LLNL-JRNL-656057

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Derivative-free optimization for parameter estimation in computational nuclear physics

S. Wild, J. Sarich, N. Schunck

June 23, 2014

Journal of Physics G: Nuclear and Particle Physics

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Derivative-free optimization for parameter estimation in computational nuclear physics

Stefan M Wild,[1] Jason Sarich,[1] and Nicolas Schunck[2]

[1] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

[2] Physics Division, Lawrence Livermore National Laboratory, Livermore, CA, 94551, USA

E-mail: `wild@anl.gov`

Abstract. We consider optimization problems that arise when estimating a set of unknown parameters from experimental data, particularly in the context of nuclear density functional theory. We examine the cost of not having derivatives of these functionals with respect to the parameters. We show that the POUNDERS code for local derivative-free optimization obtains consistent solutions on a variety of computationally expensive energy density functional calibration problems. We also provide a primer on the operation of the POUNDERS software in the Toolkit for Advanced Optimization.

Submitted to: *J. Phys. G: Nucl. Phys.*

1. Introduction

Much of the intellectual capital in nuclear physics is invested in *forward problems* whereby a theory or model is posited, assumptions are added (to improve accuracy) and relaxed (to improve universality), and hypotheses are tested. An example can be seen with the *ab initio* approach to nuclear structure. Here, the form of the Hamiltonian is derived from chiral effective field theory [1]. One of the basic assumptions is that the nuclear many-body problem can be solved non-relativistically with nucleons as basic degrees of freedom [2]. The truncation in chiral perturbation and the inclusion or neglect of three- and N-body forces are some of the hypotheses that can be tested by comparing model predictions with experimental data [3]. Today, work on forward problems invariably extends along a computational axis as well: models are made computationally tractable and numerically implemented, and computational performance and efficiency are improved.

Equally important is the *inverse problem*: given data (experimental or otherwise) and a forward model, free parameters for the model are determined based on the data. This aspect is especially important in the context of the nuclear shell model or nuclear density functional theory (DFT). Indeed, these approaches to the nuclear many-body problem are a notch more phenomenological than *ab initio* theory: they rely on an effective interaction, or alternatively an effective energy density, that is not predetermined from some underlying theory [4, 5]. Obtaining a robust and reliable estimate of the free parameters is essential since nuclear DFT is widely used in a number of applications, from large-scale surveys of nuclear properties [6] to fission [7], and will play a critical role in the physics explored at the future Facility for Radioactive Ion Beams [8]. In this paper we focus on numerical optimization, one aspect of inverse problems that often presents a bottleneck when working with computationally expensive forward models.

Formally, we assume a collection of n_d components of scalar data $\mathbf{d} = (d_1, \ldots, d_{n_d})$ based on which we must determine values of n_x real parameters $\mathbf{x} = (x_1, \ldots, x_{n_x})$. It is often convenient to think of a model m as generating the observable d_i based on the set of real parameters $\mathbf{x}$ and a set of hyperparameter values, $\boldsymbol{\nu}_i \in \mathbb{R}^p$, which represent known values needed to compute the forward problem (such as the number of protons and the number of neutrons). Thus the inverse problem is to determine the value(s) $\mathbf{x}_*$ such that

$$m\left(\mathbf{x}_*; \boldsymbol{\nu}_i\right) \approx d_i \qquad i = 1, \ldots, n_d. \tag{1}$$

The level of agreement dictated by "$\approx$" can depend on the uncertainties in the model m, the parameters $\mathbf{x}_*$, and/or the data $\mathbf{d}$.

Parameter estimation typically depends on the distribution of the errors between reality and the data. Given an assumed distribution of these errors, a common approach in both Bayesian and frequentist parameter estimation is to determine the maximum likelihood estimate (or maximum *a posteriori* estimate for Bayesians) for the parameters,

namely, those values that, given the values of the data, are most likely under the assumed distribution(s).

Regardless of the distribution or whether the errors are independent of one another, one generally arrives at an optimization problem. For example, if the model m is correct, the errors are independent, and the errors are Gaussian with mean zero and known variance $w_i^2 > 0$, then maximizing the log-likelihood (and hence the likelihood) is equivalent to solving

$$\min_{\mathbf{x}\in\mathbb{R}^{n_x}} \sum_{i=1}^{n_d} \left(\frac{m\left(\mathbf{x};\boldsymbol{\nu}_i\right) - d_i}{w_i} \right)^2 . \tag{2}$$

If the errors are correlated, then (2) becomes

$$\min_{\mathbf{x}\in\mathbb{R}^{n_x}} \sum_{i=1}^{n_d} \sum_{j=1}^{n_d} w_{ij} \left(m\left(\mathbf{x};\boldsymbol{\nu}_i\right) - d_i\right) \left(m\left(\mathbf{x};\boldsymbol{\nu}_j\right) - d_j\right), \tag{3}$$

where w_{ij} captures the (inverse) covariance between the errors of observables i and j.

The objective in (2) differs from χ^2 objectives by a constant factor (related to the degrees of freedom, $n_d - n_x$), and hence the solution of (2) with an appropriate w (see [9]) arises throughout computational science. Similar objective functions to be optimized can be derived for a wide variety of other distributions, including cases where the variances $\{w_i^2 : i = 1, \ldots, n_d\}$ are unknown or specified only by a diffuse prior. These latter cases are especially relevant to nuclear DFT, since there is little *a priori* information about the errors on computed observables. Likewise, if constraints on the parameters are imposed (e.g., to break symmetries or satisfy physical realities), the optimization problem can be modified to consider the restriction $\mathbf{x} \in \Omega \subset \mathbb{R}^{n_x}$.

As we will see, the derivatives $\frac{\partial}{\partial x_j} m\left(\mathbf{x};\boldsymbol{\nu}_i\right)$ play a crucial role in identifying solutions to such optimization problems. The solution of these problems is especially difficult when such derivatives are not made available to the optimization solver; such "derivative-free" situations are pervasive when evaluating $m(\cdot;\cdot)$ entails running a legacy computer simulation. In Section 2 we review methods for solving problems of the form (2) in both the unconstrained and bound-constrained case. We focus on derivative-free approaches for calibrating energy density functionals and review the POUNDERS method for solving such problems. In Section 3, we examine some of the optimization problems from the UNEDF0, UNEDF1, and UNEDF2 parameterizations [10, 11, 12]. We show that despite the potentially multimodal nature of the objective function considered, the solutions obtained by POUNDERS are surprisingly robust to the choice of starting point. Our results also offer a further empirical validation of the sensitivity analysis conducted in the UNEDF studies. Section 4 returns to the matter of derivatives. Through a specific example involving nuclear masses, we show that the availability of derivatives with respect to even a few parameters can improve the efficiency of the optimization. In Section 5 we provide details on the usage of the POUNDERS method as well as general tips for solving such problems.

