

PARALLEL SIMULATION OF CAVITATED FLOWS IN HIGH PRESSURE SYSTEMS

Panagiotis Adamidis¹⁾, Frank Wrona³⁾, Uwe Iben³⁾,
Rolf Rabenseifner¹⁾, Claus-Dieter Munz²⁾

¹⁾ High Performance Computing-Center Stuttgart (HLRS)

²⁾ Institute for Aero- and Gasdynamics (IAG)

³⁾ Robert Bosch GmbH, Dept. FV/FLM

Contact: Panagiotis Adamidis (adamidis@hlrs.de)

Frank Wrona (frank.wrona@de.bosch.com)

PARCO 2003, Sep. 2–5, Dresden, Germany

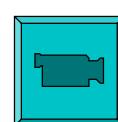
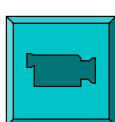


CAVITATED FLOWS IN HIGH PRESSURE SYSTEMS

What is cavitation

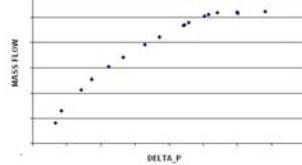
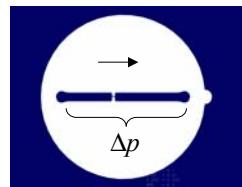
- If the pressure drops beneath a certain threshold the fluid begins to evaporate
- That happens there where the fluid is accelerated

Cavity formation behind a backward facing step

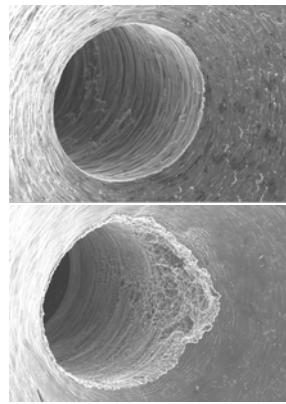


Effects of Cavitated Flows

Limitation of mass flow rate



Erosion



PARCO 2003

Slide 3

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

H L R I S



Governing Equations

- The fluid is treated as a homogeneous mixture (single phase fluid)
- All physical properties (pressure, temperature, ...) of both phases are the same
- The phase transition is modelled by the mass fraction

$$\mu = \frac{m_G}{m_L + m_G} \equiv \mu(p, T)$$

- Mixture relations and for the density and the internal energy

$$\frac{1}{\rho} = \frac{\mu(p, T)}{\rho_G(p, T)} + \frac{1 - \mu(p, T)}{\rho_L(p, T)}$$

$$e = \mu(p, T)e_G(p, T) + (1 - \mu(p, T))e_L(p, T)$$



PARCO 2003

Slide 4

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

H L R I S



Governing Equations

- Navier-Stokes equations for the homogeneous mixture

$$\mathbf{u}_t + (\mathbf{f}^{conv}(\mathbf{u}, p) - \mathbf{f}^{visc}(\mathbf{u}, T))_x + (\mathbf{h}^{conv}(\mathbf{u}, p) - \mathbf{h}^{visc}(\mathbf{u}, T))_y = 0$$

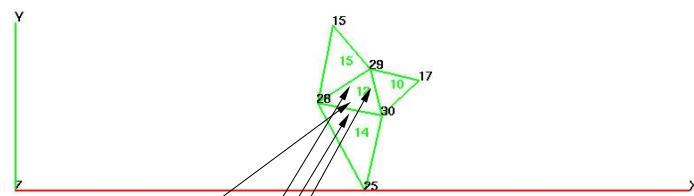
$$\frac{d}{dt} \int_V \mathbf{u} dV = - \oint_V (\mathbf{f}^{conv}(\mathbf{u}, p) - \mathbf{f}^{visc}(\mathbf{u}, T)) n_1 + (\mathbf{h}^{conv}(\mathbf{u}, p) - \mathbf{h}^{visc}(\mathbf{u}, T)) n_2 ds$$

$$\mathbf{u} = (\rho, \rho v, \rho w, \rho(e + 1/2(v^2 + w^2)))^T$$

- Main problem: Fluxes depend on pressure and temperature
- They are given only implicitly by density and internal energy



