Evolution of mesh refinement rules for impact dynamics

Daniel Howard
Software Evolution Centre,
Software & Systems Engineering Centre,
Defence Evaluation & Research Agency,
Malvern, WORCS WR14 3PS, UK.
dhoward@dera.gov.uk

Simon C. Roberts
Software Evolution Centre,
Software & Systems Engineering Centre,
Defence Evaluation & Research Agency,
Malvern, WORCS WR14 3PS, UK.
scroberts@btinternet.com

Abstract-

Genetic programming (GP) was used in an experiment to investigate the possibility of learning rules that trigger adaptive mesh refinement. GP detected mesh cells that required refinement by evolving a formula involving cell quantities such as material densities. Various cell variable combinations were investigated in order to identify the optimal ones for indicating mesh refinement. The problem studied was the high speed impact of a spherical ball on a metal plate.

1 The problem of optimal mesh refinement

It is sometimes impossible or too costly to carry out physical experiments to investigate an engineering design. Instead, fast computers can undertake a computational experiment, or simulation of the Physics of a real experiment, which implements a numerical method to approximate the solution of the partial differential equations (PDEs) that describe the physical situation.

Usually PDEs are numerically solved with the Weighted Residuals Method (WRM) [3]. Examples of WRM are the Finite Element Method (FEM) [1] and the Finite Difference Method (FD) [2]. These divide the geometry of the experiment into a number of points or cells that are collectively known as a 'grid' or a 'mesh'. Over these points the PDE derivatives are defined and integrated using approximating functions. The WRM approximation should converge to the analytical solution of the PDE as the spacing between grid points decreases and also as the order of the functions increases (h-p method).

By definition the WRM requires a choice for its 'test' or 'weighting' function and a choice for its 'trial' or 'shape' functions. However, the most significant choice is whether the test function itself is symmetric or skewed and this is completely determined by the type of PDE that needs to be solved.

The PDEs that describe structural engineering problems are elliptic and contain even order derivative terms. With these, WRMs with a symmetric test function possess a mathematical property that ensures that they converge in an 'optimal' fashion to the analytical solution. Examples of these WRMs are the FD with central differencing and the Galerkin FEM. Under some assumptions the Galerkin FEM can be shown to be equivalent to the Ritz method [5]. The approximation is called 'optimal' as it equates to the minimization of a quadratic functional - the Hessian matrix H in the system of FEM equations $Hx-b=0$ is positive definite, making the FEM approximation x a global minimiser of the functional $f=\frac{1}{2}x^THx-b^Tx+c$.

The structural engineering PDEs tend to have smooth solutions. And so consequently, the WRM that discretises the domain into a number of equally spaced points, the 'uniform grid' or 'uniform mesh', converges to an accurate result.

Uniform grids, however, are not appropriate for PDEs that describe Fluid Dynamics, Heat Transfer, and Combustion because these PDEs contain first order derivatives. The equivalence of the Galerkin FEM and the Ritz method breaks down for these problems as the Hessian H is no longer positive definite. Two strategies exist to alleviate this problem: (a) refining the grid by reducing the distance between the grid points in places where the solution gradient is large - to reduce the contribution of the discrete advection terms to H, and/or (b) skewing the test function along characteristic directions, streak lines or streamlines, to recover a more elliptic system. Failure to adopt either or sometimes both strategies results in unstable and inaccurate solutions.

Strategy (b) presents mathematical difficulties. Though we know how to construct the optimal test function for a linear ODE case (1D), i.e. the Hemker function for the steady-state convection-diffusion equation, an optimal test function for the linear PDE case, i.e. 2D, 3D, is not a tensor product of the 1D function. Nor in CFD, save for trivial examples, is it is a function skewed along a streamline. As showed by Morton [4], the optimal test function is extremely complex. FD practicing engineers have never used it confusing the issue with discussion of truncation error. FEM schemes, e.g. SUPG [6], are equally sub-optimal in general. Solving the PDEs with a sub-optimal test function means solving the wrong PDE. Usually it gives a solution which is deceptively smooth [8] losing the power of prediction, i.e. a numerical result will not match an experimental result.

Strategy (a) refines the grid selectively and is usually implemented iteratively. For example, the transport terms in the Navier-Stokes equations are usually linearized via a Picard iteration or Newton method, and a grid refinement rule is evaluated at each iteration based on current gradients. In time-transient PDEs - a pseudo time stepping explicit method, or

a time stepping implicit method - the mesh refinement rule can be embedded in the time stepping.

