# Some advice relating to buckling of shells

David Bushnell

This file, maintained by Allen Waters of NASA Langley Research Center, contains seven CHAPTERS, CHAPTERS 1 and 2 are of general interest. CHAPTERS 3 – 7 are oriented specifically toward users of the computer program PANDA2 (minimum-weight design of imperfect, composite ring and stringer stiffened flat or cylindrical panels and shells under multiple load sets).

**CHAPTER 1**
**HOW TO DETERMINE IF A STRUCTURE IS BUCKLING-SAFE**
**------- 22 October 2007 (modified, January 9, 2008: --------------**

Here is a message in which I describe what I feel is a generally valid procedure to determine whether or not a structure previously modeled with any general-purpose computer program for a stress analysis is buckling-safe.

The following letter, reproduced below, was written to my colleague, Frank Weiler, at Lockheed Martin in Palo Alto, CA. Frank had asked me what he should do to assure that a structure he was analyzing for one of the engineering divisions would not buckle. He already had a very detailed ABAQUS finite element model, and he had performed a stress analysis.

Now, I have never used ABAQUS and don't know what its capabilities are. However, despite this gross ignorance on my part, I wrote Frank the following message. In my opinion a procedure such as that described below should work in any context in which an engineer wants to determine if a structure is buckling-safe.

------------------ Letter to Frank Weiler follows (with some modifications made on January 9, 2008) ----------------------

October 18, 2007

Dear Frank:

I've been thinking about your structural buckling problem while I lie abed. I have come up with the following suggestions.

First, let me introduce a few definitions applicable to a **linear** buckling formulation:

1. A linear buckling eigenvalue equation is:

[K1]{q} = lambda[K2]{q}                              (1)

in which [K1} is the stiffness matrix for the structure as loaded by "Load Set B"; [K2] = Load-geometric matrix for the structure as loaded by “Load Set A”; {q} = buckling eigenvector, that is, the buckling mode shape; lambda = buckling eigenvalue (buckling load factor). Example: suppose you want to find the buckling pressure of a cylindrical shell under uniform external pressure (“Load Set A”) and under uniform axial tension that has a “fixed” value, that is, axial tension that you know in advance and that is not to be multiplied by the eigenvalue, lambda (“Load Set B”).

2. “Load Set A” = loads from which [K2] is computed, that is, the loads corresponding to which you want to find a buckling load factor (eigenvalue). In the example “Load Set A” = uniform external pressure.

3. “Load Set B” = loads which affect [K1], that is, the loads that are “fixed” in the sense that they represent part of the structural systsem in the same way that stiffnesses and boundary conditions represent the structural system. In the example “Load Set B” = uniform axial tension.

The following suggestions are based on the assumption that all the loads are either in "Load Set A" (that is, all the loads are "eigenvalue" loads) or any loads that happen to be in "Load Set B" (that is, "non-eigenvalue" or “fixed” loads) are **stabilizing**. I'm assuming that there are no "fixed" pre-loads, such as fixed **destabilizing** thermal loads and/or fixed **destabilizing** pressure or other "mechanical" loads. If it is hard to tell whether Load Set B loads are stabilizing or destabilizing, then do what you would do assuming that the loads in Load Set B are destabilizing. If there are significant destabilizing loads in Load Set B, then first perform a linear buckling analysis with all the loads in Load Set A set equal to zero and all the loads in Load Set B transferred to Load Set A. Check to see if this case yields any eigenvalues that are less than unity. If not (or if all eigenvalues less than unity are negative) then proceed as described below. If so, then the structure will buckle under Load Set B acting by itself and it must therefore be redesigned (or there might be something wrong with your modeling or with your specification of Load Set B).

Assuming that there are no "Load Set B" problems, proceed as follows:

**Item no.1**. Run a **linear** buckling analysis at 1g. (NOTE: In Frank Weiler’s particular structural problem the “Load Set A” loading was weight. This weight acted sort of like a nonuniform external pressure on a short cylindrical shell.) I expect you'll probably find a reasonably high (safe) buckling load factor (eigenvalue), but that might possibly not be so in your case (see **Item no. 3**).

