SLAC-R-758
December 2008

A Proposed Global Optimum Algebraic Iterative Solver for Modeling of Lattice Element Errors

by

Martin J. Lee

SLAC National Accelerator Laboratory
2575 Sand Hill Road
Menlo Park, CA 94025

Work supported by US Department of Energy contract DE-AC02-76SF00515

Prolog

On the day after Thanksgiving, it is the perfect time for me to give thanks to all of the people I've known and worked with since joining "Project M" in 1962. Without their interest and support, I would never have been able to accomplish much of the work I did, particularly in accelerator lattice modeling for design and control applications—work I have been doing for nearly three decades.

More recently, I feel fortunate that I was able to return from retirement in 2005 to complete work that I began on OASIS Pathfinder. The publication of this SLAC report brings to a close over a year of intense work by bringing this project to a conclusion. My objective in submitting this report is to share with all my friends and colleagues at SLAC and other laboratories and research centers around the world a original and simple approach to nonlinear programming and systems optimization.

I started to write this report after OASIS Pathfinder was invented about five years ago and have only returned to finish it recently, after submitting a proposal in November 2008, for an SBIR Phase I grant. As I now complete this report on my work, I realize that unlike the many small businesses that are unwilling to share information deemed proprietary, I am eager to share my work, my project, with my collaborators, colleagues and friends here at SLAC. In sharing my work this way, I hope to mark and celebrate all the good times I've had at the laboratory over the past 46 years—I am thankful to my SLAC family for making all that I have worked for possible.

Abstract

Model-based electron accelerator control is the maintenance of optimal parameters of an electron beam such as its orbit, size, and shape, as well as machine parameters such as tunes. It works well when the model reflects reality. SLAC pioneered this technique in SPEAR about thirty decades ago. Similar techniques are now employed in particle accelerator and synchrotron laboratories around the world. There is still an inherently complex problem related to the employment of such techniques to manage the operation and analysis of accelerators and storage rings. The problem arises from the use by those techniques of complex numerical algorithms commonly known as nonlinear solvers that are difficult to control and operate. Lessons learned at SLAC have led to the development of a new, simple-to-use, and iterative nonlinear solver that holds much promise not only in advancing the derivation of errors in accelerators and storage rings at SLAC, but also in its ability to tackle a range of complex engineering problems. We intend to further develop and validate this nonlinear solver for robust SLAC accelerator and other applications.

A. Statement of How the Problem is Being Addressed in this Propose Project

Features in the new nonlinear solver—OASIS Pathfinder—will be tested with simulated data from a SPEAR3 error-resolving code. The performance of the new solver's convergence parameters will be analyzed and compared with existing SLAC lattice modeling codes. How lattice expertise can be incorporated into the new nonlinear solver will be studied. Requirements in running the new nonlinear solver directly on measured orbit data will be identified. The incorporation of lattice expertise and an ability to work directly with real-time orbit data allow the definition of requirements in turning the new nonlinear solver into a prototype code, to be developed on Phase II, which is capable of finding global minimum solutions in accelerator lattice error-resolving applications.

B. Applications and Benefits

Existing nonlinear solvers are difficult to use. How they may get to a solution is often far from being transparent to a user. The new nonlinear solver to be developed is expected to introduce a simple-to-use approach that is based on an iterative procedure guided by just two solution-path-seeking and convergence parameters irrespective of the size of the problem. The solution-path-seeking graphical approach also provides a high level of intuition for a user. After development, the new solver is expected to find its place next to existing math tools that are used worldwide such as MATLAB and Mathematica. The developed solver is also expected to be found useful in synchrotron storage rings around the world.

Analytical tools that can be used to efficiently resolve accelerator errors in U.S. synchrotron light sources (or high-brightness X-ray laboratories) will increase the up-time of those complex facilities which currently serve thousands of users from all scientific and engineering fields. The proposed project intends to develop such a tool that is based on a new method to solve complex equations in a simplified way. The developed tool also has many commercial applications, e.g., as a math-solution software toolkit and as a solver for complex engineering systems.

C. Identification and Significance of the Problem or Opportunity, and Technical Approach

C.1 Current Model Based Control at SLAC

One common theme in the operation of an electron storage ring is the use of a magnet lattice model for precision control of ring parameters. When storage ring components operate normally, the task of maintaining quality electron beam storage with a long lifetime and small emittance is challenging yet straightforward. However, component drift and other factors (temperature, ground motion, etc.) tend to detune the operating conditions away from optimal values. As a consequence, on-going error identification and correction of lattice/beam parameters such as optical functions, beam emittance, lifetime, and coupling require intervention of experts to resolve discrepancies from their optimal values.

C.2 Response Matrix Analysis

Response Matrix Analysis, known as the Resolve technique[1,2], is used to compare measured data against data generated from the accelerator model in order to diagnose discrepancies or faults among a large array of system parameters. Typically, the following steps are used to diagnose errors in the accelerator elements such as focusing errors in the quadrupole magnets:

1) The orbit response matrix is measured. Each element in the response matrix is a measurement of the orbit change at one beam position monitor (BPM) due to a unit orbit kick at one corrector. Hence, if there are *N* BPMs and *M* orbit correctors, the measured response matrix is an *N* x *M* matrix.

2) A simulated response matrix is then “computed” using an accelerator modeling code. The analysis begins with the computation of the response matrix using the ideal magnet strength values in the lattice model.

3) The analysis proceeds by varying the strength values of a chosen set of magnet components, such as normal and skew quadrupole components, sextupole feed-downs, and dipole correctors, in the model to make the computed response matrix agree with the one measured. As the model and measured data converge, the model yields a progressively more accurate representation of the operational parameters of the actual accelerator and can be used for both machine control and predictive analysis leading to fault identification and machine parameter optimization for the production of more stable and brighter beams.

A common problem occurs when the beam orbit shifts from the reference position. In this case the Resolve technique can specifically locate the cause of the orbit shift. Similarly, to diagnose faults in a group of magnet components the accelerator model trajectory/orbit data can be analyzed in the following manner:

1. Define the observed orbit change by the difference between measured and reference orbits.
2. Simulate an ensemble of one-pass beam trajectories within different regions of the accelerator. Look for the largest “error-free” region where there is a match between the simulated trajectory and the orbit change.
3. Continue analysis by varying the strength values of the magnet components located outside of the error-free region found in step 2) until a match between the simulated orbit and the orbit change is found.

C.3 Significant Problem

There is an inherently complex problem in the current operation and analysis of storage rings. Though the aforementioned steps appear relatively straightforward, they involve the use of complex numerical algorithms commonly known as nonlinear solvers that are difficult to control and to operate. For this reason, only accelerator modeling experts are able to use the Resolve method to analyze data and diagnose problems. Given these circumstances, the turn-around time (the period between when the problem is first discovered and when it is finally resolved) for any given diagnosis can vary from hours or days to weeks. In most cases, operators resort to fixing the problem themselves. They may be successful if the fix is simple enough, but their solutions may not address the underlying cause of the problem. Very often, they fail and are left with no choice but to wait for the assistance of experts. As long as the problem goes unresolved, the ring will not operate optimally and the user program is compromised.