Numerical Solution: Transient Finite Volume Approach



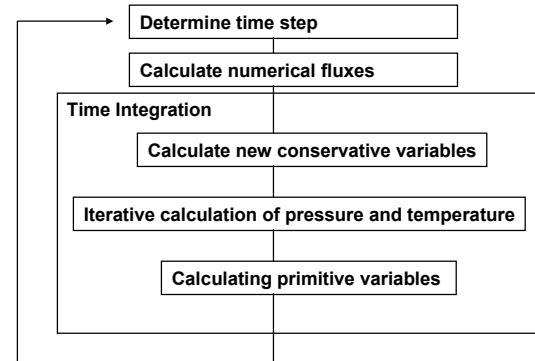
$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{V_i} \left\{ \sum_{j \in N(i)} \mathbf{F}_{kony}(\mathbf{u}_i^n, \mathbf{u}_j^n) l_j - \sum_{j \in N(i)} \mathbf{F}_{visc}(\mathbf{u}_i^n, \mathbf{u}_j^n, \mathbf{u}_{k \in M(i,j)}^n (K(k))) l_j \right\}$$

- Iterative solution for pressure and temperature from \mathbf{u}_i^{n+1}

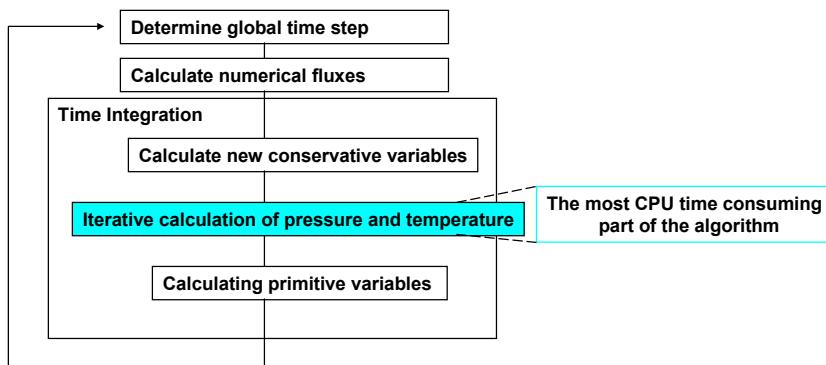
- System of conservative variables is calculated at each time step explicitly



Flow Chart of Serial Algorithm



Flow Chart of Serial Algorithm



Main Problem and Solution

Main Problem

- Pressure and temperature can only be calculated iteratively
- This iterative process is the most CPU time consuming part of the whole simulation
- Especially the time needed for cavitating cells is much more than for others, because there are more iterations executed in order to determine pressure and temperature
- This leads to enormous computation times for large realistic problems, which are in the range of several days or weeks

Solution

- Parallelization of the algorithm in order to take advantage of more processors and memory space of a parallel computing environment
- Target computing platform is a PC cluster