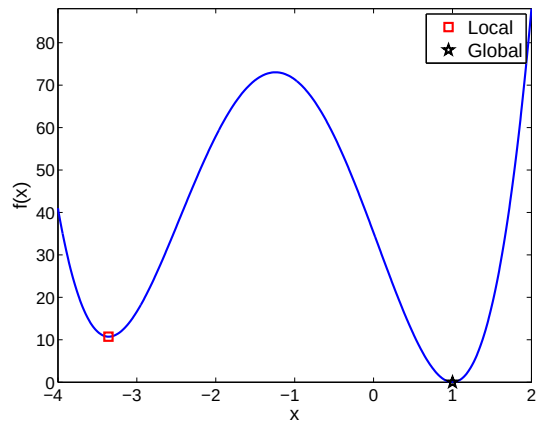

Figure 1. Example of the one-dimensional nonlinear least squares problem $f(x) = \sum_{i=1}^{n_d} F_i(x)^2$ with $n_d = 3$, $F_i(x) = (x - \nu_i)^2 - d_i$, $\boldsymbol{\nu} = (1, 1.1, 1.5)$, and $\mathbf{d} = (1 - \boldsymbol{\nu})^2$.

2. Optimization-Based Approaches for Parameter Estimation

We will restrict our focus to (2), the most common form of optimization problem encountered in parameter estimation, but we note that much of our discussion applies for more general objectives. In many practical applications, in particular the optimization of energy densities in DFT, the model m is a nonlinear function of the parameters $\mathbf{x}$; hence the problem in (2) is that of *nonlinear least squares* (NLS),

$$\min_{\mathbf{x}\in\Omega} f(x) = \sum_{i=1}^{n_d} F_i(\mathbf{x})^2, \tag{4}$$

where the vector mapping $\mathbf{F} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_d}$ captures the weighted residuals and Ω can correspond either to all of $\mathbb{R}^{n_d}$ (called the "unconstrained" case) or to some subset of $\mathbb{R}^{n_d}$ (e.g., when non-negativity, $x_i \geq 0$, is imposed for some parameter x_i).

Solutions to (4) are referred to as *global* minimizers, and such points $\hat{\mathbf{x}}$ have the property that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \Omega$. However, finding global solutions for arbitrary functions $\mathbf{F}$ is generally intractable. Consequently, optimization methods that promise global solutions are either making problem-specific assumptions (e.g., that $\mathbf{F}$ is a linear function of $\mathbf{x}$ or that Ω contains a finite number of points), guaranteeing global optimality only asymptotically (and thus never achieved in practice), or overstating their claims.

As a result, we follow the approach of seeking *local* minimizers, which cannot be improved upon locally: $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \Omega$ close to $\hat{\mathbf{x}}$. Figure 1 illustrates that even simple, one-dimensional ($n_x = 1$) NLS problems can have multiple local minimizers, with potentially all but one of these being nonglobal minimizers. Hence, one must apply local optimization methods to such problems with caution; Section 3 returns to this topic.

2.1. Derivatives and Methods for Nonlinear Least Squares

When the residual vector $\mathbf{F}$ is differentiable, the gradient of f with respect to the parameters $\mathbf{x}$ is $\nabla_{\mathbf{x}} f(\mathbf{x}) = 2\sum_{i=1}^{n_d} F_i(\mathbf{x})\nabla_{\mathbf{x}} F_i(\mathbf{x})$ and plays a crucial role in local optimality conditions. In the unconstrained case, a necessary condition for local minimizers is that the gradient of the function disappear, $\nabla_{\mathbf{x}} f(\mathbf{x}) = 0$. In the constrained case, things are slightly more complex. Here, we focused on one of the simplest cases, when bound constraints

$$\Omega = \{\mathbf{x} \in \mathbb{R}^{n_x} : l_i \leq x_i \leq u_i, i = 1, \ldots, n_x\} \tag{5}$$

are the only ones present. In the bound-constrained case, a necessary condition is that $\hat{\mathbf{x}} \in \Omega$ and that the components of the gradient satisfy

$$\frac{\partial f(\hat{\mathbf{x}})}{\partial x_i} \quad \begin{cases} = 0 & \text{if } l_i < \hat{x}_i < u_i \\ \geq 0 & \text{if } \hat{x}_i = l_i \\ \leq 0 & \text{if } \hat{x}_i = u_i \end{cases} \qquad i = 1, \ldots, n_x.$$

We say that a bound (or parameter, in this case) is "active" if the parameter attains the bound (e.g., $\hat{x}_i = l_i$ or $\hat{x}_i = u_i$).

In both the unconstrained and constrained cases, the derivatives $\nabla_{\mathbf{x}} f$ (and hence $\nabla_{\mathbf{x}} F_i$) play a vital role in guaranteeing decrease of the objective f, accelerating convergence, and recognizing a solution. In most practical problems, the residual $F_i(\mathbf{x})$ invariably depends on the output of a numerical or physical simulation, and hence such derivatives may not readily be available.

When these residuals are defined by a computer code free of proprietary libraries and control flow logic that may introduce discontinuities, *algorithmic differentiation (AD)* [13] can be an invaluable technique. AD tools generate source code—often automatically—by propagating the chain rule through the original code. Under infinite-precision arithmetic, derivatives from AD are exact. Alternatively, one can apply numerical differentiation (ND) to obtain approximate derivatives. With ND, however, one must take great care in selecting an appropriate finite-difference stepsize on noisy simulations [14]; also, the cost of obtaining a full gradient using ND is generally at least n times the cost of a function evaluation, a potentially significant expense.

When derivatives are not available from the simulation or through AD, an alternative to ND is to employ a *derivative-free* optimization method [15], that is, one that relies only on evaluations of the function F_i (or the aggregate objective function, f). Because they are provided less information about the objective, such methods generally require a greater number of function evaluations than do derivative-based methods.

