PARALLEL COMPUTATION OF MULTI-PASSAGE CASCADE FLOWS ON PARTITIONED OVERSET GRIDS

Ismail H. Tuncer*
Middle East Technical University
06531 Ankara

ABSTRACT

Multi-passage cascade flows are computed on overset grids in a parallel computing environment. PVM message passing library routines are utilized in a PC cluster running Linux operating system. A courseseparallel algorithm is implemented by partitioning overset grids and solving the governing Navier-Stokes equations on each subgrid on a separate processor. The computation of a flow through a four-passage cascade is speeded up about 3.9 times in comparison to serial computations, and a 98% processor efficiency is attained.

INTRODUCTION

In Computational Fluid Dynamics applications, multiple workstations on a local network may be combined into a parallel computer, and this virtual parallel computer may be used to solve a single flow problem. Such a parallel processing environment is an attractive alternative to a supercomputer or to a dedicated computer with multi-processors. It also improves the productivity of otherwise idle workstations. On the other hand, computational domains discretized with overset grids provide an easily exploited parallel solution algorithm by computing the solution on each grid in parallel on a separate workstation, and by exchanging the boundary information on grid interfaces.

The current trends in gas turbine design towards higher aerodynamic blade loading and slender blades demand an accurate and detailed study of aeroelastic behavior of compressor and turbine blades. Considerable experimental and computational efforts are currently being made to predict unsteady cascade flows and blade flutter in turbomachinery. It is, therefore, of great interest to develop solution methods to compute dynamic response of blades with an acceptable level of accuracy and speed.

Unsteady cascade flows as the blades undergo in and out-of-phase vibrations are currently being computed by solving the Euler or Navier-Stokes equations. In

Fig. 1 Overset grid system for a five-passage compressor cascade

*Assoc. Prof., Dept. of Aeronautical Engineering,
Tel: (312) 210-4289, Fax: (312) 210-1272,
E-mail: tuncer@ae.metu.edu.tr
the computation of out-of-phase cascade flows, the periodic boundary conditions are implemented either by discretizing the multi-passage domain based on the Inter-Blade Phase Angle (IBPA) or by imposing temporal periodicity in a single passage domain. The latter method is known as direct store[1]. Although the direct store method reduces the computational domain to a single blade passage, it, in fact, linearizes the periodic boundary conditions in time. In addition, a periodic convergence may take a large number of periods to be computed on a single passage domain, and a large storage may be required to store the periodic boundary information in time[2].

In this study, a Navier-Stokes solver is modified to compute multi-passage cascade flows on overset grids in a PC based parallel processing environment. The multi-passage flow domain is discretized with a rectangular background grid and viscous grids around each blade which are overset onto the background grid. The background grid covers the whole multi-passage domain. The main advantage of this approach lies in its versatility to resolve the multi-passage computational domain with high quality, mostly orthogonal subgrids, and in the application of simple periodic boundary conditions. Furthermore, overset grids readily lend themselves to parallel processing. Figure 1 shows a discretized four-passage flow domain over the turbine cascade, STC No. 4, which has been investigated experimentally and numerically, and extensive data are available[3,4]. In an earlier study[5], parallelization of the code was implemented for each overset grid, where the load imbalance increases and the parallel efficiency decreases as the size of the background increases with increasing number of passages. In the present parallel implementation, the background grid is first partitioned into as many subgrids as the number of passages, and the solution on each partitioned subgrid and the blade grids may be assigned to a separate processor.

NUMERICAL SOLUTION METHOD

2-D unsteady Navier-Stokes equations are solved in a distributed parallel environment. Parallel Virtual Machine (PVM) message-passing library routines are used in a cluster of PC processors connected by Ethernet. Master-worker paradigm is employed. The solution on each subgrid is assigned to a separate worker process. Intergrid boundary conditions are applied on the background grid partition.

Solution of Navier-Stokes Equations

An unsteady, implicit, thin-layer Navier-Stokes solver with the third order accurate Osher’s upwind biased flux difference splitting scheme[5] is solved on each subgrid. The strong conservation-law form of the 2-D, thin-layer Navier-Stokes equations in a curvilinear coordinate system, (ξ, ζ), along the axial and circumferential direction respectively, is given as follows

\[ \partial_t \mathbf{Q} + \nabla \cdot \mathbf{F} + \partial_\zeta \mathbf{G} = Re^{-1} \partial_{\zeta} \mathbf{S} \]  

where \( \mathbf{Q} \) is the vector of conservative variables, \( 1/J(p, \rho_u, \rho_w, \epsilon) \), \( \mathbf{F} \) and \( \mathbf{G} \) are the inviscid flux vectors, and \( \mathbf{S} \) is the thin layer approximation of the viscous fluxes in the \( \zeta \) direction normal to the airfoil surface. The flowfield is assumed to be fully turbulent and the Baldwin–Lomax turbulence model is implemented only on the blade grids.

