J Indian Inst Sci., May–June 1990, **70**, 231–248.
© Indian Institute of Science.

An expert system approach for the design of composite laminates

K. P. RAO
Aerospace Engineering Department, Indian Institute of Science, Bangalore 560012,

AND

S. VISWANATH, S. SRIDHARA MURTHY AND C. JAYATHEERTHA
Structures Division, National Aeronautical Laboratory, Bangalore 560017.

Received on January 31, 1989; Revised on May 29, 1989.

Abstract

An engineering expert system to function at the level of a human expert should integrate artificial intelligence techniques with computer-based engineering tools. In the context of structural engineering applications, an expert system should integrate the design, analysis or database tools with the expertise in the knowledge base of the system. Based on this requirement, an expert system has been developed for optimal design of composite sandwich panels from strength and buckling considerations. Two analysis programs, LAMRANK and BUCLAM for strength and buckling, respectively, are interfaced with an expert system framework written in Turbo Prolog. The expertise has been represented as rules in the knowledge base. The resulting expert system, LAMDA (Laminate Design Assistant), serves as a useful design tool to the engineer in the design of composite sandwich panels.

Key words: Expert system, design, composite structures, laminate.

1. Introduction

Application of artificial intelligence (AI) techniques to engineering problems is a relatively recent development[1]. Traditional computing is unable to represent and interpret knowledge in an explicit form. On the other hand, design is concerned with concepts, ideas, judgements and experience, all of which are outside the realm of traditional computing. The aim of the present research is to develop an expert system where both numerical computation and knowledge-based problem-solving capability are together applied to perform optimum composite panel design from both buckling and strength points of view. In order to function at the level of an expert structural engineer, an AI system must be able to predict structural behaviour on the basis of both heuristic knowledge and the results of using the same computer programs as the engineer. In particular, programs for structural analysis and synthesis are necessary to design a structure. When the results of an analysis of the structure are needed to enable a rule to be applied, the expert system must be able to access and execute a computer code to extract the required results. This capability of using already available or newly developed computer programs implies broader range of expertise than is commonly associated with current expert systems. Some of the benefits that accrue by using such an expert system are[1]:

- permanent availability of expertise,
- wider accessibility of expertise,
- increase in productivity,
- training of new designers/analysis,
- a second opinion to that of a practicing expert, and
- more rapid response than a human expert.

While the AI techniques have been fairly successfully applied in civil engineering design problems[2,3] like that of multistorey buildings, multispan bridges, etc, their application to aerospace structures has been very limited[1,4,5]. Fibre-reinforced composites, because of their high-specific stiffness and strength, are extensively used now-a-days in aerospace industry. Development and application of expert systems to design problems of fibre-reinforced composites are very meagre. First attempt, it appears, to develop an expert system for design and analysis of composite structures is by Zumsteg and coworkers[4,5]. Here, an engineering expert system framework has been integrated with existing analysis and database programs to result in a 'composite design assistant'. Interaction between the expert system and the analysis program is established by means of limited, special-purpose interface routines.

In this paper, we present the work done towards developing an engineering expert system for the design of sandwich panels with composite facings. The design is based on either strength or buckling or both the criteria. The rules that govern the optimum design based on either criterion are identified and are built into the rule database. We present in detail the development of highly interactive computer program BUCLAM by Rao[6] and also the details of a program called LAMRANK developed by Tsai[7]. These two programs form major part of the computational element of the envisaged expert system, a long-term objective of the present exercise, using which one will be able to design a composite laminate from both buckling and strength points of view.

2. Engineering expert systems

Research efforts by computer scientists in the application of AI methods to intelligent problem solving have led to the development of expert systems. An expert system is a computer program that produces the same solution to a problem, in a limited problem domain, as would a human expert. In other words, an expert system aims to emulate the ability of human experts to ask pertinent questions, to explain why they are asking them and to justify their conclusions in providing a solution to the problem. The knowledge and experience of an expert in a problem domain is built into the knowledge base of an expert system in the form of rules and facts in symbolic form (in non-numeric as well as numbers).

The expert system manipulates these rules in making decisions about a problem in a given situation and arrives at the solution. Expert systems are different from that of algorithmic programs[1] in that they:

- are capable of making complex decisions within their knowledge domain,
- manipulate both symbols and numbers,
- use heuristic techniques in addition to algorithmic methods,
- are able to explain their results,
- can perform a variety of functions like diagnosis, interpretation, prediction, planning, design, monitoring and instruction,
- are able to work with probablistic knowledge, and
- are able to intelligently interact with the user.

Among the several expert systems that are developed for different classes of problems, the most popular ones are those for medical diagnosis. These systems, in general, rely on rules of experience of the experts and need very little capability to use any of the existing algorithmic programs. On the other hand, in structural engineering applications, design or analysis cannot be based purely on experience of the designer; hence mathematical modelling becomes mandatory. These mathematical models are efficiently implemented in algorithmic computer programs. A number of computer codes which perform the structural analysis and design are developed with several man-years of effort. An engineering expert system should be capable of integrating the relevant structural analysis and design programs to provide information needed by rule. Some of the engineering expert systems that are developed can be found elsewhere[1,2-5,8].