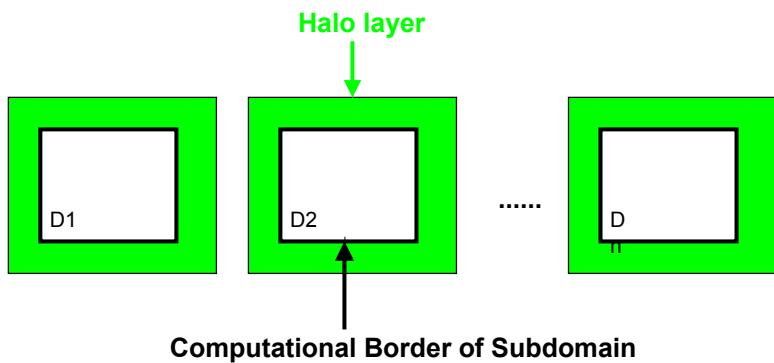


PARCO 2003

Slide 9

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

Parallelization using Domain Decomposition



- Communication is needed in order to update the halo cells

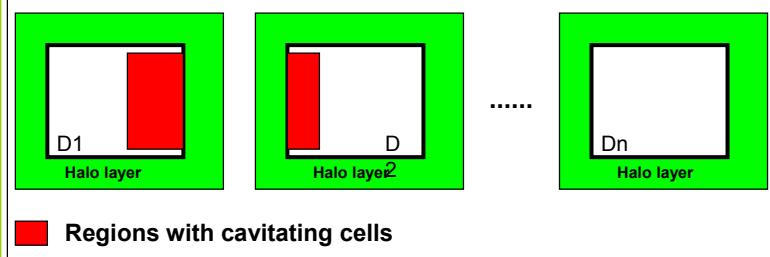


PARCO 2003

Slide 10

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

Load Imbalance due to Cavitation



- The number of cavitating cells is changing in each time step
- Cavities are moving across subdomains



PARCO 2003

Slide 11

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

H L R I S



Reassigning Work done during PT Calculations

Approach

- Detecting cavities in each subdomain and **reassign only the computations** done on the cells during the iterative procedure of calculating pressure and temperature from subdomains with more workload to subdomains with less workload
- **The cells are not moved** to other subdomains, thus keeping the initial partitioning
- Only the information needed to determine pressure and temperature of a cell is moved to another process and the result is recollected to the original location of the cell

How to detect cavities

- Using the void fraction criterion
- Using the CPU time spend during the iterative PT calculation in a cell as criterion



PARCO 2003

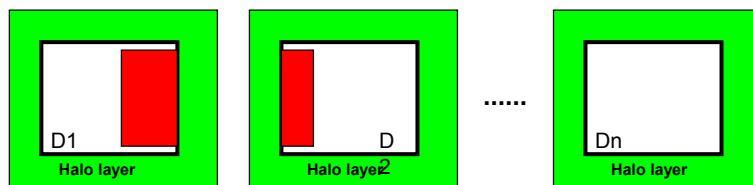
Slide 12

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart

H L R I S



CPU Time Criterion



t_i = iteration time needed by each cell i

t_{pj} = iteration time needed by each subdomain j

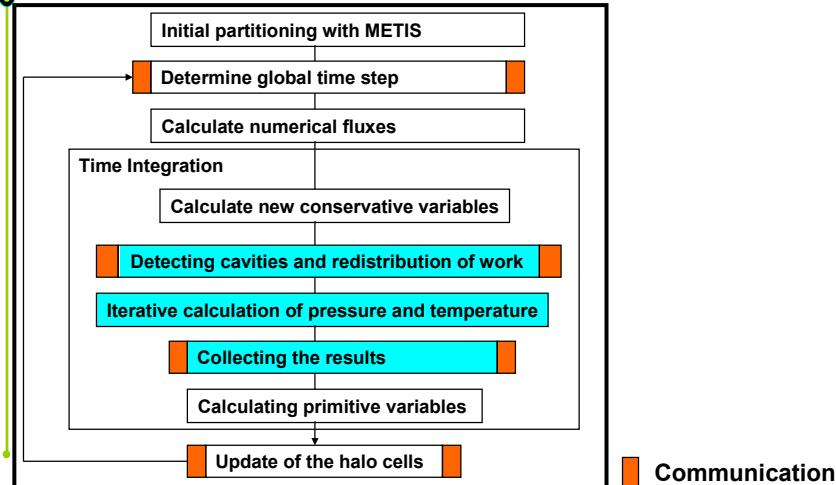
t_{opt} = average iteration time over all subdomains

CPU Time Criterion

$$t_{diff,j} = t_{pj} - t_{opt}$$

- $t_{diff,j} > 0$ process j is classified as “sender” of workload
- $t_{diff,j} < 0$ process j is classified as “receiver” of workload
- $t_{diff,j} = 0$ process j will not reassign any work