Strategy (a) without strategy (b) requires excessive mesh refinement to control sharp boundary layers. And for problems that involve shocks, strategy (b) must be in force if we wish to get any result at all, and is popularly referred to as an artificial viscosity scheme.

On some PDEs with certain boundary conditions the grid may need to be refined gradually - the cell size should not change abruptly - or may need to avoid distorting the shape of the computational cells as this could promote wave reflections and introduce artificial numerical effects. It is also usually required to design the grid for computational efficiency, i.e. the implementation minimizes indirect memory addressing, and/or inter process communication for solution of the PDEs on parallel supercomputers.

2 Discovery of mesh refinement rule with Genetic Programming

A grid refinement rule [9] compares the magnitude of the gradients in the current solution to the local grid spacing and decides how and where to refine the mesh: either in the next iterative step, or perhaps refine the grid and re-compute the current step. It may also de-refine a grid, i.e. remove grid points or move them farther apart.

The grid is called 'structured' when grid cells are of the same geometry (e.g. quadrilateral, triangular) and each cell is always surrounded by the same number of neighbouring cells. And a grid is 'unstructured' when cells have various numbers of neighbouring cells. However, a structured grid of square cells can be locally refined in a nested fashion by sub-dividing square cells into internal square cells, and the resulting mesh is 'unstructured' as described in reference [9].

Popular grid refinement rules are not sophisticated. They compare the current localised solution gradient in a cell to a threshold to decide whether to refine or de-refine. Grid refinement rules are ideal candidates for optimisation with an evolutionary method. Rules need to be more sophisticated to address the following list which is not exhaustive: (a) computational efficiency, (b) non-linear continuation and accuracy. In the rest of the paper Genetic Programming (GP) is used to discover a rule that satisfies a modest example objective from this list.

3 Test Problem

Numerical solution of a system of non-linear time-transient and axi-symmetric PDEs, simulates the experiment whereby a tungsten ball impacts on a steel plate at great speed. Its solution for a number of time steps was approximated on the Cray supercomputer using a FD WRM with a uniform mesh of 60 by 60 square cells. The time integration scheme produced visual output after each time-frame, i.e. a *time-frame* is a plot of results on the 3600 cells in the mesh after a given number of time steps, see Fig. 6.

We explored the idea of asking the investigator (physicist, engineer) to judge the suitability of the grid at a given time step, i.e. to produce a 'truth' of grid cells that merit further refinement for the numerical solution at the following time frame. These cells lie in areas where important physical phenomena, e.g. shocks or tensile waves, are recognised by the expert, as opposed to areas where numerical phenomena, e.g. internal wave reflections at cell interfaces or noisy boundary conditions creating numerical waves, are evident.

The experiments discussed in the rest of the paper required that Genetic Programming reverse engineer a grid refinement rule that should be more elaborate than the usual grid refinement rule based on a local density gradient [9], as it accounts for the bias of a human expert.

The PDE system of momentum and energy equations is described in reference [9] and involves a quantity set, q_s for the three materials corresponding to air, ball, and plate. For example, densities for the three materials are denoted by the symbols ρ_a, ρ_b and ρ_p.

The pressure plots for the first six *time-frames* were shown to our expert who quickly identified a number of physical phenomena, e.g. shocks and tensile waves, together with other features that amounted to numerical noise. Our expert helped us to label the areas of the mesh, the cells, where the interesting physical phenomena occured for each of the six time-frames. We considered whether a general purpose evolutionary technique such as GP could discover the rule or relationship between gradients of the nodal values of density, velocity and internal energy that could mark the cells containing the interesting features as candidate cells for mesh refinement, as had been indicated by our expert. The rule as evolved with training data from a 'training' time-stage would also need to be general enough to predict the marked cells at a 'test' time-frame as explained in section 3.2.

3.1 Parameters for Genetic Programming

Details of our steady-state GP implementation are summarized in Table 1. Tournament selection was used to select an individual with a relatively low fitness for replacement (a kill), and to select one or two individuals with relatively high fitness for regeneration (breeding). The regeneration operators were randomly selected as prescribed by set probabilistic odds. Cross-over involved combining two randomly selected sub-trees, one from each parent individual. Mutation involved randomly selecting a constant terminal in a single parent and randomly resetting its value.

