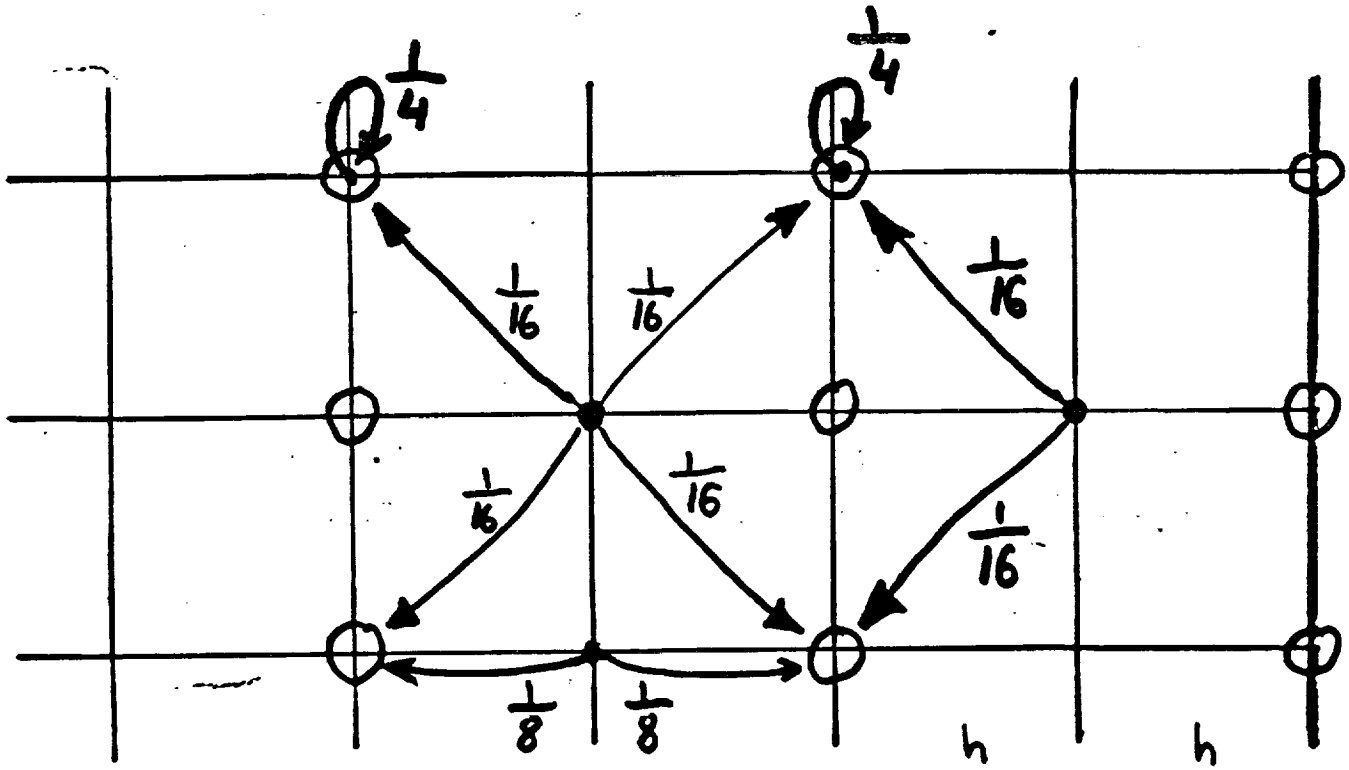
Weights in 2-dimensions

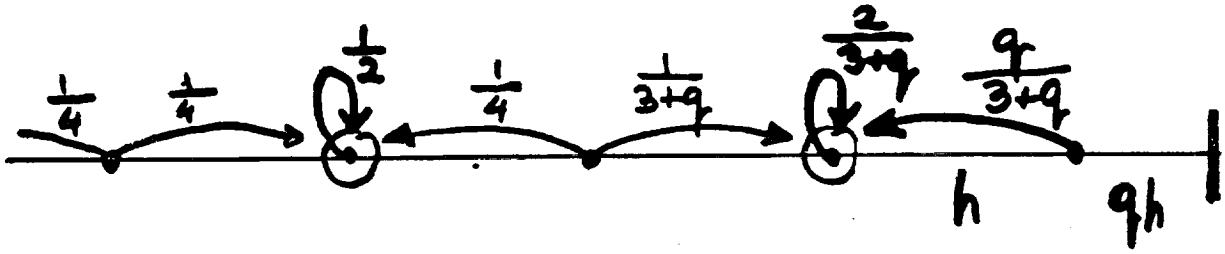
$$\begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \boxed{\frac{1}{4}} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad \square$$

B.C.

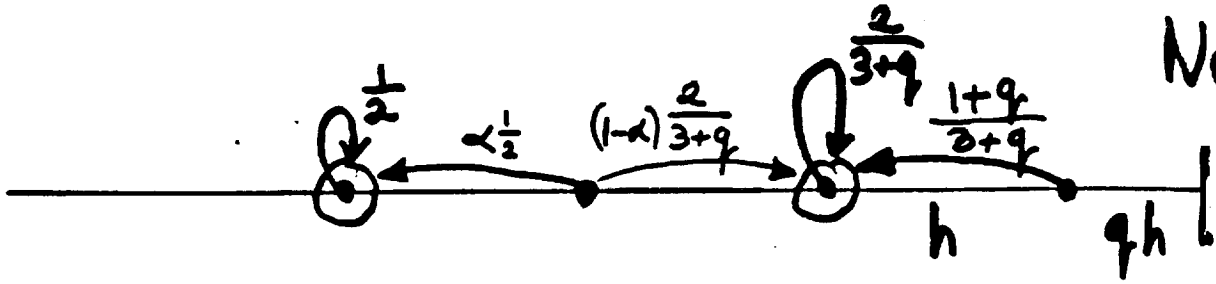
Dirichlet







Neumann



Residual Weighting: Rules

Delta functions: $\delta_j^h(x) = \begin{cases} 1 & : x = jh, \\ 0 & \end{cases}$ $\quad j = (j_1, \ldots, j_d)$

Adjoint weights: $I_h^H \delta_j^h = \sum w_{ij} \delta_i^H$

Volume represented by the j-th equation: a_j^h

Distance i from boundary: d_i

Interior Rule: $\sum_i w_{ij} = \left(\frac{h}{H}\right)^d = \frac{a_j^h}{a_i^H}$

Near Dirichlet boundary: (Green function $\sim d_x$)

$$\sum_i w_{ij} d_i a_i^H = d_j a_j^h$$

Near Neumann or interior boundary

$$\sum_i w_{ij} a_i^H = a_j^h$$

Cycle C for extreme $L_h = a\partial_{xx} + \partial_{yy}$

a: odd points ($\supset G^{2h}$)	a: even points	y-line relax.	weighting: res.	weighting: a	$\mathring{\mu}$ $\left(\begin{smallmatrix}\delta=.3\\ \eta=.6\end{smallmatrix}\right)$	
			i	i	.94	$\approx \mu_{relax}$
.01	1	F	i	w	.95	$\approx \mu_{relax}$
			w	i	3	fast div.
			w	w	.66	
1	.01	F	i	i	1+	slow div.
			i	w	1+	slow div.
			w	i	.74	
			w	w	.64	
G^{2h}	$G^h - G^{2h}$					
1	.3	F			.51	
1	.1	F			.57	
1	.01	F	w	w	div	?
.1	.01	F			.64	?
1	.01	F*			.73	?

w = weighting $\frac{1}{16}\begin{pmatrix}1&2&1\\2&4&2\\1&2&1\end{pmatrix}$. i = injection. F = forward. F* = coarse columns before fine

18. DEBUGGING TECHNIQUES.

1. Standard Output Checks

1.1. $r_F^{2h} < r_0^h$ but should not be much different.