An engineering expert system shown in fig. 1 consists of the following components[5]:

- User interface
- Knowledge base
- Inference engine
- Attribute or context database
- Computational element consisting application programs and their databases.

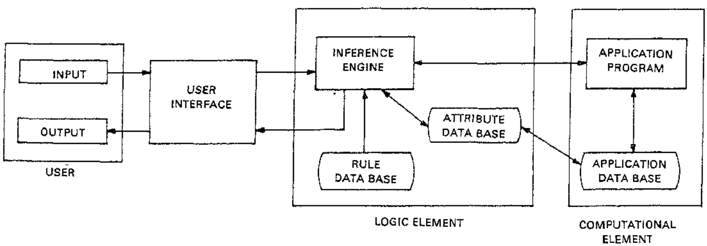

FIG. 1. Components of an engineering expert system.

Ideally, an user interface should provide an efficient communication between the user and the expert system by employing English-like natural language by means of a highly interactive screen-oriented display (menus and graphics) to convey user requests to the system and to report results back to the user. The logic element of an expert system consists of: i) a knowledge base consisting of general facts and heuristic (rules of thumb) knowledge provided by experts, ii) an inference engine (or mechanism), and iii) an attribute database that contains facts about a particular problem domain. The knowledge may be deep or surface. In the context of structural design, knowledge based on principles of mechanics is considered 'deep', whereas the heuristic knowledge developed from experience is termed 'surface knowledge'. A number of formalisms such as production rules, frames or concepts and semantic nets are available for representing knowledge. In the extensively used production rule representation, knowledge is represented as IF-THEN rules or 'premise–action' pairs: the 'action' is taken if the 'premise' evaluates to be true. Obtaining knowledge from a human expert and representing it as rules for the expert system to use is known as 'knowledge engineering'.

The facts about the problem being solved are stored in the attribute or fact database. The data in the attribute database reflect the current state of the problem at hand, and they are erased at the end of the session. The inference engine monitors the execution of the program by using rules in the knowledge base to manipulate the data in the attribute database. It may query the user or cause an application program to be run to acquire additional information. The mechanism used in the inference engine for manipulating the knowledge base is of two types: (1) forward, and (2) backward-chaining systems. In the forward-chaining or data-driven system, rules are searched to determine what conclusions could be made from the information given by the user, facts stored in the knowledge base (KB) and the previous conclusions. As conclusions are reached, premises of other rules are satisfied, such process continues until no more searches can be made. In the backward-chaining or goal-directed system, a hypothesis is accepted from the user, and satisfy it on the basis of information provided by the user, facts in the KB and other rules.

The last part of a generic expert system is the computational element which contains numeric-intensive and proven application programs together with their databases. These codes are generally written in an algorithmic programming language like FORTRAN, and are required to carry out design or analysis calculations. These codes are used by the knowledge-based system (KBS) in the same manner as a human expert to get the response of the structure being designed.

Thus, the following software is needed to build a KBS that integrates an expert system with an application program[1]:

- a domain-independent expert system including an user interface, reasoning explanation facility and the capability to include user subroutines.
- software incorporating domain-specific rules which represent the experience of experts, including the knowledge of when and how to use computational programs.
- interface program to provide link between logic and computational elements, to execute application programs, and to transfer the selected data from an application database to the attribute database or to the user.
- application software and their associated database (s).

The expert system for optimum design of composite laminates developed here is given the acronym LAMDA (Laminate Design Assistant), the details of which are presented in the next section.

3. The composite laminate design assistant—LAMDA

In this section, we describe the procedure for the design of composite panels (with sandwich or solid core), the method of knowledge acquisition and its representation in a rule database, and the components that constitute LAMDA along with their functional specifications. Described in detail are two application programs, LAMRANK and BUCLAM, which are used for optimum design of composite panels based on 'ranking technique'[7] with different design constraints. The design procedure adopted in LAMDA differs from the conventional design procedure[5] in that the laminate design is based on the concept of 'repeated sublaminate construction'. The general steps[9] involved in the design of sandwich panels are:

1. Define panel geometry and boundary conditions
2. Define external loads
3. Select safety factor
4. Select skin material
5. Select core material and type.

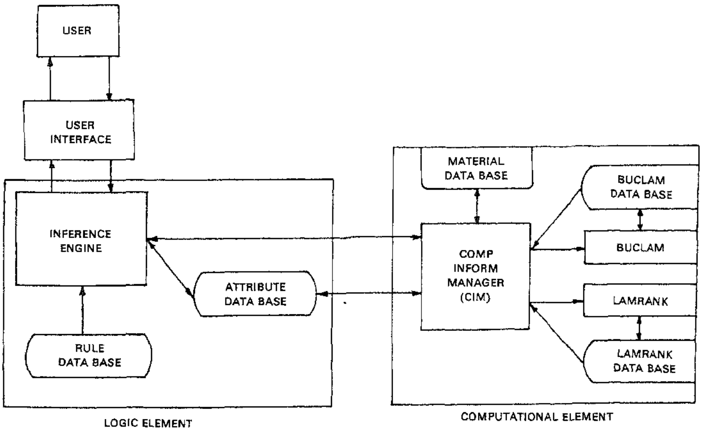

FIG. 2. Laminate Design Assistant (LAMDA).

