

Lecture Notes in Control and Information Sciences

Edited by A.V. Balakrishnan and M. Thoma



38

Neural Models of Language Processes,

M.A. Arbib et al. eds, Academic Press, NY, 1982

*Language Processing and Other Organisms, S.-J. Segalowitz et al.,
Hillsdale, NJ, 1983*

System Modeling and Optimization

Proceedings of the 10th IFIP Conference
New York City, USA, August 31 – September 4, 1981

Edited by R.F. Drenick and F. Kozin



Springer-Verlag
Berlin Heidelberg New York 1982

Applications of Advances in Nonlinear Sensitivity Analysis
 Paul J. Werbos, U.S. Department of Energy
 Forecast Analysis and Evaluation Team

The following paper summarizes the major properties and applications of a collection of algorithms involving differentiation and optimization at minimum cost. The areas of application include the sensitivity analysis of models, new work in statistical or econometric estimation, optimization, artificial intelligence and neuron modelling. The details, references and derivations can be obtained by requesting "Sensitivity Analysis Methods for Nonlinear Systems" from Forecast Analysis and Evaluation Team, Quality Assurance, OSS/EIA, Room 7413, Department of Energy, Washington, DC 20461.

Context of the Work

The Energy Information Administration (EIA) provides data and analysis on all aspects of energy supply and demand. It uses dozens of models, including econometric (statistical, empirical) models, linear programming models based on technological data, a nonlinear micro equilibrium model solving for thousands of variables simultaneously across a 50-year span, hybrids and combinations of these, etc.

Many users of EIA's analyses do not accept EIA's conclusions at face value, especially when reports from other sources disagree. Thus the Forecast Evaluation and Analysis Team of EIA and its predecessors have carried out a broad program to evaluate and explain the qualitative assumptions of EIA models and forecasts. This program includes the development of tools to characterize the properties of large models, studies of estimation methods which are robust against outliers or model misspecification (i.e., correlated errors), proofs of convergence and existence properties, and many other projects. The first part of this paper describes how a small part of this work - the minimum cost calculation of first and second order derivatives of nonlinear systems - makes an essential contribution to the rest. The second part elaborates on another application, a method for stochastic optimization which becomes feasible only with the help of low-cost derivatives. This method opens up a wholly new approach to the field of artificial intelligence and neuron modelling; it is especially efficient with the new generation of "parallel" computers.

- $\underline{x}(t+1) = \underline{f}(\underline{x}(t), \underline{u}(t))$
- N components of \underline{x}
- m terms per equation f_i
- T time periods
- cost of simulation = mNT
- not a "simultaneous" (implicit) model

$$\frac{\partial^2 x_i(T)}{\partial x_j(1)}$$

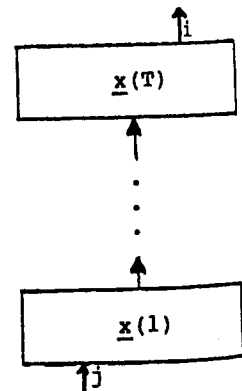


Figure 1: A Simple Example

Figure 1 shows a simple example of the kind of "derivative" we are trying to compute. Suppose that we have a nonlinear system, with a

vector
ables.
Figure
is mNT,
for each
terms.
small
large
The
constan
differ
coeffi
multip
sponse
"order
reason
usuall
more c
We
deriv
(i.e.,
the la
a func
for se
examp
may b

B
C

in t

The
der
con
upw
obs

vector \underline{x} of N endogenous variables and a vector \underline{u} of exogenous variables. Suppose that the system is governed by the equation shown in Figure 1. The cost of simulating the model over the whole time range is mNT , because in each of the T time periods we compute a forecast for each of the N variables in \underline{x} , and each such forecast involves m terms. Please note that N is often much larger than m . Given a small change in the variable x_i in time period 1, we want to know how large the resulting change in \underline{x}_i is in the final time period T .

The change in $x_i(T)$ per change in $x_j(1)$, holding the rest of $\underline{x}(1)$ constant, is a fundamental quantity of the system. It goes by many different names. In modelling, it is often called a "sensitivity coefficient." In economics, it is traditionally called an "impact multiplier." Electrical engineers often call it a "transient response," or "constrained derivative." Here we will call it an "ordered derivative," using the notation shown in Figure 1, for two reasons: (1) the notation is somewhat more explicit than what is usually used; and (2) the concept of ordered derivative is somewhat more general and rigorous, as will be seen.

