Status of Coupled Fluid-Structure Dynamics Code SEURBNUK

B.L. Smith  
*Swiss Federal Institute for Reactor Research, CH-5303 Würenlingen, Switzerland*

A. Yerkees  
*Commission of the European Communities, J.R.C. Ispra Establishment, Department A, I-21020 Ispra (Varese), Italy*

J. Adamson  
*UKAEA, Safety Studies Group, AEE Winfrith, Dorchester, Dorset DT2 8DH, U.K.*

Summary

The computer code SEURBNUK2 is used collaboratively for the study of fast reactor containment integrity. Continuous extension and improvement of the numerical modelling has been required to match the performance of the code against the COVA series of scale model experiments and the requirements of reactor safety analysis. This article outlines the present capabilities of SEURBNUK2 and summarises the most recent development topics.

For internal structures amenable to thin shell treatment, a recent addition to the code permits these to be perforated, which is useful in modelling dip-plates and above-core structures in the reactor. Representation of thick structures can now be provided by means of a logical interface to the finite element code EURDYN. The coding of this link includes an automatic element generator to streamline the structural data preparation.

The work hardening of structural materials has hitherto been represented by the mechanical sublayer model. Recently an alternative method has been provided - the isotropic strain hardening model. Results using both methods are compared.

In safety analysis much attention is paid to the response of the roof structure to impact loading from a rising coolant slug. The typical relationship between duration of the loading and the natural period of the roof shows that a coupled fluid/structure analysis is required. This must include the roof hold-down device which can introduce a low frequency component that considerably modifies the response of the closure system. A recent major extension to the SEURBNUK modelling is the installation of a moving roof option which, together with development of the logic to link structures external to the containment vessel, provides such coupling.

The motions of marker particles which define free surfaces are estimated by extrapolating from the neighbouring fluid velocity field. The choice of appropriate velocities close to a structure can be subjective, particularly at a free end or a corner, and this has often led to instability in the shape of the surface. A systematic method recently adopted to reduce this problem is described.

As the free fluid surface sweeps out through the cells, cells that were boundary cells change their status to become fluid cells. In this case care must be taken to initialise the unknown quantities, otherwise large errors can easily be introduced if the field variables are changing rapidly. Examples are given to illustrate this point.

An improvement in the portability of the code has been achieved by attending to Fortran protocol and non-standard features such as unassigned variables, generally modifying the program source in the direction of ANSI standard Fortran 77. This is consistent with the recently extended collaboration on development of the SEURBNUK code.

Validation relating to several aspects of the development has been pursued via the COVA experiments and is well reported. For the very recent development related to the flexible roof, although considerable further effort will be required, there are already on record a few successful comparisons with experiments.
1. Introduction

SEURBNUK-2 is a two-dimensional (axisymmetric) computer code capable of performing hydrodynamics/structural mechanics calculations to assist in the evaluation of fast reactor containment integrity. The reactor coolant is modelled as a compressible fluid and an implicit finite difference algorithm similar to ICE [1] is used to compute the fluid motions. A complementary structure calculation taking into account material and geometrical non-linearities is performed to simultaneously compute the deformations of the various internal plates and shells (which may be perforated) and the main tank. Representation of massive internal structures is provided by means of a link to the finite element code EURDYN [2]. Fluid/structure interaction effects are included.

In a typical reactor configuration related to a HCDA the fluid will enclose a gas bubble representing the expanding vapourised core products and the fluid upper surface will be in contact with the cover gas blanket. As the core bubble expands the reactor fluid is driven upwards and the cover gas gap may close sufficiently to bring the fluid in contact with the underside of the reactor roof. The response of the roof to the high impact pressures generated is computed along with the fluid pressurisation. The development of SEURBNUK-2 is shared internationally and progress has been reported regularly at previous SMIRT conferences [3,4,5]. Development topics are defined partly by the needs of the various European reactor safety analysts and partly to correct shortcomings in the modelling techniques that may be identified by the parallel COVA experimental validation programme [6]. Developments aim to extend code modelling capability by introducing new features or by updating existing coding to achieve greater reliability. Active liaison maintained between the safety analysis, experimental and theoretical groups ensures that the overall trend in code development is consistent with the general needs of the various European reactor safety programmes. Some of the more recent topics of SEURBNUK code development and validation which have arisen by these means are summarized in the following sections.