Two types of terminal input were devised in order to enable GP trees to process time-frame data directly: half-radial gradients (HRGs) and full-radial gradients (FRGs). An HRG was the absolute difference between a quantity in a cell and the same quantity in an adjacent or diagonal neighbour. An FRG was the absolute difference between a quantity in a cell neighbouring a given cell and the same quantity in the neighbour on the opposite side of the given cell, thus be-

Parameter	Description
Terminals	integers (-128 to 127); +ve reals (0.005 to 1.0 in 0.005 steps); -ve reals (-1.0 to -0.005 in 0.005 steps).
Functions	min, max, +, -, *, protected division.
Regeneration	95% cross-over; 5% mutation of cons.
Fitness	figure of merit (FOM - see text).
Population	5000
Max nodes	1000
Kill tour.	size 2: for steady-state GP,
Breed tour.	size 4: for steady-state GP.
Max gens.	20

Table 1: GP parameters.

ing similar to an HRG but involving a distance of two cells. Therefore, a cell was associated with 8 HRGs and 4 FRGs. However, these gradients were not used directly as terminal inputs in order to avoid problems regarding rotation invariance. Instead, the statistics for each type of gradient shown in Table 2 were defined as terminal inputs.

Gradient type	Statistic	Symbol
HRG	average	HA
HRG	standard deviation	HS
HRG	minimum	HL
HRG	maximum	HH
FRG	average	FA
FRG	standard deviation	FS
FRG	minimum	FL
FRG	maximum	FH

Table 2: Terminal inputs based on gradient statistics.

The cell quantities pertaining to individual materials (i.e. the densities and energies) were set to zero if the associated material was absent from a given cell. The terminal inputs accounted for this by setting a gradient to zero when either of the two associated cells had the relevant quantity equal to zero. For example, every cell in a time frame had a value for the density of the ball but only cells on the ball had a non-zero value for this quantity. Hence, if the gradients pertained to the density of the ball, a cell in the centre of the ball would be associated with 8 non-zero HRGs, a cell on the edge of the ball would be associated with less than with 8 non-zero HRGs and a cell away from the ball would not be associated with any non-zero HRGs. Therefore, the gradient statistics were only calculated from non-zero gradients and were themselves set to zero in the event of all gradients being zero.

3.2 Training procedure

All of the experiments discussed below used the same training procedure. Mesh pressure plots were manually scrutinised for a number of time-frames in order to judge which cells required refinement. The time-frames TF2 and TF3 were chosen as the training time-frame and the validation time-frame respectively. Table 3 gives the number of manually designated refinement cells which corresponded to each material in these time-frames.

Time-Frame	Ball	Plate	Overlap	Total
training (TF2)	38	46	3	81
validation (TF3)	52	67	9	110

Table 3: Number of refinement cells manually designated for the different materials in the training and validation time-frames. The overlap column gives the number of refinement cells containing the ball and the plate.

Training involved processing all the manually designated refinement cells in TF2 and 1500 non-refinement cells. The latter were chosen such that $x < 30$ (i.e. positioned in the left half of the time-frame which was where the impact was located) and such that each cell was at least one cell away from any refinement cell. Non-refinement cells which were neighbours to refinement cells (including diagonal neighbours) were termed as *limbo cells* because they represented 'indifferent' cells, and GP neither rewarded nor punished their detection.

Each tree output a real value r_{tc} in response to each training cell and the result for each training cell was classified according to the conditions in Table 4. For example, if $r_{tc} \geq 0$ and the training cell was a refinement cell then the result was deemed to be a *true positive* (TP). These classifications

Is $r_{tc} \geq 0$?	Refinement cell?	Class
true	true	true positive (TP)
true	false	false positive (FP)
false	true	false negative (FN)
false	false	true negative (TN)

Table 4: Result classifications depending on GP tree output and training cell type.

enabled each tree to be assigned a figure of merit (FOM) as,

$$\text{FOM} = \frac{N_{TP}}{N_{ref} + N_{FP}} \tag{1}$$

where N_{TP} and N_{FP} are the TP and FP counts over all training cells and N_{ref} is the total number of refinement cells in the training time-frame. Therefore, the minimum FOM was 0 which occured when a GP tree failed to identify any refinement cells, and the maximum FOM was 1 which occured when a tree successfully identified all refinement cells and did not detect any non-refinement cells as requiring refinement. (Note that the limbo cells did not influence the FOM.) Hence, the FOM was used as the fitness measure.