<u>Error sources:</u> RESCAL scale; RELAX residual-norm not in differential scale; RESCAL not compatible with RELAX (check 3.1)

h	r_{-1}^h	..
h	r_0^h	...
$2h$	r_F^{2h}	
	CGC	
$2h$	r_L^{2h}	...
h	r_1^h	...
h	r_2^h	...

1.2. Should be: $r_1^h \overset{?}{<} r_0^h$, $r_2^h \ll r_0^h$.

<u>Error sources:</u> RELAX - RESCAL incompatible (3.1); RESCAL and/or INTERP bugged or too low (3.3 - 3.4); Insufficient smoothing before CGC ($\Rightarrow r_2^h \not\ll r_0^h$; 3.1-3.2).

1.3. If above bad behavior appears stronger for coarse h - it may be a boundary problem (check 2.1, 2.3; 3.1, 3.2; boundary interpolation; boundary residuals are transferred in differential scale? Try res. weighting near boundary and/or other boundary relaxation).

1.4. $r_0^h > \eta r_{-1}^h$ for all equations (in a system), for all regions (3.2). If not, introduce partial relaxation? Or change relax. scheme

1.4.1 η test made with comparable errors? (from relaxing in same directions?)

1.5. $r_L^{2h} < \delta_h r_0^h$

1.6. ν = number of h-sweeps per ~~cycle~~ cycle. If $\bar{\mu}^\nu \lesssim .1$ efficiency deteriorates.
<u>Source for large ν:</u> too many directions

(use fixed algorithms, distributed relax.);
or 1.4.1; wrong η.

1.7. Check efficiency per cycle (4., 4.4)
per h-sweeps (4.1), etc.

1.8. FAS output should be identical (upto
round-off) with CS, for linear problems,
line by line.
Error sources: PUTU not compatible with
SUBTRCT; COARSRES not compatible
with RELAX (2.4).

2. Simple tests.

2.1. Very small δ_h, to eliminate the possible source of not solving accurately on 2h. & Check efficiency per h-sweeps (4.1)

2.2. Experimental $\bar{\sigma}$ by ~~set~~ shutting off relaxation (4.5).

2.3 M (# levels) = 2. check precisely trouble specific to coarse levels (1.3).

2.4. Insert 0 instead of CGC correction. Should get μ relax.

2.5. FAS: Insert 0 instead of residuals transfer. Should get $r_F^{2h} = 0$. If not: COARSRES incompatible with RELAX (2h)

2.6. Various δ, various η. Check efficiencies per cycle, per h-sweeps.

3. Additional Printouts

3.1. Printout $\|\text{RESCAL residuals}\|_{L_2}$ and $\|\text{static RELAX residuals}\|_{L_2}$.

The two should be identical.

3.2. Printout $\|\text{boundary residuals}\|$. (Check the scale).

3.3. Printout table of RESCAL residuals, to locate local troubles, (non-smoothness. correct by residual weighting and/or (local) smoothing improv

3.4. Do 3.3 before and after CGC, for h and 2h. Check wher m.g. logic breaks down.

3.5. Comparison, line by line, of a special simple case to a former special program.

4. Comparison to theory

4.1. Small δ_h, efficiency per h-sweep should be j
or:

4.2. Comparison to $\left\{\text{weighted average of } \mu(\theta)^{\nu}\right\}^{\frac{1}{\nu}}$
or:

4.3. One-mode comparison to $\bar{\mu}$.

4.4. (One-mode) comparison, line by line, to full m.g. analysis. For small δ_h there should be (full) agreement, except for developmen from boundaries.

