SOCRATES Teaching Staff Mobility Program
2000 - 2001

Technical University of Budapest
1782
Department of Fluid Mechanics

DMA-URLS

Finite Element Method
for Turbomachinery Flows

Fluid model and solution strategy

Alessandro Corsini

Dipartimento di Meccanica e Aeronautica, University of Rome "La Sapienza"

***BUDAPEST University of Technology and Economics* - 28 November 2000**

MODEL EQUATIONS AND FLUID DYNAMIC CLOSURE

- MODEL EQUATIONS FOR TURBULENT, INCOMPRESSIBLE AND ROTATING FLOWS (*PRIMITIVE VARIABLES*)

- FLUID DYNAMIC CLOSURE ⟶ EDDY VISCOSITY MODELLING EVM

  REYNOLDS DECOMPOSITION

  TURBULENT STRESS TENSOR CONSTITUTIVE RELATION $\overline{u'_i u'_j} = F_{ij}(\bar{u}_{i,j}, k, e)$

  F^{st}_{ij} linear and isotropic Newtonian like relation (Jones and Launder, 1972)

  F^{nl}_{ij} non-linear 3rd order polynomial relation (Craft et al., 1993)

- DYNAMIC RESPONSE OF *INCOMPRESSIBLE* TURBULENT FLUIDS BY BV PROBLEM

  3-D averaged navier stokes equations
  turbulent kinetic energy k, dissipation ε

TURBULENCE CLOSURE

standard EVM *(Jones and Launder, 1972)*	*non-linear EVM* *(Craft et al., 1993)*	*EASM*
tensor representation with **gradient approximation**	tensor representation with ***kinematic* polynomial expansion up to 3rd order terms**	tensor representation with **polynomial expansion up to 3rd order terms**
Newtonian-like stress-strain relationship up to 1st order term	non-Newtonian stress-strain relationship	explicit algebraic stress components equations
dependence from strain invariant	dependence from strain and vorticity invariants	dependence from strain and vorticity invariants
	empirical expansion coefficients *calibration* with experiments and numerical data *physical consistency* conditions	expansion coefficients derived consistently **to full differential RSE**
isotropy of stress normal components	**anisotropy** of stress components	**anisotropy** of stress components
	built-in sensitivity to curvature and rotation effects	built-in sensitivity to curvature and rotation effects

BOUNDARY CONDITIONS

- BOUNDARY CONDITIONS
  - INLET: DIRICHELET CONDITIONS VELOCITY, k AND ε
  - OUTLET: NEUMANN CONDITIONS HOMOGENEUS (k AND ε) AND NON HOMOGENEUS p
  - SOLID BOUNDS: WALL FUNCTION (WALL SHEAR STRESS, k AND ε) on stationary and moving walls
  - PERIODIC BOUNDS: DEGREES OF FREEDOM EQUALITY

ITERATIVE SOLUTION STRATEGY

- *NON-LINEAR* ALGEBRAIC SYSTEM $K(\Theta)\,\Theta = F$
  - LINEARIZATION BASED ON SUCCESSIVE SUBSTITUTION PROCEDURE (DOF RELAXATION)
  - LINEARIZED SYSTEM SOLUTION *VIA* A *SEMI - ITERATIVE* SOLVER (GMRESR)

- SEMI-ITERATIVE SOLUTION ALGORITHM
  - K STIFFNESS MATRIX COMPACT STORING - 2 ADDRESSING VECTORS STRUCTURE

- *GMRES* SOLVER- RESIDUAL PROJECTION (r_o) OVER A KRYLOV SPACE
  $\left[r_o, K\, r_o, K^2 r_o, \ldots, K^m r_o\right]$ ® *m* SOLVER VECTORS FOR EACH OUTER ITERATION GMRESR 10(10)

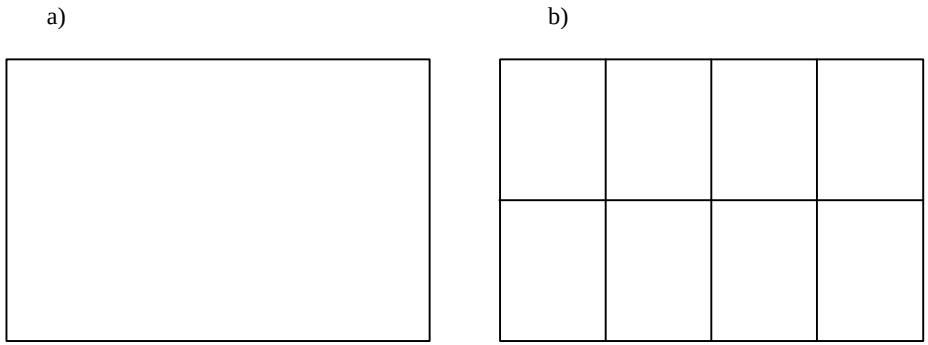
- PRECONDITIONING TECHNIQUE AS CONVERGENCE ACCELERATOR (GAUSS – SIEDEL LIKE)