We illustrate this concept by examining a typical example of an unconstrained NLS problem, `chwirut1.c`, which is included in the TAO distribution; see Section 5. This problem is based on the chwirut1 dataset [16], with the extended version consisting of $n_x = 6$ parameters and $n_d = 428$ observables:

$$\min_{\mathbf{x}\in\mathbb{R}^6} f(\mathbf{x}) = \sum_{i=1}^{214} \left(\frac{e^{-x_1\nu_i}}{x_2 + x_3\nu_i} - d_i \right)^2 + \sum_{i=1}^{214} \left(\frac{e^{-x_4\nu_i}}{x_5 + x_6\nu_i} - d_i \right)^2, \tag{6}$$

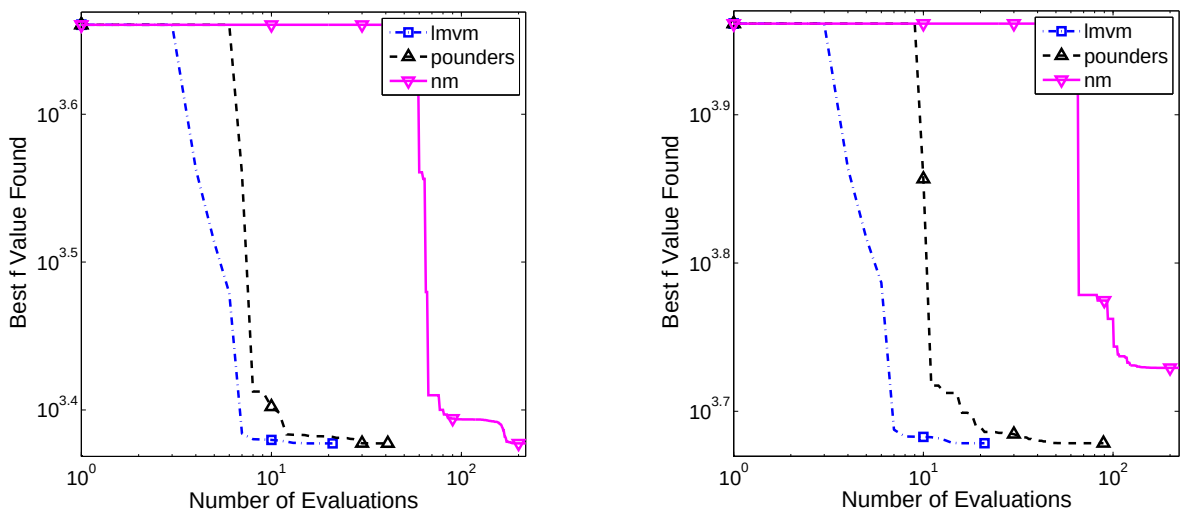

Figure 2. Comparing the performance (log log scale) of three TAO solvers (limited-memory variable metric, POUNDERS, Nelder Mead) on NLS test problems of the form (6). The left plot is the original version with $n_x = 3$ variables; the right plot is the extended version with $n_x = 6$ variables.

where $\{\nu_i : i = 1, \ldots, 214\}$ are hyperparameters (metal distances) and $\{d_i : i = 1, \ldots, 214\}$ are experimental ultrasonic responses. The original version does not include the second sum and thus has $n_x = 3$ parameters and $n_d = 214$ observables.

We solved both versions using several of the algorithms available in TAO; the results are shown in Figure 2. The limited-memory variable metric (LMVM) algorithm is a quasi-Newton method that utilizes first-derivative information, and Nelder-Mead is a simplex-based derivative-free method; neither method takes into account the sum-of-squares structure present in (4). POUNDERS is a derivative-free method that exploits the availability of the residual vector $\mathbf{F}$ rather than just the single aggregate f; we refer the reader to [17] for a mathematical description of the algorithm. Figure 2 shows that when measured in terms of the number of function evaluations, the derivative-based method LMVM reduces the f value considerably faster than do the derivative-free methods. If the combined expense of a function and gradient evaluation is roughly the same as two function evaluations, the advantage of LMVM over POUNDERS persists. However, if the combined cost is roughly the same as $n_x + 1$ function evaluations (as would happen if using LMVM with gradients approximated by ND and forward differences), then POUNDERS is faster. In all these scenarios, the derivative-free method that does not exploit the structure inherent in (4) performs significantly worse.

2.2. POUNDERS for Calibrating Energy Density Functionals

Under the UNEDF [18] and NUCLEI [19] collaborations, a wide variety of parameter estimation problems arose where derivatives of the residuals were unavailable [10, 11, 12, 20, 21]. Here we focus on some of the results obtained when calibrating Skyrme

Table 1. Results for unconstrained and bound-constrained derivative-free methods starting from SLy4 on the UNEDF0 problem ($n_x = 12$, $n_d = 108$, $n_N = 72$) [10].

		UNEDFnb [10]	**UNEDF0** [10]	
Method	Nelder-Mead	POUNDERS	POUNDERS (bounds)	$\boldsymbol{\sigma}$
ρ_c	0.16155537	0.15104627	0.16052598	0.001
E^{NM}/A	-16.115363	-16.063211	-16.05589	0.055
K^{NM}	234.64613	337.87808	<u>230</u>	–
a_{sym}^{NM}	31.919478	32.454973	30.54294	3.058
L_{sym}^{NM}	46.186671	70.218532	45.080413	40.037
$1/M_s^*$	1.4306113	0.95727984	<u>0.9</u>	–
$C_0^{\rho\Delta\rho}$	-78.133526	-49.513502	-55.260592	1.697
$C_1^{\rho\Delta\rho}$	4.4779896	33.52886	-55.622579	56.965
V_0^n	-240.42409	-176.79601	-170.37424	2.105
V_0^p	-252.81184	-203.25488	-199.20225	3.351
$C_0^{\rho\nabla J}$	-92.272157	-78.456352	-79.530829	3.423
$C_1^{\rho\nabla J}$	-27.615105	63.993115	45.63019	29.460
$f(\hat{\mathbf{x}})$	106.23493	41.865965	67.309821	
n_f	300	268	300	

energy density functionals where HFBTHO [22] was the underlying simulator. As a reminder, HFBTHO solves the Hartree-Fock-Bogoliubov equations for generalized Skyrme functionals under the assumption of axial and time-reversal symmetry. These built-in symmetries make HFBTHO particularly adapted to large-scale surveys of nuclear properties and optimization problems [6, 23].

