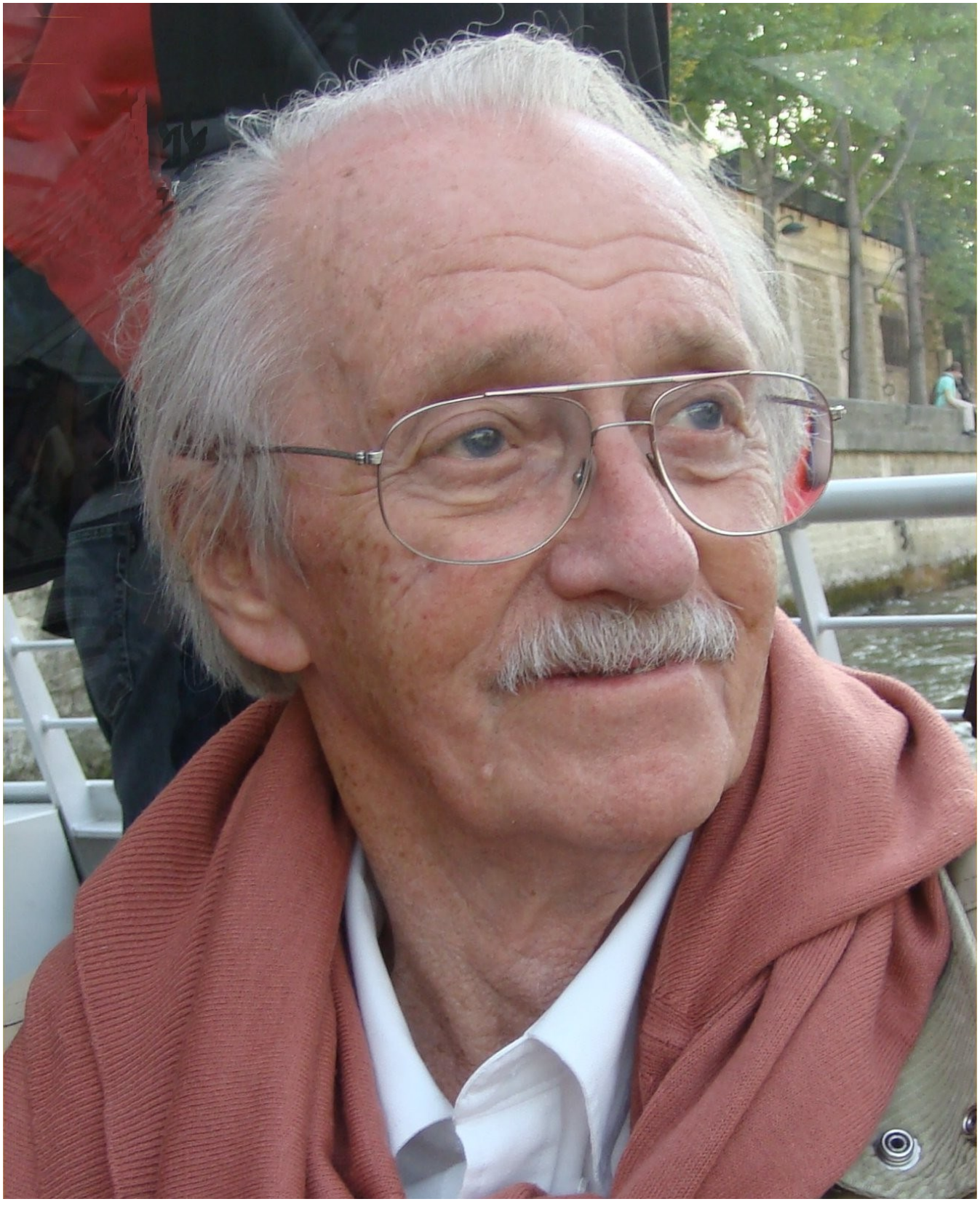
PHOENICS NEWS

CHAM

PHOENICS – Empowering Engineers Winter 2016



Professor Brian Spalding FRS, FREng

09/01/1923—27/11/2016

Professor Brian Spalding died, unexpectedly, after a short illness in November 2016.

Despite his 93 years he was still passionate about CFD, working on new ideas (including his beloved Population Model) and, when not lecturing or working abroad, at CHAM seven days a week.

He travelled to Moscow to meet with his Research Group, became ill and, despite best efforts by the medical profession in Russia, failed to recover.

We, his family and colleagues, are most appreciative of the number of people from the scientific community who have been in contact to express their sense of loss at Brian's death—thank you all.

We, at CHAM, are doing what we are pretty sure Brian would want. We are moving forward, carrying out the work he had embarked upon, launching new products that he had been involved with and ensuring that CHAM keeps his name, and his not inconsiderable reputation, alive.