**Caution 1**: Your analysis is for a PERFECT structure. The question always arises, what is the sensitivity of the buckling load factor to **initial imperfections**? This, of course, depends on the nature of the structure and the loading. If you find that the prebuckling state in the area where buckles first occur indicates relatively uniform axial compression in a thin cylindrical shell, then you probably should apply a knockdown factor of at least 2.0. It is to be hoped that the linear/nonlinear buckling analyses will yield buckling load factors that are high enough that you don't have to worry about initial imperfections.

**Caution 2**: I think yesterday you said that there are contact elements in your structure. Suppose you find that buckling occurs in a region where considerations of contact are important. A question arises: "How does ABAQUS model bifurcation buckling in a region where contact elements exist? For example, suppose you have a thin cylindrical shell (a "liner") encased in a relatively soft material, all under external pressure. During the buckling process the thin cylindrical shell is free to separate from the surrounding soft material where inward buckles develop, but pushes into this surrounding soft material where outward buckles develop. How does ABAQUS handle this? Perhaps a linear eigenvalue analysis is not sufficiently accurate.

**Item No.2**. Run a **nonlinear** buckling analysis at 1g. There probably won't be much difference from the linear result if the linear result yields a reasonably high buckling load factor. If the nonlinear buckling load factor agrees fairly well with the linear (which is the case in most designs), and if the buckling load factor for a loading of 1g is reasonably high (that is the structure is safe from buckling), then you are done. If the nonlinear buckling load factor disagrees with the linear buckling load factor, is **higher** than the linear buckling load factor, and the linear buckling load factor is reasonably high, you are done. A nonlinear buckling load that is **higher** than the linear buckling load indicates that either the destabilizing (negative) prebuckling internal loading of the structure increases more gradually (more slowly) with increasing load factor than does that corresponding to the linear theory, or the prebuckling deformation is of such a nature as to stabilize the structure, or both. If the nonlinear buckling analysis yields a significantly **lower** buckling load factor than does the linear buckling analysis, but the nonlinear buckling load factor is still rather high (indicating a buckling-safe structure), then you are done. A nonlinear buckling load that is **lower** than the linear buckling load indicates that either the destabilizing (negative) prebuckling internal loading of the structure increases more steeply than does that corresponding to the linear theory, or the prebuckling deformation is of such a nature as to destabilize the structure (such as prebuckling flattening of a cylindrical shell in the region where it buckles), or both. It is possible (though unlikely) that even though the linear buckling load factor indicates a buckling-safe structure, the nonlinear buckling load factor does not. If that unlikely event holds, proceed as in **Item 3**. Otherwise, you are done.

**Item No. 3**. If the linear buckling buckling analysis at 1g yields a buckling load factor that is unacceptably low (taking possible imperfection sensitivity and the effects of contact into account), then there are two possibilities:

a. It is possible (although unlikely) that a nonlinear buckling analysis will solve the problem for you. (Believe it or not, this actually happened in one case in my past!) The nonlinear buckling eigenvalue may be higher than the linear buckling eigenvalue by enough to render the structure buckling-safe. In that unlikely case you are done. (Although I'll bet you'd have a difficult time persuading the customer that the structure is buckling-safe under this circumstance!)

b. More likely, nonlinear effects are harmful, that is, they "soften" the structure, making it appear to be less stable than does the linear theory or nonlinear effects are not significant enough to change the verdict, "buckling-unsafe structure". In this most likely of findings under this item (**Item no. 3**) the best solution is first to redesign the structure before you attempt any more precise buckling analyses, such as nonlinear load stepping. If it is not feasible or possible to redesign the structure, and if you must obtain a more precise prediction of the structural behavior, I suggest you do the following:

i. Obtain one or more imperfection shapes from linear buckling theory. Use as imperfection shapes one or more linear buckling modes. Assign as amplitudes of these buckling modal imperfection shapes values that are in keeping with the engineering tolerances specified in your project.

ii. Perform nonlinear equilibrium analyses of the imperfect structure by load stepping. Be careful to use as imperfection amplitude(s) the proper algebraic sign(s), that is, the algebraic sign(s) that render the imperfection shape(s) which are most harmful. If you don't know what sign(s) to use, you just have to do at least two nonlinear analyses for each imperfection shape, one with positive and the other with negative sign of the amplitude of the initial imperfection(s).

iii. Look for nonlinear collapse or excessive stresses that occur during the nonlinear load steps.