In addition, there is another subtle problem in the current state of storage ring lattice modeling. Since the development of RESOLVE[3] more than two decades ago, nearly a half-dozen modeling codes have been developed by various teams of experts. Each of them uses a different nonlinear solver to solve for model errors. To see the differences between these codes, consider a modeling code as a process with the orbit data as input and the magnet component strengths as output. Most of the codes have different inputs yet they are looking for the same output. For example, the input data for one of the latest modeling codes, LOCO[4], are the orbit response matrices computed from a modern lattice modeling code[5]. The objective function is defined by:

$$f_{obj}^{LOCO}(a_1, a_2 ...) = \sum_i \sum_j \{m_{ij}^{\mathrm{mod}\,el}(a_1, a_2 ...) - m_{ij}^{meas}\}^2$$

where $(a_1, a_2, ...)$ are the magnet component strengths. Elements in the measured and model-predicted response matrix, $m_{ij}^{\mathrm{mod}\,el}$ and m_{ij}^{meas}, are defined to be, respectively, the ratio of the observed and computed change in the closed orbit at the ith beam position monitor due to a small kick from the jth dipole orbit corrector. Similarly, the other existing codes, MIA[6] and RESOLVE, minimize different objective functions, f_{obj}^{MIA} and $f_{obj}^{RESOLVE}$, that are defined in terms of turn-by-turn or single turn betatron oscillation values. **Since the objective functions in LOCO, MIA, and RESOLVE are different, different possible outcomes can be expected:**

$$(a_1, a_2 ..., a_k, ...)^{LOCO} \neq (a_1, a_2 ..., a_k, ...)^{MIA} \neq (a_1, a_2 ..., a_k, ...)^{RESOLVE} .$$

Only when the values of these solutions are the same will all three codes have found the global minima of their corresponding objective functions. In practice, LOCO, MIA and RESOLVE have not been able to consistently find the same global- minimum solution, thus making it difficult to compare their solutions.

One approach to finding the global-minimum solution[I] is to define an aggregate objective function such as a linear combination of the three objective functions:

$$f_{obj}^{AOF} = f_{obj}^{LOCO} + f_{obj}^{MIA} + f_{obj}^{RESOLVE} .$$

In addition, a nonlinear program could be specifically developed to find the global minimum. **The development of a simple, easy-to-use nonlinear program is a necessary step toward developing such an error solver for modeling of linear accelerators, storage rings, transport systems, and insertion devices.**

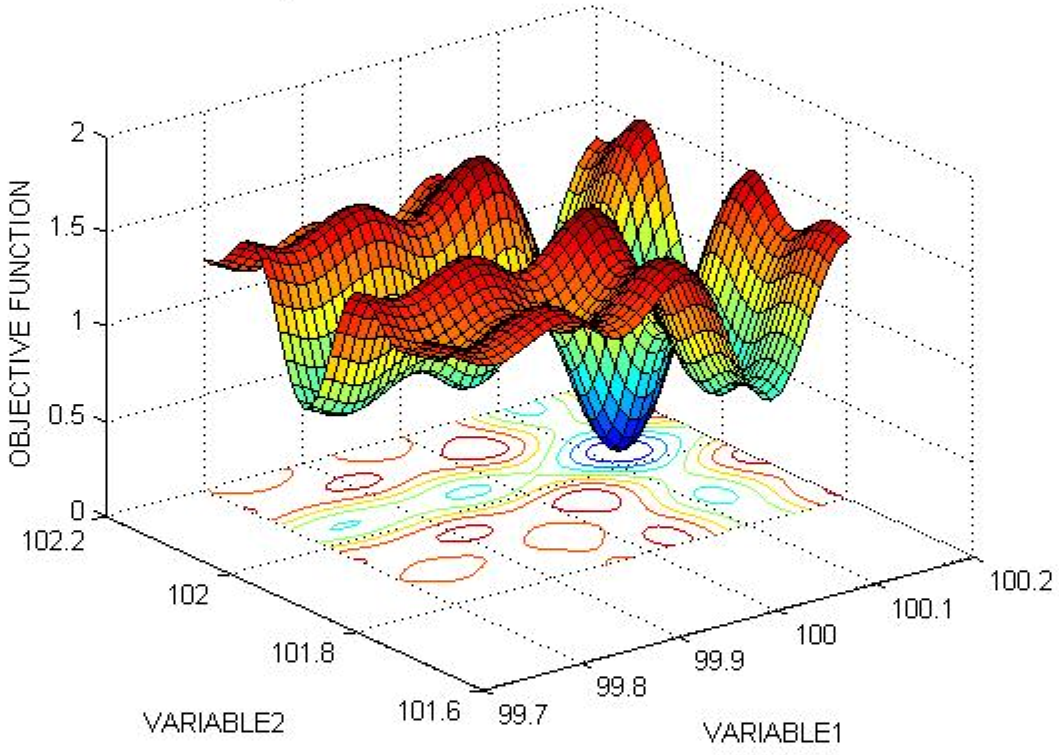
C.4 Opportunity

Existing nonlinear programs can be classified into two basic types: One uses an analytical iterative method and the other relies on a stochastic search method such as a genetic algorithm. The inherent difficulty of using an iterative method to find the global-minimum solution is well known. In general, an iterative method requires an initial guess solution. If this start solution is too far from the global-minimum solution, the program will find only a local-minimum solution. As an illustration, a surface plot of the objective function for a minimization problem with two

[I] *The global-minimum solution refers to the bounded multi-variable point having the overall lowest objective function value that may be difficult to find due to measurement noise in the orbit data.*

variables is shown in Fig. 1. This figure shows the locations of local-minimum points and the global-minimum point.

Fig. 1:OBJECTIVE FUNCTION SURFACE PLOT



A common approach to address this shortcoming is to use an ‘exhaustive search’ or a genetic algorithm. However, there are notable limitations when using such methods—they are often difficult and time consuming to use, particularly when used to find the global-minimum solution to a large scale problem as is the case for accelerator modeling.
How to find a way to overcome these limitations in the use of conventional nonlinear programs remains to be a challenge. The nonlinear programming method proposed in this project is an attempt to mitigate these limitations.

Furthermore, there is another problem in lattice modeling with existing iterative nonlinear programs. In practice, the ‘start’ solution is often taken to be the ideal magnet component strengths. On the one hand, this practice eliminates the need to search for a start solution. However, on the other hand, it may create another problem—there is an inherent difficulty known as the basin-of-attraction limit (BOA) to overcome. A BOA is defined to be the biggest region around a given minimum solution. The problem with the existing iterative nonlinear programs is that they will only find the actual solutions for a special case in which the start solution is inside the BOA corresponding to the global-minimum solution. We propose to collaborate with a small business company to develop a nonlinear program that does not have the inherent BOA limitation of conventional nonlinear programs. **One intended goal is to find a solution path that ends at the actual lattice element errors even when the start solution is not inside the BOA corresponding to the global-minimum solution**. The next subsection explains how a **new** technical approach can be developed to address the opportunity of reaching the goal.