Quantity	Gradient type	FOM		N_{TP}		N_{FP}	
		av.	s.d.	av.	s.d.	av.	s.d.
ρ_a, ρ_b, ρ_p	HRG	0.746	0.039	96.9	6.4	19.9	7.2
ρ_a, ρ_b, ρ_p	FRG	0.701	0.046	93.7	7.4	23.9	11.1
ρ_a, ρ_b, ρ_p	HRG, FRG	0.778	0.044	98.0	8.2	15.8	6.0
I_a, I_b, I_p	HRG	0.593	0.069	92.0	4.2	46.8	16.2
I_a, I_b, I_p	FRG	0.616	0.026	86.5	3.1	30.6	4.6
I_a, I_b, I_p	HRG, FRG	0.573	0.072	88.4	4.1	46.3	16.9
v_r, v_z	HRG	0.511	0.031	80.1	4.4	46.8	4.8
v_r, v_z	FRG	0.504	0.019	84.8	4.6	58.3	7.3
v_r, v_z	HRG, FRG	0.513	0.041	82.3	5.6	50.9	7.9
ρ_a, ρ_b, ρ_p, I_a, I_b, I_p, v_r, v_z	HRG	0.731	0.079	100.1	2.8	28.9	18.5
ρ_a, ρ_b, ρ_p, I_a, I_b, I_p, v_r, v_z	FRG	0.721	0.079	910.5	6.0	29.4	14.4
ρ_a, ρ_b, ρ_p, I_a, I_b, I_p, v_r, v_z	HRG, FRG	0.707	0.078	100.4	4.9	33.9	19.3

Table 5: FOM, TP and FP results produced by processing the validation time-frame using various terminal inputs.

3.3 Detector performance

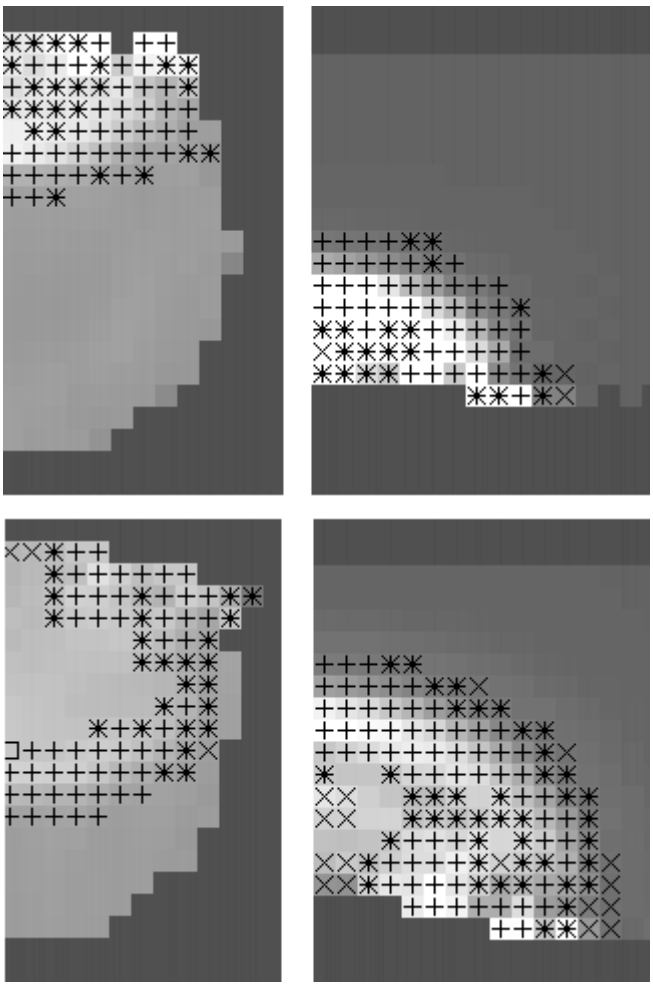
Various terminal input sets were investigated for detector evolution by running GP with 20 different randomiser seeds per set. The results produced by processing the validation time-frame with the fittest detector from the population for each seed are given in Table 5. The results were averaged over 20 randomiser seeds in each case. The densities ρ are in $1000kgm^{-3}$, and I_a, I_b, and I_p are internal energies of air, ball and plate; v_r and v_z are the radial and axial velocities respectively.