Table 1 summarizes the solutions obtained during UNEDF0 computational experiments [10]. Each of the three runs was started from the SLy4 parameterization [24, 25] and, because of the computational expense of evaluating $n_d = 108$ observables across $n_N = 72$ even-even nuclei, run for a maximum of 300 evaluations. The first two columns represent the solutions from the Nelder-Mead and POUNDERS codes in TAO when solving the unconstrained problem, whereas the "POUNDERS (bounds)" column shows the POUNDERS results when bound constraints (see Table 2) are enforced for the 6 parameters that correspond to nuclear matter properties, for which relatively strict constraints exist. These bounds were added after it was noticed that the nuclear incompressibility parameter in the unconstrained optimization had a large value that was incompatible with experimental data.

As seen from the number of function evaluations, n_f, performed, only the unconstrained POUNDERS terminated short of the budget (because of a measure of criticality, similar to $\|\nabla f(\hat{\mathbf{x}})\| \leq \epsilon$, being satisfied); however, the bound-constrained POUNDERS was also seeing negligible decreases at the time the budget was exhausted. Since the bound-constrained problem involves a smaller parameter space, the associated global minimum will necessarily have a larger function value; this is borne out in the best functions values, $f(\hat{\mathbf{x}})$, obtained by POUNDERS on these two problems. In contrast, as with the test function in Figure 2, the Nelder-Mead performance is markedly worse. At the time of the UNEDF0 runs, each evaluation of f required 12 minutes of wall time on

Table 2. UNEDF bound constraints and scaling intervals.

	UNEDF0 [10]		**UNEDF1** [11] **UNEDF2** [12]		**Scaling Bounds**	
	l	**u**	**l**	**u**	$\mathbf{s}^l$	$\mathbf{s}^u$
ρ_c	0.15	0.17	0.15	0.17	0.14	0.18
E^{NM}/A	-16.2	-15.8	-16.2	-15.8	-17	-15
K^{NM}	*190*	*230*	*220*	*260*	170	270
a_{sym}^{NM}	28	36	28	36	27	37
L_{sym}^{NM}	40	100	40	100	30	70
$1/M_s^*$	0.9	1.5	0.9	1.5	0.8	2.0
$C_0^{\rho\Delta\rho}$	$-\infty$	∞	$-\infty$	∞	-100	-40
$C_1^{\rho\Delta\rho}$	$-\infty$	∞	$-\infty$	∞	-100	100
V_0^n	$-\infty$	∞	$-\infty$	∞	-350	-150
V_0^p	$-\infty$	∞	$-\infty$	∞	-350	-150
$C_0^{\rho\nabla J}$	$-\infty$	∞	$-\infty$	∞	-120	-50
$C_1^{\rho\nabla J}$	$-\infty$	∞	$-\infty$	∞	-100	50

72 cores; thus each 300-evaluation run required 2.5 days.

For the bound-constrained problem, two of the $n_x = 12$ parameters (K^{NM} and $1/M_s^*$) were active and hence restricted by the enforced bounds. Parameter values that are active are <u>underlined</u> in each of the tables in this paper. We note that for subsequent studies, the bound on K^{NM} was relaxed based on this analysis; see Table 2. The final column in Table 1 shows the standard deviations $\boldsymbol{\sigma}$ computed for each optimal parameter value; see [10] for details of the computation of $\boldsymbol{\sigma}$.

For UNEDF1, the number of nuclei and number of observables were increased, with the resulting solution shown in the last column of Table 3. A similar run (the UNEDF1ex column) was performed with an additional parameter $0 \leq \alpha_{ex} \leq 1$ multiplying the exchange Coulomb part of the functional. This parameter was added with the intent of simulating many-body correlation effects for the Coulomb term, and early work suggested it could significantly improve reproduction of masses [26]. The parameter α_{ex} was treated as a free parameter (with bound constraints corresponding to $[0, 1]$). Adding an additional parameter without increasing the amount of data should result in an objective value no worse than when that parameter is held fixed. Although this result cannot be guaranteed in practice when doing local optimization from arbitrary starting points, Table 3 shows that this was indeed the case for POUNDERS runs starting from UNEDF0. However, the improvement of the fit was deemed too marginal to justify introducing an empirical parameter. We note that moving from UNEDF0 to UNEDF1, the active parameters changed; see the discussion in [11].

3. Consistency of Local Solutions

As discussed in Section 2, using local optimization methods has the benefit of substantially reducing the number of expensive simulations performed, when compared with global optimization methods. This benefit, however, must be weighed against risks

Table 3. UNEDF1 ($n_d = 115$, $n_N = 79$) results obtained by POUNDERS starting from the UNEDF0 parameterization. α_{ex} was fixed at its nominal value of 1.0 for UNEDF0 and UNEDF1 and treated as a free parameter (restricted to [0, 1]) in UNEDF1ex.

	UNEDF1ex [11]	**UNEDF1** [11]	$\boldsymbol{\sigma}$
ρ_c	0.15836673	0.15870677	0.00042
E^{NM}/A	-15.8	-15.8	–
K^{NM}	220	220	–
a_{sym}^{NM}	28.383952	28.986789	0.604
L_{sym}^{NM}	40	40.00479	13.136
$1/M_s^*$	1.0018717	0.99242333	0.123
$C_0^{\rho\Delta\rho}$	-44.601636	-45.135131	5.361
$C_1^{\rho\Delta\rho}$	-180.95647	-145.38217	52.169
V_0^n	-187.46859	-186.0654	18.516
V_0^p	-207.20942	-206.57959	13.049
$C_0^{\rho\nabla J}$	-74.339131	-74.026333	5.048
$C_1^{\rho\nabla J}$	-38.837179	-35.658261	23.147
α_{ex}	0.8135508	1.0	–
$f(\hat{\mathbf{x}})$	49.341359	51.058424	
n_x	13	12	
n_f	253	218	

associated with being dependent on the initial point from which a local run is started. We now revisit some of the runs in the previous section and test the robustness of POUNDERS under changes to the starting point, the simulation code, and the data.

In each case, we find that POUNDERS obtains consistent (relative to the original reported uncertainties) solutions. Possible explanations of this (beyond being sufficiently "lucky") include the following

- POUNDERS is relatively robust and tends to avoid getting stuck in poor local minimizers.
- The starting points are in reasonable parts of the parameter space and are thus conducive to yielding the same local minimizer/basin of attraction for POUNDERS.
- The data $\mathbf{d}$ and model m result in an objective function that is not very multimodal in this part of the parameter space.