C.5 Technical Approach[II]

In this section, the framework of the new nonlinear program to be applied to SLAC's accelerator modeling is presented. A nonlinear program is a solver typically employed to find the global minimum of a given objective function subject to certain conditions known as constraints. An objective function of *n* variables $(a_1, a_2, ..., a_n)$ is generally written as $f_{obj}(a_1, a_2, ..., a_n)$. In the accelerator lattice modeling, the variables are the strengths of a chosen set of magnet components. The complexity of finding the global minimum for the objective function depends on the degree of nonlinearity of the objective function and the number of variables.

For this proposed project, two of the main components of the new nonlinear program to be developed for finding actual lattice element errors are **Original and Simple Iterative Solution (OASIS) and Max-Jump-Look (MJL)**. OASIS is a non-derivative based algorithm that finds the values of the variables iteratively for an initially-guessed start solution. The iterative process in OASIS is shown in Fig. 2. When the values of the variables converge, the set of values of the variables at the end point is a solution that corresponds to a minimum, maximum, or saddle point of the objective function. This unique feature has led to the development of MJL—a new search method working in conjunction with OASIS to lead a warm start solution to the global minimum of the objective function.

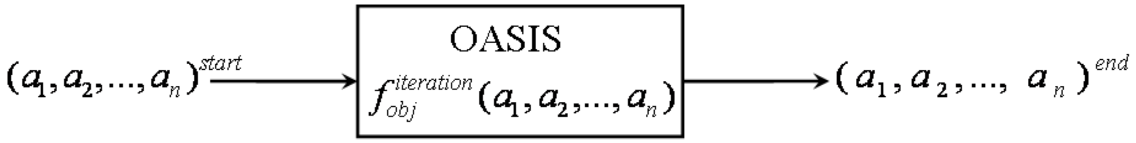

$(a_1, a_2, ..., a_n)^{start}$
OASIS
$f_{obj}^{iteration}(a_1, a_2, ..., a_n)$
$(a_1, a_2, ..., a_n)^{end}$

Fig. 2: Block Diagram of OASIS —an iterative process to find a solution that minimizes the value of a given objective function.

The new solver, **OASIS Pathfinder**, to be developed for SLAC's needs is composed of OASIS and MJL components that are linked together by the following three functional blocks: Start Solution Search (SSS), Path Search (PS), and Framework Set Up (FSU). The functional relationships among these five elements in OASIS Pathfinder are shown in Fig. 3.

[II] *Material in this section has been disclosed to SLAC Office of Technology Transfer.*

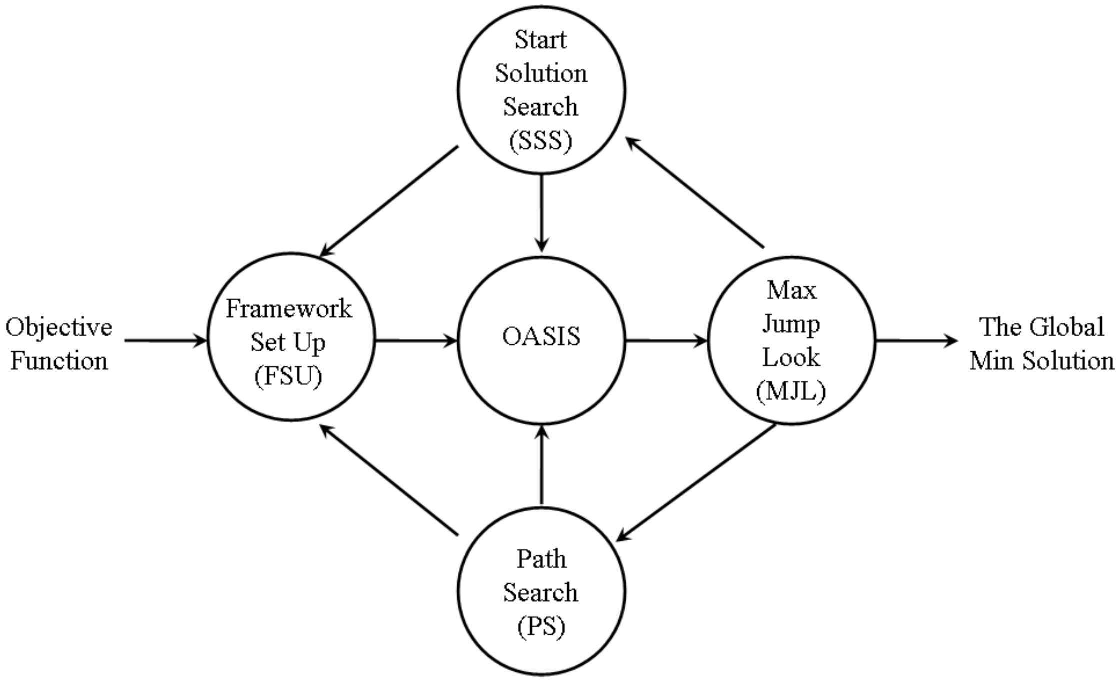


Fig. 3: Conceptual Layout of the Proposed Nonlinear Solver to be named **OASIS Pathfinder.**

A main feature of this proposed solver is that using OASIS it can find the global-minimum solution even when the start solution is not within the BOA corresponding to the global-minimum solution. Another salient feature is that using MJL it can search for a path that ends at the global-minimum solution independent of the size and complexity of a given problem, i.e., no matter how many variables an objective function may have and how nonlinear the problem may be.

OASIS Element

The key component in OASIS Pathfinder—the **OASIS** element— is an original and simple iterative algorithm[III] developed at SLAC from work performed on earlier CRADAs between SLAC and a few small businesses on their SBIR and STTR projects supported by DOE Office of Science (see Section H.2). This new algorithm was developed for solving a set of equations of the form,

$$\begin{aligned} &y_1 = f_1(x_1, x_2, ..., x_m) \\ &y_2 = f_2(x_1, x_2, ..., x_m) \\ &\ldots \\ &y_m = f_m(x_1, x_2, ..., x_m), \end{aligned}$$

[III] *A small business can license from SLAC the use of this technology for further development and eventual commercialization.*

to find the values of $(x_1, x_2, ..., x_m)$ for given values of $(y_1, y_2, ..., y_m)$ or $y_k^{desired}$'s for $k = 1,2,\ldots,m$. This formalism, after some algebra, leads to an iterative procedure that is briefly summarized in the appendix and is shown in the equation immediately below.