6. Specify design criteria:
   i) Strength criterion
   ii) Buckling criterion
   iii) Combined strength and buckling criteria
7. Merit list the sublaminates from either strength or buckling or both the considerations
8. Choose the best laminate sequence which meets the factor of safety requirement, while satisfying the specified design criteria
9. Check the design for:
   i) face wrinkling and intracell dimpling
   ii) compatibility of core with skin
   iii) plate effects
   iv) environmental effects.

The designer provides the information required in steps 1 through 6, and the program operates in an iterative fashion to arrive at an optimum laminate construction in terms of repeated sublaminates to satisfy the specified design criteria and the factor of safety.

The components of LAMDA are shown in fig. 2. The direction of information flow is indicated by arrows. Before we describe the special programming features of LAMDA in section 5, it is pertinent to describe the details of two design programs which form the computational element of LAMDA.

4. BUCLAM and LAMRANK software

The BUCLAM and LAMRANK codes are based on the concept of repeated sublaminate construction of composite laminates and the philosophy of laminate ranking as a design optimization tool. A brief description of these concepts is presented.

4.1. *Repeated sublaminate construction*

Repeated sublaminate construction of composite laminates is employed in industry to reduce manufacturing errors and also to produce more damage-tolerant laminates. In this type of construction, basic sublaminate has a smaller number of plies, for example, 8, 6, and 4 and the full laminate is obtained by repeating the basic sublaminate. Figure 3 shows the symmetric and unsymmetric types of repeated sublaminate construction commonly used.

The location of a ply of a particular orientation in a sublaminate has great influence on the stiffness of the sublaminate and the load-carrying capacity of the laminate. So, in order to optimize the number of plies and the orientations of plies in a sublaminate, a coding has been developed[7] which is described below.

The number of plies in a sublaminate are taken to be 8, 6, 4 or 2. The number of different orientations of plies permitted is assumed to be 4 and 3 for 8- and 6-ply sublaminates, respectively, and two orientations for 4- and 2-ply sublaminates. Thus, the following 1128 possible lamination schemes of 8-, 6-, 4- and 2-ply laminates are examined.

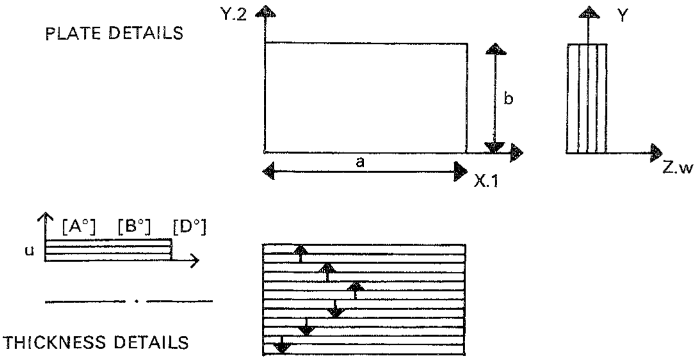

FIG. 3a. Repeated sublaminate construction (symmetric).

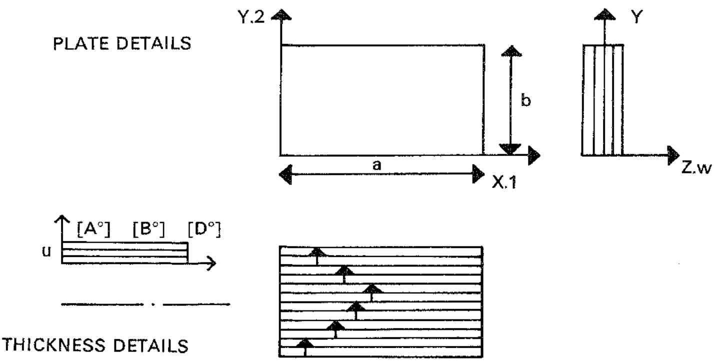

FIG. 3b. Repeated sublaminate construction (unsymmetric).

(i) 840, 8-ply, quadri-directional lamination schemes starting with [11111234], [11111243], [11111423],…,[44444321],
(ii) 240, 6-ply, tri-directional lamination schemes starting with [111123], [111132],…, [444432],
(iii) 36, 4-ply, bi-directional lamination schemes starting with [1112], [2111],…,[4443],
(iv) 12, 2-ply, bi-directional lamination schemes starting with [12],…, [43],

where the numbers 1, 2, 3 or 4 in the laminate code designate one layer of 0, 90, 45 or −45 degrees ply. For example, [11111234] designates [0/0/0/0/0/90/45/−45] and [444432] designates [−45/−45/−45/−45/45/90] ply sequence in a sublaminate, respectively.

