CHT/CFD for heat-exchanger design;
and some recent developments
by Brian Spalding

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design;
past, present and future

Contents

Part 1: Heat exchangers

- Pre-CFD;
- Space-averaged CFD
- Detailed-geometry CFD
- Space-averaged and detailed geometry CFD in combination

Part 2. Something about Suhas

Part 3: Algorithmic novelties

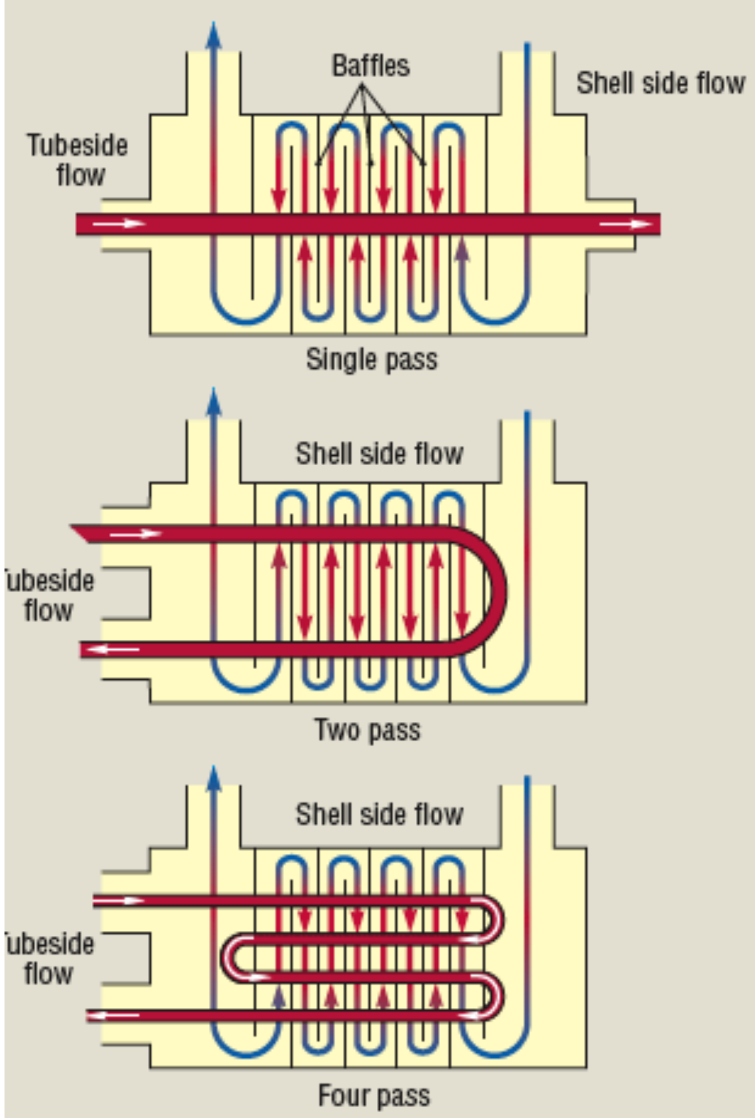
- SIVA and SIMPLE combined
- Odd-even segregation
- Simultaneous fluid flow and solid stress
- Free-surface posibilities

Pre-CFD : **guess** flow patterns; assume **uniform** coefficients

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design;
past, present and future

- Text-books contain formulae for effectiveness & pressure drop. In parallel-, counter-& cross-flow.
- The true flow pattern is complex, but unknown; so formula-fitting idealised guesses are used, as shown on the right.
- Heat-transfer and friction coefficients in fact **vary** from place to place. Unknowably; so therefore assumed **not** to vary.
- Most heat-exchangers are **still** designed in the same way: **assume** or **guess. No CFD/CHT!**

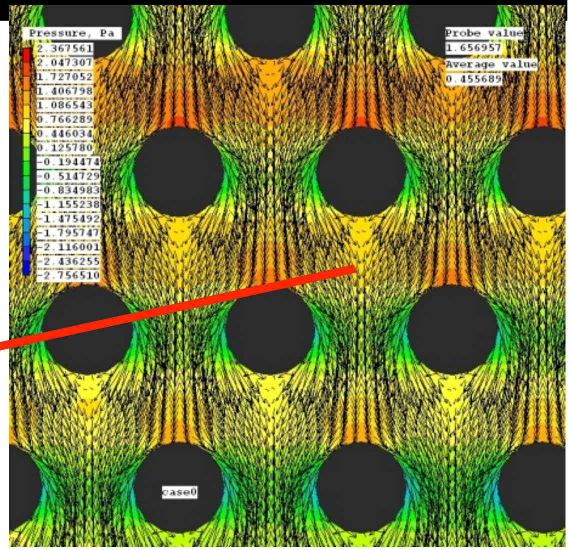


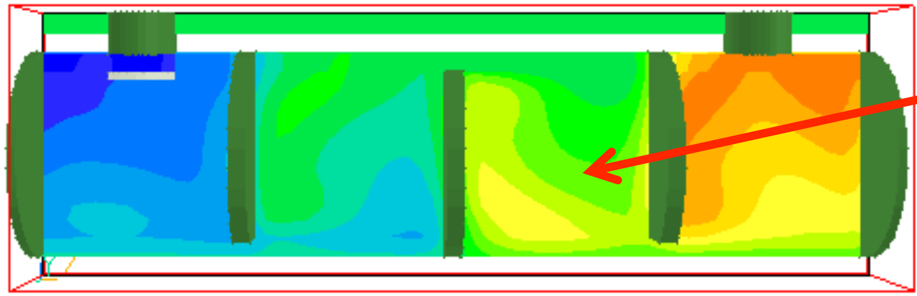
What is CFD?
There are two kinds

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design;
past, present and future

Detailed-geometry **CFD** (DGCFD) uses a **fine** grid for a **part** of the whole domain**,** *e.g***. a tube-bank.**





Space-averaged CFD (SACFD) uses a **coarser** grid for the **whole** domain, *e.g.* a shell-and-tube **heat exchanger.**

SACFD represents the small-scale behaviour by way of **formulae** for **volumetric** friction and heat-transfer **coefficients** *etc*.

Formulae may be derived from **experiments** or from **DG**CFD studies. The overall-prediction realism **depends on their** accuracy.

4 decades ago, SACFD enabled flow patterns to be **calculated** not guessed

- The first shell-side fluid-flow pattern was calculated by Suhas Patankar and myself in **1973.** He drew this -->
- Because computers were then small and slow, the computational **grids were coarse**, as shown on the right.
- Nevertheless, the **principle was proved**; and the practice helped in late 1970s to solve severe problems experienced by the **nuclear-power industry.**
- They involved **two-phase flow**, *i.e.* steam and water, inter-mingled, each with different velocity components.

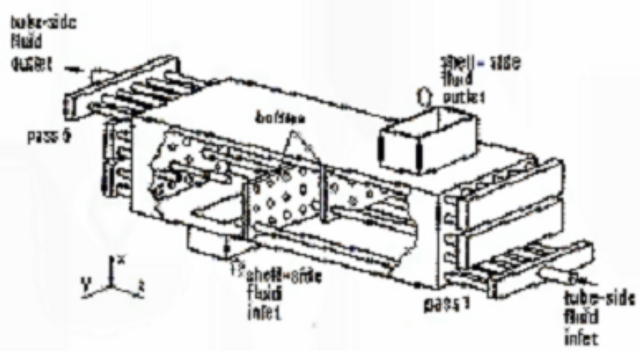


Fig. 3.1-1 The geometry of the heat exchanger.

Fig. 3.2-1 Velocity vectors in the central xz plane.