I hope this is helpful, Frank!

Dave

---

**CHAPTER 2**
**CONCERNING FACTORS OF SAFETY**
----------------- 25 August 2007: -----------------------------------

Suppose you have two computer programs, Program A and Program B, that supposedly do the same thing: find the minimum-weight design of an elastic stiffened cylindrical shell.

**Assume that Program A** is based on the simplest possible model: membrane prebuckling behavior (that is, uniform prebuckling stress resultants, Nx, Ny, Nxy), smeared stiffeners (without any compensating knockdown factors to account for the inherent unconservativeness of a smeared-stringer and/or smeared-ring model), stiffener cross sections not permitted to deform, perfect shell (no buckling modal imperfections), and linear bifurcation buckling model (no nonlinear effects at all).

**Assume that Program B** accounts for some prebuckling bending, has knockdown factors to compensate for the inherent unconservativeness of smearing stiffeners, discrete stiffeners that can deform locally, buckling modal imperfections, and a "quasi-linear" bifurcation bucking model that accounts for the nonlinear prebuckling growth of the initial buckling modal imperfections with increasing applied load, Nx, Ny, Nxy, p.

If you use the same factors of safety and the same design load for both computer Program A and computer Program B, you will usually obtain a heavier optimum design with Program B than with Program A.

The purpose of this email is to emphasize that you have to give a lot of thought to what factors of safety and what loading you want to use in order to obtain your optimum design.

What I like to do whenever I obtain an optimum design is the following:

1. Use for the **design load** combination the **ULTIMATE** load, that is, the load combination, Nx, Ny, Nxy, p, that the structure would have to survive in its most rigorous static test. (This applies to multiple load combinations that the structure would have to survive with each ultimate load combination applied separately).

2. **In PANDA2 generally use factors of safety for buckling equal to 0.999.** (In PANDA2 when you use 0.999 for a buckling factor of safety PANDA2 does **not** automatically change the factor of safety to 1.1 the way it does if you use a buckling factor of safety between 1.0 and 1.09999). If you wish to allow local buckling at a lower load than the design load, then you must use a factor of safety for local buckling that is less than 0.999, say, 0.7 for a small amount of local post-buckling or 0.1 for a large amount of local post-buckling, for examples. If you do **not** want to permit any local post-buckling but you still for some reason want local buckling to occur at a lower load than general buckling, then use a buckling load factor of safety for local buckling of 0.999 and a buckling load factor of safety for general buckling something like 1.3, for example. I think it's best always to optimize with use of the YES KOITER option in the *.OPT file. (See Case 5 and compare its optimum design with that for Case 4 in my most recent PANDA2 paper, AIAA Paper 2007-2216, April 2007).

Often (it seems to me) factors of safety are dictated from "on high"; the designer doesn't have a choice in the matter. This management policy is poor, in my opinion, because then computer programs that include the most sophisticated models lead to the heaviest designs. You just penalize yourself by including a very sophisticated model for maximum stress, for example, and then applying to the predictions from this sophisticated model the same "traditional" factor of safety that you would have applied to the simplest "membrane" model. Might as well use the simplest "membrane" model if your management is unbending with regard to what factor of safety to use.

For example, if you include the effect of initial imperfections in your structural model you do not also want to include a factor of safety that in the past compensated for our lack of ability to predict the effect of initial imperfections many years ago.

**CHAPTER 3**
**CONCERNING THE ISSUE OF OPTIMIZED DESIGNS POSSIBLY CONTAINING PLIES WITH THICKNESS LESS THAN THE SPECIFIED THICKNESS OF A SINGLE PLY (This chapter is specific to PANDA2)**
---------------- 3 August 2007: ----------------------------------

To answer your question about an integral number of plies, each with a specified thickness, such as 0.0052 inches:

Unfortunately there is no "trick" to do it directly in an easy manner. However, you can develop an optimum design with multiple plies of thickness 0.0052 in a step-wise procedure such as the following:

1. Start with a face sheet with more layers than you think you'll eventually end up with. Include layers with layup angles, 0, +90, +,-45, etc. Do this allowing only ply thicknesses to be decision variables, not layup angles.