The above scheme of laminate coding has been employed in the LAMRANK and BUCLAM programs for arriving at laminate rank table leading to an optimum configuration from strength and buckling considerations, respectively. The laminate ranking as a design tool is briefly presented in the next section.

4.2. *Laminate ranking as a design tool*

Laminate ranking is an alternative to the conventional optimization methods and appears to be a high-potential emerging tool for optimum design of composite laminates. This approach is reliable, consistent with the lamination theory, and above all, easy to implement. It offers definitive laminates that are achievable in practice. A detailed account of the ranking method of laminate sizing can be found in Tsai[7].

In the laminate-ranking technique, all the lamination schemes possible in a laminate with repeated sublaminate construction are evaluated for structural performance like laminate strength, stiffness and buckling load, etc. A rank table or merit list is constructed by arranging the laminates in the decreasing order of the strength or the buckling load along with the laminate code. The laminate with rank one is the optimum choice. The corresponding laminate code gives the ply sequence of the sublaminates from which the optimum laminate is constructed. A description of LAMRANK and BUCLAM software now follows.

4.3. *The LAMRANK program*

The LAMRANK program written in FORTRAN is a laminate design program based on laminate-ranking technique[7]. It is based on lamination theory and uses the quadratic failure criterion with a value of $-\frac{1}{2}$ for strain interaction term. It accounts for residual stresses due to curing of laminates and environmental conditions. Only in-plane loads are considered in LAMRANK. The LAMRANK database, created within the program using the input, can accommodate up to four different ply-angles and ten total plies in a sublaminate. For each laminate in the database, the program evaluates an 'effective' strain invariant, the strengths based on first-ply failure (FPF) and last-ply failure (LPF). The program determines the number of sublaminates required and then ranks the family of laminates for any of the following criteria: i) minimum strain, ii) maximum strength based on FPF, iii) maximum strength based on LPF, and iv) maximum strength based on the safety rule. In addition, any other rule may be built into the LAMRANK.

A factor of safety (FS) value of 1.5 is set within the program. However, user can specify his own value of FS. The design can be based on one of the first three criteria, or, alternatively, the maximum strength based on safety rule can be adopted in which the design limit is examined according to the following rule:

If (LPF stress/FPF stress) $<$ FS,

define a new design limit = LPF stress/FS.

If (LPF stress/FPF stress) $>$ FS,

set ultimate stress = FS $\times$ limit stress.

The output display from LAMRANK contains the following:

- An optimum sublaminate and the theoretical number of required plies based on the strength-ratio analysis.
- Absolute strength and stiffness of each laminate.
- Relative strength and stiffness over the quasi-isotropic laminate.

4.3.1. *Round-off procedure in LAMRANK*

In LAMRANK, a solution is given in the form $[a\ b\ c\ d]$, where the number of plies in each sublaminate, n, is equal to $a+b+c+d$. If N is the total number of plies in the solution, the repeat index is defined as, $r = N/n$. If N is not a multiple of n, then the repeat index is $r =$ INTEGER $(N/n) + 1$. This adjustment of the repeat index to the next integer value, so as to get an integral number of sublaminates, is a drawback resulting in more number of plies in the solution than actually required. A round-off procedure is adopted to rectify this drawback.

The total number of plies required for each configuration usually is not an integer number of sublaminates. A simple rounding off of the number of sublaminates to the next integer may result in more plies than required, thus sacrificing the gain made in the optimized solution. Thus, to achieve optimum laminate sizing, a supplementary sublaminate $[A\ B\ C\ D]$ is needed. The number of plies in the supplementary sublaminate is less than the number of plies in the main sublaminate. This is achieved such that the following criteria are satisfied:

- $[a\ b\ c\ d]^s$ does not meet the design criteria,
- $[a\ b\ c\ d]^s_{(r+1)}$ overshoots the design criteria,
- $\{[a\ b\ c\ d]_r + [A\ B\ C\ D]\}^s$ or
  $\{[a\ b\ c\ d]_{r+1} - [A\ B\ C\ D]\}^s$ just meets the design criteria,
- $[A\ B\ C\ D]$ has the minimum number of plies,

where the superscript 's' indicates that the laminate is symmetric.

The number of possible supplementary sublaminates increases very quickly with the number of plies in the sublaminate. In order to reduce the number of supplementary sublaminates to be ranked, a table of supplementary sublaminates that are investigated by the round-off procedure is stored in the LAMRANK database.

4.4. *The BUCLAM program*

The aim of designing with composite materials, based on buckling criterion, is to achieve maximum buckling load for given laminate thickness by optimizing the orientation and number of plies in each sublaminate. The BUCLAM program written in FORTRAN is used for this purpose. The BUCLAM has got access to the same material database as the LAMRANK. The BUCLAM database contains the codes corresponding to 1128 possible lamination schemes of 8-, 6-, 4- and 2-ply sublaminates (quadri-, tri- and bi-directional). These are used to construct ranking table based on buckling strength.