We hypothesize that the likely reason is some combination of the above, but these results provide some confidence in the use of POUNDERS for this class of problems

3.1. UNEDF0, revisited

The HFBTHO code has undergone several changes since the version used for the UNEDF0 optimization in [10]. In particular, different initialization schemes of the HFB problem have been implemented, the numerical accuracy of the direct Coulomb potential has been improved, and a small bug on the rearrangement term for the pairing field has been fixed; see [22]. These changes result in minimal differences (often at the level of only a few keV on binding energies) to most observables used in the UNEDF0

Table 4. Rerun of POUNDERS on the UNEDF0 problem ($n_d = 108$, $n_N = 72$) using HFBTHO code (Ver 201) from two different starting points. The scaled difference columns represent the difference between the final value found and the original UNEDF0 parameterization, scaled by the uncertainties σ_i in Table 1.

	Starting from SLy4		**Scaled Diff.**	**Starting from SKM***		**Scaled Diff.**
	initial	*final*		*initial*	*final*	
ρ_c	0.159539	**0.160**486	-0.03954	0.160319	**0.160**435	-0.09106
E^{NM}/A	-15.9721	**-16.0**685	-0.2285	-16	**-16.0**73	-0.3119
K^{NM}	229.901	**<u>230</u>**	–	216.658	**<u>230</u>**	–
$a_{\rm sym}^{NM}$	32.0043	**3**1.3393	0.2604	30.0324	**3**1.7221	0.3856
$L_{\rm sym}^{NM}$	45.9618	54.2493	0.2290	45.7704	60.4725	0.3844
$1/, M_s^*$	1.43955	**<u>0.9</u>**	–	1.26826	**<u>0.9</u>**	–
$C_0^{\rho\Delta\rho}$	-76.9962	**-55.2**344	0.01545	-68.2031	**-55**.7348	-0.2794
$C_1^{\rho\Delta\rho}$	15.6571	-64.1619	-0.1499	17.1094	-70.4274	-0.2599
V_0^n	-285.84	**-170**.796	-0.2003	-280	**-170**.788	-0.1966
V_0^p	-285.84	**-19**7.782	0.4238	-280	**-19**8.038	0.3474
$C_0^{\rho\nabla J}$	-92.25	**-7**7.9436	0.4637	-97.5	**-79**.2915	0.06990
$C_1^{\rho\nabla J}$	-30.75	27.4519	-0.6171	-32.5	**4**9.5737	0.1339
$f(\hat{\mathbf{x}})$	1188.75	**67**.9034		24814.1	**67**.5738	
n_f		235			150	

calibration problem. Overall, the function value obtained at UNEDF0 is roughly 67.985 for the latest version (Ver 201) of HFBTHO compared with 67.310 (see Table 1) for the version used in [10].

Although small, these differences imply that the UNEDF0 parametrization no longer satisfies the optimality conditions when computed with the new version of HFBTHO. It is, therefore, natural to ask whether additional optimization using this code version results in substantial changes. In fact, this points to the general problem of the sensitivity of optimization results on starting points: if one begins additional optimization starting from UNEDF0, or SLy4, or any other starting point, will the resulting parameterization substantially differ from UNEDF0?

In Table 4, we report the results of the optimization obtained from two very different starting points, the SLy4 parametrization of [25] used in our original UNEDF0 paper, and a starting point strongly inspired by the SkM* parametrization of [27]. Since the binding energy per nucleon of SkM* is out of our bounds, we fixed it arbitrarily at -16 MeV; similarly, SkM* does not come with any prescription for pairing strengths, which we fixed at -280 MeV for both protons and neutrons. Table 4 shows that in both cases—and despite SLy4 and SkM* being very different from UNEDF0 and from one another—similar solutions are found. In fact, as the scaled difference column $((\hat{x}_i^{\rm final} - \hat{x}_i^{\rm UNEDF0})/\sigma_i)$ shows, the two solutions are both well within a single standard deviation of UNEDF0 (based on the uncertainties $\boldsymbol{\sigma}$ reported in Table 1).

3.2. UNEDF1, revisited

We now consider the effect of changing the data $\mathbf{d}$ employed in the NLS optimization of UNEDF1. We begin by motivating an estimate of the effect of this change on the optimal parameter values $\hat{\mathbf{x}}$.

Formally, let $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ minimize $f^0(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2$ as in (4). Now suppose that the residual $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^{n_d}$ undergoes a change by $\boldsymbol{\epsilon} \in \mathbb{R}^{n_d}$, for example, because each normalized datum $\frac{d_i}{w_i}$ is changed to $\frac{d_i}{w_i} + \epsilon_i$. A second-order Taylor expansion of $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x}) + \boldsymbol{\epsilon}\|_2^2$ about $\hat{\mathbf{x}}$ is

$$\begin{aligned} f(\mathbf{x}) \approx\ & f(\hat{\mathbf{x}}) + 2\boldsymbol{\epsilon}^T \hat{J}(\mathbf{x} - \hat{\mathbf{x}}) \\ & + \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^T \left(\nabla^2 f^0(\hat{\mathbf{x}}) + 2\sum_{i=1}^{n_d} \epsilon_i \nabla^2 F_i(\hat{\mathbf{x}}) \right) (\mathbf{x} - \hat{\mathbf{x}}), \end{aligned}$$

where $\hat{J}$ denotes the Jacobian matrix $[\frac{\partial F_i(\hat{\mathbf{x}})}{\partial x_j}]_{i,j}$ and we have used the first-order optimality condition $\nabla f^0(\hat{\mathbf{x}}) = 2\hat{J}^T\mathbf{F}(\hat{\mathbf{x}}) = 0$. When $\boldsymbol{\epsilon}$ is small, this quadratic will be convex and hence minimized at

$$\begin{aligned} \mathbf{x}_\epsilon - \hat{\mathbf{x}} &= 2\left(\nabla^2 f^0(\hat{\mathbf{x}}) + 2\sum_{i=1}^{n_d} \epsilon_i \nabla^2 F_i(\hat{\mathbf{x}}) \right)^{-1} \hat{J}^T \boldsymbol{\epsilon} \\ &= 2\left(\nabla^2 f^0(\hat{\mathbf{x}})\right)^{-1} \hat{J}^T \boldsymbol{\epsilon} + \mathcal{O}(\|\boldsymbol{\epsilon}\|^2). \end{aligned}$$

When $\mathbf{F}(\hat{\mathbf{x}})$ is small, the Hessian $\nabla^2 f^0(\hat{\mathbf{x}})$ is well-approximated by $2\hat{J}^T\hat{J}$, which yields the approximation

$$\tilde{\mathbf{x}}_\epsilon = \hat{\mathbf{x}} + \left(\hat{J}^T\hat{J}\right)^{-1} \hat{J}^T \boldsymbol{\epsilon} \tag{7}$$

of the new optimal solution for $f = \|\mathbf{F}(\mathbf{x}) + \boldsymbol{\epsilon}\|_2^2$.