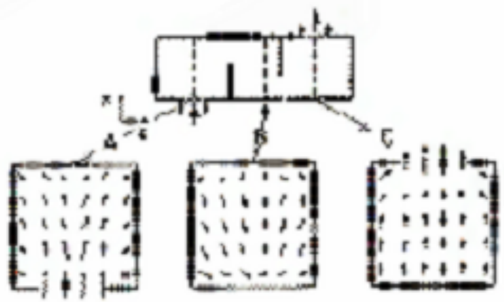


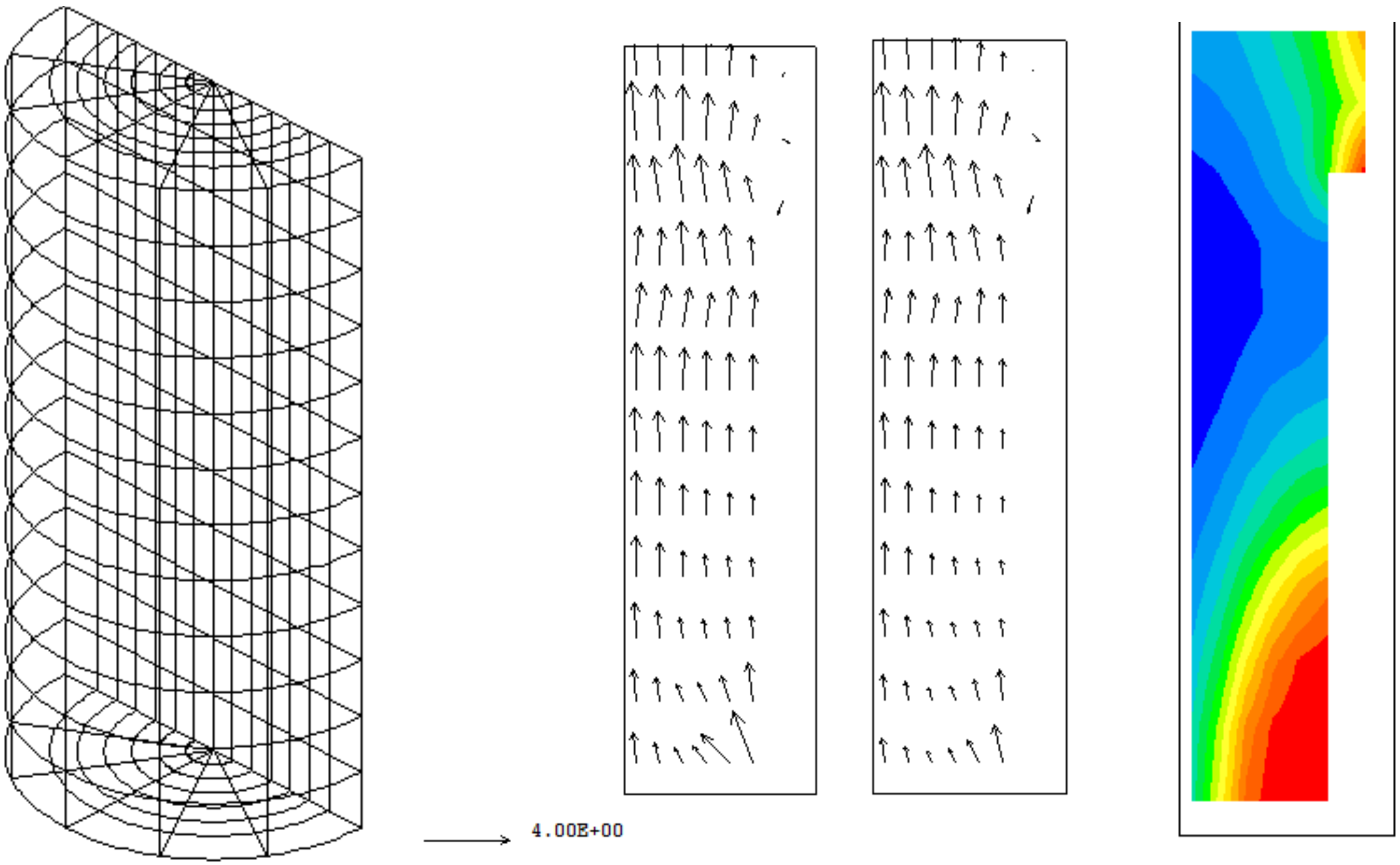
Fig. 3.2-2 Velocity vectors in the three xy planes.

IPSA (*i.e.* inter-phase Slip Algorithm) &
SACFD (*i.e.* Space-Averaged CFD)

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design;
past, present and future

IPSA applied to a nuclear steam generator (PHOENICS, 1982)

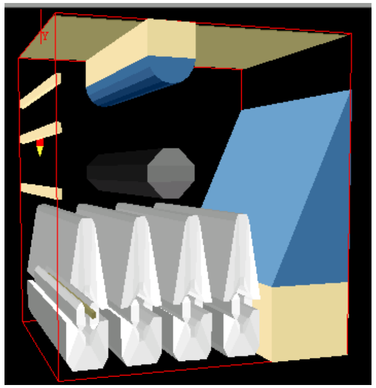


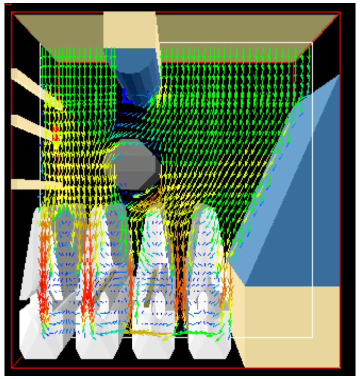
Grid Velocites: Steam Water Water Vol. fraction

Water-cooled power-station steam condensers; early 1980s

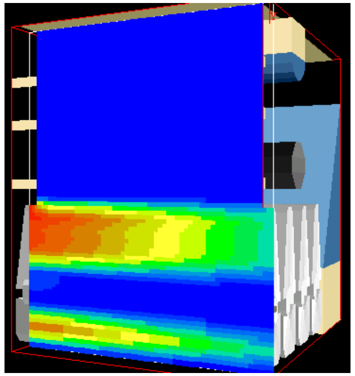
CHT for heat-exchanger design;
past, present and future

- Shell-side flow patterns in power-station steam condensers are **impossible to guess** because the **steam/air ratio** varies from very large (near entry) to very small (near exit). So velocities also vary greatly.





- Enlightened condenser designers therefore used CFD to compute **flow-patterns in order to predict** influences of tube-bundle geometry and baffles on performance.
- The image on the **left** shows the pattern by way of **calculated velocity vectors**.



- On the **right** are shown some predicted tube-side **temperature contours**, deduced from the (dependent-on-local-air-content) condensation rates. So we **can** calculate; need not guess.

Another power-station application

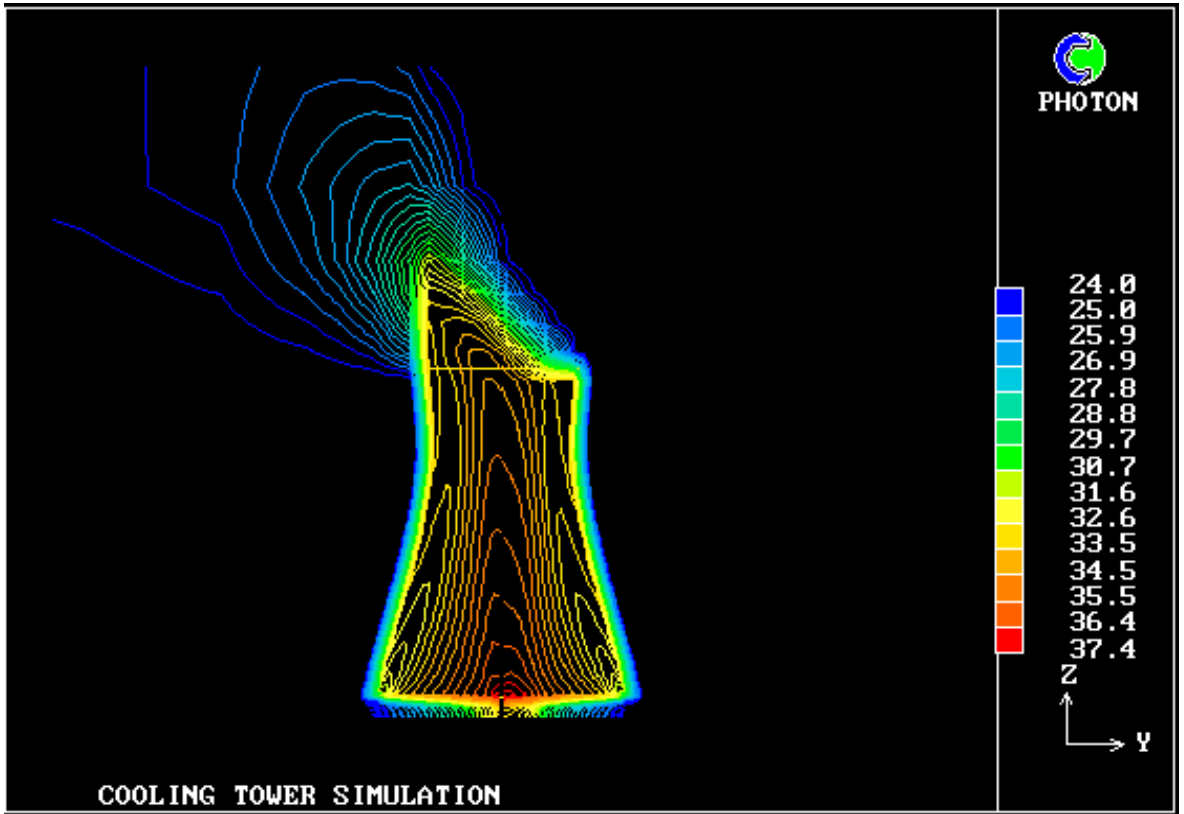
CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

In the late 1980s the effect of wind on air and water temperatures in natural-draft cooling towers was first computed by D.Radosavlevich.