Using the input consisting of the geometry, design load, factor of safety (FOS) and the material of the panel, the optimum laminate scheme satisfying the buckling criterion is achieved in the following manner:

1. Assume an initial number of sublaminates (NSUB) based on Euler's buckling strength of a column with both ends fixed as:

$$P_{\mathrm{crit}} = \frac{4\pi^2 EI}{a^2}$$

where

$$EI = \frac{E_x h^3}{12(1-\nu^2)}$$

a being the length of the panel, h the panel thickness, ν the Poisson's ratio and E_x the longitudinal Young's Modulus of the face sheet.
Using the above formula, the panel thickness required to obtain the panel-buckling strength equal to the limit-buckling load (= design buckling load × FOS) is obtained. Thus, by knowing the panel thickness, and the core thickness provided by the user, the total number of plies required for the face sheet, and hence an initial estimate of the number of sublaminates (NSUB) can be obtained.
2. For this NSUB, buckling loads are calculated for all the 1128 ply combinations with 8, 6, 4 or 2 number of plies per sublaminate.
3. These lay-up schemes are merit-listed so that the laminate with the highest buckling load is arranged at the top of the table.
4. By using this highest buckling load along with the panel-limit buckling load, the new factor of safety (FOSS) can be computed.
5. Compare the FOSS with FOS. If FOSS is less than FOS, then increase the value of NSUB as per cubic power law[6]. This NSUB is to be used and the procedure repeated from step no 2.
6. If the value of FOSS in step no. 4 is greater than FOS, then decrease the value of NSUB by the same power law[6]. This value of NSUB has to be used and the procedure repeated from step no. 2.
7. The optimum number of sublaminates is reached when the actual factors of safety corresponding to two consecutive NSUB values bracket the design factor of safety.
8. The round-off procedure described in section 4.3 is also used here to adjust the number of plies in the supplementary laminate to meet the factor of safety exactly.

5. Programming aspects in LAMDA

The computational scheme of LAMDA with different components is shown in fig. 2. The user interface plays an important role in the success of an engineering expert system. It

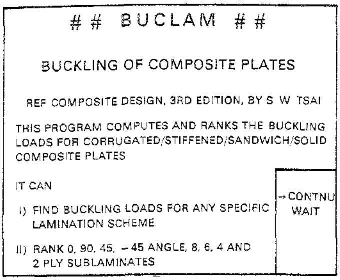

FIG. 4a.

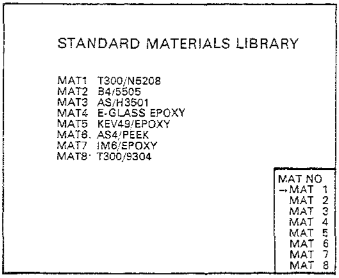

FIG. 4b.

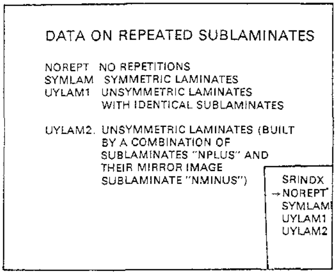

FIG. 4c.

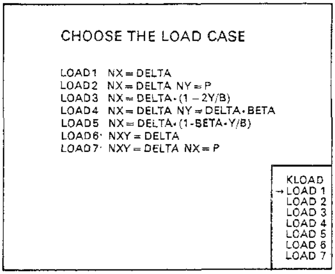

FIG. 4d.

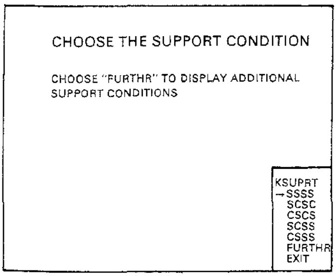

FIG. 4e.

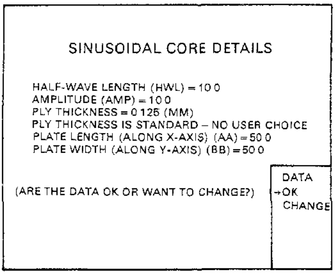

FIG. 4f.

should provide user-friendly link between the user and the expert system by means of menus and graphic displays, to convey the user requests to the system and to report results to the user.

During the development of LAMDA, the application programs were independently developed on personal computers as stand-alone software for optimum design of laminates based on different design criteria, as discussed earlier. A highly interactive user facility for problem definition and result presentation was provided through a menu-driven program written in FORTRAN using GKS software. The same is ported to LAMDA with minimum modification through a suitable interface program. An example of the user input through window dialogue is shown in fig. 4(a–f).

5.1. *Knowledge engineering in LAMDA*

The performance of an expert system depends heavily on 'knowledge engineering', *i.e.*, the acquisition and representation of knowledge that are required for solving the problem. Different types of knowledge involved in an engineering expert system for design, like LAMDA, are: the experience, equations, graphs, use of application programs and tables of data, etc.