2. EURDYN link

The continued development of SEURBNUK-2 has enabled many of the complex flow situations expected to arise under HCDA conditions to be adequately handled by the code. In particular, the ability to model slug impact, flow round corners, through porous materials and through perforated plates enables quite sophisticated reactor simulations to be attempted.

The structure modelling capabilities of the code have likewise been improved by optionally replacing the finite difference thin shell model in SEURBNUK by the finite element structure code EURDYN [2]. The axisymmetric elements available in the EURDYN code suite are a thin shell element, a constant strain triangular element and an 8-node isoparametric element.

Early work on the coupling, reported by Staniforth and Yerkes [4] has now been followed by improvements in input/output facilities. In particular, an automatic node and element generator for thin shell elements has been introduced in order to streamline data preparation. This permits for example data prepared for use with the standard version of SEURBNUK to run with only minor modifications the coupled code SEURBNUK/EURDYN.

Figure 1 shows a selection of structure linkages available using SEURBNUK/EURDYN, thus it is possible to model quite complicated reactor configurations. Although the thin shell element can be connected to triangular and quadrilateral elements with a hinged joint, a rigid connection is more appropriate for most practical applications.

3. Recent options added to the finite difference thin shell model

The isotropic strain hardening model as used in EURDYN [2] has recently been introduced into the standard version of the code to serve as a useful alternative to the mechanical sublayer model since the CPU time can be substantially reduced if there are several sublayers and many shell elements. The results obtained using these two material work hardening models for the spherical cap problem [7] are practically identical (see Fig.2), confirming the conclusion reached by Hunsaker et al. [8] that both models are equivalent if no reverse loading occurs.

Mean stresses at mid-segment points are normally calculated using the value at the mean surface of the shell (membrane approximation) in order to save CPU time. In the plastic regime this result is not applicable and the mean stresses ought to be calculated by integration through the shell thickness. A further option in the code now allows both possibilities. Fig. 2 compares the results for both methods in the spherical cap problem, they are significantly different, the more accurate integration scheme giving results in better agreement with NONSAP and EURDYN1 predictions.

4. Perforated plates

Perforated plates are represented in SEURBNUK by means of a distributed resistance model. The plates must be completely immersed in the fluid, but can deform and be of arbitrary orientation with respect to the underlying Eulerian grid, see Smith [5] for more details.
A cell-averaged transient pressure drop formula has been adopted:

$$\Delta p = \rho \frac{\partial w_n}{\partial t} + \frac{1}{2} \rho |w_n|^2$$  \hspace{1cm} (1)

to relate the fluid states on the two sides of the plate: both inertial and frictional resistance effects are included. The quantity $w_n$ in the expression refers to the (superficial) fluid velocity relative to the plate in the normal direction. The user must supply the parameters $\xi, \lambda$ as input data. These are, respectively, the steady state hydraulic resistance coefficient, obtained for example from Iдельчык [9], and an equivalent inertial length determined from the plate geometry, see for example Smith [10].

The prevailing pressure drop not only drives the fluid through the perforations in accordance with eq.(1), but moves and deforms the plate, the average material properties of which should be adjusted to allow for the decrease in mass and strength caused by the perforations. The plate motion alters the driving pressure and this computation is carried out iteratively within the code to account for the feed-back effects.

An application of the model to the analysis of the 1/6th scale test of the SNR300 has been reported by Meier et al. [11]. The test configuration, shown in Fig. 3, consists of a hemispherical tank partially filled with water and containing an explosive charge. The dip-plate, diagrid and diagrid support skirt, indicated in the figure, are perforated structures.

5. Control of spurious deformation at free surfaces

Spurious deformation of the bubble in some COVA calculations has proved to be troublesome for the logic of the code, see Figs. 4a and 4b. In this particular case, a refinement of the mesh led to an even worse situation, see Fig. 4c. Investigations have revealed that some of these spurious deformations originate during the creation of new fluid cells formed when the bubble collapses. In the past, the code has used either linear extrapolation and/or the mass conservation equation (assuming no density change during the time step) to initialise unknown velocities and momenta, trial runs using simple zero order terms proved to be better.