PARALLEL SOLUTION METHODOLOGY

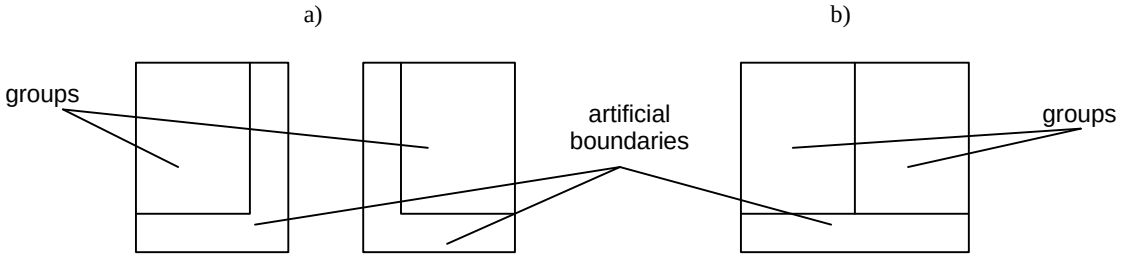
- OVERLAPPING ADDITIVE SCHWARZ METHOD
  - ONE LEVEL TECHNIQUE
  - FULLY PARALLEL
  - EXPLICIT TREATMENT OF THE EXPLICIT BOUNDARY CONDITIONS
  - COMMUNICATION RESTRICTED TO ARTIFICIAL BOUNDARY CONDITIONS UPDATING
  - RELAXATION OBTAINED *VIA* AN INEXACT LOCAL SOLUTION
- DOMAIN DECOMPOSITION ALGORITHM
  - LOAD BALANCING
  - DATA LOCALITY THROUGH MINIMIZATION OF OVERLAPPING REGIONS
- COMMUNICATION STRATEGY
  - MPI LIBRARIES
  - PORTABILITY (IBM SP2 AND CRAY T3E)

DOMAIN DECOMPOSITION (1)

- ONE LEVEL INEXACT EXPLICIT NON-LINEAR OVERLAPPING SCHWARZ METHOD

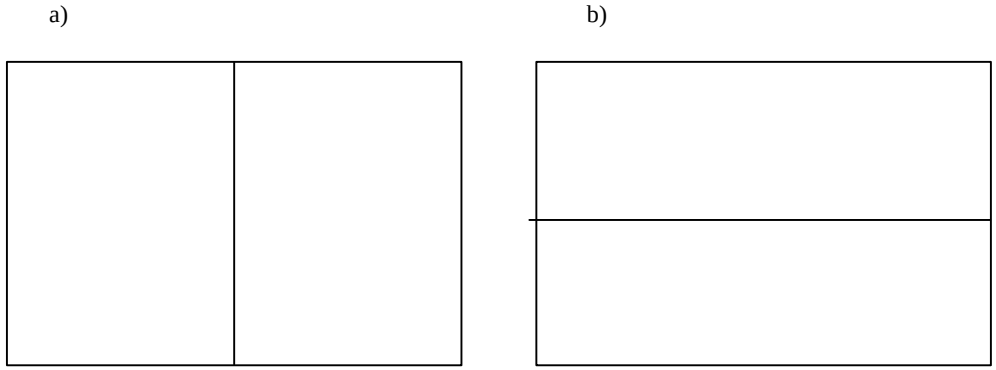


Domain Decomposition of the whole geometry (a)
in 8 groups (b)

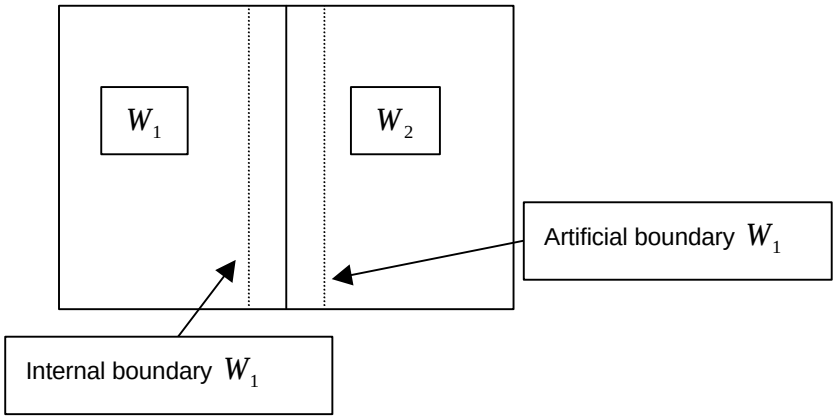


Assembling of non-contiguous (a) and contiguous (b) groups

DOMAIN DECOMPOSITION (2)



Geometry Domain Decomposition in two domains
(a) vertical cut - b) lengthwise cut



Domain Decomposition

EXPLICIT TREATMENT OF ARTIFICIAL BOUNDARY CONDITIONS – SCHWARZ METHOD

DOMAIN DECOMPOSITION (3)