CFD was used for simulating 2-phase-flow, namely upward-moving air and downward-moving water near the base of the tower.

But today's towers are still designed by way of pre-CFD methods.



Why is CFD not in everyday use for heat-exchanger design?

Some possible answers:

- Vendors of pre-CFD software wish to retain their income?
- Their customers do not read the research-oriented publications of Bengt Sunden *et al*?
- Vendors of better alternatives have not worked hard enough to prove and publicise their superiority?
- The better alternatives have not been made easy and inexpensive enough to use?

Conclusion: All the above; but (lack of) ease-of-use is the weightiest.

Example: The air-cooled steam condenser.

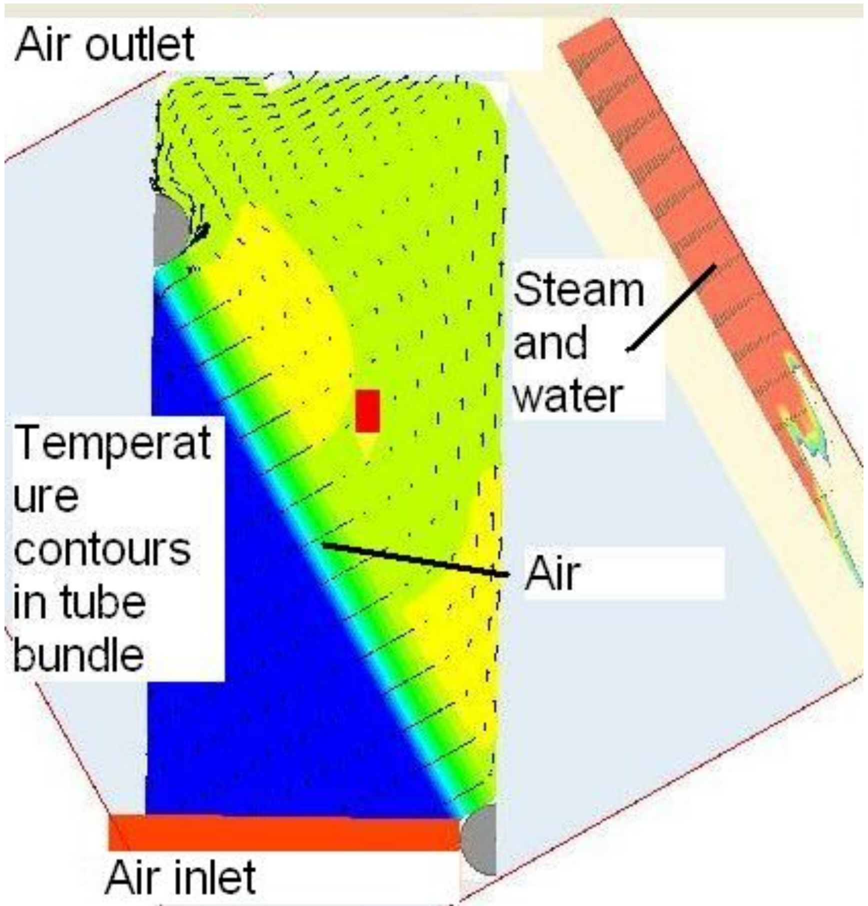
It's not easy. How would **you** begin?

2010: A multiple-grid example: (one half of) an air-cooled condenser

Most of the space in these condensers is occupied by air. Steam and water are present only inside the tubes.

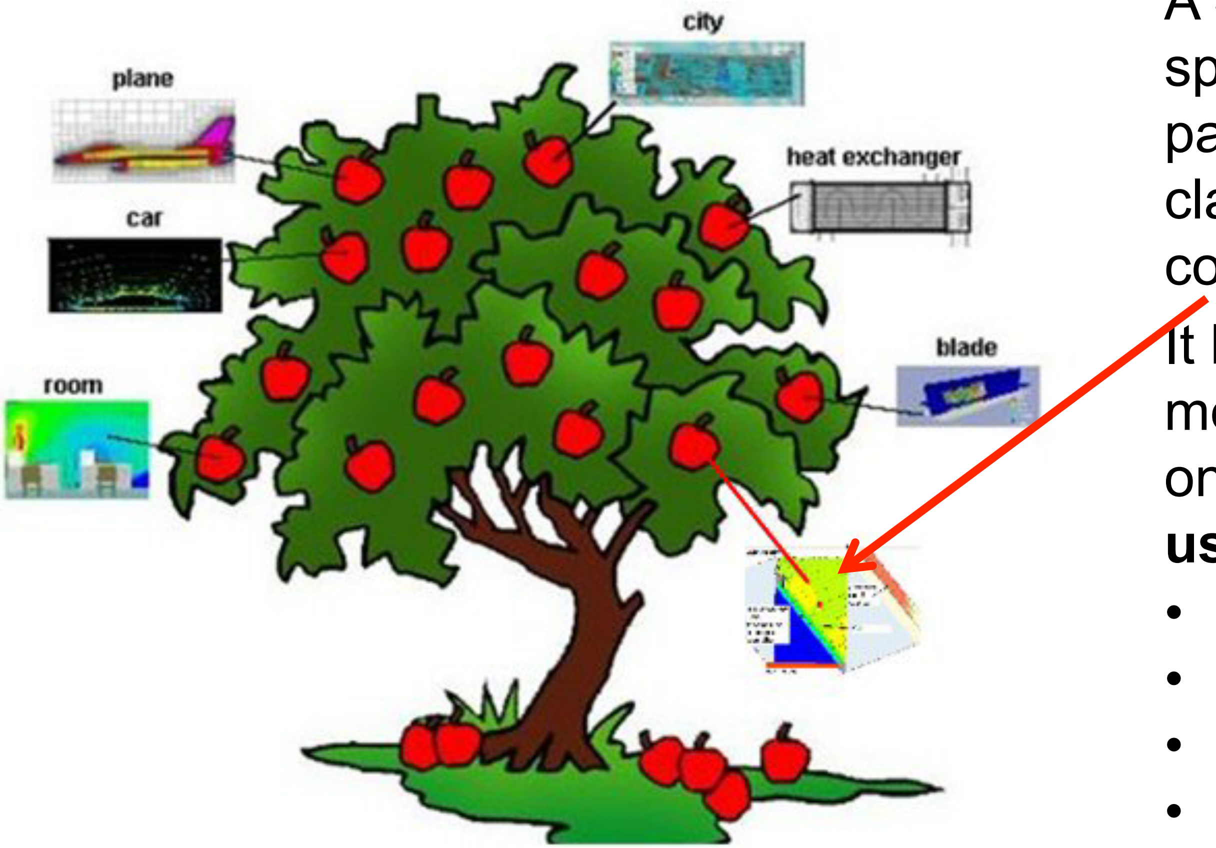
Therefore a ‘grid trick’ that PHOENICS can use is to cover the tube bundle **twice:** on left for air; on right for steam+water.

Few (even-CFD-using) designers are skilled enough to use this trick; but, if a special **SimScene** has been created, then **anyone** can do so.



What are SimScenes? They are apps for cities, rooms, heat exchangers, condensers, cooling towers, *etc*.

SimScenes are “apps on a app-tree” as shown below.



The CFD code is out of sight among the roots.

A SimScene is specific to a particular equipment class, *e.g.* air-cooled condensers.