Before OASIS is used, the set of n variables $(a_1, a_2, ..., a_n)$ is first transformed by the FSU element into a set of chosen m $(\geq n)$ "working" variables $(x_1, x_2, ..., x_m)$. **The iterative algorithm can then be formulated as**

$$x_k^{next} = x_k \left\{ \frac{y_k^{desired}}{y_k} + s \left[\frac{y_k^{desired}}{y_k} - \frac{1}{m} \sum_{j=1}^{m} \frac{y_j^{desired}}{y_j} \right] \right\}, \; k = 1,2,\ldots,m,$$

where $y_k(s, p) = f_k(x_1, x_2, ..., x_m)$ is the value of each of the functions, f_k's, evaluated at every iteration, and s and p are the two control parameters. The values of $y_k^{desired}$'s are determined for achieving the global minimum of the objective function at the end of the iteration, i.e. $\frac{\partial f_{obj}}{\partial a_j^{end}} = 0$ for $j = 1,2,...,n$. Note the changes in the working variables at each iteration (step-size) are given by

$$\Delta x_k^{iteration}(s, p) = x_k^{next} - x_k = x_k \left\{ \frac{y_k^{desired}}{y_k} - 1 + s \left[\frac{y_k^{desired}}{y_k} - \frac{1}{m} \sum_{j=1}^{m} \frac{y_j^{desired}}{y_k} \right] \right\},$$

for $k = 1,2,...,m$. The size of the first step as well as the sizes of all subsequent steps is determined by the choice of s and p values. When the values of the working variables converge, the size of the end step goes to zero: $\Delta x_k^{end}(s, p) \to 0$ for $k = 1,2,...,m$. This condition implies that $y_k^{end}(s, p) \to y_k^{desired}$ for $k = 1,2,\ldots,m$. For this reason, s and p can be considered as convergence control parameters: s is called the "Size" parameter, and p the "Path" parameter. At present, the values of *(s , p)* are chosen manually in prototype code-testing studies. In the following section, an idea is proposed for a systematic search procedure consisting of SSS and PS elements in the new solver for choosing the required values of s and p to find a path leading to the global-minimum solution. On the Phase I project, a goal is, therefore, to develop an automated method to find the values of the actual lattice errors using the new solver.

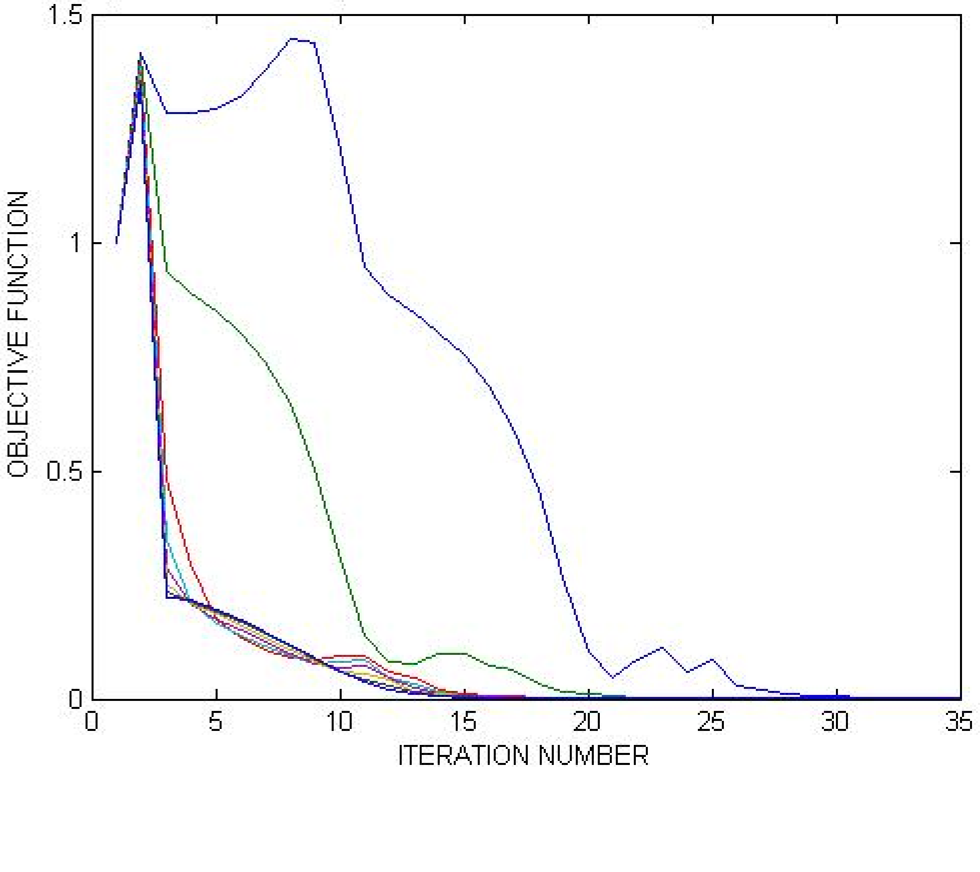
In practice, depending on the way the objective function approaches its end value, it is easy to see that the end point is a minimum. From now on, the values of $(a_1, a_2, ..., a_n)^{iteration}$ found by OASIS at each iterative step will be referred to as points along a path in the multi-variable space. To find a solution for a given minimization problem, the following steps are taken by OASIS:

1. Read initial set of guess values for variables: $(a_1, a_2, ..., a_n)^{guess}$.
2. Use $(a_1, a_2, ..., a_n)^{guess}$ values to compute $(x_1, x_2, ..., x_m)^{start}$.
3. Use $(x_1, x_2, ..., x_m)^{start}$ values in the iterative formulae to compute $(x_1, x_2, ..., x_m)^{first}$.

4. Use $(x_1, x_2, ..., x_m)^{first}$ to iteratively compute $(x_1, x_2, ..., x_m)^{\sec ond}$.
5. OASIS continues this process to find the values of $(x_1, x_2, ..., x_m)^{third}$, $(x_1, x_2, ..., x_m)^{fourth}$, and so on.
6. The iteration then stops when the values of all the x_k's converge, i.e. $x_k^{end} \cong x_k^{end-1}$ for $k = 1,2,...,m$.
7. At the end, OASIS uses the $(x_1, x_2, ..., x_m)^{end}$ values to compute the values of the end solution as well as the end value of the objective function: $(a_1, a_2, ..., a_n)^{end}$ and $f_{obj}(a_1, a_2, ..., a_n)^{end}$ or f_{obj}^{end} .

Example of a Two-Variable Problem**:** As an illustration of how OASIS works, the results obtained for a typical small-scale minimization problem with two variables a_1 and a_2 are presented. In this example, the same bounds, $\Delta = 0.2$, are imposed on the values of both bounded variables: $\Delta > (a_k - a_k^{start}) > -\Delta$ for $k = 1,2$ with $a_1^{start} = 100$ and $a_2^{start} = 102$. By running OASIS repeatedly with different (s, p) values, OASIS is able to find a set of solution paths starting at the same point and ending at the global minimum point as shown in Fig. 4A.

