REFERENCES

DE LEEUW, J. (1982). Nonlinear principal component analysis. In *COMPSTAT 1982: Proceedings in Computational Statistics* (H. Caussinus, P. Ettinger and R. Tomassone, eds.) 77–86. Physica, Vienna.

DONNELL, D., BUJA, A. and STUETZLE, W. (1994). Analysis of additive dependencies and concurvities using smallest additive principal components. Unpublished manuscript.

FRIEDMAN, J. H. (1991). Multivariate adaptive regression splines (with discussion). *Ann. Statist.* **19** 1–141.

KETTENRING, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika* **58** 433–451.

WAHBA, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia.

BELLCORE
445 SOUTH STREET
MORRISTOWN, NEW JERSEY 07962–1910

TREVOR HASTIE

AT&T Bell Laboratories

Professor Stone has done an admirable job in leading us through the difficult mathematics needed to build a firmer theoretical framework around high-dimensional nonparametric regression and density estimation techniques. ANOVA decompositions of regression surfaces are no longer confined to the case when the predictors are categorical; we can now play the same games in function spaces. Gu and Wahba (1991) describe similar decompositions in reproducing-kernel Hilbert spaces using tensor-product smoothing splines.

This comment moves us to the opposite boundary of the field and describes some computational tools for expressing and fitting tensor-product spline models of this kind in the S language [Becker, Chambers and Wilks (1988)].

In S there is a *formula language* for expressing models, primarily aimed at traditional ANOVA and linear models. For example, the formula ~ a * (b + c) expands to ~ a + b + c + a:b + a:c and expresses a model with main effects and interactions. Typically the variables a, b and c are factors. The formula is converted into a *model matrix* where the factors are coded via contrast matrices, and their interactions as matrix tensor products of these. The contrast matrix for a factor is a basis for representing the piecewise constant effect as a function of its levels; this is the default behavior for factors, and in fact a default contrast coding is used. This notion is extended by allowing the following in formulas: (i) variables representing matrices and (ii) expressions that are calls to functions, which evaluate to matrices.

We now elaborate in the context of regression splines.

There are some primitive functions in S, for example, `poly(x,...)`, `bs(x,...)` and `ns(x,...)`, for producing polynomial, *B*-spline and natural *B*-spline bases, respectively. The function `bs( )` (which we focus on here) has additional arguments relating to *knot placement* and *degree*, and returns a matrix corresponding to the specified *B*-spline basis evaluated at the values of x. For example,