We apply this estimate to the UNEDF1 problem when additional nuclear mass data is added for the 17 new neutron-rich, even-even nuclei measured in [28]. We refer to this new data as the Argonne masses (AM); further details of the new observables can be found in [29, Supplementary material].

With the data vector $\mathbf{d}$ now containing $n_d = 132$ components, we estimate the effect of including the new observables by considering the vector $\boldsymbol{\epsilon} \in \mathbb{R}^{n_d}$ consisting of zeros, except in the 17 components corresponding to the new observables. For these new observables, we take $\epsilon_i = F_i(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ is the UNEDF1 parameterization. For each of the 10 inactive parameters of UNEDF1, Figure 3 illustrates the interval corresponding to the UNEDF1 parameter value and a half standard deviation ($\pm\frac{\sigma_i}{4}$, where $\boldsymbol{\sigma}$ is reported in Table 3). The figure shows that the estimator (7) predicts the new optimal values to differ from UNEDF1 in only minor ways, each new value being within $\frac{\sigma_i}{2}$ of UNEDF1. Also shown are the actual optimal values as found by POUNDERS when the new data is included in an optimization begun from UNEDF1 (see Table 5). These actual values are also within $\frac{\sigma_i}{2}$ of UNEDF1, with the predictions in (7) generally indicating the correct direction of the change (with the exception of $C_1^{\rho\nabla J}$).

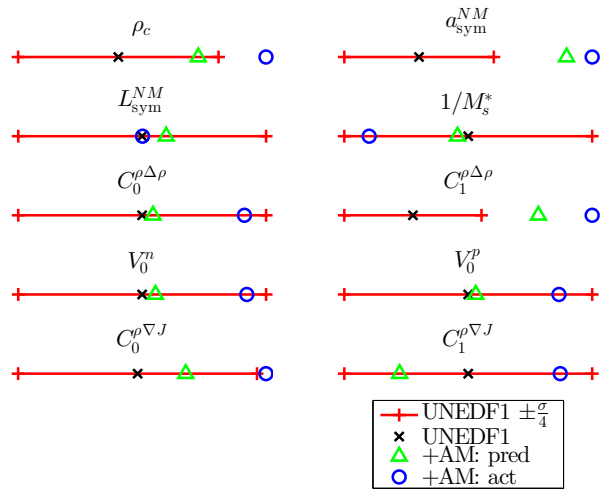

Figure 3. (Half standard deviation) Intervals for the inactive parameters in UNEDF1 [11], the optimal parameters when the AM data is included as predicted by (7), and the actual optimal parameters found by POUNDERS run from UNEDF1.

Table 5 summarizes the solutions found by POUNDERS from two different starting points. Again, the solutions found are remarkably close to one another, each parameter being with 0.07 of a standard deviation based (on the UNEDF1 uncertainties $\boldsymbol{\sigma}$ reported in Table 3). Furthermore, as predicted by the parameters remaining close relative to their uncertainties (Figure 3), the χ^2 values based on UNEDF1 ($\frac{f(\hat{\mathbf{x}})}{n_d - n_x} = \frac{51.058}{103} = 0.496$; see Table 3) and the parameterization obtained with the AM data ($\frac{f(\hat{\mathbf{x}})}{n_d - n_x} = \frac{54.01}{120} = 0.450$; see Table 5) are similar.

The final columns in Table 5 return to the topic of removing the bounds on the nuclear matter property parameters (recall the UNEDF0 case in Table 1). Here we see that χ^2 can be further reduced ($\frac{f(\hat{\mathbf{x}})}{n_d - n_x} = 0.386$) if E^{NM}/A and K^{NM} are allowed to violate the bounds, but that the changes to the parameters are substantial (up to 5 standard deviations for $a_{\rm sym}^{NM}$ alone), even when starting from the bound-constrained solution "AM Run 2." In future work we plan to examine the effect of these bounds and the inclusion of observables that better constrain the nuclear matter properties.

4. Derivatives, Revisited

The tables in Sections 2 and 3 compare differences in the initial and final values obtained after an optimization. Although the computational budget used is indicated through the reported number of function evaluations (n_f), these tables do not provide a sense of the rate of progress made the reported algorithms. Were the majority of the evaluations devoted to certifying approximate optimality? Or, were substantial reductions of the objective obtained right up until the final evaluations?

Figure 4 illustrates the rate of convergence on a $n_x = 17$-parameter problem involving the calibration of an occupation number-based energy functional from [20].

Table 5. Reruns of the UNEDF1 optimization with the inclusion of 17 new AM data ($n_x = 12$, $n_d = 132$, $n_N = 96$). The scaled difference columns are scaled by the uncertainties σ_i in Table 3, the first column being the difference between AM runs 1 and 2, the second column being the difference between AM run 2 and the run without bound constraints enforced.

	AM Run 1	**AM Run 2**	**Scaled Diff.**	**No Bounds**	**Scaled Diff.**
start	UNEDF0	UNEDF1		AM Run 2	
ρ_c	**0.1588**9255	**0.1588**6155	0.07381	0.15748674	3.273
E^{NM}/A	**-15.8**	**-15.8**	-	-15.692799	-
K^{NM}	**220**	**220.0**2317	-	221.06558	-
a_{sym}^{NM}	**29.3**44856	**29.3**36203	0.01433	26.173927	5.236
L_{sym}^{NM}	**40.**714438	**40.**014867	0.05326	13.510725	2.018
$1/M_s^*$	**0.96**859386	**0.96**78555	0.00600	0.91930059	0.3948
$C_0^{\rho\Delta\rho}$	**-4**3.980091	**-4**4.028902	0.00910	-39.479616	-0.8395
$C_1^{\rho\Delta\rho}$	**-11**4.29145	**-11**1.31777	-0.05700	-150.49163	0.7509
V_0^n	**-182.**23717	**-182.**15551	-0.00441	-174.88812	-0.3925
V_0^p	**-20**3.98073	**-20**4.19083	0.01610	-199.51881	-0.3580
$C_0^{\rho\nabla J}$	**-72.**417226	**-72.**668136	0.04970	-71.753276	-0.1812
$C_1^{\rho\nabla J}$	**-3**2.920571	**-3**1.360678	-0.06739	-31.708413	0.0150
$f(\hat{\mathbf{x}})$	54.0468	54.0140		46.3344	
n_f	76	152		74	

The different methods in this figure illustrate the benefits—in terms of convergence speed—of exploiting structural knowledge about the optimization objective. All three methods are based on the same model-based trust-region framework of POUNDERS; see [17]. The POUNDER variant assumes that the optimization algorithm does not have access to the residual vector and thus operates only with f values; POUNDERS uses the same formulation as in the previous sections, whereby an entire residual vector $\mathbf{F}$ (in this case, consisting of binding energies for $n_N = n_d = 2049$ nuclei) is passed to the optimization algorithm; and the POUNDERSM variant exploits the fact that the (first- and second-order) derivatives of each residual component are available with respect to 3 of the 17 parameters.