2. Allow lower bounds that are quite a bit smaller than 0.0052, such as 0.0005. As I recall, PANDA2 drops plies that are less than a certain small amount (I don't recall just now what that small amount is, 0.001 maybe???).

3. Set upper bounds of ply thicknesses that are equal to an integral number of plies.

4. Execute SUPEROPT followed by CHOOSEPLOT (choosing only the objective function to plot in CHOOSEPLOT).

5. Inspect the optimized face sheets carefully to see what the various thicknesses of the 0, 90, and +,-45 degree layers are. As a result of this inspection you may see that whereas you put 0-degree layers on the outside of the face sheet, PANDA2 is perhaps informing you that it would have been better to put a 90-degree layer on the outside, for example. One way to do this is to establish the face sheet laminate in Step 1 with the following layup angles: [0, 90, 0, +45,-45,0, 90,0]. Then, after your execution of SUPEROPT you may find that the first layer (the 0-degree layer) may have been dropped or may have a thickness that is much less than a single ply thickness. This means that PANDA2 is trying to put the 90-degree layer as the outermost layer (layer no 1). Or you may find that the first three layers, 0-degree, 90-degree, and 0-degree, have very small thicknesses. This means that PANDA2 is trying to put the +45, -45-degree layers on the outside. You have to use your judgment at this stage.

6. My intuition tells me that the top face sheet layup should always be the mirror image of the bottom face sheet layup. You will doubtless want to force this by appropriately linking the thicknesses of the layers in the bottom face sheet with those in the top face sheet. You could also force each of the face sheets to have a balanced layup, although off-hand I don't see the advantage of this. Still, you will have to link the thicknesses of the layers in the bottom face sheet to those of the appropriate layers in the top face sheet. Otherwise, you'll end up with an unbalanced sandwich wall.

7. As a result of the inspection described in Step 5, establish a brand new laminate for the top face sheet, probably with fewer layers. In doing this you will be guided by the results from your first execution of SUPEROPT. Layers that end up with very small thicknesses (let us say layers with a thickness less than half your specified 0.0052-inch ply thickness) you will probably want to drop from the face sheet laminate. After you have decided what to do, start over with a brand new case. Execute BEGIN and DECIDE again, this time with the new "proposed" face sheet laminate. (You can generate a new *.BEG file by appropriately editing the old *.BEG file, but give the new *.BEG file a different name, so that you will have preserved documentation of the entire process of arriving at a final optimum design).

8. Execute SUPEROPT followed by CHOOSEPLOT for your new case as in Step 4.

9. Inspect the results and proceed as in Step 5.

10. You may wish to set up yet a third case as described for the second case in Steps 7-9.

11. Possibly perform the Step 4 - Step 5 - Step 7 - Step 8 "loop" yet one more time, etc.

12. Suppose after you have completed this somewhat tedious iteration process you have a face sheet laminate that is reasonably satisfactory except that one or more layers are less in thickness than 0.0052 inches and other layers are of thickness between n and n+1 plies of thickness 0.0052 inches, where n is a small number. At this point the best thing to do is to reset the lamina thicknesses so that each layer is of thickness equal to one or more plies of thickness 0.0052 inches. Use your judgment as to how to do this. For example, you may want to round up all the lamina thicknesses to the next integral number of plies. Or you may want to round the thickness of some lamina up and other lamina down. Or you may want still to allow the thickness of one or two or three layers to vary during optimization cycles. After you have made your decisions with regard to face sheet thicknesses, execute DECIDE again such that the thicknesses are no longer decision variables (except for those few layers that you may want still to change during optimization cycles). Optimize again (with SUPEROPT/CHOOSEPLOT) allowing other variables such as stiffener spacings and cross section dimensions and core dimensions to vary during optimization cycles.