For unsteady cascade flows, vibration modes of blades with constant amplitude and constant interblade phase angle result in a pitch-wise spatially periodic flow with a multi-passage domain. Therefore, the number of passages depends on the interblade phase angle. An interblade phase angle of 0° (in-phase vibration) requires only one blade passage, phase angles of 180° and 90° (out-of-phase vibrations) two and four blade passages, respectively.

At the inlet, total pressure, total temperature and the flow angle are given. At the outflow boundaries the exit static pressure is specified. The remaining flow variables are obtained using isentropic flow relations and the zeroth-order Riemann invariant extrapolation. At the periodic boundaries, the solution is overlapped. No-slip conditions are applied at the blade surfaces. At the intergrid boundaries, the conservative flow variables are interpolated from the neighboring subgrid at every time step[6-9]. Intergrid

![Fig. 2 Flowchart of the master program](image-url)
program MASTER
WORKEXE = 'pvm/src/worker'
c..start up pvm
    call SYSTEM("pvm hosts")
c..enroll this program in PVM
    call PVMFMYTID(mytid )
c..spawn the worker
    open(unit=1, file='hosts')
    read (1, (a, end=-10, err=10) (nodes(n), n=1,65)
    10 mode = n-1
    do n = 1, nby
        call PVMFSPAWN(WORKEXE, pvmhost.
            + nodes(n),1, tids(n),err)
    enddo
c..Partition the background grid
    mdelk=kgx(1))/nbl)
    kgb(s)=1
    do i=2,nbl
        kgb+1= kg(s)+1+ mdelk
        kgb(s)= kg(s)+1
    enddo
    kgb(i)= kgx(1)
c..Broadcast init data, grid and q arrays to the nodes
    call SENDINIT
    do n = 1,nbb
        call SENDGRQ(n)
    enddo

subroutine SENDGRQ(nb)
if(nb,le,nbl) then
    ng = 1
    k = kgb(s)
    kmx= kg(s)-1+mdelk
    ikmx= ra%kmx
else
    ng = nb - nbl + 1
    k = 1
    kmx= kgx(ng)
    ikmx= ra%kmx
endif
msgid=2
call PVMFINITSEND(PvmDataRaw, ibuf)
call PVMFPACK(int4, nb, 1, info)
call PVMFPACK(int4, kgx, 1, info)
call PVMFPACK(int4, kmx, 1, info)
call PVMFPACK(real4, kgx, 1, info)
call PVMFPACK(real4, kgx, 1, info)
do l=1,4
    call PVMFPACK(real4, kgx, 1, info)
    do l=1,4
        call PVMFSEND(tids(nb), msgid, info)

nia : Maximum array size in i
imgx, kmx : Grid sizes in i and k
kgb, kgb : partition index of the background array
kg, z : global array for x and y-coordinates
kg : global array for solution variables

Fig. 3 Excerpts from the master program

boundary points are first localized on the donor subgrid using a directional search algorithm. Intergrid boundary conditions are applied in the master process.

The numerical integration on each computational subgrid is performed using an upwind biased, factorized, iterative, implicit, numerical scheme[5]. The hole points in the background grid are accommodated by replacing the corresponding coefficient matrix by an identity matrix and by setting the right hand side to zero. This process essentially imposes no change in the flow variables for the hole grid points. One sided differencing is used at the fringe points next to the hole boundary points.

Parallel Solution Algorithm

The parallel solution algorithm is based on the master-worker approach. The master process performs all the input-output, starts up PVM, spawns worker processes and sends initial data to the workers (Figure 2). The excerpts from the master program are given in Figure 3. The system call "pvm hosts" starts the pvm with the pvm nodes given in the file "hosts".

Worker processes first receive the initial data, apply the flow boundary conditions, exchange intergrid boundary conditions with the neighboring/overest grids at each time step, and solve the flowfield with the these boundary conditions. Aerodynamic loads are also evaluated in the worker processes.

The intergrid boundary conditions are applied on the partitioned background grid boundaries and the overest grid boundaries between the blade grids and the background grid. The partitioned background grids are overlapped at their upper and lower boundaries by one grid point. Thus, they exchange the solution at these grid lines.