The availability of an expert's experience is a very essential requirement to develop an expert system. For engineering problems, the experience includes knowledge like the procedure involved in solving the problem, the equations and data needed, relevant handbooks and programs (and how to use them), etc. The experience obtained from the expertise is formated to suit the knowledge representation scheme of an expert framework. In LAMDA, using a TURBO PROLOG shell, circumstances are described by IF (antecedent) – THEN (consequent) type rules. A fact database is used to represent information that does not depend on the situation. The representation of different kinds of knowledge in an expert system for engineering design has been discussed by Pecora *et al*[5]. Here, we give a typical example of a rule as employed in rule database of LAMDA.

While designing a laminate with LAMRANK, the design limit is redefined based on the ratio LPF stress/FPF stress, as discussed in section 4.2. This is represented as two rules in LAMDA:

```
rule:
        assign (N) if
          design_ values (N,_ , Lpfs, Fpfs, Fos,_),
          Fscalc = Lpfs/Fpfs,
          Fscalc < Fos,
          Nd1 = Lpfs/Fos,
          write ("\n The new_ design_ limit is\n", Nd1).
```

```
assign (N) if
   design_ values (N, _, Lpfs, Fpfs, Fos, Ls)
   Fscalc = Lpfs/Fpfs,
   Fscalc > = Fos,
   Us = Fos* Ls,
   write ("\n The ultimate stress for the design is\n", Us).
```

Here, N is associated with the name of the 'current_design' in progress and is represented symbolically in database. Due to the legal nature of PROLOG, each rule must include antecedent clauses to retrieve from the fact database all items occurring in calculations before they are used.

5.2. *Representation of equations*

Mathematical expressions are frequently used during the design process to calculate certain parameters and they must be evaluated by rules. As an example, we consider the Euler beam formula in section 4.4 which is used to calculate approximate ply thickness (h). This is achieved by the following rules.

```
rule:
        /* calculation of skin_ thickness */
        value (X, Y) if
           ask_value (X, Y); xvalue (X, Y).
        ask_ value (X, Value) if
           write (X," has value \n"),
           readreal (Value),
           asserta (xvalue (X, Value)).
        clear_ facts if
           write ("\nPlease press the space bar to Exit"),
           readchar (_).
        access_ cal (X) if
           xvalue (pi, Pi),
           xvalue (euler_ critical_ load, PC),
           xvalue (poissons_ ratio, PR),
           xvalue (plate_ length, PL),
           xvalue (long_ youngs_ modulus, LY),
           Hnum = 3* PL* PC* (1 - PR* PR),
           Hden = Pi* Pi* LY,
           Hfract = Hnum/Hden,
           X = exp ((ln(Hfract))/3).
```

```
answer_ is (X) if
  value (name_ of_ the_ current_ design,_),
  value (euler_ critical_ load,_),
  value (poissons_ ratio,_),
  value (plate_ length,_),
  value (long_ youngs_ modulus,_),
  access_ cal (X).
```

In LAMDA, the user can specify the type of design criterion in step 6 of the design procedure, or alternatively the rules in the rule database will try to satisfy the relevant design criterion by calling either LAMRANK or BUCLAM or both, and keep the user informed of the choice made and the reason for the particular choice. The choice of the design program and control of information from and to the computational programs are managed through the interface program, Computation Information Manager [CIM], written in FORTRAN, to act as a link between the logic element and the computatinal element (see fig. 2) of LAMDA.

In addition to providing access to design program(s), the CIM also executes an algorithmic program LAMSTIF to get effective laminate properties for a composite sandwich panel and links the output file from LAMSTIF to LAMRANK and BUCLAM. This is not explicitly shown in fig. 2.

The LAMRANK and BUCLAM have their own databases, apart from a common material database which they share through the CIM. The material database presently has eight material systems and can be easily extended. Any new material defined by the user is inherited by the program and is made a part of the material database for future use. Table I shows the details stored for a typical material in the material database.

Sample output from LAMDA based on strength and buckling criteria are shown in figs 5 and 6, respectively. A summary of the design of optimum laminate from buckling consideration is shown graphically in fig. 7. The figures are self-explanatory. The effect of round-off procedure, by addition/dropping of plies, on factor of safety is clearly shown in both the cases.

The expert system LAMDA provides the user with an optimum design for the composite sandwich panel satisfying the safety factors and the design criteria specified by the user. The optimum design is specified in terms of the number of sublaminates required along with the number of plies in the supplementary laminate. The number of plies in the sublaminate along with the ply sequence within a sublaminate is clearly specified in the output.