Flow Chart of the Parallel Algorithm



Reassigning Work of PT Calculations vs Migrating Cells

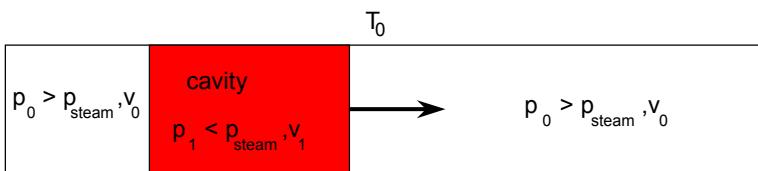
Migrating Cells

- Repartitioning the mesh or using diffusion schemes means that redistribution of cells would have to be carried out at every time step
- For **each migrated cell** the amount of data which would have to be communicated is **1536 Bytes (32 integers and 176 doubles)**, which means a communication time of about **140 µs** on a 100 Mbit/s Ethernet

Reassigning Work done in the iterative PT part

- Each “sending” process must send **32 Bytes (4 doubles)** of data per cell to a “receiving” process and must receive **16 Bytes (2 doubles)** of results, which is a total of **48 Bytes per cell**.
- On a 100 Mbit/s Ethernet a communication time of about **4.3 µs** is needed, which is about **32 times less** than the above mentioned approach

Benchmark: Shock Tube Problem



- Length of tube 1 m
- Mesh consisting of 10000 cells (1000 in x-direction, 10 in y-direction)
- Cavity is embedded in pure liquid
- **Computing platform:** PC cluster consisting of dual-CPU PCs with Intel Xeon 2GHz processors