Well-known applications which require the use of such first-order derivatives are sensitivity analysis, maximization of a system result (i.e., "deterministic optimization"), and statistical estimation. In the last two cases, one actually is concerned with the derivatives of a function of $\underline{x}(T)$ or of $\underline{x}(t < T)$ rather than the derivatives of $x_j(T)$ for some j , but it is easy to make this extension of the methods; for example, the function to be differentiated or a running total for it may be added to the list of system variables.

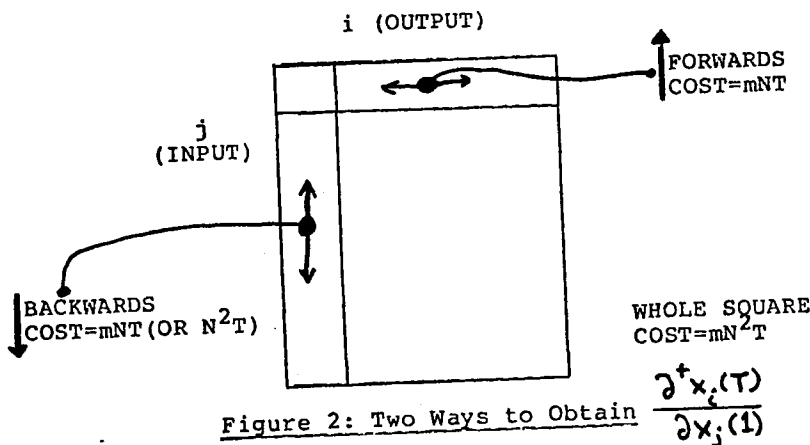


Figure 2: Two Ways to Obtain $\frac{\partial^+ x_i(T)}{\partial x_j(1)}$

Figure 2 describes two methods for computing ordered derivatives in the example above. The corresponding equations are:

$$\uparrow: \underline{z}(t) = \frac{\partial^+ \underline{x}(t)}{\partial \underline{x}_j(1)}; \underline{z}(t+1) = f'(t)\underline{z}(t)$$

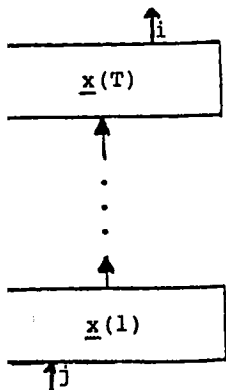
$$\downarrow: \underline{z}^*(t) = \frac{\partial^+ x_i(T)}{\partial \underline{x}(t+1)}; \underline{z}^*(t) = f^{-T}(t)\underline{z}^*(t+1) \text{ (or transpose)}$$

The large square in Figure 2 represents the entire matrix of ordered derivatives of all $x_i(t)$ with respect to all $x_j(1)$. The conventional or "forwards" method (indicated by an arrow pointing upwards) is based on perturbing one of the initials values $x_j(1)$, and observing the impact on all the final results, i.e., on the vector

Analysis

and application and include the or economic and neuron be obtained ar Systems" urance, C 20461.

data and uses dozens of models, a nonlinear simultaneous of these, etc. conclusions at disagree. Thus predecessors in the qualitative program includes of large against outputs), proofs of objects. The this work - derivatives to the rest. method for with the help of new approach to ng; it is "level" computers.



derivative" we are system, with a

$x(T)$. Each time we apply this method, we perturb only one of the initial values; thus we obtain only one row of the matrix of ordered derivatives, as shown in Figure 2. This costs us mNT calculations, as shown. Often the initial value $x_j(1)$ is actually changed, and the model resimulated. (This costs mNT operations, as did the original run of the model.) However, this leads to problems with the numerical accuracy of the results, because one computes each derivative by subtracting two numbers very close to each other in size. MIT's Troll System uses the forwards closed-form Jacobian formula, shown at the bottom of Figure 2, which has the same cost but is more accurate. The backwards method, shown with a downwards pointing arrow in Figure 2, computes an entire column of the matrix, using only mNT calculations. In engineering, this sort of method has been used for many years with "constrained derivatives," but has not been applied more generally.