Application of the overest grid boundary conditions between blade and background grids is done in the partitioned background grid. It involves first the localization of intergrid boundary points on the donor grid, and then the interpolation of flow variables. As given in the subroutine STEP, at each time step, a partial solution close to the outer boundary of the blade grid is sent to the corresponding background grid process. Only the solution array on the outer 8 k-grid lines are sent to the background grid process. The outer 8 grid lines of the blade grids cover enough depth for localizing the hole boundary points of the background grid. The intergrid boundary conditions are then applied[4,5], the new boundary information is sent back to the blade grid process.

Parallel Computing Environment

The parallel computing environment consists of networked PCs running Linux operating system; 4 Intel
Pentium-II 300Mhz processors with 64Mb memory, 3 Intel Pentium 200Mhz processors with 64Mb memory. PCs are connected to a 10Mhz Ethernet network. PVM version 3.4, which is the latest version currently available, is installed.

RESULTS AND DISCUSSION
The parallel algorithm was applied to the flows through two- and four-passage turbine cascades, and a five-passage compressor cascade. It is first validated that the parallel solutions are the same as the serial solutions computed earlier[8-10] in terms of aerodynamic loads and the flowfields. The speed-up and parallel efficiency figures are then evaluated based on the wall clock time per timestep (time/ts) of the computations.

<table>
<thead>
<tr>
<th># of proc</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>time/ts (sec)</td>
<td>2.64</td>
<td>1.34</td>
<td>0.86</td>
</tr>
<tr>
<td>speed-up</td>
<td>1</td>
<td>1.97</td>
<td>3.07</td>
</tr>
<tr>
<td>efficiency</td>
<td>1</td>
<td>0.99</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Table 1 Parallel efficiency for the two-passage flow computations

First, the steady flow through a two-passage turbine cascade is computed in parallel at $\alpha_{inlet} = 50.0^\circ$, $P_0 = 205800 \text{ Pa}$, $T_0 = 330.1^\circ K$, and $P_{ext} = 121200 \text{ Pa}$.

Figure 4 Flowchart of the worker program

```c
program WORKER
    data first ./true/
    if(first) then
        first = .false.
        call PVMFMYTID( mytid )
        call PVMFPARENT( mstid )
        call RECVINIT
        call RECVGRQ
        endif
    c..Apply BC
    do itr = 1, nstep
        if(mynb .le. nb) then  !Background grid
            call SENDBC(np,1)
            call SENDBC(nn,1)
            call RECVBC
            call RECVBC
            call EBCG
        else  !Blade grid
            call SENDBC(mynb-nbl,0)
            call BCO
            call RECVBC
        endif
    c..Solve the flowfield
        call SOLVER
        if(save) call SENDQ(loq)
    enddo
... 
```

subroutine RECVGRQ
    msgid = 2
    call PVMFRECV(mstid, msgid,info)
    call PVMFUNPACK(int4, myrb, 1, 1, info)
    call PVMFUNPACK(int4, imx, 1, 1, info)
    call PVMFUNPACK(int4, kmx, 1, 1, info)
    call PVMFUNPACK(real4, x(1,1), kmx, 1, info)
    call PVMFUNPACK(real4, y(1,1), kmx, 1, info)
    do l = 1, 4
        call PVMFUNPACK(real4, q(l,1,1), kmx, 1, info)
    enddo

mynb : Index for the local block
imx, kmx : Extend of local grid in i and j
x, y : local array for x and y-coordinates
q : local array for solution variables

Figure 5 Excerpts from the worker program

(Figure 6.) The blade grid is of $276 \times 25$ size, and the background grid is of $91 \times 61$ size. The parallel efficiency figures are given in Table 1. As seen in the case of two processors 99% efficiency is obtained, whereas in the case four processors, the efficiency drops to 77% due to the load imbalance and the intergrid communication overhead. It should be noted that the most of
the communication is done between the blade grid and the corresponding background grid. Parallel efficiency is higher when the solutions on these two grids are computed in the same processor. The ratio of the grid sizes is $\frac{S_{\text{blade}}}{S_{\text{background}}} = 0.80$, which means that in the case of four processors, two processors are idle for about 1/5 CPU time. It suggests that the communication overhead for this case is about $(0.80 - 0.77) = 0.03$.