13. Keep iterating "by hand" until all the layers of the face sheets are of thickness equal to an integral number of plies and until you are satisfied with the final optimum design. NOTE: If any face sheet layer is too thick, that is, it consists of n plies each of thickness 0.0052 inches in which n is a rather large integer, then it would be a good idea to split this layer up by "interleaving" it with single-ply layers of different layup angle(s). For example, if you have a 0-degree layer with lots of plies, you will probably want to chop it up into several sub-layers interspersed with 90-degree plies each of thickness 0.0052 inches.

AN ALTERNATIVE PROCEDURE:

Make the layup angles decision variables in addition to the layer thicknesses. Don't forget to set up the appropriate linking relations in DECIDE: The thickness of a +theta degree layer must be equal to that of the appropriate -theta degree layer!

I suggested the method described first with the Step numbers implying that you would not use the layup angles as decision variables because PANDA2 results are sometimes "jumpy" from design iteration to design iterations when layup angles are included as decision variables.

A SUGGESTION

If you want some help on this rather difficult process, I could execute a case with laminated composite face sheets and see what happens. Send me the proper material properties, loading, etc. and I'll try going through the tedious process described above and then send the results to you as an example for you to follow in the future.

**CHAPTER 4**
**CONCERNING THE INITIAL MODELING OF RINGS (This chapter is specific to PANDA2)**

----------------- 3 August 2007: -----------------------------------

Perhaps the best way to establish the initial simplified one-layer model of the rings is to first do the following:

1. Set up a very simple **flat plate model, not a sandwich wall, no stiffeners**. The purpose of this very simple model is to establish an "effective" Young's modulus for the simplified one-layered rings to be used in the first model of your ring-stiffened sandwich cylindrical shell.

Have the flat plate be laminated composite, with the following layup:

[0,90,+45,-45,-45,+45,90,0] = total wall (quasi-isotropic).

Use the same material properties for each of the layers of this simple flat plate wall that you intend to use for the ring segment walls in your final complex model of the rings.

Have very simple, very small loading, say axial compression Nx only, and a very small value for Nx so that there will be no buckling or other failure.

2. Set NPRINT = 2 in the *.OPT file. Set ITYPE = 2 (fixed design).

3. Run PANDAOPT once.

4. Inspect the *.OPM file. What you want to search for in the *.OPM file is near the beginning of that file, the part where the 6 x 6 integrated constitutive Cij matrix is printed out. (The Cij are printed out to the *.OPM file when NPRINT = 2

5. Locate the value, C11, in the *.OPM file.

6. Compute the effective modulus, E(effective), from the equation,

C11 = E(effective) * t/(1.-nu**2)

in which t is the total thickness of the 8-layered wall and nu is Poisson's ratio. You can assume that nu = 0.3.

7. Use your computed value of E(effective) and nu = 0.3 for the material properties of the simplified one-layer rings in your initial model of the ring-stiffened cylindrical sandwich shell.

8. Optimize using the simplified one-layered ring model. Let the ring segment thicknesses and web height and outstanding flange width be decision variables Use SUPEROPT/CHOOSEPLOT, perhaps multiple times.

9. After you get an optimum design you are reasonably happy with, then re-optimize using the fancy laminated composite rings.

---

## CHAPTER 5
## CONCERNING THE USE OF SUPEROPT (This chapter is specific to PANDA2)

---------------- 4 August 2007: ----------------------------------

This is about the use of SUPEROPT. The SUPEROPT processor of PANDA2 is used to find a **"global"** optimum design.

With the use of SUPEROPT, I almost always specify 5 for the number of design iterations in the *.OPT file. (This specification occurs near the end of the *.OPT file. You were using 20, but I have urged you in previous emails to use 5 instead of 20.)