The key point about these methods is that the forwards method is often used when the backwards method would be more appropriate. This can multiply costs (by a factor of N) to the point where it becomes infeasible to do what one wants to do. For example, it has long been known that economic data, like engineering measurements, are fraught with many errors, and that these errors invalidate conventional estimation methods. Statisticians like Hannan observed years ago that white noise converts a simple econometric model (like our example, but linear) into a "vector mixed autoregressive moving average process." In other words, one can account for such errors in data by estimating the corresponding vector ARMA process. However, because of the sheer cost of such estimation, it has rarely been done in economics. Instead, an approximation suggested by Hibbs has become popular of late: a conventional model is estimated by regression, and then simple univariate ARMA ("Box-Jenkins") modeling is used on the residuals, and the process may be iterated. Yet in statistical estimation, one only needs a single column of the derivative matrix (i.e., the derivatives of error), not the whole matrix; using the backwards method, one can compute all the derivatives needed in an iteration at the cost of only mNT , which is what it takes to exercise the model. This method was applied to vector ARMA estimation in the early 1970's, but has yet to receive wide application in economics. It now appears that vector ARMA estimation (and thus Kalman filtering estimation, which is formally equivalent to it) may have less value in social science than other more robust methods based on a generalization of Hartley's simulation path approach; however, those methods, too, require a set of derivatives, as part of minimizing a complicated loss function. Likewise, in sensitivity analysis, a user often wants to know the sensitivity of a few key results to all the initial values, or to be sure he knows the largest of these sensitivity coefficients. Again, only a few columns of the matrix are required; it is wasteful to pay for the whole matrix.

With large models or network systems, N may range from the hundreds to the millions or more. Thus cutting the cost of computing derivatives by a factor of N is often crucial to feasibility. One may be sure that the cost of exercising the system (mNT) is affordable, or the system would be of no interest; more than this, by a multiple of N , may be unacceptably expensive.

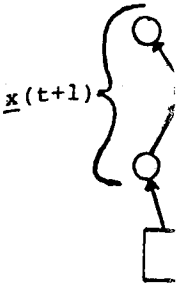


Figure 3: A

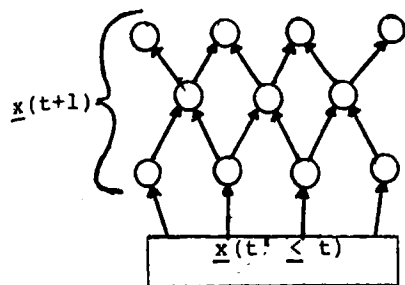
Now let nonnegative variables. model has be call this a sive system. be ordered i the vector x

Figure components, circle repre

The for Figure 3. TR variable-numl conventional there are on still only ne column of ne not previous constrained c to use N by the generaliz work systems with conventi

The methc "parallel" co operating in common. With culation time of x). With period, 4 pro none of the 4 period, the m method shown derivatives o first period bottom tier.

Large sca relatively spe works, made up ture. To opt is essential t measure with r feasible, it backwards meth derivatives fa



"CHAIN RULE" (DYNAMIC FEEDBACK):

$$\frac{\partial^+ x_i}{\partial x_j} = \sum_{k=j+1}^i \frac{\partial^+ x_i}{\partial x_k} \cdot \frac{\partial f_k}{\partial x_j} \quad i > j$$

CONVENTIONAL PERTURBATION:

$$\frac{\partial^+ x_i}{\partial x_j} = \sum_{k=j}^{i-1} \frac{\partial f_i}{\partial x_k} \cdot \frac{\partial^+ x_k}{\partial x_j} \quad i > j$$

Figure 3: A More General Example: $\underline{x}(t+1) = f(\underline{x}(\text{all } t' \leq t+1), \underline{u}(\text{all } t'))$

Now let us consider the more general model shown in Figure 3. All nonnegative lags - including zero - are permitted in the endogenous variables. However, we still assume here (as in the paper) that the model has been reduced to "explicit" form. (In economics, one would call this a recursive model; in mathematics, one calls it a nonrecursive system.) We assume that the functions f_i , which make up \underline{f} , can be ordered in such a way that we can use them one by one to calculate the vector $\underline{x}(t+1)$. The dynamic feedback "chain rule" has not been published before.

Figure 3 illustrates an example where $\underline{x}(t+1)$ has eleven components, each represented by a circle; the arrows flowing into a circle represent inputs required to compute that component of \underline{x} .

The forwards and backwards methods are generalized as shown in Figure 3. The subscripts here refer to an ordered index of all time/variable-number combinations; the formulas are given in more conventional form in the main paper. The key thing to note is that there are only m calculations per time/variable combination. Thus we still only need to make mNT calculations to get a complete row or column of ordered derivatives, as in our earlier example. This has not previously been published. With conventional matrix methods for constrained derivatives, based on our earlier example, one would have to use N by N matrices f' , which would not usually be sparse; thus the generalization here makes it feasible to differentiate large network systems which would have been too expensive to differentiate with conventional methods.

The methods shown in Figure 3 remain efficient even if one uses "parallel" computers. Parallel computers - based on many processors operating in parallel rather than one CPU - are becoming increasingly common. With a conventional computer, it would take roughly 11 calculation times to compute $\underline{x}(t+1)$ in our example (1 for each component of \underline{x}). With a parallel computer, it need only take 3: in the first period, 4 processors would calculate the lower tier in parallel, since none of the 4 lower components depends on the others; in the second period, the middle tier would be calculated; etc. The backwards method shown here allows similar economies: one can calculate ordered derivatives of a model result with respect to the top tier in the first period of calculation, then to the middle tier, and then to the bottom tier. The forwards method is similar.