- APPROXIMATE LOCAL SOLUTION OF OSEEN PROBLEM

  - GMResR SOLVER
  - GAUSS-SIEDEL OR JACOBI PRECONDITIONING

- VARIATIONAL APPROACH OF ADDITIVE SCHWARZ METHOD

find $V_i^{h,k} \in H_0^{1h}(W_i)$,

$$c\left(\bar{u}_i^{h,k}; V_i^{h,k}, w^h\right) + s\left(V_i^{h,k}, w^h\right) + P_i^k\left(p^h\right) = \left(B_i^h, w^h\right) + \left(j_{iN}, w^h_{|\partial W_N}\right)_{\partial W_N \cap \partial W_i} - F_i^k\left(w^h\right)_{\partial W_{iD} \cup_j G_{ij}}$$

$$\forall w_q^h \in V_{qi}^h,\ \forall w_l^h \in V_{li}^h$$

$$U^{h,k+1} = U^{h,k} + \sum_i \hat{V}_i^{h,k}$$

PARALLEL ALGORITHM

- SIMD (SINGLE INSTRUCTION, MULTIPLE DATA) PARADIGM
- GLOBAL SOLUTION DUE TO THE COUPLING OF DOMAINS SOLUTIONS
- LOCAL SOLUTION

1) Governing equations linearisation in the local domains

2) Definition of the artificial boundary conditions set

3) Restart cycle

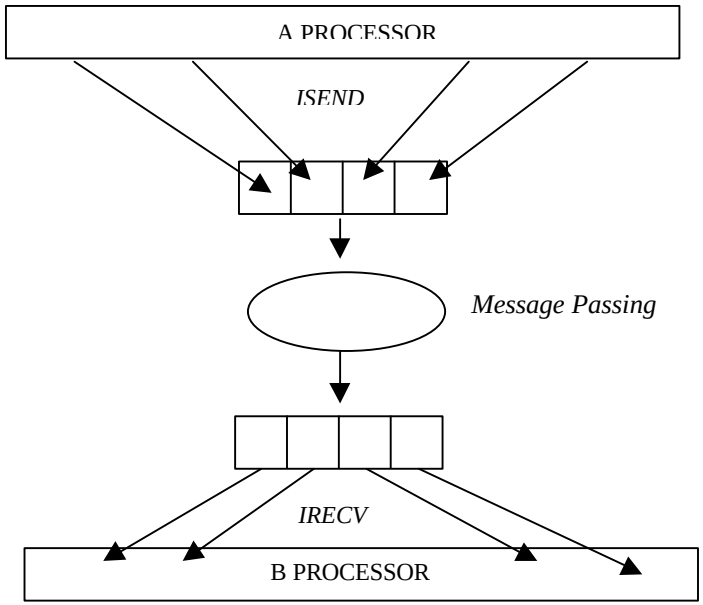
4) GMRes cycle

if local domain convergence is not reached goto 3)

if convergence on the interface coupling is not reached goto 2)

if Oseen problem convergence is not reached goto 1) else stop.

- COMMUNICATION PROCEDURE



Communication between processors A and B

PARALLEL ALGORITHM (2)

for each sub-domain W_i (i=1,n)

- *assemble several geometrically contiguous groups into sub-domains balancing the computational load among processors and minimizing the interface region extension*
- *build-up of the overlapping region and set the artificial boundary conditions*
- *create sub-domain data structures for communication*
- *renumbering of nodes within the sub-domains*

Domain Decomposition algorithm

for each processor (i, i=1,n)

- *read own data-base*
- *for each step of solution procedure until the convergence is attained*
  - *compute local (sub-domain) solution*
  - *performe communication between processors (sub-domains) and update the sets of artificial boundary conditions*

SIMD parallel solution algorithm

PARALLEL ALGORITHM (3)

for each processor (i, i=1,n)

- *extract data from variable array (gather vector ISEND) and put them on sending array DSEND*
- *for each processor j (j=1,n, j ¹ i)*
  - *check for the existence of common interfaces using IPOINT, establishing position and length of data to be transferred*
  - *send data to the processor*
  - *wait for reply*
  - *put sent data in the correct position using IPOINT in the receiving array DRECV*
- *allocate DRECV data in the corresponding position of the array of the variable using the scatter vector IRECV*
- *performe the synchronization of the processor before restarting the computation*

Communication strategy

PARALLEL SUB-DOMAIN SOLUTION

for each sub-domain W_i (i=1,n)

1) linearization of sub-domain problem

2) updating of artificial boundary conditions on G_{ij}

3) G-S Preconditioned GMResR restart loop (compact skyline assembling of stiffness matrix)

If interface convergence is achieved goto 1) else goto 2)

If non-linear problem convergence is achieved stop else goto 1)

Nested cycles approach

for each sub-domain W_i (i=1,n)

1) Linearization of sub-domain problem and updating of artificial boundary conditions on G_{ij}

2) G-S Preconditioned GMResR restart loop (compact skyline assembling of stiffness matrix)

If non-linear problem convergence is achieved stop else goto 1)

Condensed cycles approach