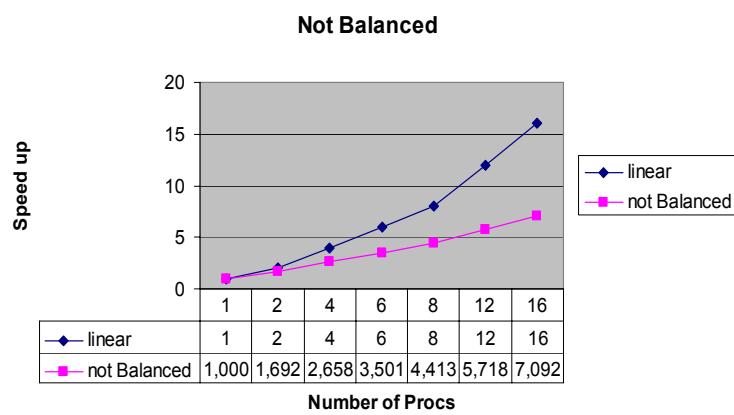


PARCO 2003

Slide 17

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart**H L R I S**

Speed up



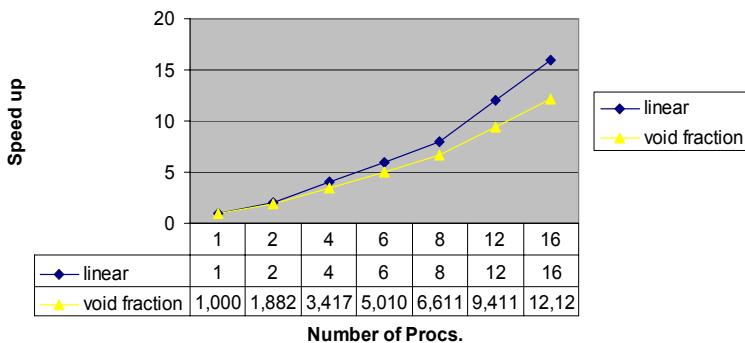
PARCO 2003

Slide 18

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart**H L R I S**

Speed up

Void Fraction Criterion



PARCO 2003

Slide 19

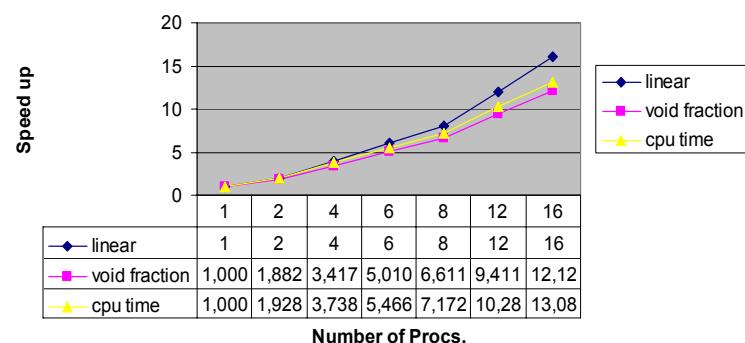
Panagiotis Adamidis

Höchstleistungsrechenzentrum Stuttgart



Speed up

CPU Time Criterion



PARCO 2003

Slide 20

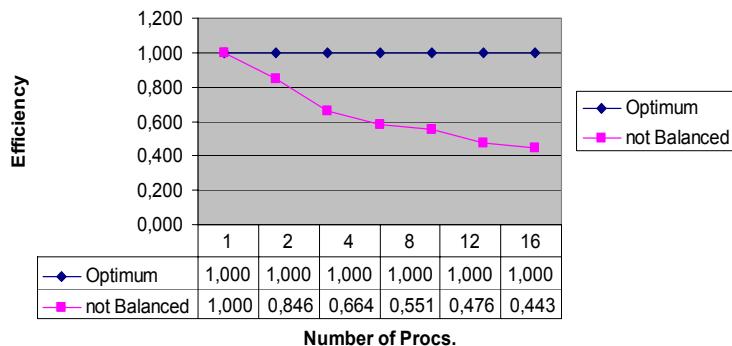
Panagiotis Adamidis

Höchstleistungsrechenzentrum Stuttgart



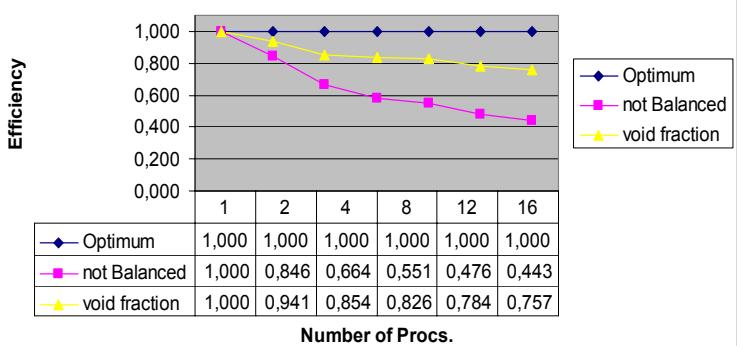
Efficiency

Not Balanced

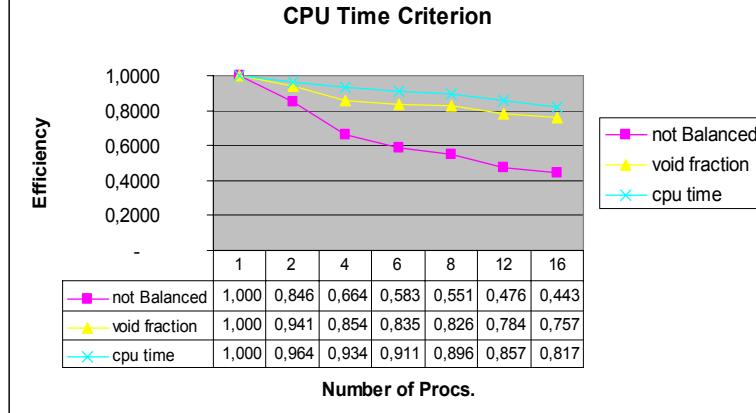


Efficiency

Void Fracrtion



Efficiency



PARCO 2003

Slide 23

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart**H L R I S**

Summary and Outlook

Summary

- Cavitation occurs in many industrial applications and cannot be calculated detailed in a proper response time on a single processor computer
- To solve large problems parallelization is unavoidable
- The load imbalance introduced by the cavities in the parallel algorithm has been treated by reassigning only the work done on cells during the most CPU time consuming part (the iterative PT calculations) of the algorithm
- The initial mesh partitioning is not changed
- The results show a significant gain in efficiency

Outlook

- Future work will be concentrated in treating 3D problems and
- on finding better criteria to detect cavities



PARCO 2003

Slide 24

Panagiotis Adamidis
Höchstleistungsrechenzentrum Stuttgart**H L R I S**