The table shows that the highest FOMs were achieved by basing terminal inputs on material densities alone. Terminal inputs based on material energies produced slightly greater FOMs than for inputs based on velocities. It can be seen that no benefit was gained from processing the densities, energies and velocities together - in fact, this produced less consistent results as shown by the relatively high FOM standard deviations.

The average FOM for inputs based on HRGs was typically greater than that for inputs based on FRGs, possibly because more information could be gained at material edges. For example, consider a cell located on a material edge such that 3 of its 8 neighbours are outside the material. For a material specific quantity, this cell would be associated with 5 non-zero HRGs but only a single non-zero FRG (due to the definition of these gradients).

3.4 Comparison to GP evolving a simple threshold

We compared the results in Table 5 with other results we will call exp_2 and reported in [11]. In exp_2 GP evolved a numerical threshold ϵ for the density gradient rule, i.e. $max[\Delta\rho] > \epsilon$. The comparison revealed that evolved detectors generally improved upon the FOMs gained using the ones evolved in exp_2. exp_2 results achieved a maximum FOM of 0.796 for the validation time frame (TF3), which was obtained by processing ρ_a, ρ_b and ρ_p.



(a) (b)

Figure 1: TF2 (top) and TF3 (bottom) results for best evolved detector for (a) ball and (b) plate. Symbols: + (TP), × (FP), square (FN), ⋆ (positive result on a limbo cell). Cell density is proportional to cell brightness.

A greater FOM resulted for 8 out of the 20 runs (with different randomiser seeds) when these densities were processed as HRGs and FRGs together. The maximum FOM in this case was 0.852 ($N_{TP} = 109$ and $N_{FP} = 18$) and the results for the corresponding detector are shown in Fig. 1. (The operation of this detector is analyzed in Section 3.5). Comparing these figures with similar ones for exp_2 in [11] shows that the FPs were largely a subset of those produced in exp_2, and that the two approaches had many positive returns on limbo cells in common.

Reference [11] explains that in exp_2 a maximum FOM of 0.802 could be gained by using individual gradient thresholds for different material densities. Therefore, detector evolution successfully produced greater FOMs than could be achieved by using optimum thresholds for different materials. We also note that the evolved detectors consistently produced much greater FOMs than the simple threshold approach of exp_2 when energies and velocities were processed.

3.5 Detector interpretation

An advantage of GP over other fuzzy approaches (e.g. artificial neural networks) is that evolved solutions can be interpreted to discover the constituent building blocks. This section interprets the evolved detector which processed ρ_a, ρ_b and ρ_p as HRG and FRG statistics, giving a FOM of 0.852.

The detector comprised 65 nodes but analysis of the active paths through the tree during processing revealed that many of these nodes were redundant. Redundant components in GP solutions are known as *introns* and, although these components do not contribute to a solution's fitness, they do benefit the survival of a solution by reducing the chances of the regeneration operators from mutilating the active components. For example, the detector contained a max function where the first argument was always greater than the second, thus the sub-tree constituting the second argument was redundant. However, its existence reduced the likelihood of a regeneration operator (i.e. cross-over or mutation) from selecting a node in the sub-tree constituting the first argument, thereby increasing the survival chance of the active components.

The detector was reduced to 23 nodes after removal of the redundant components. The detector is represented by $\min(s_1, s_2)$ where s_1 and s_2 are returns from the following sub-trees,

$$s_1 = HA_b + HH_p - 0.025 \quad (2)$$

and

$$s_2 = (HH_b + FH_p - 0.08)(FL_b + FA_p + 0.425) + FH_b + HA_p - 0.015 \quad (3)$$

where the symbolic representation for the terminal inputs introduced in Section 3.1 has been augmented with a subscript to denote the relevant material, e.g. HA_b represents an average HRG of the density of the ball.

The detector did not include any nodes pertaining to air density because none of the manually designated refinement cells in the training time-frame contained air alone, and so the evolved solution learned to ignore air density gradients. The absence of nodes relating to standard deviations of gradients suggests that this statistic was not useful for detecting refinement cells.

Equation 2 clearly represents a thresholding measure, using a single threshold of 0.025 for the density gradients on both the ball and the plate. However, from experiment exp_2 we know that the optimum threshold for the ball was greater than that for the plate when considering maximal cell-boundary gradients. The sub-tree s_1 allowed for this by using the maximum HRG for the plate but the average HRG for the ball. Equation 3 constitutes a more complex thresholding measure which will be discussed further below.

