NLPy Documentation

Release 0.1

Dominique Orban

July 21, 2009

CONTENTS

Release 0.1

Date July 21, 2009

CHAPTER

ONE

INTRODUCTION

NLPy is a Python package for numerical optimization. It aims to provide a toolbox for solving linear and nonlinear programming problems that is both easy to use and extensible. It is applicable to problems that are smooth, have no derivatives, or have integer data.

NLPy combines the capabilities of the mature AMPL modeling language with the high-quality numerical resources and object-oriented power of the Python programming language. This combination makes NLPy an excellent tool to prototype and test optimization algorithms, and also a great teaching aid for optimization.

NLPy can read optimization problems coded in the AMPL modeling language. All aspects of a problem can be examined and the problem solved. Individual objective and constraint functions, together with their derivatives (if they exist) can be accessed transparently through an object-oriented framework.

NLPy is extensible and new algorithms can be built by assembling the supplied building blocks, or by creating new building blocks. Existing building blocks include procedures for solving symmetric (possibly indefinite) linear systems via symmetric factorizations or preconditioned iterative solvers, and much more.

1.1 Overview

NLPy is a collection of tools and interfaces for implementing and prototyping optimization algorithms. It is a set of Python modules and classes that support sparse matrices, nonlinear optimization methods, and the efficient solution of large sparse linear systems, especially those occuring in the course of optimization algorithms (e.g., symmetric indefinite systems).

The purpose of NLPy is to offer an environment in which implementing, testing, prototyping, experimenting with, and modifying and creating innovative optimization algorithms for large-scale constrained problems is a moderately easy task. We feel that the environment should be useful, simple, and intuitive enough that programmers need only concentrate on the logic of the algorithm instead of the intricacies of the programming language. We believe that NLPy is appropriate for teaching, for learning, and also for bleeding edge research in large-scale numerical optimization. This is achieved by providing the heavy-duty number-crunching procedures in fast, low-level languages such as C, Fortran 77, and Fortran 90/95.

NLPy aims to

- represent the bleeding edge of research in numerical (differentiable or not) optimization,
- provide a number of low-level tools upon which algorithms may be built, with the intent of solving potentially large problems,
- use sparse matrix data structures to do so,
- provide tools that are useful in the assessment of the performance of optimization algorithms.

1.2 Structure of NLPy

NLPy is designed as an object-oriented Python layer over lower-level subprograms. The subprograms, written in C, Fortran 77, and Fortran 90/95, implement the most numerically-intensive tasks.

These low-level subprograms include MA27 and MA57 for the multifrontal solution of symmetric linear systems, ICFS for preconditioned conjugate gradients with limited-memory Cholesky factorization, and GLTR for the solution of trust-region subproblems, to name a few.

All the sparse matrix capabilities in NLPy are based on the *Pysparse <http://pysparse.sf.net>* package.

The main purpose of NLPy is to facilitate access and manipulation of optimization problems. Therefore, an interface to the AMPL modeling language was designed. It allows access to components of models and to take advantage of the automatic differentiation abilities of AMPL.

The above and the design of the Python language combine with interfaces written in C and in which only pointers are exchanged, leads to an environment suitable for the efficient solution of large-scale problems.

Some pages of this documentation display equations via the jsMath package. They should look reasonably good with most setups but the best rendering is obtained by installing the TeX fonts. Please refer to http://www.math.union.edu/~dpvc/jsMath/users/welcome.html.

CHAPTER

TWO

INDICES AND TABLES

- *Index*
- *Module Index*
- *Search Page*

MODULE INDEX

N

INDEX

N