# Introduction

## The state of CFD

## The problems of grid generation

## Structured vs. Unstructured grids

# The Unified Coordinates

## The history of UCS

### Hui 2-, 3-D

#### Unsteady BCs

### Hui Viscous

#### Recirculation

#### Shear layers

### Other Equations

#### Multi-material flows

#### MHD flows

#### Kinetic flows (BGK)

#### Space-marching Euler

### Avalanche modeling

## The UCS transformation

### Grid velocity

#### Hui’s methods

##### Lagrangian

##### Grid-angle-preserving

##### Jacobian-preserving

##### Skewness-preserving

#### Limitations to Hui’s work

##### Do not follow boundaries exactly

##### Does not pin grid points to singular points

##### Elastic “pressure-based” grid possibilities?

### Compatibility conditions

#### Derivation

#### Typical solution methods

##### Finite difference?

##### Boundary value methods?

##### Impacts on solution stability

#### The full UCS equations

##### Strong conservation form for flow variables and metric components

##### Source terms for physical coordinates

##### Free specification of grid velocity, as long as transformation remains well-behaved (J>0?)

### Assumptions

#### Eta, Zeta are material coordinates (Lagrangian-esque)

#### … at least mostly. Perhaps some kind of balance between material-ness of Eta, Zeta and distance from boundaries?

#### Or maybe better to just pin grid point motion at boundaries.

#### Xi is determined such that a well-behaved grid is obtained. In 2-D, grid-angle preserving is the preferred method.

### Solution algorithm

#### Time-step-Eulerian

#### Dimension Splitting vs. Finite-Volume methods

#### 1st-order Godunov solver

#### 2nd-order MUSCL update

#### Godunov dimensional splitting

##### Describe Godunov algorithm

#### Finite-Volume

##### Describe algorithm and rationale, as well as drawbacks.

### Specification of grid motion

#### Grid-angle-preserving

##### Include derivation here

#### Jacobian-preserving

##### Include derivation here

#### Elastic boundary-conforming?

# Verification of Codes – Almost copy/paste from SciTECH

## Order of convergence verification

### Measure rate of convergence of code to complex, exact, solution

### Compare convergence rate with that expected based on algorithm.

### Highly sensitive to even subtle errors

## Method of Manufactured Solutions

### Exact solutions are rarely complex enough to yield useful verification tests.

### Boundary conditions for exact solutions are typically simplistic.

### MMS begins with a manufactured solution and an analytically computed source term

#### Manufactured solution can be as complex as desired, but must be differentiable.

#### Boundary conditions are arbitrary, based on the value of the solution at the boundaries.

#### Source term is computed using computer-aided algebra (CAS) systems and code generation tools.

## Integrative Method of Manufactured Solutions

### Numerical integration accuracy

### IMMS performance (Roy’s method)

### Multidimensional integration with discontinuities

## Verification of BACL-Streamer & UCS

### Choice of exact & manufactured solutions

### Verification of Euler solver

### Verification of UCS solver under various forms of grid motion

### Verification of BCs?

# Challenges of Moving Grids

## The equations are solved in unsteady, computational coordinates.

## The boundaries are defined in physical coordinates.

## How do you efficiently apply unsteady BC’s in this way?

### Convert physical BCs to computational coordinates

### Determine computational coordinates from grid indices (d-xi = 1)

### Requires conversion of BCs to computational coordinates, but no loop over boundary points.

## How do you accurately resolve singular points in the flow?

## How do you ensure accurate tracing of boundary surfaces?

# BACL-Streamer

## Code design

### Space & time looping

### Mixed-Language Programming

### Modularity

## Unit tests

### Direct testing of expected outcome

### Grid convergence studies

### “Does routine \_\_X\_\_ do what I want it to do?”

## Algorithmic implementation

### Dimensional splitting

### U = h u

#### Lagrangian-esque grid motion

#### Grid-angle-preserving grid motion

# Conclusion