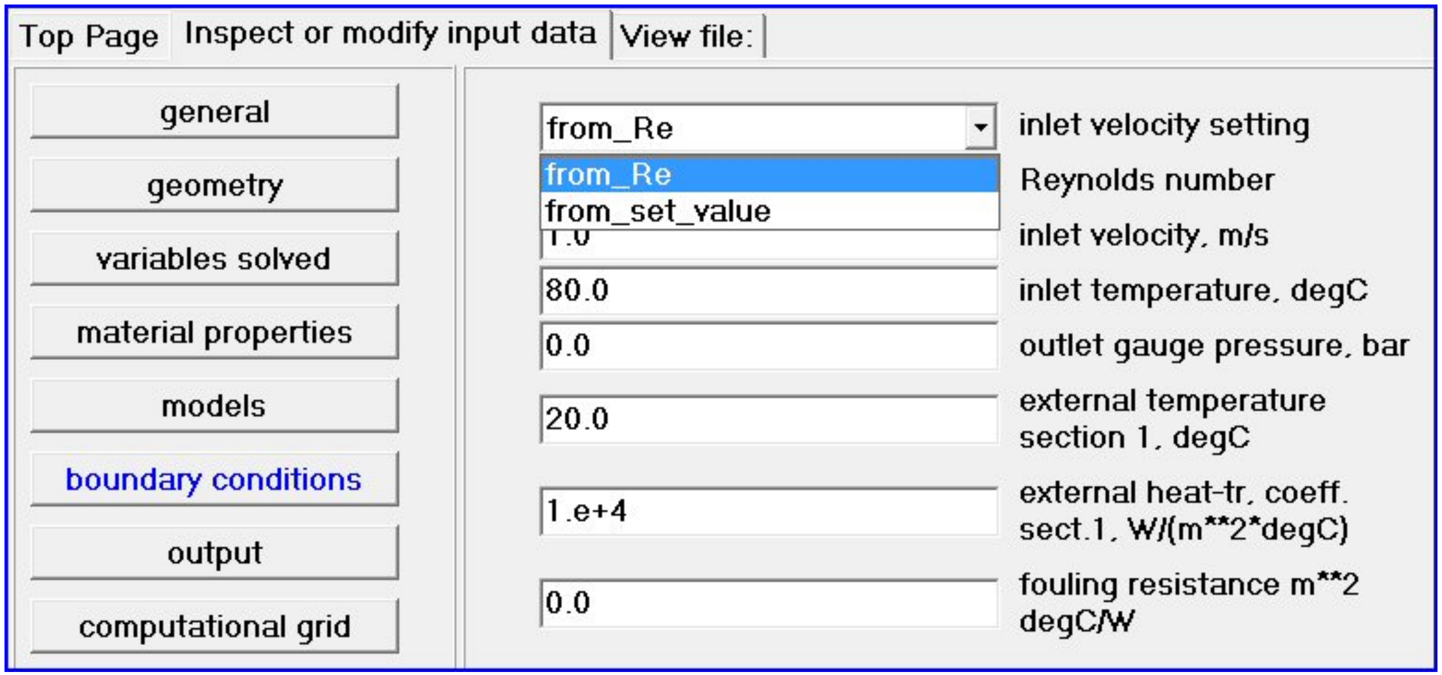
It has appropriate menus, for making only **meaningful-to-user** choices; *e.g.*

- Air temperature
- Steam pressure
- Air-in-steam %
- Tube diameter number, spacing
- Fin dimensions.

Part of the SimScene menu set for an air-cooled condenser

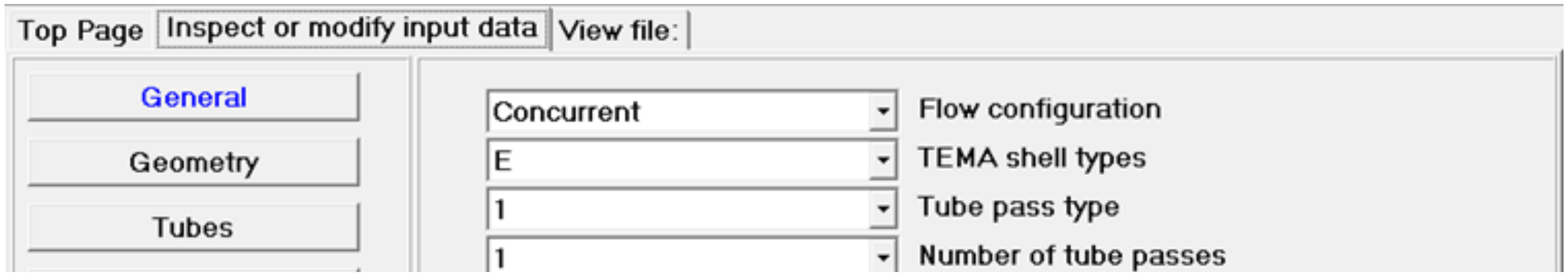
A SimScene is like a (CFD-based) car. Its user knows as much about CFD as a car-driver knows about thermodynamics.

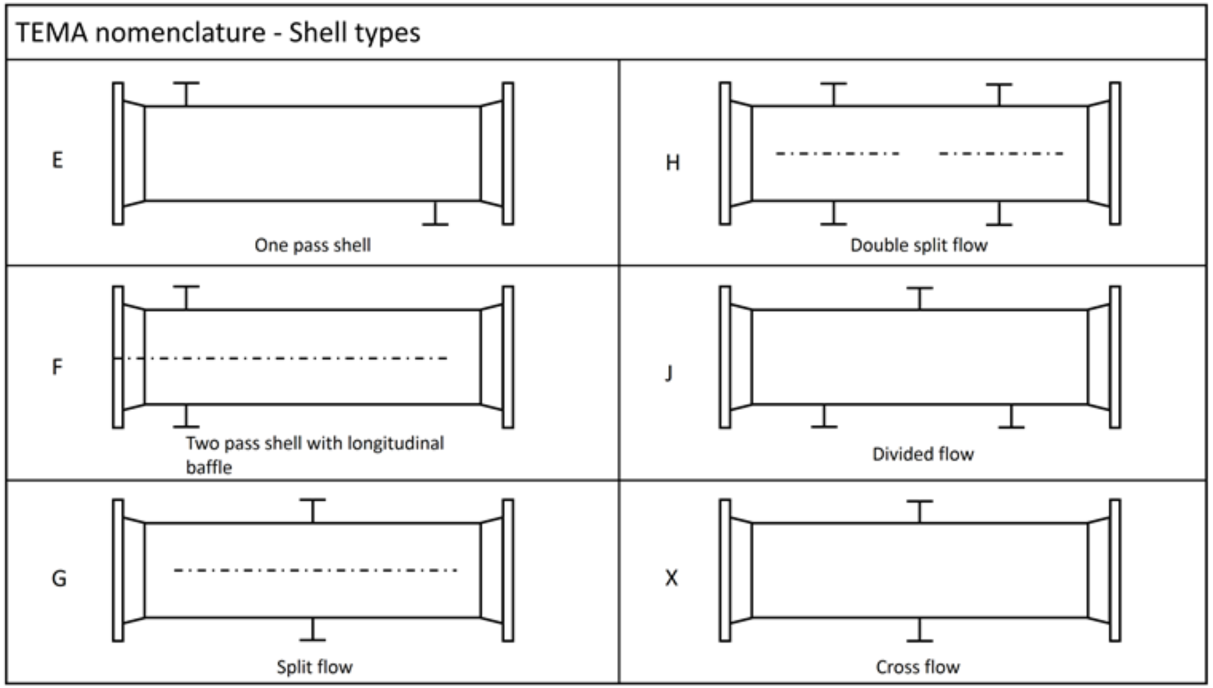
But he can choose from menus, written in **language which he does understand**. Below, the sub-menu names are on the left. The one for air-cooled-condenser boundary conditions is shown.



Part of the SimScene menu set for all shell-and-tube heat exchangers

Designers know what TEMA type they want; so they choose here.





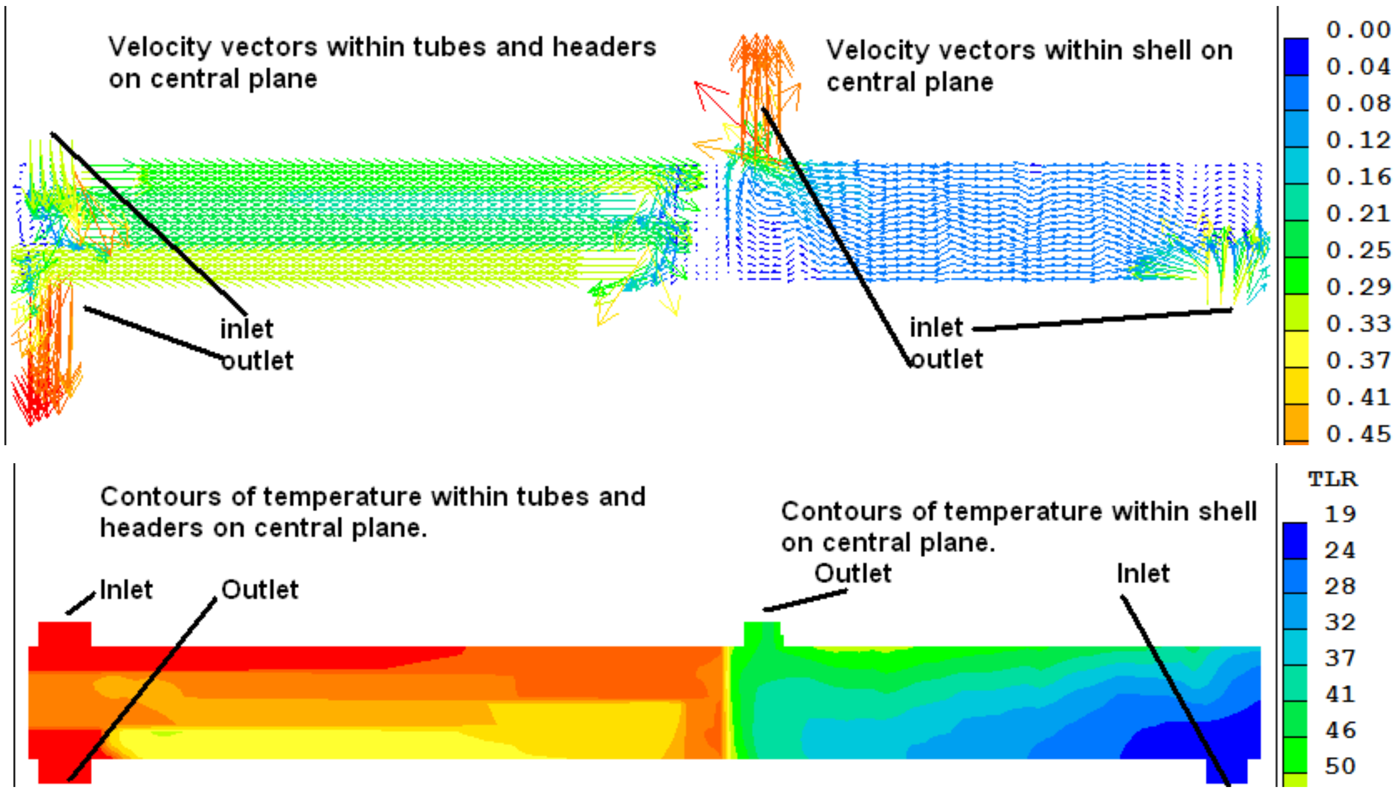
They choose just one letter, because **all** the types have been built into the SimScene.