4.5. Theoretical $\bar{\sigma}$ (2.2). If $\bar{\sigma} > 1$, correct res. weighting and/or interpolation.

4.6. Boundary mode analysis?

ICASE WORKSHOP ON MULTI-GRID METHODS

JUNE 19 - 23, 1978

NAME LIST

Raymond Alcouffe
MS 269
Los Alamos Scientific Lab
Los Alamos, NM 87545

Alvin Bayliss
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

Robert M. Bennett
MS 340
NASA Langley Research Center
Hampton, VA 23665

Marsha Berger
Serra House
Computer Science Department
Stanford University
Palo Alto, CA 94301

Dana Brewer
George Washington University
MS 169
NASA Langley Research Center
Hampton, VA 23665

Dennis Bushnell
MS 163
NASA Langley Research Center
Hampton, VA 23665

Fred Carlson
MS 473
NASA Langley Research Center
Hampton, VA 23665

Aivars Celmins
BRL - BMD, Building 394
Aberdeen Proving Ground, MD 21005

Jagdish Chandra
Director, Mathematics Division
U.S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709

Hai-Chow Chen
29 Newport Key
Bellevue, WA 98006

Ivan Clark
MS 473
NASA Langley Research Center
Hampton, VA 23665

Joel Cohen
Department of Mathematics
University of Denver
Denver, CO 80208

K. R. Czarnecki
TAB
MS 360
NASA Langley Research Center
Hampton, VA 23665

Wolfgang Dahmen
Institut Für Angewandte Mathematik
University of Bonn
Wegelerstraße 6
53 Bonn , WEST GERMANY

Nathan Dinar
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

Craig Douglas
Department of Computer Science
Yale University
10 Hillhouse Avenue
New Haven, CT 06520

J. P. Drummond
MS 168
NASA Langley Research Center
Hampton, VA 23665

Douglas L. Dwoyer
MS 360
NASA Langley Research Center
Hampton, VA 23665

Jim Epperson
Department of Mathematics
Carnegie-Mellon University
Pittsburgh, PA 15213

F. Farassat
George Washington University
MS 461
NASA Langley Research Center
Hampton, VA 23665

Donald D. Fisher
School of Math Sciences
Oklahoma State University
Stillwater, OK 74074

Hartmut Foerster
GMD
Schloss Birlinghoven
Postfach 1204
D-5205 St. Augustin 1
WEST GERMANY

John Gary
Computer Science Department
University of Colorado
Boulder, CO 80309

Gary L. Giles
MS 243
NASA Langley Research Center
Hampton, VA 23665

Peter A. Gnoffo
MS 366
NASA Langley Research Center
Hampton, VA 23665

Randy Graves
MS 366
NASA Langley Research Center
Hampton, VA 23665

Anne Greenbaum
1062 Catalina Dr., #58
Livermore, CA 94550

Chester E. Grosch
Institute of Oceanography
Old Dominion University
Norfolk, VA 23508

Fred Gustavson
IBM
T. J. Watson Research Center
Yorktown Heights, NY 10598

Mohamed M. Hafez
Flow Research Company
21414 68th Avenue South
Kent, WA 98031

Harris Hamilton
MS 366
NASA Langley Research Center
Hampton, VA 23665

R. J. Hayduk
MS 243
NASA Langley Research Center
Hampton, VA 23665

Forrester Johnson
3826 S. W. 313th
Federal Way, WA 98003

D. J. Jones
26 Tiverton Dr.
Ottawa
CANADA

Gershon Kedem
Computer Sciences Department
Math Sciences Building
University of Rochester
Rochester, NY 14627

E. B. Klunker
MS 360
NASA Langley Research Center
Hampton, VA 23665

Jay Lambiotte
MS 125
NASA Langley Research Center
Hampton, VA 23665

Hsi-Nan Lee
Atmospheric Science Division
Building 51
Brookhaven National Laboratory
Upton, NY 11973

Chen-Huei Liu
MS 460
NASA Langley Research Center
Hampton, VA 23665

Nan-suey Liu
George Washington University
MS 460
NASA Langley Research Center
Hampton, VA 23665