There is a section in my very long 1987 paper on PANDA2 entitled "PANDA2 - Program for minimum-weight design of stiffened composite locally buckled panels", Computers & Structures, Vol. 25, No 4, pp 469-605 (1987), about the optimization strategy used in the PANDA2 computer program. Please read Section 21.4 on pp 579 - 582 of that paper. That section of the paper explains why multiple executions of pandaopt are required for convergence to a **local** optimum design. SUPEROPT is used to find a **"global"** optimum design. "Global" is in quotes because SUPEROPT cannot rigorously determine the actual unique global optimum design but comes close to it in value (hopefully) by computing various local optima obtained from many different "starting" designs during a given execution of SUPEROPT. Each new "starting" design in a SUPEROPT execution is determined via the processor called "autochange". AUTOCHANGE establishes each new "starting" design by changing the decision variables in a random manner, consistent with lower and upper bounds, linking relationships, and inequality relationships specified by the PANDA2 user in DECIDE.

When I execute SUPEROPT, I almost always choose 5 pandaopts per autochange. This means that the SUPEROPT script executes pandaopt 5 times in succession, then executes autochange. SUPEROPT repeats this pattern until a total of about 470 design iterations has been reached. In other words, when the PANDA2 user specifies 5 pandaopts per autochange, the superopt UNIX script accomplishes the following succession of executions:

```
pandaopt
pandaopt
pandaopt
pandaopt
pandaopt
autochange

pandaopt
pandaopt
pandaopt
pandaopt
pandaopt
autochange

pandaopt
pandaopt
pandaopt
pandaopt
pandaopt
autochange
```

etc, etc, until a total of about 470 design iterations has been reached. If you inspect the list output near the end of the *.OPP file, you will see there a list of the objective function for this succession of executions carried out while SUPEROPT is in process.

Occasionally a case may come along for which you want to execute more than 5 pandaopts per autochange. You can tell by using chooseplot to plot the objective function vs design iterations after completion of each superopt execution.

How do you tell how many pandaopts per autochange is appropriate? By inspecting the plot of objective vs design iterations. Each "spike" in this plot represents a new "starting" design. Each "starting" design is obtained via execution of autochange within the superopt script. If, from the plot, it appears that PANDA2 has difficulty obtaining a converged result with 5 pandaopts per autochange, then in the next execution of superopt, try 6 pandaopts per autochange or 7 pandaopts per autochange, or perhaps even 8 pandaopts per autochange.

Generally 5 pandaopts per autochange is adequate, but not always.

Sometimes it may be necessary to execute superopt more than once in order to obtain a truly "global" optimum design. **Please note this important point: each execution of superopt must be followed immediately by an execution of chooseplot before you can execute superopt again.**

Hence, a typical runstream with the use of multiple executions of superopt would be:

superopt
chooseplot
diplot

superopt
chooseplot
diplot

superopt
chooseplot
diplot

With each "chooseplot", choose to **plot only the objective vs design iterations**. You can also plot margins and/or decision variables, but the plots are very messy looking so it's usually better not to.

---

**CHAPTER 6**
**CONCERNING GENERATION OF BOSOR4/BIGBOSOR4/STAGS INPUT VIA PANDA2**
---------------- 5 August 2007: ----------------------------------

In an earlier email you mentioned that you made an attempt to generate a bosor4 (or bigbosor4) model automatically. The purpose of this message is to inform you that bosor4 (or bigbosor4) is not very good for predicting the behavior of shells with sandwich wall construction. That is primarily because BOSOR4 (or BIGBOSOR4) does not account for the weakening effect of transverse shear deformation (t.s.d.). t.s.d. is especially significant in the case of sandwich wall construction.

Also, BOSOR4 (or BIGBOSOR4) does not deal with the many types of failure possible only with sandwich wall construction, such as face sheet wrinkling, core crushing, core shear failure, face sheet pull-off, etc. - those many "sandwich" modes of failure covered approximately by PANDA2. Hence, BOSOR4 will yield very unconservative predictions for sandwich shells.

In brief, I advise against using BOSOR4 (or BIGBOSOR4) for validating optimum designs of sandwich shells obtained by PANDA2. BOSOR4 merely handles the sandwich shell as an ordinary layered shell with the core simply modeled as just another layer and in which t.s.d. effects are not accounted for. You should use STAGS for the validation, but read the next paragraph.