Results from Heat-Exchanger-Design SimScene: E-type, counter-current, 4-pass, 6-baffles

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

Heat-transfer effectiveness and pressure drops are calculated; but also **much more** about internal processes, such as:



Sometime contours show that there must be **mistakes**, such as here: at tube outlet, the temperature is **the same as at inlet**!

SimScenes will change how engineering enterprises use CFD

- No longer will firms have to **employ expensive** CFD specialists.
- Nor purchase licences for **more-powerful-than-needed** general-purpose CFD software.
- Their SimScene-aware engineers will access **Internet**; upload a file containing the **scenario-defining input data** to 'the Cloud'; accept the **pay-as-you-go** cost estimate; **download the files** containing the computed output; if satisfied **authorise payment** of the bill.
- Only after many such experiences may they **perhaps** decide to buy a licence allowing their engineers to do the calculations for themselves.
- The technological basis for this mode of CFD exploitation is **available right now**.
- But wait: all the applications shown have been of **Space-Averaged** CFD. Has **Detailed-Geometry** CFD no contribution to make?
- Yes, it has; as four excerpts from a 2011 lecture explain.

Excerpt 1. Critique of the formulae used by **both** current methods and SACFD

Finned-tube bundles are in common use; and handbooks contain formulae **purporting** to represent their behaviour; for example this, from Rohsenow and Hartnett:

$$\frac{h_0 D_r}{k} = 0.134 \left(\frac{D_r G_{max}}{\mu}\right)^{0.681} \left(\frac{c\mu}{k}\right)^{1/3} \left(\frac{s}{l}\right)^{0.2} \left(\frac{s}{t}\right)^{0.113}$$

where h_0 is mean heat transfer coefficient Btu/hr-ft^2 external area-°F; D_r is root diameter of tube, ft; k is thermal conductivity; G_{max} is mass rate of flow at minimum cross section, lb/ft^2-hr; s is distance between adjacent fins, in.; l is fin height, in.; t is fin thickness, in.; c is specific heat; μ is viscosity at bulk temperature, lb/ft-hr

$$\frac{\Delta P g_c \rho}{n G_m^{\,2}} = f = 18.93 \left(\frac{D_r G_m}{\mu}\right)^{-0.316} \left(\frac{P_t}{d_r}\right)^{-0.927} \left(\frac{P_t}{P_l}\right)^{0.515}$$

Such formulae are copied slavishly from handbook to handbook; but can they truly be relied upon? Surely not because…

Excerpt 2. Critique of the formulae used by **both** current methods and SACFD

(1) The number of dimensionless parameters needed for finned-tube bundles (geometry, material properties, velocities) should be at least **12**.

(2) The army of experimentalists needed systematically to explore the corresponding **12-dimensional space** was surely never mobilised.

(3) Even if it had been, it is **not probable** that its findings would so conveniently have fitted the invariably-offered **power-law** forms:

- *Euler_number = a * Reynolds_number* ^{b}

and

- *Nusselt_number = d * Reynolds_number* ^{e} * *Prandtl_number* ^{f}.

About such subversive thoughts, **the heat-transfer community maintains a conspiracy of silence**.

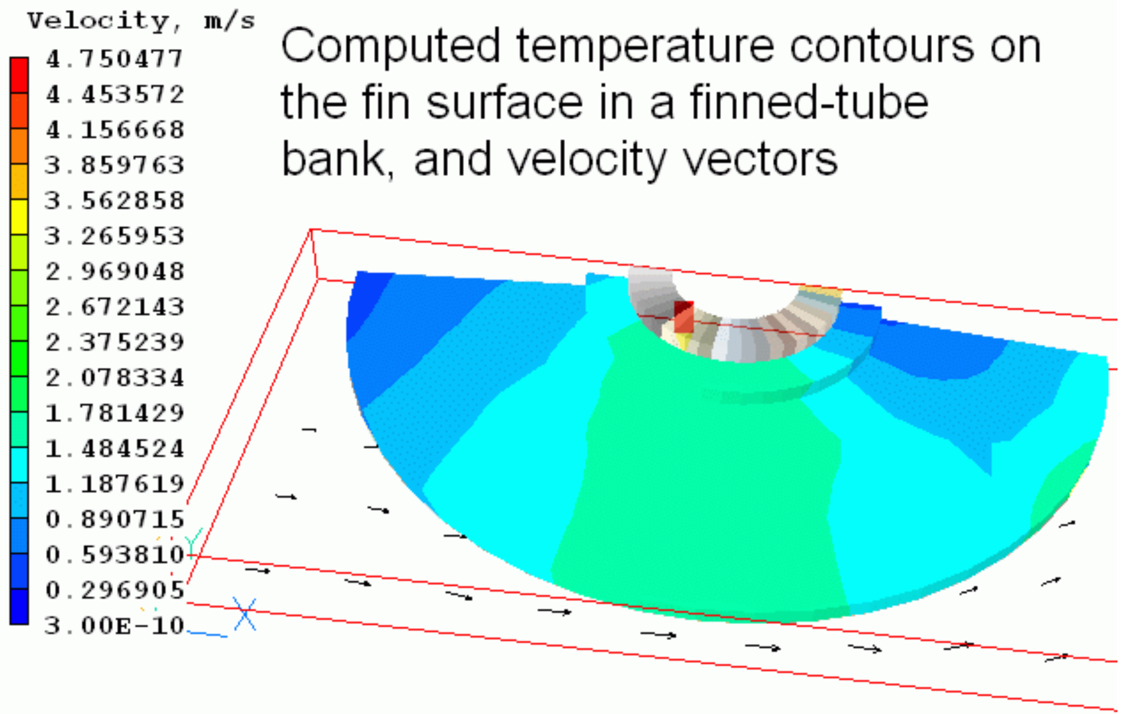
In the future it will not need to do so; for Detailed-Geometry CFD, applied to a few-tube segment of the bundle, will **compute** friction and heat-transfer coefficients **for the precise conditions in question**.

Excerpt 3. Outline of a not-yet-existing but wholly practicable Internet service:

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

- Log in to ‘Heat-Transfer-On-Line’. Click on ‘finned-tube bundle’.
- Select or type in the material, geometric and boundary-condition information which concerns you.
- Click submit.
- Within a few seconds you should receive a stream of graphical and alpha-numeric files, in a form your eye can enjoy and your computer software can employ in its own computations.



Computed temperature contours on the fin surface in a finned-tube bank, and velocity vectors

Here is an example.