Wayne Mastin
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

Stephen F. McCormick
Department of Mathematics
Colorado State University
Fort Collins, CO 80523

David S. McDougal
MS 325
NASA Langley Research Center
Hampton, VA 23665

Gunter H. Meyer
School of Mathematics
Georgia Tech
Atlanta, GA 30332

Ahmed K. Noor
George Washington University
MS 246
NASA Langley Research Center
Hampton, VA 23665

Youn Hwan Oh
103 Sleepy Hollow Lane
Yorktown, VA 23692

Joseph Oliger
Serra House - Serra St.
Stanford, CA 94305

Seymour V. Parter
Mathematics Department
Van Uleck Hall
University of Wisconsin
Madison, Wis. 53706

Joe Pasciak
Brookhaven National Laboratory
Building 515
Upton, Long Island, NY 11973

T. Craig Poling
1645 Ardmore Blvd., Apt. 8
Forest Hills, PA 15221

John Radbill
JPL 125/128
4800 Oak Grove Dr.
Pasadena, CA 91103

Robert Reklis
Launch and Flight Division
Building 120
Aberdeen Proving Ground, MD 21005

J. C. Robinson
MS 395
NASA Langley Research Center
Hampton, VA 23665

M. E. Rose
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

D. H. Rudy
MS 163
NASA Langley Research Center
Hampton, VA 23665

John W. Ruge
200 E. Laurel
Ft. Collins, CO 80521

Bob Smith
MS 125
NASA Langley Research Center
Hampton, VA 23665

Jerry South
MS 360
NASA Langley Research Center
Hampton, VA 23665

Brooke Stephens
Naval Surface Weapons Center
Code R44
Building 427, Room 530
White Oak, MD

John Strikwerda
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

Klaus Stueben
GMD, Schloss Birlinghoven
Postfach 1204
D-5205 St. Augustin 1
WEST GERMANY

Charles Swanson
MS 280
NASA Langley Research Center
Hampton, VA 23665

Frank C. Thames
Vought Corporation
MS 360
NASA Langley Research Center
Hampton, VA 23665

J. W. Thomas
Department of Math
Colorado State University
Ft. Collins, CO 80521

Sylvester Thompson
Babcock & Wilcox Company
P. O. Box 1260
Lynchburg, VA 24505

J. S. Tripp
MS 238
NASA Langley Research Center
Hampton, VA 23665

John Van Rosendale
31 F Digital Computer Lab
University of Illinois
Urbana, IL 61801

Veer N. Vatsa
Old Dominion University
MS 163
NASA Langley Research Center
Hampton, VA 23665

Robert G. Voigt
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 23665

H. H. Wang
1530 Page Mill Road
Palo Alto, CA 94303

Willie R. Watson
MS 460
NASA Langley Research Center
Hampton, VA 23665

Deene J. Weidman
MS 190
NASA Langley Research Center
Hampton, VA 23665

Jim Weilmuenster
MS 366
NASA Langley Research Center
Hampton, VA 23665

Carl Weiman
Department of Math and
Computer Science
Old Dominion University
Norfolk, VA 23508

P. R. Wohl
Department of Math and
Computer Science
Old Dominion University
Norfolk, VA 23508

Stephen F. Wornom
MS 360
NASA Langley Research Center
Hampton, VA 23665

Eleanor C. Wynne
MS 340
NASA Langley Research Center
Hampton, VA 23665

E. Carson Yates, Jr.
MS 340
NASA Langley Research Center
Hampton, VA 23665

Warren Young
MS 340
NASA Langley Research Center
Hampton, VA 23665

Csaba K. Zoltani
BMD
Ballistic Research Lab
Aberdeen Proving Ground, MD 21005

Late Registrants

Lois Mansfield
Department of Mathematics
North Carolina State University
Raleigh, NC 27606