Table 6 shows which sub-tree contributed to the overall detector output for the training and validation time-frames. It can be seen that s_1 gave the detector output for most TPs, i.e. $0.0 \leq s_1 < s_2$ for most manually designated refinement cells. This can also be seen from Figures 2 and 3 which show the individual outputs from each sub-tree. Note that s_1 was very close to 0.0 (equal to -0.0005) for the missed refinement in cell in TF3.

The detector output was given by s_2 for most TNs because these cells corresponded to air alone or to uniform local densities, both of which yielded zero gradient statistics and thus $s_1 = -0.025$ and $s_2 = -0.049$.

TF2				**TF3**			
N_{TP}	N_{FN}	N_{FP}	N_{TN}	N_{TP}	N_{FN}	N_{FP}	N_{TN}
61	0	0	33	70	1	3	41
20	0	3	3267	39	0	15	3185
81	0	3	3300	109	1	18	3226

Table 6: Sub-tree results for the training and validation time-frames. The first row is sub-tree s_1, the second is sub-tree s_2, and the third row is totals. Note that the sum of the totals for a given time-frame is less than the time-frame size (60*60=3600) due to limbo cells and the fact that the extreme time-frame boundaries $y = 0$, $y = 59$ and $x = 59$ were not processed. The boundary $x = 0$ was processed because it acted as a line of symmetry.

Analysis of the individual sub-tree outputs revealed that the main purpose of s_2 was to reduce the FPs resulting from s_1 alone. This can be seen from Figure 4 where s_2 has effectively reduced N_{FP} from 10 to 3 for the training time-frame, by virtue of the fact that s_1 and s_2 had opposite sign for most of these non-refinement cells. Figure 5 shows that s_2 effectively reduced N_{FP} from 33 to 18 for the validation time-frame.

Referring to Equation 3, it can be seen that the sub-tree s_2 achieved this FP reduction by increasing its threshold whenever HH_b or FH_p was less than 0.08, due to the product being negative. Interestingly, when the training time-frame was

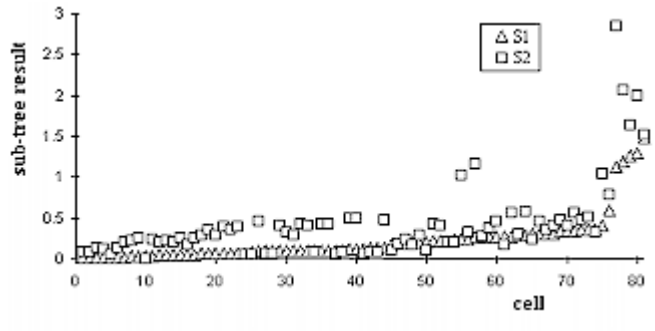


Figure 2: Sub-tree outputs in response to the manually designated refinement cells in the training time-frame. The cells are ordered by ascending s_1.

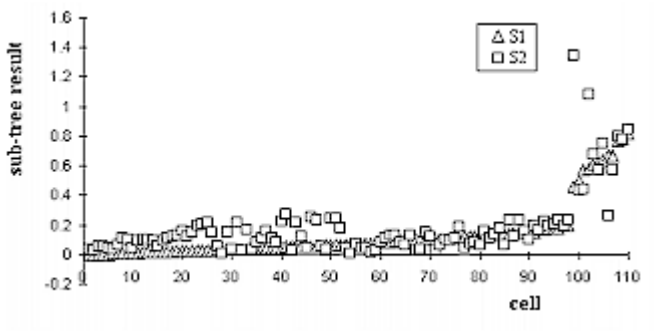


Figure 3: Sub-tree outputs in response to the manually designated refinement cells in the validation time-frame. The cells are ordered by ascending s_1.

processed with the s_1 sub-tree alone, HH_b and FH_p were typically greater than 0.08 for correctly detected refinement cells, i.e. TPs, whereas they were typically less than 0.08 for incorrectly detected refinement cells, i.e. FPs (providing that the corresponding material was present in the relevant cell, e.g. HH_b was only less than 0.08 for a refinement cell when the cell did not include the ball). In other words, it appears that this threshold manipulation in s_2 evolved in order to correct for the short-comings of the s_1 sub-tree.