Large scale models or systems typically can be represented as relatively sparse networks, as in this example. Actual physical networks, made up of units operating in parallel, have a similar structure. To optimize such a system (except in unusual special cases) it is essential to know the derivatives of the desired performance measure with respect to all parameters in the system; for this to be feasible, it is essential to use a method such as the generalized backwards method which does not multiply the cost of getting the derivatives far beyond the cost of exercising the system.

In this paper, we have discussed derivatives with respect to initial values of the variables only; however, the EIA report does consider parameters, and the case of exogenous variables is a trivial extension of the endogenous variable case. To avoid making a complicated discussion even more complicated, the EIA report only mentions our earlier example when discussing second derivatives; however, it is trivial to substitute the general formulas in Figure 3 for those in Figure 2, whenever they apply in the second derivative calculation, to arrive at more general methods. The section on stochastic optimization provides a partial example of the possibilities.

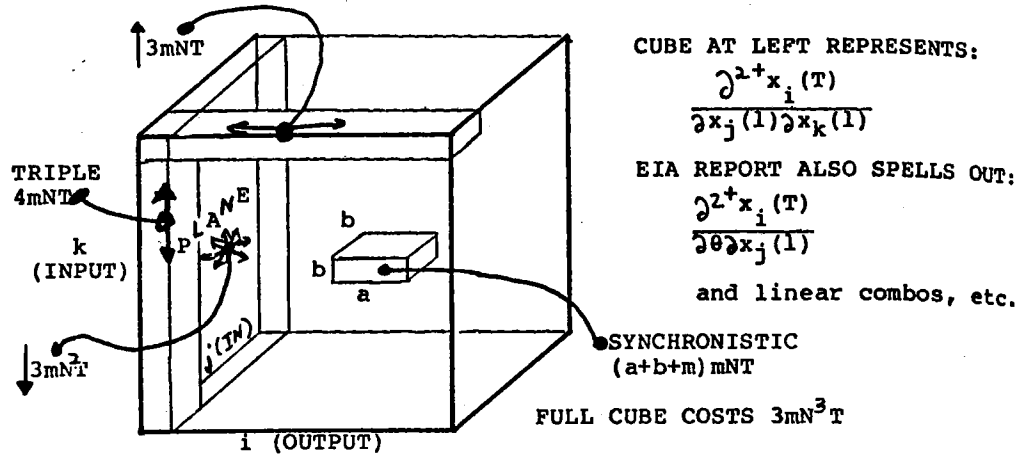


Figure 4: Costs of Obtaining Various Sets of Second Derivatives

Figure 4 provides a summary of the properties of the four variable-variable second derivative calculation methods provided in the EIA report. The set of ordered derivatives of $x_i(T)$ to $x_j(1)$ and $x_k(1)$ form an N by N by N cube, as shown; each method computes a subset of the cube, at approximate costs shown. Again, in practice, the key point is to compute only the subset required, and not pay for the entire cube. The five methods for computing variable-parameter second derivatives offer the same subsets (except that an upwards column and a row pointing backwards count as two separate cases) for the same rough costs.

The EIA report notes that variable-parameter second derivatives provide meaningful information about a model, essentially equivalent to what MIT provides for linear systems by looking at changes in eigenvalues. In effect, they tell us, for a change in a parameter of the system, how its dominant dynamics (revealed in the matrix of ordered derivatives to variables) change.

Among the possible applications is the use of Newton's method in estimation and optimization. It is straightforward to use the full backwards approach here for parameter-parameter derivatives; this allows computation of all the second derivatives one needs in order to use Newton's method, for a rough cost of only $3mN^2 T$, about the same as what people have paid to get only first derivatives when using forwards methods.

Applications of Stochastic Optimization Over Time

The remainder of this paper will discuss a way to implement "GDHP," a previously published approach to stochastic optimization

over t.
feasibi.
possibi.
Gi

Applica
decisio
output
and (4)
Th
be the
by the
issue,
Th
economy
many th
(devisi
however
mental
general
solutio
efficie
The EIA
In .

straint:
conside
must dev
the opt.
even wit
variable
finding
of the M
recognit
intellig
does not
that one
is well
a factor
period,
explicit
actions

Ther
abandone
the work
neuron (
scriptio
only two
and u ab
has show
"volleys
optimiza
differen
function
Simi
Like obj
and will