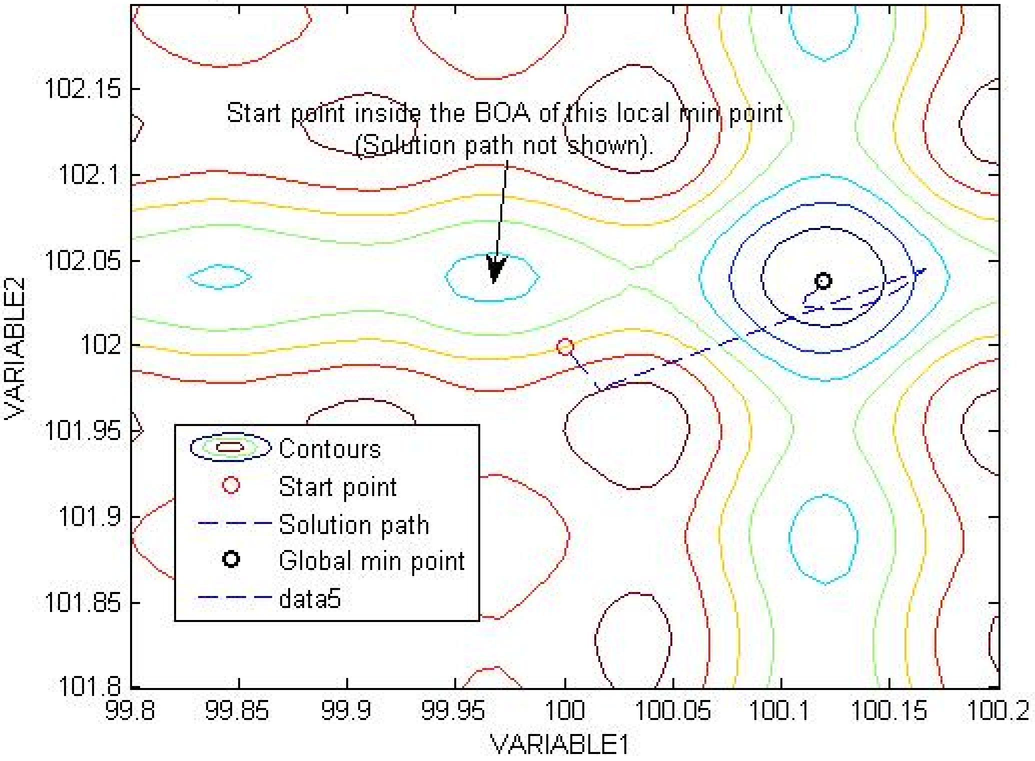
Fig. 4A Variation of Objective Functions on Solution Paths to the Global Min.



It can be seen from this plot that, because OASIS Pathfinder is a non-derivative-based algorithm, the objective function values it finds for points on the solution paths first rise above the objective function value at the start point before falling toward zero. One of these solution paths is shown in Fig. 4B. (In contrast, the objective function values found by using conventional derivative-based algorithms will always be less than the objective function value at the start point.) Since all paths end at the same global-minimum point, the variable values at the global-minimum point are given by $a_1^{end} = 100.13$ and $a_2^{end} = 102.04$.

Figure 4B also shows another special feature of OASIS Pathfinder—Its unique ability to find the global-minimum solution when the start point is outside of the BOA of the global-minimum point. At present, for a given start point, a manual method is used to search for the appropriate values of the control parameters (*s*, *p*) to find solution paths that end at the global-minimum point. Using this manual search method, OASIS has been able to find the global-minimum solutions for small-scale problems having less than ten variables. Based on this experience, We are confident that OASIS is capable of addressing larger scale problems.

Fig. 4B Plot of a Solution Path that ends at the Global Min Point



Max-Jump-Look Process

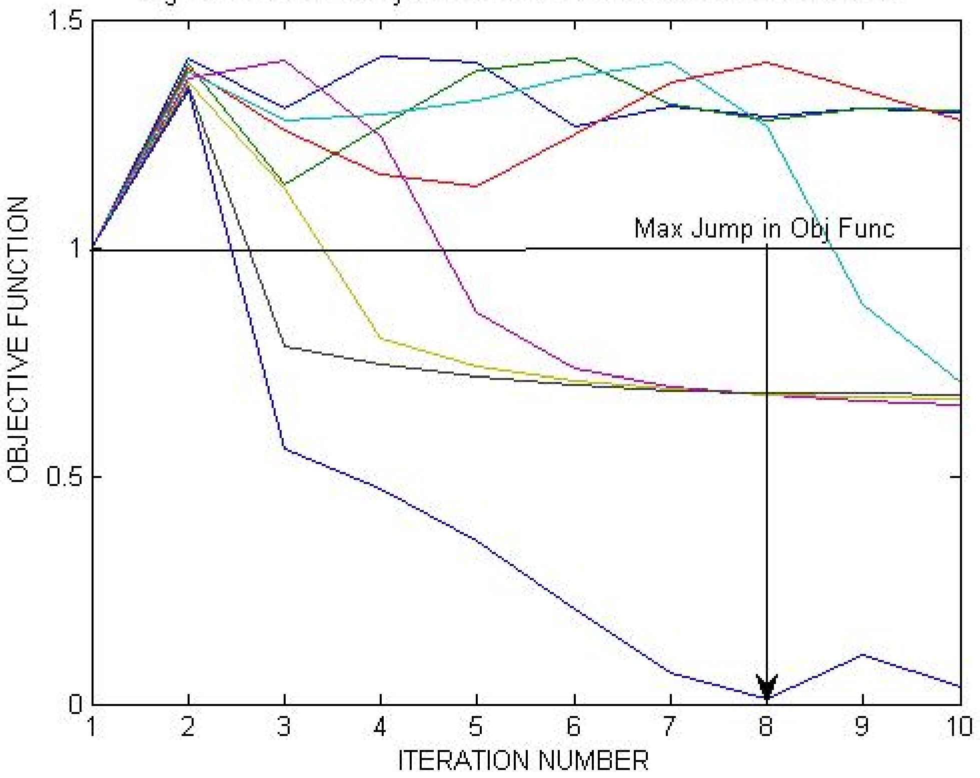
The development of MJL is guided by the following observations:

1. The value of the objective function at a local minimum point is proportional to the distance between that local minimum point and the global minimum point

2. When the objective function value at any given point is less than the objective function value at a certain local minimum point, that given point is inside the BOA of another local minimum point which is closer to the global minimum point than the original local minimum point.
3. For a given start point, there is a region in the control parameters space such that every pair of (*s*,*p*) values inside the region produces a path that ends at a minimum point of the objective function. As this given start point gets closer to the end point, the larger is this region and easier it is to find the values of the control parameters (*s*,*p*) for a path that ends at the global-minimum point.

When there are multiple solution paths, MJL looks for and records the pair of (*s*,*p*) that refers to the path having the overall largest jump (downward) of the objective function value as shown in Fig. 5. Hence, this process is named: Max-Jump-Look (MJL). The recorded solution will be used as the start point when the MJL process is repeated. Using the solution at the first max-jump point to repeat MJL, OASIS finds the value of the objective function at the second local minimum point can be found**.** It is easy to see that each time the MJL process is repeated, the point with the largest downward jump in objective function it finds will get closer to the global minimum point. **Thus far, the result of using the MJL manually in conjunction with OASIS has demonstrated that this combined process can find the global-minimum solution for small scale problems (less than 10 variables with a small number of local minimum points surrounding global minimum).** On this Phase I project, another goal is to further develop the MJL's dual search methods (SSS and PS) so that they can find a path leading to the global-minimum solution automatically.