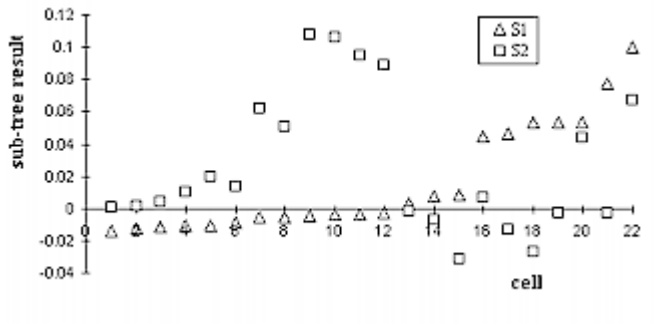


Figure 4: Sub-tree outputs in response to manually designated non-refinement cells in the training time-frame when $\max(s_1, s_2) \geq 0.0$). The cells are ordered by ascending s_1.

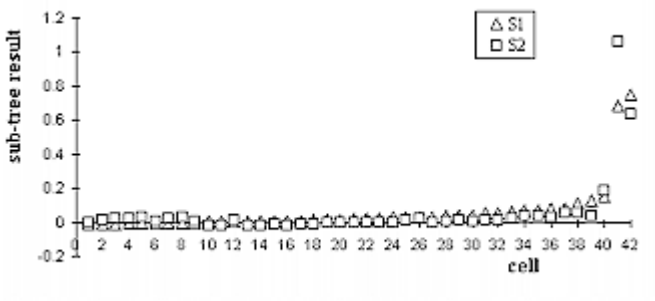


Figure 5: Sub-tree outputs in response to manually designated non-refinement cells in the validation time-frame when $\max(s_1, s_2) \geq 0.0$). The cells are ordered by ascending s_1.

4 Conclusions

This was a proof of concept problem to determine whether the Genetic Programming method can be applied to the field of numerical methods for impact analysis for non-destructive testing.

Evolved detectors generally gave better results than could be achieved by a simple threshold approach, e.g. $max[\Delta\rho] > \epsilon$, even when the latter used different optimum thresholds for different materials.

The simple threshold and detector evolution experiments both revealed that the individual material densities were the most useful cell properties. The figure of merit (FOM) in response to densities was at least twice that for the other cell properties in the threshold optimisation experiment. There was a less marked distinction between processing the different cell properties in the detector evolution experiment because the evolved detectors greatly improved upon the simple threshold results in response to energies and velocities. Nevertheless, the evolved detectors produced the greatest FOMs in response to material densities alone. Producing velocities alone consistently gave the lowest FOMs.

Similar results were obtained by using HRGs and FRGs individually but there was a slight tendency for HRGs to give greater FOMs. This suggests that refinement cells were more reliably identified using very localised gradients.

It has been shown that evolved detectors can be interpreted in order to gain understanding of a solution's underlying principles. Interpretation of the best detector revealed that detection was based on a combined thresholding for the different materials, which was significantly more sophisticated than the simple threshold approach. The interpretation also indicated which dimensions of the problem space could be neglected due to their absence in the evolved solutions, e.g. air densities and standard deviation of gradients.

Efficient mesh refinement for impact problems will remain a topic of research for many years.

Acknowledgments

The authors acknowledge assistance from Dr. Ian Cullis from DERA Fort Halstead for acting as our expert, suggesting the problem area, and for providing the impact simulation; and assistance from Dr. Richard W. Brankin from DERA Malvern for essential help with Fortran and engineering data manipulation.

Bibliography

[1] Zienkiewicz O.C.*The Finite Element Method in Engineering Science*, 2nd edition, McGraw-Hill, New York, 1977.

[2] Mitchell A.R., Fairweather G.*The Finite Difference Method in Partial Differential Equation*, Wiley-Interscience, New York, 1980.

[3] Fletcher C.A.J.*Computational Techniques for Fluid Dynamics*, Springer Series in Computational Physics, Springer-Verlag, 1988.

[4] Morton K.W. *Finite Element Methods for Non-Self-Adjoint Elliptic and Hyperbolic Problems: Optimal Approximations and Recovery Techniques*, Numerical Analysis Report 7/83, Dept. of Mathematics, Reading University, UK, 1983.

[5] Axelsson O., Barker V.A., *Finite Element Solution of Boundary Value Problems*, Academic Press, 1984.

[6] Brooks A.N., Hughes T.J.R. *SUPG formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations* Comp. Meths. in Applied Mechanics and Engineering, vol. 32, pp. 199-259, 1982.