An even more serious difficulty in the numerical scheme exists because some of the quantities used to initialise the new cell variables are themselves derived from linear extrapolation and/or the mass conservation equations during the calculation of momenta on the Eulerian bubble boundary. Normally, random errors can be tolerated because they tend to balance each other out as the solution evolves. However, systematic errors introduced during the creation of a new cell can accumulate during the creation of other fluid cells. The use of zero order terms to calculate momenta further improved the solution, see Fig. 4d. These investigations have demonstrated that, when the flow is non-uniform and changing rapidly both in time and space, the most simple schemes often prove to be the best. Note, however, that this simple scheme is inadequate during the initial expansion phase of a bubble which is small in relation to the underlying mesh [4], for this reason the code will now allow the user to change the method of extrapolation during a computation.

6. The motions of free surfaces

The motions of marker particles which define free surfaces are estimated by extrapolation from pairs of mesh velocities in the neighbouring velocity field. In order to determine which pairs of mesh velocities to use the programme examines the neighbourhood, defined as the cell containing the particle and the eight surrounding cells, and identifies the contents and relative positions of each cell. Assuming that the centre cell is subdivided into four quadrants and that a marker particle can occupy any quadrant, then four possible mesh velocity selections are considered. These choices were made manually in the past by the users, but this task has now been superseded by a systematic method. The steps in this procedure are briefly as follows:

1) identify the contents of each cell in the neighbourhood and check to see if a particle is less than half a cell distance from a wall cell;
2) scan the nine-cell neighbourhood in a specific order and identify one pair of grid velocities for each velocity component. Fig. 5a specifies the scanning order in the case of the radial component for a particle positioned in the quadrant indicated;
3) provided that the particle is not within half a cell distance from a wall cell (noted in step 1), scan a further six cells in another particular order to identify the second pair of grid velocities. Figs. 5b and 5c are examples of the scanning order for the radial component, in the first case the pair of radial velocities chosen in step 2 are in the central column, in the second case they are in the right hand column. The order in which the cells are scanned is principally decided on the grounds of proximity and the preference to limit the selection to the nine-cell neighbourhood. All possible combinations of the selections shown in Figs. 5a to 5c can easily be generated by applying the geometrical properties of reflection and inversion.
For marker particles within half a cell distance from a wall boundary only one pair of grid velocities is used if they are pointing in the same direction as the wall, this helps the marker particle to slip along a boundary and follow the flow around a corner or free-end without crossing the wall boundary. Of course, as with all discretisation techniques, there are bound to be anomalies in realising the desired effects for every situation, for this reason the code still retains the possibility of manual user intervention. Figs. 6a and 6b are examples to indicate which grid velocities would be selected using the automatic procedure.

7. Flexible roof

Much of the effort undertaken in HCDA containment analysis concerns the loading and response of the reactor cover following an impact by a rising coolant slug. Traditionally, roof loading and response studies have been decoupled. Scoping studies now suggest, however, that roof deflection during the impact process could alter significantly the characteristics of the roof loading, in particular the evaluation of the total impulse, see Smith [12]. As a consequence, SEURBNUK, originally programmed to deal only with impact on a rigid roof, has been extended to more general configurations of fluid impact on moving, deformable structures, and includes the effects of fluid/structure interaction. This proved to be a major task because the simple rigid roof logic was not readily extendable.

Figure 7a depicts a typical fluid-structure configuration just prior to fluid contact with the roof (considered rigid in this example). In the original treatment the horizontal roof line LM was restricted to lie on an Eulerian grid line. The fluid surface, described by a series of massless Lagrangian marker particles, is connected to the axis of symmetry by a sliding junction point, 'J', in the figure. In the example chosen, impact occurs first on the axis and is detected by the code when one or more of the fluid surface markers cross the roof datum line LM. When this occurs, the impacted markers are brought vertically downwards to take up position slightly below the roof line LM, see Fig. 7b, and the impacted cells are flagged (shaded in the figure). Further vertical movement of the impacted markers is prevented and impact pressures are calculated in the affected fluid cells using the rigid boundary condition rather than the free surface boundary condition.

In the new scheme the code explicitly recognises fluid/roof contact. The impacted markers are removed and the vertical sliding junction, 'J', originally on the axis of symmetry, is now situated on the roof and is defined as the intersection point between the fluid surface and the roof, see Fig. 7c. The sections LJ and JM are subjected to fluid and cover gas pressure loadings, respectively. For a movable roof these pressures are used to drive the roof motion in a simultaneous computation.

Although the description here is for a simple, domed impact in which the fluid hits first along the axis of symmetry, more general fluid/roof impact configurations have been coded.