There will be a gathering in London, to celebrate his life and his memory, later this year. Details will be circulated by email, and put on the CHAM website, in due course and all who would like to attend will be warmly welcomed. If any one would like to share their remembrances of Brian at this event please get in touch.

Brian was scheduled to give a keynote lecture at CHT-2017 on Advances in Computational Heat Transfer in Naples May 28—June 2 2017. There will now be a session to remember him as a scientist and as a person. Details can be found at http://www.ichmt.org/site/4/cht-17.

It is expected that many of those present will speak of Brian's contribution to Computational Fluid Dynamics (and perhaps his contributions to Combustion, Heat Transfer, Fluid Flow, Turbulence Models and other aspects of the science he loved).

It is planned to set up a memorial fund in Brian's name to support students who want to enter the fields of computational fluid dynamics and engineering. More details will be provided in due course.

The last two lectures Brian delivered were at ETTM in Sicily (September 2016) and an OpenFoam Meeting in Cologne (October 2016). Extracts are contained below.

Those of us working at CHAM say a fond, and sad, farewell to our founder and inspiration. He is, and will be, much missed.

"Turbulent-chemical-reaction models: old and new; and which way forward?"

Brian Spalding

Ercoftac ETMM11, Palermo, Sicily, September 2016 (Absract)

Background

Much of the world's power is generated by way of furnaces and engine combustion chambers; and many of its materials are created in large paddle-stirred chemical reactors. The flow in all these equipment types is invariably turbulent. Yet, of current research on turbulence, the proportion devoted to chemically-reacting flows is very small.

Probably this is because such research has little success to proclaim; and the cause of that, the author suggests, is that it has been restricted to the employment of models of the kind introduced by Kolmogorov three-quarters of a century ago. That seeks to describe turbulent flows by way of time-average distributions of statistical quantities such as energy *k,* dissipation rate *e,* Reynolds stresses such as *u'v'*, concentration fluctuations c'^2_i *etc*. These distributions are hypothesised as being governed by partial differential equations, which, when discretised in respect of space and time, can be solved numerically.

The reason for the 'little success' is that chemical reaction rates depend on time-average values of more complex statistical quantities such as $rc_ic_jT^n$; and no-one is ever likely to be bold enough to propose partial differential equations for them. Therefore researchers' interest in the subject has understandably dwindled.

There is however a quite different approach to turbulence; and one which is at least as successful for turbulent chemical reaction as the Kolmogorov one is for the hydrodynamics. It treats a turbulent fluid as a discretised population; and its hypotheses concern the interactions between the members of that population

The discretised-population approach (called DP in what follows) is also not new. It was used by Smoluchowski a whole century ago to quantify the implications of Einstein's theory of Brownian motion for the composition of colloidal dispersions. Its use for turbulent combustion first made its appearance in 1973 as the 'eddy-break-up model' (*i.e.* EBU); for DP terms EBU can be described as involving a two-member population. Not until twenty-five years later, however, was it followed by its four-member, fourteen-member and finally multi-member successors. Sometimes the pace of science is glacial.

The two members postulated for the EBU population were: (1) a wholly unburned fuel-air mixture: and (2) its fully-burned counterpart of the same enthalpy and atomic composition. The composition of fluid present in all parts of the domain differed only in the relative proportions of these two components. Typical present-day multi-member populations still use the extent-of-burnedness and atomic composition as member-distinguishing dimensions; these dimensions are now sub-divided into as many intervals as are needed for at least qualitative realism. CFD simulations based on DP modelling, therefore explore not only the four space-and-time dimensions, but also two more.

The present contribution.

This **multi-dimensional** aspect of the DP approach may appear daunting. Therefore a prime aim of the author's contribution to ETMM11 is to explain that, on the contrary, the computational task associated with DP can be less burdensome than that of the current practice. This comes about because:

- the two approaches can be used in combination, with Kolmogorov's only for velocities and DP only for chemical composition;
- the whole calculation can be conducted in steady-state mode, the more expensive LES for the hydrodynamics adding nothing to the overall realism;
- the linkages between the hydrodynamic and the chemical-reaction parts of the total model are simple, in that the former supplies the mixing rate as an *e/k* distribution;
- it then receives from the latter distributions of time-average density, temperature and composition;
- the DP calculation needs to be carried out only in those parts of the domain in which the time-average temperature is high and therefore the volumetric release rates of energy (good) and NOX and smoke (bad) require accurate calculation;
- attention to relatively few typical locations suffices, when the uncertainty of the underlying chemical-kinetics data is acknowledged.