Another thing: I have never created the software required automatically to set up STAGS input files for panels or shells with sandwich wall construction, in which the core is modeled truly as a shear-deformable element in the model. I think that now STAGS is capable of handling sandwich shells in a rigorous manner. After you have obtained an optimum design for a sandwich shell with PANDA2, you should create a STAGS model, not from STAGSUNIT or STAGSMODEL, but "by hand" in order to validate the PANDA2 optimum design of a sandwich shell.

---

## CHAPTER 7
## CONCERNING “ICONSV” (This chapter is specific to PANDA2)

----------------- 20 August 2007: -----------------------------------

This note is about the relatively new input datum, ICONSV (ICONSV = 1 or 0 or -1) that the PANDA2 user now provides during the MAINSETUP interactive session (*.OPT file).

My latest PANDA2 paper, "OPTIMIZATION OF AN AXIALLY COMPRESSED...", AIAA Paper 2007-2216, AIAA 48th SDM meeting, Honolulu, Hawaii, April 2007, describes ICONSV as follows:

--------------------------------------------

Item 676: A new input datum, ICONSV, has been introduced into the PANDA2 mainprocessor, MAINSETUP/PANDAOPT (*.OPT file). Therefore, old *.OPT files will no longer work. ICONSV can have values 1 or 0 or –1, the recommended value being ICONSV = 1. As set forth in the PROMPT.DAT file (prompting file for interactive PANDA2 input), The three permitted values of ICONSV are defined as follows:

ICONSV = 1 (recommended model) means:
a. Include the ARBOCZ theory [1D] along with the “PANDA2 theory” [1E] when computing knockdown factors for local, inter-ring, general buckling of imperfect shells.
b. Use a conservative knockdown factor to compensate for the inherent unconservativeness of smearing stringers for models in which the stringers are smeared (Item 676 in [1L]).
c. Use the computed (conservative) knockdown factor for smearing rings (Item 605 in [1L]).
d. The Donnell shell theory is used in SUBROUTINE STRIMP, where imperfection sensitivity is being computed. This is done because the ARBOCZ theory [1D] is based on Donnell theory.
e. PANDA2 will use the non-zero slope of the buckling nodal lines in the computation of prebuckling bending and twisting, Wxx, Wyy, Wxy, of shells with general, inter-ring, and local buckling modal imperfections. (panda2.news Items 620 and 645 are cancelled).

ICONSV = 0 (less conservative model) means:
a. Do NOT include the ARBOCZ theory [1D] when computing knockdown factors for local, inter-ring, general buckling of imperfect shells. Use only the “PANDA2 theory” [1E].

b. Use a less conservative knockdown factor for models in which the stringers are smeared (Item 676 in [1L]).
c. Use the computed knockdown factor for smearing rings (Same as for ICONSV = 1).
d. The user-selected shell theory is used in SUBROUTINE STRIMP, where imperfection sensitivity is being computed.
e. Panda2.news Items 620 and 645 are cancelled. (Same as for e. under ICONSV = 1).

ICONSV = -1 (still less conservative model) means:
a. Do NOT include the ARBOCZ theory [1D] when computing knockdown factors for local, inter-ring, general buckling of imperfect shells. Use only the “PANDA2 theory” [1E]. (Same as for ICONSV = 0).
b. Use a less conservative knockdown factor for models in which the stringers are smeared. (Same as for ICONSV = 0)
c. Do NOT use the computed knockdown factor for smearing rings (Knockdown factor for smearing rings = 1.0 when ICONSV = -1, EXCEPT when there exists significant local deformation in the outstanding flange of the ring in the “skin”-ring single discretized module general buckling model, in which case the knockdown factor is computed in the same way as for ICONSV = 0 and ICONSV = 1).
d. Set the knockdown factor for truncated double-trigonometric series expansion (ALTSOL) models [1G] to RFACT = 0.95. (RFACT=0.85 for "ALTSOL" models in which there are smeared stiffeners if ICONSV = 0 or 1).
e. The user-selected shell theory is used in SUBROUTINE STRIMP, where imperfection sensitivity is being computed. (Same as for d under ICONSV = 0).
f. Panda2.news Items 620 and 645 are in force, that is, a non-zero slope of buckling nodal lines will probably be set equal to zero for the computation of prebuckling bending and twisting, Wxx, Wyy, Wxy, of an initially imperfect panel. (Different from e. under ICONSV = 0 and ICONSV = 1).