6. Conclusions

The design procedure for the optimal design of composite sandwich panels, built of repeated sublaminates, has been incorporated into an expert system framework using TURBO PROLOG expert shell. The resulting program LAMDA (Laminate Design Assistant) plays

Table I
Property data used by LAMRANK showing title block and corresponding T300–5208 graphite-epoxy values as a sample data block

Title block	NAME [SI]	Ex. GPa(*a*)	Ey. GPa(*b*)	nu/x(*c*)	Es. GPa(*d*)	Em. GPa(*e*)	T/cure(*f*)
	X. MPa(*g*)	X'. MPa(*h*)	Y. MPa(*i*)	Y'. MPa(*j*)	S. MPa(*k*)	rho/m(*l*)	T/opr(*m*)
	F^*_{xy}(*n*)	ho.E-6m(*o*)	Vf(*p*)	rho/ply(*q*)	eta/y(*r*)	a, Em(*s*)	b, eta(*t*)
	alph/x.E-6(*u*)	alph/y.E-6(*v*)	beta/x(*w*)	beta/y(*x*)	eta/s(*y*)	f, Ef(*z*)	h, Xf(*aa*)
	EIso. GPa (*bb*)	XIso. MPa(*cc*)	E'u/E'l(*dd*)	X'u/X'l(*ee*)			
Data block	T3/N52[SI]	181	10.3	0.28	7.170	3.40	122.000
	1500	1500	40.0	246.00	68.000	1.20	22.000
	−0.5	125	0.7	1.60	0.500	0.30	0.100
	0.2	22.5	0.0	0.60	0.400	0.004	0.004
	69.7	325	0.916	1.56			

(*a*) E_x, 0° elastic modulus, (*b*) E_y, 90° elastic modulus, (*c*)ν, Poisson's ratio, (*d*)E_s, shear modulus, (*e*)E_m, matrix modulus, (*f*) T_{cure}, cure temperature, (*g*)X, 0° tensile strength, (*h*)X', 0° compressive strength, (*i*) Y, 90° tensile strength, (*j*) Y', 90° compressive strength, (*k*)S, shear strength, (*l*)ρ_m, matrix density, (*m*) T_{opr}, operating temperature, (*n*)F^*_{xy}, normalised interaction term, (*o*)h_o, unit ply thickness, (*p*) V_f, fiber volume fraction, (*q*)ρ_{ply}, ply density, (*r*)η_s, stress-partitioning parameter for transverse stiffness, (*s*) a, E_m hygrothermal exponent for matrix modulus, (*t*)b, η, hygrothermal exponent for stress-partitioning parameter, (*u*) α_x, 0° coefficient of thermal expansion, (*v*) α_y, 90° coefficient of thermal expansion, (*w*)β_x, 0° moisture expansion, (*x*)β_y, 90° moisture expansion, (*y*)η_s, stress-partitioning parameter for shear modulus, (*z*)f, E_f, hygrothermal modulus of fiber, (*aa*)h, x_f, hygrothermal exponent for fiber strength, (*bb*) EIso, quasi-isotropic stiffness, (*cc*) XIso, quasi-isotropic uniaxial strength, (*dd*)$E'u/E'l$, elastic modulus ultimate/elastic modulus limit, (*ee*)$X'u/X'l$ uniaxial strength ultimate/uniaxial strength limit.

LAMRANK

LAMINATE RANKING
LAMINATE LAYUP
NEXT LARGER REPEAT INTEGER
R^{iso}/R^{lam} at lim
h/R^{lim}
R^{iso}/R^{lam} at lim*
h/R^{lim*}
CONTROLLING LOAD

No	Laminate		Limit			Limit*		Critical load	
	Layup	Repeat	Relative	# plies	Ult/FPF	Relative	# plies	Lmt	Lmt*
1	44	14	1.91	212.5	1.41	1.91	212.5	1	1
2	53	16	1.66	244.1	1.12	1.66	244.1	1	1
3	35	16	1.62	250.5	1.79	1.93	210.1	1	1
4	1034	16	1.6	253.9	1.61	1.72	236.2	1	1
5	134	16	1.6	253.9	1.61	1.72	236.2	1	1

No	Laminate with round-off	# plies	R-Limit	R-Limit*	R-Ult
1	[(44)X13 + (2)]s	212.0	1.00	1.00	1.50
2	[(53)X15 + (2)]s	244.0	1.02	1.02	1.52
3	[(35)X15 + (30)]s	246.0	1.02	1.18	1.77
4	[(1034)X15 + (30)]s	246.0	1.00	1.04	1.57
5	[(134)X15 + (30)]s	246.0	1.00	1.04	1.57

$[(a\ b\ c\ d)_{repeat} + (A\ B\ C\ D)]_S$
TOTAL PLIES
R-VALUES

FIG. 5. Explanation of results of laminate ranking. All laminates are CFRP T300/5208, 8-ply sublaminates, $\pi/4$ layup, degradation factor of 0.3, and safety factor of 1.5.

the role of an expert in carrying out the design of composite sandwich panels satisfying the strength or buckling or both the criteria. The panel optimization is based on ranking of laminates with different ply-orientations within a sublaminate. Two analysis programs, LAMRANK (for strength) and BUCLAM (for buckling) have been incorporated into LAMDA.