Fig. 5 Variation of Objective Functions on a set of Solution Paths.



OASIS Working with Max-Jump-Look Method

In this section, the role the *s*- and *p*-parameters play in the MJL process to search for a 'warm' start solution and a proper path that leads to the global-minimum solution is described. When OASIS is used to find the real lattice model of an accelerator or storage ring, the first step is to select a set of variables and define an objective function. In practice, the variables $(a_1, a_2, ..., a_n)$ include the strengths of magnet components as well as the calibrations of the BPMs that are used in measurement of the orbit response matrix. The iterative process begins with the ideal strengths and calibrations as the start solution: $(a_1, a_2, ..., a_n)^{start} = (a_1, a_2, ..., a_n)^{ideal}$. The success of calibrating the real magnet strengths will depend on finding the (s, p) values for a path that ends at the global-minimum solution. Table I is a comparison of the proposed method (OASIS plus Max-Jump-Look) and the conventional methods for finding the real lattice errors in accelerators and storage rings.

Comparison	LOCO, MIA	The New Solver
Basin of Attraction around a minimum point: BOA	Larger	Smaller
Ideal start point: $(a_1, a_2, ..., a_n)^{ideal}$	Must be inside BOA of the Global-minimum solution	Can be outside BOA of the Global-minimum solution
Convergence: Ideal start point outside BOA of the global minimum point	Cannot find the global-minimum solution unless first search successfully for a warm start solution over a n –variable space, $(a_1, a_2, ..., a_n)$.	Can find the global-minimum solution through search over only a 2-parameter space, *(s , p).*

Table I: Summary and comparison of the basic differences between the proposed method and existing minimization methods to find the real magnet strength calibrations.

D. Anticipated Public Benefits

There are a number of benefits from the proposed project that will affect different U.S. constituencies which we enumerate below.

D.1 Benefits to SLAC

The proposed project intends to study a new approach in the performance of lattice modeling programs at SPEAR3, making use of currently available modeling techniques while adding functionality to it. The new functionality is expected to increase the efficiency and confidence with which accelerator-control experts can make changes to their machine and this translates directly to savings on the overall operational costs of accelerators. A more specific near-term goal is to produce an efficient means to manipulate the lattice for extremely short pulse synchrotron light. This will enable the facility to serve increasingly stringent requirements of its scientific and industrial users.

D.2 Benefits to DOE

The potential savings expected in future in SPEAR3 operations imply that the facilities funded by DOE's Basic Energy Sciences (BES) program will be able to do more for scientific and industrial research users. Specifically, it could provide a more sophisticated means to tune complex accelerator and free electron laser facilities such as SPEAR3 and the LCLS. The short bunch lengths associated with a low-α mode, for example, provide fast, broadband X-ray pulses that can be used for studies of ultra-fast dynamical systems along the line of research carried out by the BES Division of the DOE Office of Science. The developed techniques can even be transferred to other DOE light source user facilities.

D.3 Benefits to Small Businesses

An outside small business can leverage the DOE SBIR Phase I funding (and, hopefully, in a later Phase II-funded project) to also study the requirements for: 1) the development of a software toolkit to market to synchrotron light sources throughout the world; and 2) the development of a commercially viable product to specifically serve industrial needs in seeking optimized solutions for complex engineering systems. This small business also can plan to develop OASIS Pathfinder as a nonlinear solver to be widely licensed to developers of math software tools such as MatLab. When those requirements have been identified, the small business can then plan to develop a commercialization plan and seek funding to bring the above-mentioned products to their respective markets.

E. Technical Objectives

As noted in the preceding sections, the key limitation of most conventional approaches has to do with its inability to find the global-minimum solution for large-scale nonlinear programming problems. The overall objective of the OASIS-based approach is to develop a new solver that can overcome this limitation. For simplicity, this new solver will be referred to as the OASIS Pathfinder. The orbit response matrix values obtained from SPEAR3 model simulation and actual measurement will be used for validating OASIS Pathfinder's global-minimum solutions. The particular steps planned to take in developing OASIS Pathfinder are described in this section.

E.1 Overall Objective

A current requirement for running OASIS Pathfinder is that the user needs to specify the range *of s* and *p* for the program to scan uniformly to find the optimal pair. For each scanned pair of *s* and *p*, the objective functions are recorded for all iterations. Among all objective functions recorded, the one with the lowest value is identified and the corresponding pair of *(s,p)* is chosen to run OASIS Pathfinder. In the course of this study a number of alternative strategies for finding the optimal *(s,p)* more systematically will be investigated. After a workable search method is identified, the requirement for building an automated search method based on an expert-system approach[7] will be defined for further development in a Phase II project. Expert systems are computer programs that use codified expertise of human experts to provide expert-level performance without an expert present. Expert systems often involve if-then type rules and an inference engine to chain together logical reasoning that solves a problem.

For OASIS Pathfinder the expertise involves both in-depth knowledge of accelerators as well as limited knowledge in how OASIS and MJL work. Since OASIS Pathfinder is being used in an accelerator lattice error-finding problem, the knowledge needed for running OASIS Pathfinder mostly involves accelerator expertise because the various parameters involved have relationships that are known to experts in the field. For example, in solving the SPEAR3 storage ring lattice modeling problem, realistic bounds on the values of the magnet strengths will be imposed on the minimization process.

The fully developed expert system will enable OASIS Pathfinder to carry on a dialog with the user/operator, who is possibly a non-expert, to guide that person through the various tasks performed by the MJL element. Based on a logical chain of questions posed by OASIS Pathfinder and garnered through the experience of human experts, an operator will be able to safely and efficiently pilot the lattice modeling process through the steps found by OASIS Pathfinder toward the goal of finding the actual magnet strength values.

E.2 Specific Objectives

A four-step approach will be used in this Phase I project to develop OASIS Pathfinder and compare the global-minimum solution to be found by OASIS Pathfinder with some of the existing solutions obtained by the conventional methods for the lattice modeling problem in SPEAR3. The objectives to be reached in the four steps are:

1. Explore different means to search for a set of ‘warm’ start solutions and ‘max-jump-look’ paths that lead to the global-minimum solution. Simulated orbit response matrices from SPEAR3 will be used to develop automated processes with OASIS Pathfinder.

2. Develop the expert system that uses OASIS Pathfinder to find the magnet strength errors in the model. A number of test cases will be run that involve simulations using different sets of magnet strength errors in an ideal lattice.

3. Test the fully automated OASIS Pathfinder on actual measured SPEAR3 response matrices by comparing the global-minimum solution with solutions found by LOCO.

4. Investigate how the fully automated OASIS Pathfinder can be further developed to become a training tool for accelerator physicists and operators—simulated data and realistic constraints can be added to the modeling process and different aggregate objective functions can be studied. The aim is to find a way to make expert knowledge learned from simulation available to accelerator operators by capturing it in OASIS Pathfinder.

F. Phase I Work Plan

To implement the technical approach for achieving the objectives as started above, the following work plan is proposed:

1. Merge LOCO and OASIS Pathfinder into one integrated system in MATLAB.

2. Study means to develop hands-free automated search procedures to find optimal convergence control parameters *s* and *p* (Objective 1).