With ICONSV = 1 a given stiffened cylindrical shell with a given general buckling modal imperfection and given loading experiences more prebuckling bending under application of the design load than is the case with the less conservative options, ICONSV = 0 or ICONSV = -1. With ICONSV = 1 the knockdown factors for compensating for the inherent unconservativeness of smearing stringers and of smearing rings are conservative. Therefore, estimates of the general buckling load factor obtained from PANDA-type models [1B] for both the perfect shell and for the imperfect shell are lower than they would be with the less conservative options, ICONSV = 0 or -1. The amplitude of a general buckling modal imperfection is assumed to grow hyperbolically [1E] as the applied load approaches the general buckling load of the imperfect shell. Therefore, for a given applied load, the lower the general buckling load factor the more overall prebuckling bending of the imperfect shell occurs. With more overall prebuckling bending there is more stress redistribution between panel skin and stiffener segments, with the result that local buckling load factors and bending-torsion buckling load factors are lower with ICONSV = 1 than they would be with ICONSV = 0 or –1, given the design of the stiffened shell. Also, maximum stresses are higher with ICONSV = 1 than with ICONSV = 0 or –1. Figure 99 shows various stress and buckling margins for the Case 4 design (Table 4) as functions of the “conservativeness” index, ICONSV. All the margins are highest for the least conservative model, ICONSV = -1, because the general buckling load factor (margin) is highest for the least conservative model. To emphasize what has already been stated: the lower the margin for general buckling the more prebuckling bending of the imperfect shell and therefore, because of stress redistribution between panel skin and stiffener parts, the lower the margins for local stress and buckling.

The behavior sometimes changes as a result of changes in ICONSV. For example, with ICONSV = 1 the maximum effective stress in Case 4 in Table 4 occurs in the outstanding flange of a ring. (See Margin No. 6 in Part 1 of Table 11). With ICONSV = -1 the maximum effective stress for the same configuration occurs in the outstanding flange of a stringer. The difference is caused by the different amounts of prebuckling bending with ICONSV = 1 and ICONSV = -1; significantly more circumferential bending occurs with ICONSV = 1 than with ICONSV = -1.
-------------------------------------------

NOTE: There is more under Item 676 in the ...panda2/execute/PROMPT.DAT file, which exists with the latest (July 2007) version of PANDA2. Please read that text in the PROMPT.DAT file. As I recall, the new additional text in the July 2007 version of PANDA2, which you now have at NASA Langley, is provided under the ICONSV = -1 option. I may be mistaken, so check it out.

Please see Cases 2, 3, and 4 in Table 4 (p. 55) of my latest PANDA2 paper (cited above) for examples of the same system optimized with the three different values of ICONSV, ICONSV = -1 (Case 2), ICONSV = 0 (Case 3), and ICONSV  = 1 (Case 4). Also, study Case 5, which is the most highly recommended model (YES KOITER).

I'm sending this note to you now to urge you to optimize the same system using all three values of ICONSV, then select the optimum design which you feel is the best.

If you decide that the best design after all your effort corresponds to your use of ICONSV = 0 or ICONSV = -1, then you **must** validate your optimum design by constructing a general finite element model and running a general-purpose computer program such as STAGS. If you decide that the best design is that obtained with ICONSV = 1, then, in my opinion, you should still use a genera-purpose program such as STAGS to validate your design.
In general, I think it is always best to check optimum designs obtained by PANDA2 by using a general-purpose computer program such as STAGS to validate the PANDA2 optimum design.

It would be rash, in my opinion, to proceed with building a structure without first doing this.

I introduced ICONSV as a new input variable about a year ago in order to allow the user to produce optimum designs with PANDA2 that are less conservative than was previously possible. The lighter-weight optimum designs produced with the use of ICONSV = 0 or ICONSV = -1 should be okay as long as you use STAGS or some other general-purpose finite element code to check them out.