[7] Leonard B.P.*Quick Scheme* Comp. Meths. in Applied Mechanics and Engineering, vol. 19, pp. 59-98, 1979.

[8] Gresho P.M., Lee R., Sani R. *Don't suppress the wiggles - they're telling you something*, in Hughes T.J.R. ed. Finite Elements for Convection Dominated Flows, AMD vol. 34, ASME, New York, 1979.

[9] Kelmanson M. A., and Maunder S. B., *Modelling high-velocity impact phenomena using unstructured dynamically-adaptive Eulerian meshes*. Journal of the Mechanics and Physics of Solids, 47, 731-762, 1999.

[10] Koza, John *Genetic Programming: On the programming of computers by means of natural selection*, 1992. The MIT Press.

[11] Roberts S.C., Howard D., and Brankin R., *Learning the rules that trigger adaptive mesh refinement*, DERA Malvern Technical Report: DERA/CIS/SEC/TR990441, 1999.

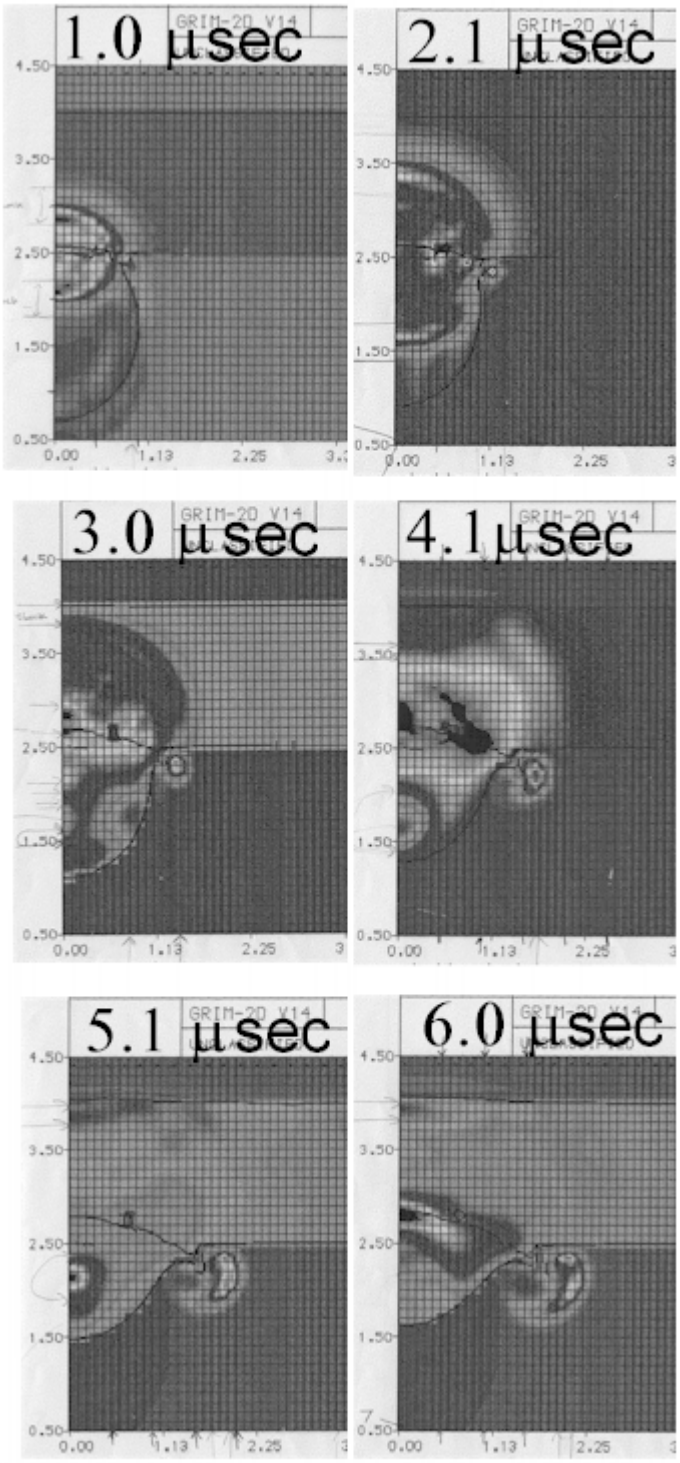


Figure 6: Ball impacting plate numerical simulation: pressure plots at different time frames.