These arguments will be explained in sufficient detail during the presentation to enable any proficient CFD specialist to implement them in their computer code. Further, illustrations will be provided which should persuade their combustor-designing colleagues to encourage them to do so, having concluded that the discretised population approach can already provide the guidance to design improvement which they need.

The latter is indeed the author's main aim; with air pollution and global warming among the main threats to humanity, any advances in turbulent-reacting-flow science should be exploited promptly by engineers.

CHAM

The Shape of CFDs to Come

Brian Spalding,

OpenFoam, Cologne, October 2016 (Extracts)

The Past: Early 1970s: open-source was the rule. Mid and late 1970s: semi- then fully-closed source. Industry recognised CFD's promise and sought assistance. CHAM (founded 1974) first provided client-specific codes; then, for QA and lower cost, variants of one General-Purpose code. Hence, 1981, part-open-source PHOENICS; whereafter closed-source Fluent, Star-CD, Fire, FIDAP, *etc.*

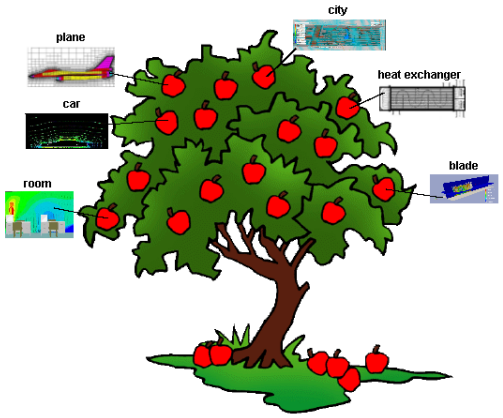
Up to the Present:

What's good: CFD is now widely used. Open-source practices have been revived (OpenFoam). New techniques have been developed, *eg* unstructured grids**.** Others are vanishing, *eg* finite-elements for fluids.

What's bad: Costly 'brute-force' (**one-grid-for-all**) methods prevail.

Much of industry still uses pre-CFD methods.

The most-widely-used codes have the fewest advanced features. What CFD can and cannot do is not widely understood.



The Future: Will general-purpose codes survive? Yes, but mainly out of sight. CFD-apps will apply CFD to classes of equipment, via application-specific menus (ie Simulation Scenarios or SimScenes).

App users will need know even less about CFD than apple-eaters about arboriculture.

CFD-apps will occupy attention. Apps and apples can be equally healthy if the tree-roots are well nourished…. by the underlying CFD code.

What are CFD-apps? And why will they prevail?

Why? General-purpose CFD codes simulate many classes of scenario; users need just one. To particularize a general-purpose code requires specialist skills which users can ill afford to learn.

What? A CFD-app is a one-scenario-class user- interface. Its creators provide the particularization.

CFD-apps ask only for inputs that users know about in application-specific language *eg* 'air-change/hour'.

CFD-apps create grids without user intervention; and set numerical parameters likewise.

CFD-apps supply results-displaying macros; and automatically write results-interpreting reports.

The three CFD-app communities: 1 Users 2 Creators 3 CFD-service providers

1) Users: choose the app; state their requirements; receive the results; enjoy benefits; pay the money; call the tune.
2) App creators: major innovators; speak 2 languages: users' & CFD's; receive most of the money.
3) CFD-service providers (ANSYS, ESI, CHAM*, etc):* supply simulation features which creators call for; crunch numbers at minimum cost; are paid in proportion to use.

What's the difference from today ? From the User's viewpoint:

Now: Users choose one service provider; pay significant money, get more than they need, which they may suppose to be the whole of CFD; but which is very often less than they need.

They must themselves create the grids; and make other numerical settings, optimal or not; run the code; display and interpret the results.

Future: Users choose the currently-needed app; pay less; and for no more than they need; do have access to the whole of CFD; rely on settings made by the app-creator for that application; but may use different creators for other applications.

From the app-creator's view-point :

1) His important middle-man role (*ie* facing both ways) is acknowledged and rewarded.
2) His special-application know how is the more important
3) But he must also know which general CFD code has the special features that his particular app needs, (no service provider's code possesses them all) *eg:*
   - o Parabolic-solution
   - o Data input *via* formulae,
   - o sub-divided Cartesian grids,
   - o PARSOL (*ie* objects embedded in structured grids),
   - o simultaneous solid-stress-solving option, *etc*

From the CFD-service provider's view-point :

1) What a relief! To concentrate on what he knows best, no longer trying to be all things to all men;
2) And a pleasure, to know that users of any apps may use his services …. so long as they are easily accessed; and competitively priced.;
3) To encourage use he documents his code-feature-activation protocols and publicises them widely.
4) And of course makes his software available *via* the 'cloud' on a pay-by-use basis.

CHAM