Excerpt 4. Why DGCFD predictions will be better than any handbook

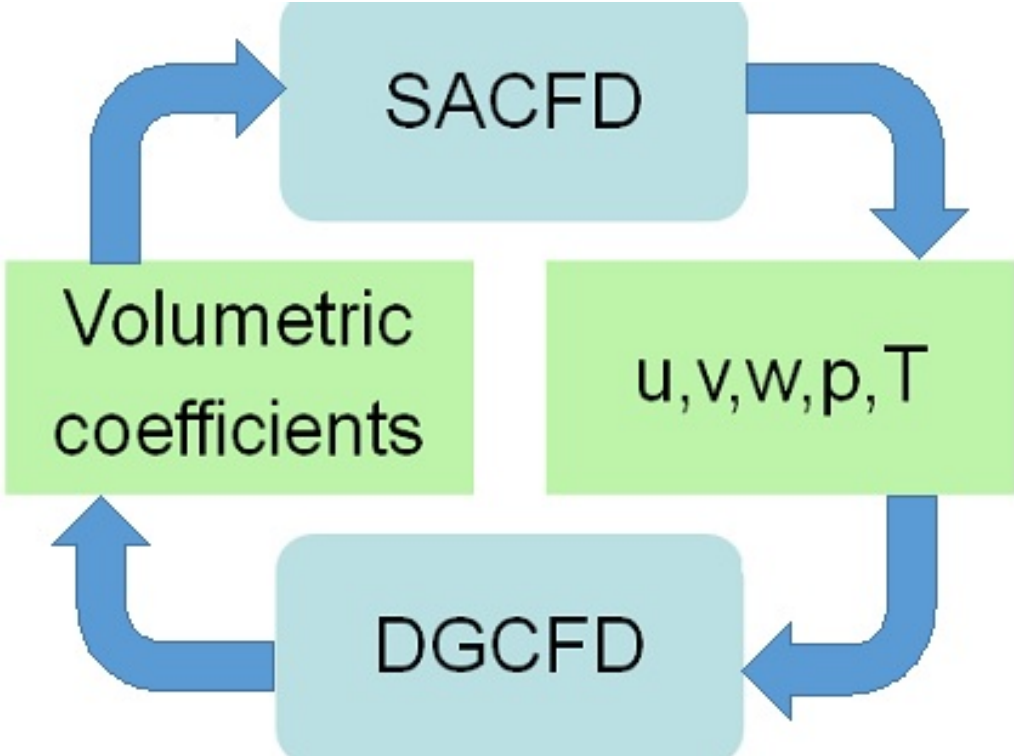
- Despite the **doubt about all turbulent–flow** predictions, finned-tube flows are **near-wall** ones about which doubt is least.
- The **detailed fin shape** can be imported from a CAD file; so even small-scale corrugations can be captured.
- Because only a few-tube segment is considered, the grid can be fine enough to **minimise numerical inaccuracy**.
- The grid-fineness will also allow the fluid and metal **properties to vary** through the integration domain under the influence of **temperature**, as they do in practice.
- The considered approach-flow directions can be arbitrarily **oblique** to the tube axes, with effects handbooks do not even mention.
- Computer times will be small enough to provide the **linearised expressions** which, rather than single-situation constants, are needed for building into SACFD models.

Present lecture resumed:
What advances since 2011?

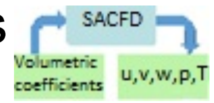
CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

There is a new concept: **Connected Multi-Run**, *i.e.* CMR.
In the heat-exchanger context, it involves execution of SACFD and DGCFD runs **in succession**, thus:

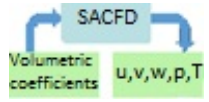


1. SACF **e**, outputs **mean v** oints.



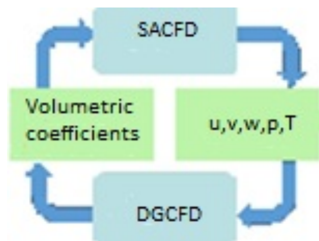
2. DGCF **ures** and outp **rmulae**.

3. SACF improved mean **ve** ints.



4. DGCF **eratures** and outp e.

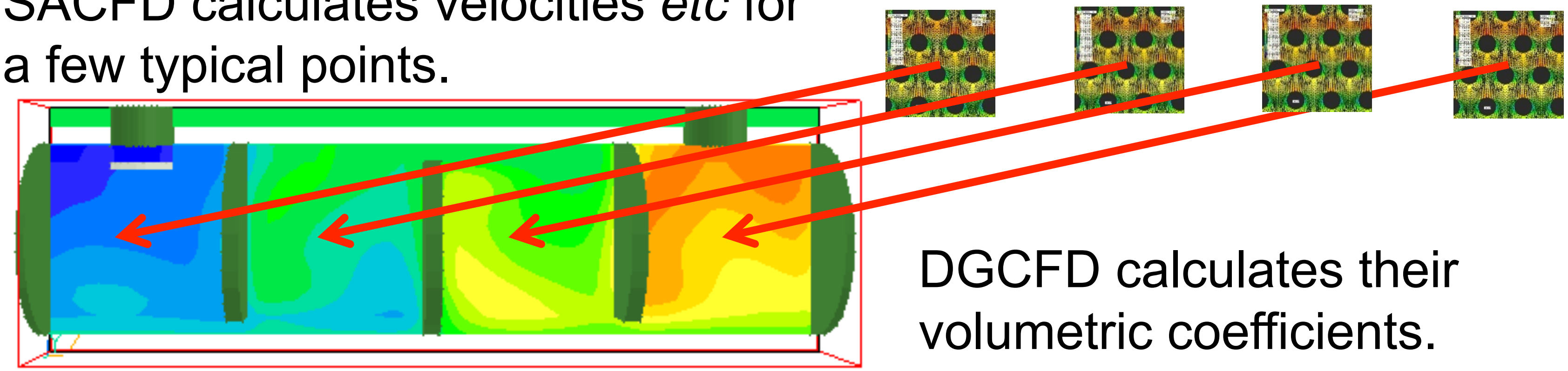
5. And s
The idea



So it is being **automated** in a **CMR SimScene** for all.

SACFD and DGCFD combined in Connected Multi-Runs

SACFD calculates velocities *etc* for a few typical points.



DGCFD calculates their volumetric coefficients.

There are many possible **variants**: number of points selected; frequency of information exchange; exchanged-information format.

It might be said that **CMR is a new kind of CFD**, still scarcely explored.

Thus heat-exchanger engineering, as well as profiting from CFD, is **contributing** also to **its development**.

CHT/CFD applied to heat-transfer equipment: **conclusions**

At long last, user interfaces are being created which allow design of all kinds of heat-exchange equipment to be based on physics rather than guess-work.

This is true of **two-phase** equipment (condensers, steam-generators, direct-contact cooling towers) as well as single-phase ones.

Experimental studies are still needed; but their main purpose in now **validation,** *i.e.* testing, and leading to improvements in **Simulation-Scenario packages (***aka* **apps)**.

New CFD techniques (multiple inter-linked grids, connected multi-runs) are of assistance; but the user **need not** know about them. The SimScene activates them **automatically**.

General-Purpose computer codes (PHOENICS, Fluent, Open-Foam) are now just CFD engines, **under the hood.** Their **drivers** are the **Apps.** **End of part 1 .**

Part 2: Suhas at Imperial College

This is how he looked in 1964.

It was from him that I first heard the word 'guru'. **I**, he told me, was **his** guru.

Only later, Wikipedia showed me how I appeared to him;

and that he should have told me that he was my **shishya**; and could assure me that he:

- was "not attached to any impermanent thing";
- had "renounced the desire for a son";
- had "no desire for wealth"; and
- was "at at peace with himself".

This is how it **should** have been.

Nevertheless our guru-shishya relationship continued successfully for many years.

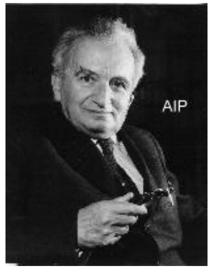
Prehistory: How we stumbled on the Finite-Volume Method

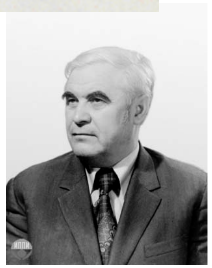
CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

When Suhas arrived, I was using **boundary-layer theory**, for heat and mass transfer to solid and liquid surfaces. My own PhD had concerned the combustion of liquid fuels.

The **prediction method** which I adopted was that used by **von Karman** (1921) for aerodynamic friction,





and by **Kruzhilin** (1936) for heat transfer.

This integral-profile method involved:

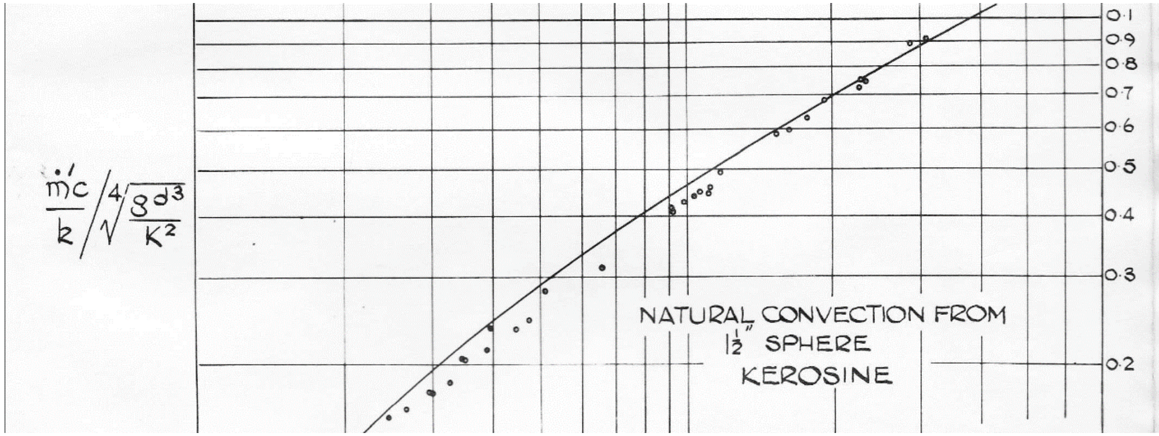
- **assuming a formula** for the **profile** of velocity or temperature;
- **determining its free parameters** from the boundary conditions;
- multiplying the partial differential equations by arbitrary **weighting factors,** before integration to form ordinary differential equations;
- solving these numerically to deduce **momentum** and **energy** thicknesses.