3. Incorporate lattice expertise together with an automated OASIS Pathfinder to produce prototype code for testing OASIS Pathfinder's ability to find the global-minimum solution with simulated data from a SPEAR3 model (Objective 2).

4. Investigate extensions to OASIS Pathfinder so it runs directly on measured orbit data (Objective 3).

5. Investigate, and identify the requirements, in the use of OASIS Pathfinder as a training tool for operators so that expert analysis can be brought into the control room at all times for finding faults without expert interventions (Objective 4).

6. Define requirements for further research and development leading to Phase II proposal.

7. Write final report.

G. Phase I Performance Schedule

The Phase I contract will be executed over a nine month time period as indicated below.

```
Task                                        Months From Project Start
                                            0   1   2   3   4   5   6   7   8   9
1. Integrate LOCO and OASIS Pathfinder       |-------|

2. Study means to find warm start solution      |------|
and max-jump-look path (Objective 1)

3. Make OASIS Pathfinder prototype
 and test OASIS Pathfinder on
 simulated data (Objective 2)                      |----------|

4. Test multiple lattice errors                          |--------------------|
 using measured orbit data (Objective 3)

5. Study OASIS Pathfinder training tools                        |--------------|
 (Objective 4)
```

6. Identify requirements Phase-II R&D |--------|

7. Write final report |--------------|

H. Related Research and R&D

While Dr. Martin Lee was an employee at SLAC, he developed the ideas of OASIS Pathfinder based on work performed on earlier CRADAs between SLAC and a few small businesses on their SBIR and STTR projects supported by DOE Office of Science. Those DOE SBIR/STTR projects are shown below.

H.1 Phase I Projects

Optimal Model-Based Fault Estimation and Correction for Particle Accelerators and Industrial Plants Using Combined Support Vector Machines and First Princiles— Pavilion Technologies, Inc. (2007).

Self-Validating Knowledge-Guided On-Line Optics and Orbit Correction in an Electron Storage Ring Using a Combined Neural Network and First-Principles Model—Pavilion Technologies, Inc. (2004).

Adaptive, Nonlinear Model Predictive Control for Accelerator Feedback Control Systems—Pavilion Technologies, Inc. (2000).

An Automatic Beam-Based System for Analyzing Accelerator Misalignment Problems—Sandia View Software, Inc. (1999).

Automatic Component Calibration and Error Diagnostics for Model-Based Accelerator Control—Vista Control Systems, Inc. (1998).

H.2 Phase II Projects

Development of PUNDA (*Parametric Universal Nonlinear Dynamics Approximator*) *Models for Self-Validating Knowledge-Guided Modeling of Nonlinear Processes in Particle Accelerators and Industry*—Pavilion Technologies, Inc. (2005).

Robust Optimal Adaptive, System Identification & Nonlinear Model Predictive Control Strategy for Accelerator Feedback Control System—Pavilion Technologies, Inc. (2001).

Intelligent Automated Tuning Systems Based on Hybrid Neural Networks—Physical Optics Corporation (1998).

Appendix - Setting Up the Framework for OASIS Pathfinder

In this proposal, the global-minimum solution refers to the bounded multi-variable point $(b_1, b_2, ..., b_n)$ having the overall lowest objective function value. To use OASIS Pathfinder, the first step is to transform the set of *n* bounded variables $(b_1, b_2, ..., b_n)$ to a set of free variables $(a_1, a_2, ..., a_n)$. The second step is to transform this set of free variables into a set of chosen m $(\geq n)$ "working" variables $(x_1, x_2, ..., x_m)$. Both of the transformations are done in the SFU element. In this section, a brief description of the FSU element of the OASIS Pathfinder is described.

In practice, the set of *n* variables bounded by constants can be transformed into another set of equal number of free variables that are bounded automatically. For example, the bounded variable, $\pi/2 > b_k > -\pi/2$, can be transformed into a free variable, $a_k = \tan(b_k)$. Similarly, the set of functional constraints can be absorbed by introducing another set of equal number free variables. Each of the free variables is accompanied by an auxiliary function which is added to the given objection function to become an aggregate objective function f_{aof}. Therefore, a square extended (m x m) Hessian matrix can always be formed with *m* equal to *n* plus the number free variables added, i.e., $m \geq n$.

Let the extended Hessian matrix evaluated at the start point $(a_1, ..., a_m)^{start}$ be denoted as H with its elements given by $h_{jk} = \dfrac{\partial^2 f_{aof}^{start}}{\partial a_j \partial a_k}$. Singular Value Decomposition of H gives

$$H = U \Lambda V^T ,$$

where U and V are unitary matrices, i.e., $U^{-1} = U^T$ and $V^{-1} = V^T$, and the elements of the diagonal Λ-matrix are the singular values[IV].

FSU first transforms the variables $\vec{a} = (a_1, ..., a_m)^T$ into the working variables $\vec{x} = (x_1, ..., x_m)^T$ such that $\vec{x} = V^T \vec{a}$. Then FSU introduces the path-control parameter *p* into the OASIS framework:

$$\vec{y} = U^T (\vec{J} - \vec{J}^{start}) + pU^T H \vec{a}^{start} ,$$

where $\vec{J}$ is the Jacobian of the objective function with its k^{th} element given by $\dfrac{\partial f_{aof}(a_1, ..., a_m)}{\partial a_k}$

for $k = 1, 2, ..., m$. Furthermore, since $\vec{a} = V\vec{x}$, the above equation can be expressed in terms of a set of functions that are used in the OASIS algorithm

[IV] *Degeneracy in the Hessian can be avoided by shifting the start point* $(a_1, ..., a_m)^{start}$ *slightly.*

$$\vec{y} = \vec{f}(\vec{x}) = U^T(\vec{J}(\vec{x}) - \vec{J}(\vec{x}^{start})) + pU^T HV\vec{x}^{start}\ .$$

Since $\vec{y}$ depends on the value of the path parameter, *p*, variations of the objective function along a path also depend on *p*. In addition, in order for the objective function value to be minimized, the Jacobian evaluated at the end point must approach zero, i.e., $\frac{\partial f_{aof}^{end}}{\partial a_k} \to 0$ for $k = 1,2,...,m$. To satisfy this condition, the desired value of $\vec{y}$ is chosen to be

$$\vec{y}^{desired} = U^T(-\vec{J}(\vec{x}^{start})) + pU^T HV\vec{x}^{start}\ .$$

Epilog

For completeness, a simple derivation of the OASIS Algorithm is described in this section.