Figure 7 shows the steady state pressure distribution for the four-passage flow computed at $\alpha_{\text{inlet}} = 50.0^\circ$, $P_0 = 219600 \text{ Pa}$, $T_0 = 329.68^\circ K$, and $P_{\text{exit}} = 91400 \text{ Pa}$. The blade grids and the partitioned backgrounds grid are of the same size as the previous case. The parallel efficiency of the computations with up to five processors is given in Table 2. Similarly when a blade grid and the corresponding background grid are assigned to the same processor, the efficiency is above $98\%$. The load imbalances and the processor power in

---

**Fig. 6** Steady flow through a two-passage turbine cascade

**Fig. 7** Steady flow through a four-passage turbine cascade
Fig. 9 First harmonic of the unsteady surface pressure distribution on the pitching blade of the five-passage compressor cascade

the case of 3 and 5 processor computations accounts for the drop in the efficiency.

<table>
<thead>
<tr>
<th># of Procs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time/td (sec)</td>
<td>5.30</td>
<td>2.68</td>
<td>2.16</td>
<td>1.36</td>
<td>1.70</td>
</tr>
<tr>
<td>Speed-up</td>
<td>1.00</td>
<td>1.98</td>
<td>2.45</td>
<td>3.90</td>
<td>3.53</td>
</tr>
<tr>
<td>Efficiency</td>
<td>1.00</td>
<td>0.99</td>
<td>0.82</td>
<td>0.98</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 2 Parallel efficiency for the four-passage flow computations

Next, the steady flow through a five-passage compressor cascade is computed at $\alpha_{\text{inlet}} = 62.0^\circ$, $P_\text{o} = 166000 P\alpha$, $T_\text{o} = 301.01^\circ K$, and $P_{\text{inlet}} = 122000 P\alpha$. The blade and background grids are of 276 x 25 and 91 x 123/244 size, respectively.

<table>
<thead>
<tr>
<th># of Procs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time/td (sec)</td>
<td>4.10</td>
<td>2.20</td>
<td>1.76</td>
<td>1.32</td>
<td>1.04</td>
</tr>
<tr>
<td>Speed-up</td>
<td>1.00</td>
<td>1.86</td>
<td>2.33</td>
<td>3.10</td>
<td>3.94</td>
</tr>
<tr>
<td>Efficiency</td>
<td>1.00</td>
<td>0.93</td>
<td>0.78</td>
<td>0.78</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 3 Parallel efficiency for the five-passage flow computations

The Mach number contours are shown in Figure 8. The efficiency figures are given in Table 3. The expected higher efficiency with five processor computations is not materialized due to the fifth processor being slower than the others. However, the %93 efficiency in the case of 2 processor computation suggests that the communication overhead increases with the
number of cascade passages.

Next, an unsteady flow is computed as the mid-blade undergoes periodic pitch oscillations about its mid-chord. The periodic pitch oscillation of the mid-blade is defined as

$$\alpha = A \sin(2\pi f t)$$

where $A$ is the pitch amplitude and $f$ is the pitching frequency. The unsteady flowfield is computed at $A = 0.5^\circ$ and $f = 320 \text{Hz}$ for more than four periods of the plunging motion with about 10000 timesteps/period. It takes almost 10 hours of computation time in parallel. Figure 9 shows the first harmonic, and the phase shift of the unsteady surface pressure distribution on the pitching blade, and they are compared to SAFESI predictions. SAFESI is a Navier-Stokes solver developed at Institut für Strahlantriebe und Turboarbeitmaschinen in Aachen, Germany[4]. It solves multi-passage flows in single-passage domain using direct store interblade boundary conditions. The parallel processing cuts the turn-around time for the computation of an unsteady cascade flow more than 3 days short in comparison to a serial computation on a single processor.

Figure 10 summarizes the speed-up and the parallel efficiency of the present computations of multi-passage cascade flows. It is observed that the maximum parallel efficiency is achieved when the blade grid and the corresponding background grid is assigned to the same processor. The maximum speed-up is still obtained when twice as many processors as the number of passages are employed.

**CONCLUDING REMARKS**

Multi-passage unsteady cascade flowfields, which are discretized with partitioned overset grids, can be computed efficiently on a small cluster of networked PCs using PVM massage passing library routines. Computation of a flow through a four-passage cascade is speeded up about 3.9 times in comparison to serial computations. PVM version 3.4 runs reliably on networked PC’s under the Linux operating system. It is expected that employment of a 100Mbps Ethernet switch connecting the PCs would significantly drop the communication overhead between the processors.

**ACKNOWLEDGMENT**

This work is being supported by the Middle East Technical University Research Fund under the contract AFP-99.03.13.03

**REFERENCES**