The questions: What profiles? What weighting functions?

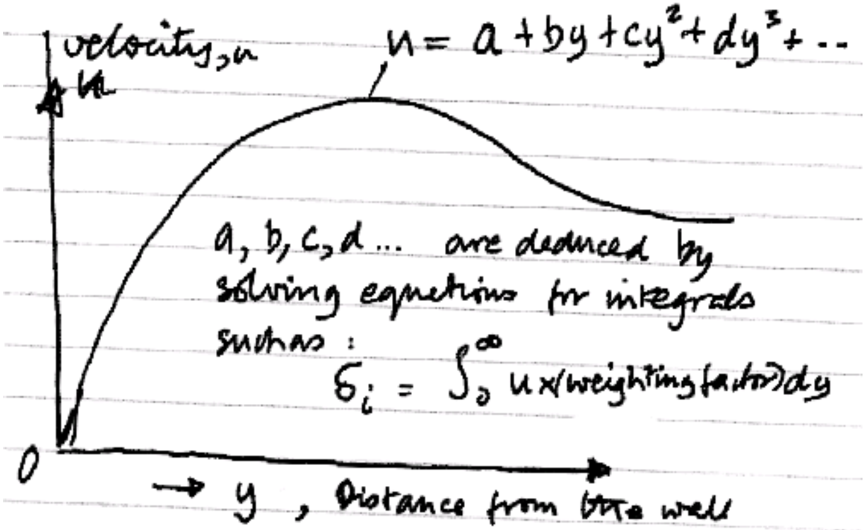
CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

It sometimes worked quite well, as shown here, by dimensionless **burning rate** *vs.* ‘**transfer number**’.



However, my students and I made systematic efforts to invent **more flexible formulae** for the **profiles** of velocity, temperature and concentration.



We also developed more-elaborate ‘**weighting functions**’ by which the equations were **multiplied before integration**.

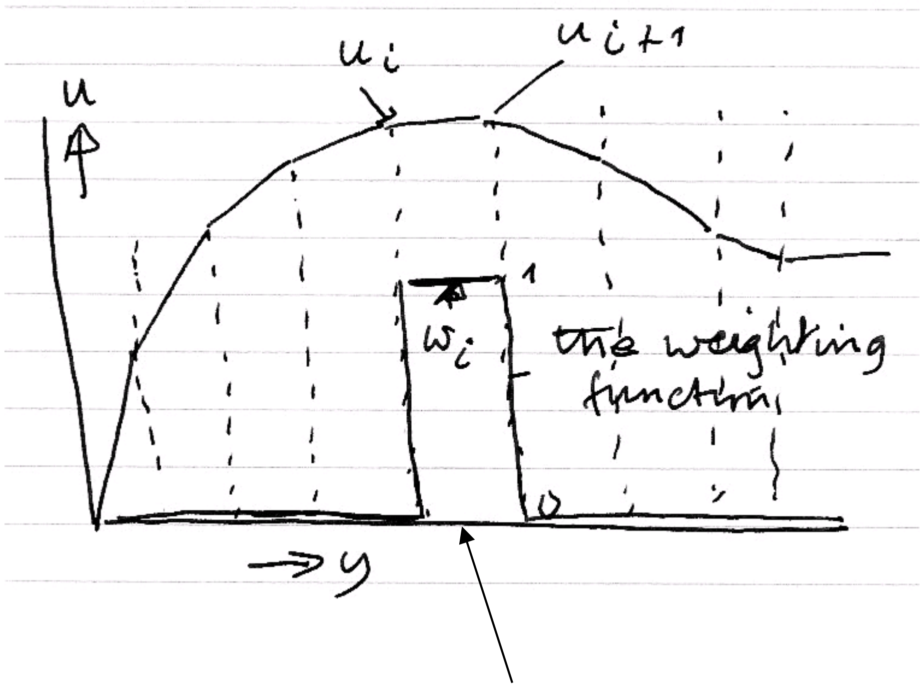
A lucky speculation

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

Then, by good fortune, I expressed during a lecture the following speculation:

- that probably the **most flexible** formula of all would be a **piece-wise linear one**;
- that if the **widths** of the 'pieces' were the limits of the integration, to calculate their **heights** would be easy, because



- the weighting functions could then be **extremely simple**, *viz.* **unity.**

The finite-volume method emerges

Why was this lucky?

Because, Suhas Patankar was in the audience;
and **listening!**

The next day he told me that he had written, overnight, a small Fortran program which embodied my suggestions;

and they **seemed to work**!

From that day on, he and I, and soon my other students (Akshai Runchal and Micha Wolfshtein among them) entirely abandoned integral-profile methods.

The finite-volume method (**our** version, that is) had lurched clumsily into life!

But that was not all!

- Our first FVM application was the **2D boundary layer**. The **computer program**, soon published, enabled researchers at Imperial College and elsewhere to test **turbulence models** for wakes, jets and other **parabolic** flows.
- The FVM was just as useful for **elliptic flows**. Runchal and Wolfshtein used it for the **vorticity~stream-function** equations of 2D flows. Whereafter, Caretto, Tatchell, Gosman and I began to solve for **pressure** and **velocities** directly.
- I had developed the **SIVA** (= Simultaneous Variable Adjustment) method for this. But this was swept into near-oblivion when Suhas returned in a post-doctoral capacity in 1971. Soon was unleashed the '**SIMPLE** Tsunami'.
- We both co-authored the publication, but it was **his** perceptive study of the works of Harlow and Chorin which led him to propose the **Semi-Implicit Method for Pressure-Linked Equations**, which the world soon adopted. End of Part2.

Part 3 .
Algorithmic novelties

The final service which Suhas did me was…

to move to Waterloo.

The consequence was that I had to become a programmer myself!

He soon replaced **SIMPLE** with **SIMPLER**; I went a different way with **SIMPLEST**, then on to **IPSA** for coupled Navier-Stokes equations, to **PARSOL** and to **simultaneous solid stress**.

And there were new computer programs too, starting with **GENMIX** and leading to **PHOENICS**.

All used, of course, the Finite-Volume ;and the **segregated-solution** procedure introduced by SIMPLE.

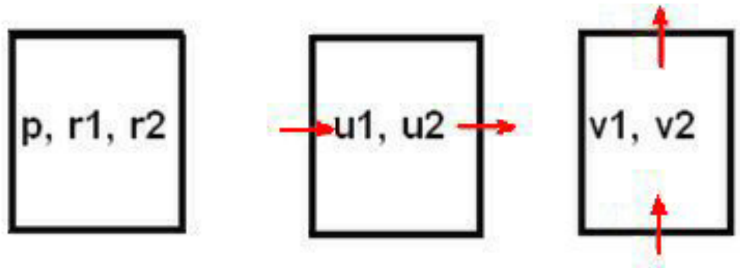
But the contest **SIVA** *vs.* **SIMPLE**, *i.e.* Simultaneous *vs.* Segregated, was not yet over.
Often they can best be **combined** as we shall see.

CHT for heat-exchanger design; past, present and future

IPSA (Interphase Slip Algorithm) has elements of both SIMPLE and SIVA

CHT15
Rutgers U
May, 2015

IPSA solve 9 fields of variables: p, r1, r2, u1, u2, v1, v2, w1 and w2 in SIMPLE-like segregated manner.



But frictional interactions between u1 and u2, v1 and v2,*etc.* are handled by SIVA-like **partial-elimination**.