Figure 4 can be placed into broader context by recalling Figure 2. As more residual derivatives are available, there is a tendency to approach the derivative-based case (where the NLS structure is exploited). In the other extreme, when only f is available, the performance of the POUNDER variant is generally expected to be slightly better than the Nelder-Mead code (see, e.g., [30]).

5. Using POUNDERS

The POUNDERS algorithm can be a powerful tool for scientists to evaluate and improve computationally expensive theoretical models so they have better agreement with experimental data. We now outline some of the typical requirements and usage of POUNDERS for solving applications involving NLS problems.

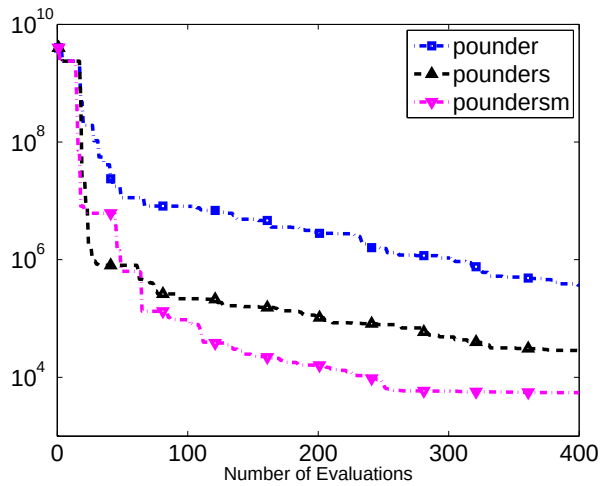

Figure 4. Best f value found as a function of the number of f evaluations for different model-based algorithms on a 17-dimensional parameter estimation problem from [20].

The POUNDERS algorithm is included in the distribution of PETSc/TAO, an open-source software package developed at Argonne National Laboratory and available free of charge at [31]. The software library can be built on most common architectures and operating systems, using almost any modern `C` compiler.

In order to use the POUNDERS algorithm, an objective function routine must be written (in `C/C++` or `Fortran/Fortran90`) that can separately compute each component, $F_i(\mathbf{x})$, of $\mathbf{F}(\mathbf{x})$ given a vector of parameters $\mathbf{x}$. POUNDERS is a derivative-free method, so no gradient information needs to be computed. In order to start the algorithm, an initial set of parameters $\mathbf{x}^0$ and an initial step length must also be provided.

There are a number of features that can be used to improve the performance and utility of POUNDERS. One of these is the aforementioned enforcement of bound constraints. Finite bounds can be provided for a subset (or all) of the parameters; these bounds can be one-sided, with only one of the lower or upper bound values being finite. We note that POUNDERS assumes that these bound constraints are *unrelaxable*, meaning that the algorithm will never attempt to evaluate the residual vector outside of the bounds. The benefit of this restriction is that one can ensure that the underlying simulation is not run in regions of parameter space where it may be error-prone or where its output may not be defined. A limitation of this restriction is that, provided the residual vector is well-behaved outside of these bounds, in some cases a derivative-free algorithm requires fewer evaluations when these bounds can be relaxed.

Whether finite bounds are provided or not, scaling of the variables is an important consideration when calling POUNDERS. By default, POUNDERS fundamentally assumes that the objective f experiences similar changes under a unit change to each of the parameters. Consequently, we recommend that the user apply an affine transformation to the parameters in the function that POUNDERS calls. For example, Table 2 lists the scaling bounds used throughout the HFBTHO-based optimizations reported in this

paper. In the layer between HFBTHO and POUNDERS, we apply a transformation $\mathcal{T}$ that maps the scaling rectangle in Table 2 to the unit hypercube, $\mathcal{T}([\mathbf{s}^l, \mathbf{s}^u]) = [0,1]^{n_x}$; the bound constraints being scaled by the same transformation.

Other features include the ability to initialize the internal model of the application using precomputed parameter sets and their objectives (warm-starting) to improve performance. There are also a number of features common to all PETSc/TAO programs provided by the PETSc framework; these include robust error handling, portability, command-line argument parsing, and performance profiling [32].

More detailed instructions for using POUNDERS are available from [31] or [33], as well as example programs, implementation details, and contact information.

Acknowledgments

This work was supported by the SciDAC activity within the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Nuclear Physics programs under contract numbers DE-AC02-06CH11357 (Argonne) and DE-AC52-07NA27344 (Lawrence Livermore). We gratefully acknowledge high-performance computing resources operated by the Laboratory Computing Resource Center at Argonne.

References

[1] E. Epelbaum and U.-G. Meißner. Chiral dynamics of few- and many-nucleon systems. *Ann. Rev. Nucl. Part. Sci.*, 62(1):159, 2012.

[2] B. R. Barrett, P. Navrátil, and J. P. Vary. Ab initio no core shell model. *Prog. Part. Nucl. Phys.*, 69:131, 2013.

[3] P. Navrátil, V. G. Gueorguiev, J. P. Vary, W. E. Ormand, and A. Nogga. Structure of A=10–13 nuclei with two- plus three-nucleon interactions from chiral effective field theory. *Phys. Rev. Lett.*, 99(4):042501, 2007.

[4] M. Bender, P.-H. Heenen, and P.-G. Reinhard. Self-consistent mean-field models for nuclear structure. *Rev. Mod. Phys.*, 75(1):121, 2003.

[5] P. Ring and P. Schuck. *The Nuclear Many-Body Problem.* Springer-Verlag, 2000.

[6] J. Erler, N. Birge, M. Kortelainen, W. Nazarewicz, E. Olsen, A. M. Perhac, and M. Stoitsov. The limits of the nuclear landscape. *Nat.*, 486(7404):509, 2012.