Powerful and user-friendly input-output programs along with the application programs written in FORTRAN have been linked to the logic element of LAMDA through special-purpose interface programs. The expertise in the form of experience, design equations, constraints, etc., are represented in knowledge base of LAMDA by means of

BUCLAM OUTPUT #

Rectangular composite plate

Material	: T300/N5208
B.C.	: Simply supported
Loading	: Nx = N critical = 0.1×10^8 N/m
Plate dimensions (in or m)	: Length = 0.254; Width = 0.254
Layup angles	: 0° 90° 45° −45°

Sl no.	Laminate with round-off	Safety factor	Critical load
1	$[(12333444) \times 14 + (0)]_S$	1.2281	0.12281 + 08
2	$[(12333444) \times 16 + (0)]_S$	1.8292	0.18292 + 08
3	$[(12333444) \times 14 + (1)]_S$	1.2721	0.12721 + 08
4	$[(12333444) \times 14 + (12)]_S$	1.3161	0.13161 + 08
5	$[(12333444) \times 14 + (123)]_S$	1.3976	0.13976 + 08
6	$[(12333444) \times 14 + (1233)]_S$	1.4760	0.14760 + 08
7	$[(12333444) \times 14 + (12333)]_S$	1.5657	0.15657 + 08

$[(\text{a b c d})\ \text{repeat} + (\text{A B C D})]_S$

FIG. 6. Sample output from BUCLAM for an SS rectangular composite plate with a factor of safety of 1.5.

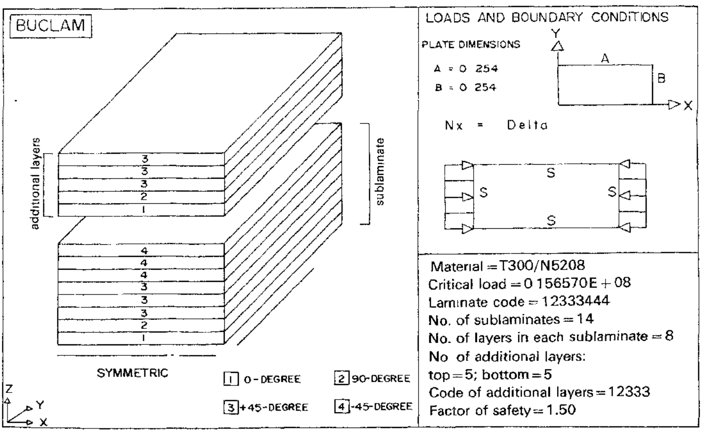

FIG. 7. Optimum composite laminate design.

rules. The novel feature of LAMDA is that it has the ability to execute the conventional design or analysis software, like a human expert, to determine the response of a structure in order to apply the various design rules in the knowledge base. This capability of LAMDA has the distinct advantage in that the user need not be familiar with the use of the analysis codes.

With the various functional requirements of an expert system having been identified, the scope of LAMDA can be extended to include several other features not presently implemented. An expert system like LAMDA appears to be a very promising tool of high potential for the optimal design of composite structures.

Acknowledgements

KPR wishes to thank Prof. Stephen W. Tsai, AFWAL/MLBM, WPAFB, USA, for many helpful suggestions he received during his tenure at WPAFB. The authors thank Mr D. D. Ravindranath, Research Fellow, NAL, for assistance in computer programming.

References

1. Zumsteg, J. R. and Flaggs, D. L. Knowledge-based analysis and design systems for Aerospace structures, *Proc. ASME Winter Annual Meeting*, Miami, Florida, Nov. 17–22, 1985, pp. 67–80.

2. Sriram, D., Maher, M. L. and Fenves, S. J. Knowledge-based expert systems in structural design, *Computers Struct.*, 1985, **20**, 1–9.

3. Nabir, A. B. Y. and Ahmed, H. El-Hajj A knowledge-based approach for the design of spread footings, *Computers Struct.*, 1988, **30**, 729–735.

4. Zumsteg, J. R., Pecora, D. and Pecora, V. J. A prototype expert system for the design and analysis of composite structures, *Proc. 1985 ASME Inter. Computers in Engng Conf.*, Boston, Massachusetts, August 4–8, 1985.

5. Pecora, D., Zumsteg, J. R. and Crossman, F. W. An application of expert systems to composite structural design and analysis, *Proc. ASME Winter Annual Meeting*, Miami, Florida, Nov. 17–22, 1985, pp. 135–147.

6. Rao, K. P. BUCLAM, A computer program for the study of elastic stability of corrugated/sandwich/stiffened/solid composite rectangular plates (Users' Manual), US Air Force Materials Laboratory. Wright Patterson Airforce Base, Dayton, USA, Feb. 1988.

7. Tsai, S. W. *Composite design*, 3rd edition, Section 26, 1987, Think Composites.

8. Bennet, J. *et al* SACON: Knowledge-based consultant for structural analysis, *Stanford Heuristic Programming Project Memo HPP-78-23*, September 1978, Stanford University.

9. Lubin, G. (ed.) *Handbook of composites*, Ch. 21, 1982, Von-Nostrand Reinhold.