In general, it is possible to transform a set of equations of the form

$$\begin{aligned}
&y_1 = f_1(x_1, x_2, ..., x_m)\\
&y_2 = f_2(x_1, x_2, ..., x_m)\\
&\cdots\\
&y_m = f_m(x_1, x_2, ..., x_m),
\end{aligned}$$

where $\vec{f} = (f_1, f_2, ..., f_m)^T$, into a matrix equation of the form $\vec{y} = F(\vec{x}, r)\vec{x}$. The transformation is accomplished via the introduction of a parameter, *r*, in this special way:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} \frac{tf_1}{x_1} & \frac{rf_1}{x_2} & \frac{rf_1}{x_3} & \cdots \\ \frac{rf_2}{x_1} & \frac{tf_2}{x_2} & \frac{rf_2}{x_3} & \cdots \\ \frac{rf_3}{x_1} & \frac{rf_3}{x_2} & \frac{tf_3}{x3} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix}$$

with $t = 1 - (m-1)r$, where *m* is the number of equations. Because of its simple form, this matrix equation can be easily inverted. After some algebraic manipulations, the resulting inverse matrix equation is given by the “Good Old” algebraic identity:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} \frac{\alpha x_1}{f1} & \frac{\beta x1}{f2} & \frac{\beta x1}{f3} & \cdots \\ \frac{\beta x2}{f1} & \frac{\alpha x2}{f2} & \frac{\beta x2}{f3} & \cdots \\ \frac{\beta x3}{f1} & \frac{\beta x3}{f2} & \frac{\alpha x3}{f3} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix}$$

where $\alpha = \frac{1+r}{1+mr}$ and $\beta = \frac{r}{1+mr}$. The form of this matrix equation suggests a simple recipe to find the solution for a given set of $y_k^{desired}$'s for $k = 1,2,...,m$—Look for the values of $(x_1, x_2, ..., x_m)$ such that when they are put into the inverse matrix and multiplied by the vector, $\vec{y}^{desired} = (y_1^{desired}, y_{21}^{desired}, ..., y_m^{desired})^T$, the computed x_k's values for $k = 1,2,...,m$ are the same as the given values, i.e.,

$$\begin{pmatrix} \frac{\alpha x_1}{f_1} & \frac{\beta x_1}{f_2} & \frac{\beta x_1}{f_3} & \cdots \\ \frac{\beta x_2}{f_1} & \frac{\alpha x_2}{f_2} & \frac{\beta x_2}{f_3} & \cdots \\ \frac{\beta x_3}{f_1} & \frac{\beta x_3}{f_2} & \frac{\alpha x_3}{f_3} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} y_1^{desred} \\ y_2^{desired} \\ y_3^{desired} \\ \vdots \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix}$$

which can be simplified to become

$$x_k \left\{ \frac{y_k^{desired}}{f_k} + s \left[\frac{y_k^{desired}}{f_k} - \frac{1}{m} \sum_{j=1}^{m} \frac{y_j^{desired}}{f_j} \right] \right\} \Rightarrow x_k \text{ for } k = 1,2,\ldots,m,$$

where $s = \frac{r}{1-mr}$. It is this observation that has led to the iterative formula in OASIS (section C.5).

Making OASIS Algorithm Adaptive

The aforementioned matrix formalism offers one unique feature for the iterative algorithm in OASIS. It is easy to vary the value of s adaptively along a solution path by considering *r* as an additional variable, $x_{m+1} = r$, and by adding another constraint equation to the given set of m equations such as

$$y_{m+1} = 1 + f_{obj}(x_1, x_2, ..., x_m).$$

an algorithm for varying the value of r at each iterative step has been developed. Note that the added function in the constraint (*m+1*) equation is defined such that the desired value of y_{m+1} is equal to 1, i.e., $y_{m+1}^{desired} = 1$ when $f_{obj}(x_1, x_2, ..., x_m) \to 0$.

In this variable r option, it can be shown (after some algebra) that the value of r from one step to the next is given by

$$r^{next} = r \frac{g_{m+1} - (m+1) r g_0}{1 - (m+1) r} .$$

where

$$g_k = \frac{y_k^{desired}}{f_k}$$

$$g_0 = \tfrac{1}{m+1} \sum_{k=1}^{m+1} g_k .$$

The corresponding iterative value of each component of $\vec{x}$ is given by

$$x_k^{next} = x_k \frac{g_k - (m+1) r g_0}{1 - (K+1) r}$$

for $k = 1, 2, ..., m$. It has been found that when the sequence of r^{next} values converges, the corresponding sequence of s^{step} values also converges. For the constraint function f_{m+1} shown above, at the end point, $s^{end} \to 0$. Under this condition, the changes in the working variables at each iteration are given by

$$\Delta x_k^{end}(o, p) = x_k^{end} - x_k = x_k (g_k - 1) \to 0 .$$

This condition implies that $g_k \to 1$, or

$$y_k \to y_k^{desired} \text{ for } k = 1, 2, ..., m .$$

Because of the simplicity of the iterative algorithm, the benefit is clear—the iterative process can be easily integrated into many of the existing modeling codes. Furthermore, since the OASIS technique is not limited to the number of equations and can be implemented in parallel with parallel processors, the technique lends itself naturally to parallelization. Another useful feature of the iterative algorithm is that it is equally applicable to finding the global-minimum solution for a set of complex functions as it does for a set of real functions.

References

[1] M.J. Lee, J.C. Sheppard, M. Sullenberger, M.D. Woodley, "*Models and Simulations*", LECTURE NOTES IN PHYSICS, Vol. 215, pp. 465-472 (1984).

[2] W.J. Corbett, M.J. Lee, V. Ziemann, "*A Fast Model Calibration Procedure for Storage Rings*", SLAC-PUB-6111, May 1993. 3pp. Presented at 1993 Particle Accelerator Conference (PAC 93), Washington, DC, 17-20 May, pp. 108-110 (1993).

[3] Martin J. Lee, Stephen Kleban, Scott Clearwater, Walter Scandale, Thomas Pettersson, Hartmut Kugler , Alain Riche , Michel Chanel , Eric Martensson, In-Ho Lin, "*Analysis of the Orbit Errors in the CERN Accelerators Using Model Simulation*", Presented at the Europhysics Conf. on Control Systems for Experimental Physics, Sep 28 - Oct 2, Villiars, Switzerland, (1987).

[4] J**.** Safranek and M**.** Lee**, "***Calibration of the X-ray ring quadrupoles, BPMs, and orbit correctors using the measured orbit response matrix***",** in the Proceedings of 4th European Particle Accelerator Conference (EPAC 94), London, England, 27 Jun - 1 Jul 1994, pp 1027-1029 **(**1994).

[5] A. Terebilo, "*Accelerator Modeling with MatLab Accelerator Toolbox*", Proceedings of the 2001 Particle Accelerator Conference, Chicago, IL, pp. 3203-3205 (2001).

[6] B. Sayyar-Rodsari, E. Hartman, C. Schweiger, M.J. Lee, Y.T. Yan, "*Efficient Modeling of Nonlinear Beam Optics Using Parametric Model Independent Analysis*", Particle Accelerator Conference (PAC 05), May 16-20, Knoxville, Tenn., pp. 1-3 (2005).

[7] Martin J. Lee, "*Model Based Expert Systems For Linac Computer Controls*" SLAC-PUB-4724, Sep 1988. 4pp. Invited talk at Linear Accelerator Conf., Oct 3-7, Williamsburg, Va., (1988).