8. SEURBNUK validation

The modelling of the hold-down arrangement of the roof has been found to be important in two basic systems studied so far. The first, shown in Fig. 8a, is a typical design for one of the COVA experiments [6]. Here the roof, cylindrical wall and base of the containment vessel are clamped together with several long studs passing through upper and lower clamp rings, the lower being firmly bolted down to a steel frame cast into a reinforced concrete inertia block (45 tons). A SEURBNUK model of this is shown in the line drawing of Fig. 8b in which the components are all flexible. The lower clamp ring is assumed to have a built-in end boundary condition at its outer circumference while the upper clamp ring is free. All the junction points marked 'X' are stiff joints, the stiffness being accomplished by bracing springs across adjacent limbs. Apart from enhancing realism the stiffening is necessary for stability of the calculation. Two obvious modes of vibration of the roof are: flexural vibrations of the roof with some boundary condition, and longitudinal vibrations on the clamping studs as of a mass on a spring.

Figure 9 shows the effect of modelling roof flexibility and hold-down for a roof structure consisting of a steel plate 150 mm thick together with a central plug 200 mm thick: the standard COVA roof assembly. In two of the calculations the hold-down was neglected, the roof being treated in one case as rigid and in the other as flexible with fixed edge. Little difference is seen between these; the roof is thick enough to be regarded as rigid, but the final impulse is excessive compared with the experimental result [14]. Modelling the hold-down brings the calculated final impulse quite close to the experimental value.

In the second comparison, Fig. 10, the roof is of aluminium of thickness 27.5 mm. Here, the flexural behaviour of the roof (which exhibits plasticity) dominates the structure response. It is seen that SEURBNUK reproduces the main features of the observed impulse history [14], especially the delay between the first and subsequent impacts. To show how important in producing this feature is the flexible roof modelling, a calculated result assuming a rigid roof is superimposed.

From the experience gained it is evident that further code development is required to represent the return of the coolant as it pulls back from the roof.
9. SEURBNUK portability

Considerable attention has been directed towards improving the image of the SEURBNUK program as regards code portability. One facet of this improvement has been to make the code conform as far as possible (given certain constraints in strategy) to the ANSI Fortran 77 standard [13]. The enhancement in portability is evidenced by the availability of virtually identical source programs for the virtual memory, 32-bit word computers, IBM, ICL and VAX. For a CDC machine with 60 bit word length some changes from double to single precision were warranted and program overlay was required to accommodate problems into available store. A second facet has been to identify and amend occurrences of unassigned variables and array bound violations. The end result is that the SEURBNUK program has been shown to solve a range of test problems on all four computers mentioned, with results which are the same except for rounding errors.

10. Concluding remarks

The recent developments described in this paper not only allow the extension of existing calculations but also have enhanced the structural capabilities of the code to include perforated plate and flexible roof modelling. Additions and modifications to SEURBNUK-2 to produce a more general code are continuously being made. In the near future, for example, a more general treatment of free surfaces will be added, thus removing some of the restrictions which are at present illegal. The EURDYN link will be updated to include release 3 of EURDYN, this version has new features such as a full large strain capability for the 9-node isoparametric elements and the possibility of choosing the radial return algorithm for elasto-plastic material modelling.

The foreseen activities also include a feasibility study to determine if the inclusion of moving porous structures as a means of representing above-core structures, heat exchangers and pumps is practical. A more representative treatment of the cover gas is also desirable.

Although the original COVA experiments have been completed, further supplementary experiments (e.g. roof loading/response) continue to be realised, this is a necessary and complementary task to determine the validity of the numerical models.

11. References


Fig. 1 - A selection of structure linkages in SEURBNUK/EURDY.

![Diagram](image1)

Fig. 2 - Comparison of results for the spherical cap problem.

![Diagram](image2)

Fig. 3 - Application of the perforated plate model.
Fig. 4 - Examples of spurious bubble deformation.

Fig. 5 - Examples of the hierarchy adopted in the grid velocity selection.

Fig. 6 - Examples of grid velocity selections.

Fig. 7 - Comparison of the SEURBNUK impact models.
Fig. 8 - Schematic and SEURBNUK model of containment and hold-down.

Fig. 9 - Roof impulse - effect of hold-down response.

Fig. 10 - Impulse on thin aluminium roof.