[7] W. Younes and D. Gogny. Nuclear scission and quantum localization. *Phys. Rev. Lett.*, 107(13):132501, 2011.

[8] A. B. Balantekin, J. Carlson, D. J. Dean, G. M. Fuller, R. J. Furnstahl, M. Hjorth-Jensen, R. V. F. Janssens, B.-A. Li, W. Nazarewicz, F. M. Nunes, W. E. Ormand, S. Reddy, and B. M. Sherrill. Nuclear theory and science of the facility for rare isotope beams. *Mod. Phys. Lett. A*, 29(11):1430010, 2014.

[9] J. Dobaczewski, W. Nazarewicz, and P.-G. Reinhard. Error estimates of theoretical models: A guide. *J. Phys. G: Nucl. Phys.*, 41(7):074001, 2014.

[10] M. Kortelainen, T. Lesinski, J. Moré, W. Nazarewicz, J. Sarich, N. Schunck, M. V. Stoitsov, and S. M. Wild. Nuclear energy density optimization. *Phys. Rev. C*, 82(2):024313, 2010.

[11] M. Kortelainen, J. McDonnell, W. Nazarewicz, P.-G. Reinhard, J. Sarich, N. Schunck, M. V. Stoitsov, and S. M. Wild. Nuclear energy density optimization: Large deformations. *Phys. Rev. C*, 85:024304, 2012.

[12] M. Kortelainen, J. McDonnell, W. Nazarewicz, E. Olsen, P.-G. Reinhard, J. Sarich, N. Schunck, S. M. Wild, D. Davesne, J. Erler, and A. Pastore. Nuclear energy density optimization: Shell structure. *Phys. Rev. C*, 89:054314, 2014.

[13] C. H. Bischof, H. M. Bücker, P. Hovland, U. Naumann, and J. Utke, editors. *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*. Springer, 2008.

[14] J. Moré and S. M. Wild. Estimating derivatives of noisy simulations. *ACM Trans. Math. Soft.*, 38(3):19:1–19:21, 2012.

[15] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. SIAM, Philadelphia, PA, 2009.

[16] D. Chwirut. Ultrasonic reference block study. Technical report, NIST, 1979.

[17] S. M. Wild. Solving derivative-free nonlinear least squares with POUNDERS. Preprint ANL/MCS-P5120-0414, Argonne Nat. Lab., Argonne, IL, April 2014.

[18] S. Bogner, A. Bulgac, J. Carlson, J. Enge, G. Fann, R. J. Furnstahl, S. Gandolfi, G. Hagen, M. Horoi, C. Johnson, M. Kortelainen, E. Lusk, P. Maris, H. Nam, P. Navratil, W. Nazarewicz, E. Ng, G. P. A. Nobre, E. Ormand, T. Papenbrock, J. Pei, S. C. Pieper, S. Quaglioni, K. Roche, J. Sarich, N. Schunck, M. Sosonkina, J. Terasaki, I. Thompson, J. P. Vary, and S. M. Wild. Computational nuclear quantum many-body problem: The UNEDF project. *Comput. Phys. Commun.*, 184(10):2235–2250, 2013.

[19] Nuclear Computational Low-Energy Initiative (NUCLEI) Web page. See `http://nuclei.mps.ohio-state.edu`.

[20] M. Bertolli, T. Papenbrock, and S. M. Wild. Occupation number-based energy functional for nuclear masses. *Phys. Rev. C*, 85(1):014322, 2012.

[21] A. Ekström, G. Baardsen, C. Forssén, G. Hagen, M. Hjorth-Jensen, G. R. Jansen, R. Machleidt, W. Nazarewicz, T. Papenbrock, J. Sarich, and S. M. Wild. Optimized chiral nucleon-nucleon interaction at next-to-next-to-leading order. *Phys. Rev. Lett.*, 110:192502, May 2013.

[22] M. V. Stoitsov, N. Schunck, M. Kortelainen, N. Michel, H. Nam, E. Olsen, J. Sarich, and S. M. Wild. Axially deformed solution of the Skyrme-Hartree-Fock-Bogolyubov equations using the transformed harmonic oscillator basis (II) HFBTHO v2.00d: A new version of the program. *Comput. Phys. Commun.*, 184(6):1592–1604, 2013.

[23] M. Stoitsov, W. Nazarewicz, and N. Schunck. Large-scale mass table calculations. *Int. J. Mod. Phys. E*, 18(04):816, 2009.

[24] E. Chabanat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer. A Skyrme parametrization from subnuclear to neutron star densities. *Nucl. Phys. A*, 627(4):710, 1997.

[25] E. Chabanat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer. A Skyrme parametrization from subnuclear to neutron star densities, Part II: Nuclei far from stabilities. *Nucl. Phys. A*, 635(1):231, 1998.

[26] S. Goriely and J. Pearson. Further explorations of Skyrme-Hartree-Fock-Bogoliubov mass formulas. VIII. role of Coulomb exchange. *Phys. Rev. C*, 77(3):031301(R), 2008.

[27] J. Bartel, P. Quentin, M. Brack, C. Guet, and H.-B. Håkansson. Towards a better parametrisation of Skyrme-like effective forces: A critical study of the SkM force. *Nucl. Phys. A*, 386:79, 1982.

[28] J. Van Schelt, D. Lascar, G. Savard, J. A. Clark, P. F. Bertone, S. Caldwell, A. Chaudhuri, A. F. Levand, G. Li, G. E. Morgan, R. Orford, R. E. Segel, K. S. Sharma, and M. G. Sternberg. First results from the CARIBU facility: Mass measurements on the r-process path. *Phys. Rev. Lett.*, 111(6):061102, August 2013.

[29] J. D. McDonnell, D. Higdon, W. Nazarewicz, J. Sarich, N. Schunck, and S. M. Wild. Uncertainty quantification for nuclear density functional theory, 2014. In preparation.

[30] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, 20(1):172–191, 2009.

[31] Toolkit for Advanced Optimization (TAO) Web page. See `http://www.mcs.anl.gov/tao`.

[32] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp,

D. Kaushik, M. G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, and H. Zhang. PETSc users manual. Tech. Memo. ANL-95/11 - Revision 3.4, Argonne Nat. Lab., Argonne, IL, 2013.

[33] T. Munson, J. Sarich, S. M. Wild, S. Benson, and L. C. McInnes. TAO 2.0 users manual. Tech. Memo. ANL/MCS-TM-322, Argonne Nat. Lab., Argonne, IL, 2012.