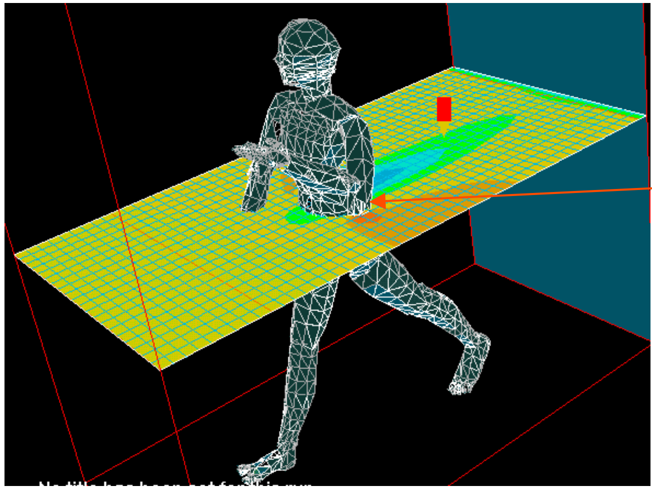
It is **2-phase** IPSA which is built into PHOENICS; but in a geyser water can flow up and down at the same location. What then?

One way: treat upward and downward-flowing water as **separate phases**. So now there are 3.

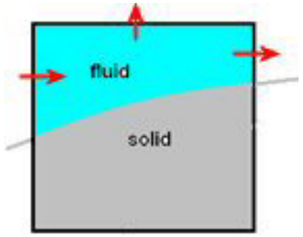
How to solve? Use more **segregation**; and further **partial elimination**.

The principle solves many problems; *e.g.* …

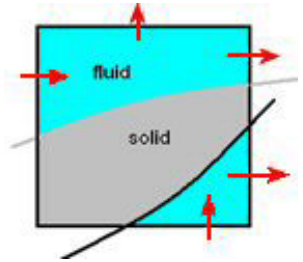
Segregation applied to PARSOL (partially solid cells)



PARSOL treats Cartesian cells cut by curved surfaces thus -->



But what about doubly-cut cells like this, with **two velocities** for the right wall?



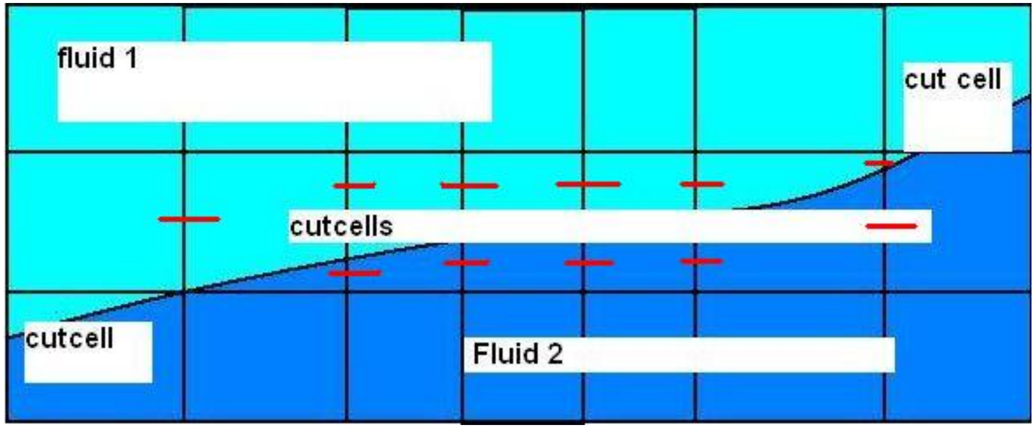
The obvious (seen first in 2014!) answer is: **segregate** a bit more, *i.e.* solve **upper** velocities on **even** sweeps and **lower** on **odd** ones.

It works.

For **conjugate heat transfer**, there are **three** temperatures per cell: 2 fluid and one solid. SIVA-type partial elimination solves for them between SIMPLE type field-wise solver sweeps. But the odd-even trick still works.

Another application of odd-even segregation: free-surface flows

Free-surface flows are often seen as **volume-of-fluid** problems; yet they are **more challenging**; for the fluids may cross the cut-cell walls with **two very different velocities**.



How solve with a code which can compute only one value for each cell? By **segregation** again. Attend to the **above-surface** and **below-surface** velocities at cut-cell faces respectively during **odd- and even-number visits** to the solver.

And make SIVA-type adjustments to the not-attended-to variable **between** visits.

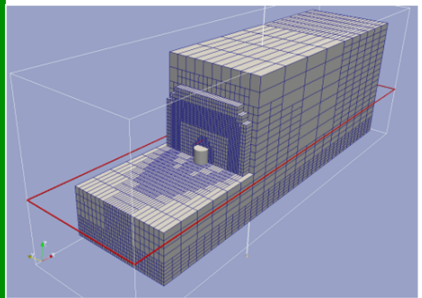
The positions of the facets defining the **surface must be updated** to fit each new velocity field; but that needs only careful interpolation.

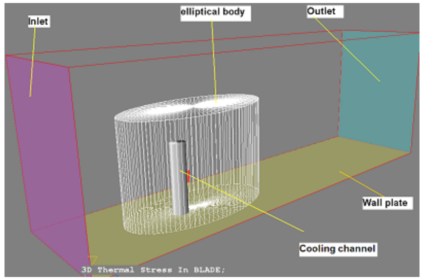
A last example; simultaneous fluid flow and solid stress for cut-cell geometries

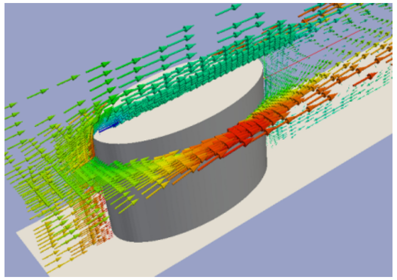
CHT15
Rutgers U
May, 2015

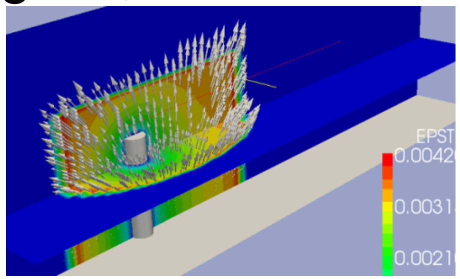
The capability of PHOENICS to compute **stresses in solids** simultaneously with **fluid flow** is rarely exercised because:

- some people still believe solid stresses **need** finite **elements**; and
- curved surfaces seemed to require unstructured grids, as shown:



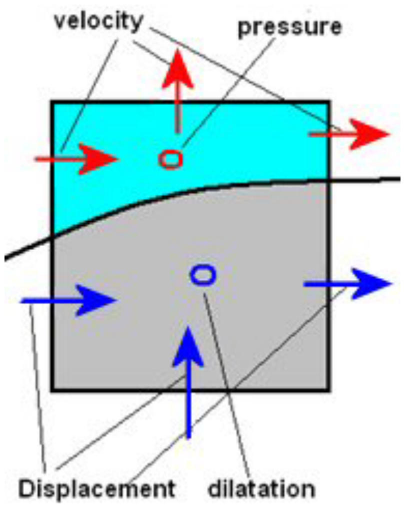






In solids, velocities are replaced by displacements and pressure by dilatation**. Were PARSOL** to be used, some cells would have both sets. But why not? They are not unlike fluid-surface cells.

Odd-even segregation would alternate **red** with **blue** solutions, with SIVA-like links between them.



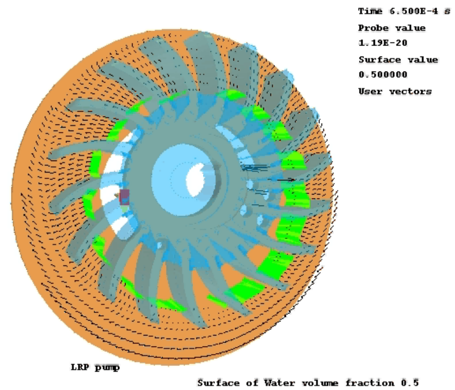
The last slide: putting it all together:
Esoteric + SimScene = Accessible

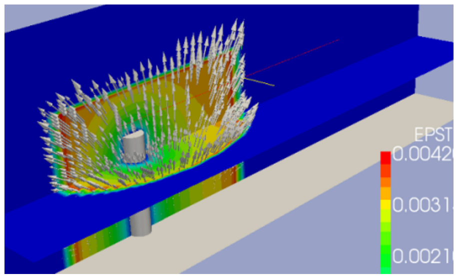
CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design; past, present and future

Heat-exchanger SimScenes enable designers to use CHT/CFD; so their equipment performs as predicted; and is cheaper to build.

A **free-surface** SimScene will be able to activate the odd-even segregation algorithm, the details of which its user needs know nothing, as in the **liquid-ring pump** on the right (click to activate).





A **Fluid-Structure-Interaction** SimScene will predict stresses in curved bodies with Cartesian Grids.

And a **multi-phase IPSA** SimScene will put SIVA and SIMPLE together, behind the scenes, to simulate the Old Faithful geyser.

The End

CHT15
Rutgers U
May, 2015

CHT for heat-exchanger design;
past, present and future

Thank you for your attention