Solving nonlinear integral equations for laser pulse retrieval with Newton's method

Michael Jasiulek [*]
Max-Born-Institut für Nichtlineare Optik, Max-Born-Straße 2A, 12489 Berlin, Germany

(Received 6 November 2020; accepted 29 April 2021; published 18 May 2021)

We present an algorithm based on numerical techniques that have become standard for solving nonlinear integral equations: Newton's method, homotopy continuation, the multilevel method, and random projection to solve the inversion problem that appears when retrieving the electric field of an ultrashort laser pulse from a two-dimensional intensity map measured with frequency-resolved optical gating (FROG), dispersion-scan, or amplitude-swing experiments. Here we apply the solver to FROG and specify the necessary modifications for similar integrals. Unlike other approaches we transform the integral and work in time domain where the integral can be discretized as an overdetermined polynomial system and evaluated through list autocorrelations. The solution curve is partially continues and partially stochastic, consisting of small linked path segments and enables the computation of optimal solutions in the presents of noise. Interestingly, this is an alternative method to find real solutions of polynomial systems, which are notoriously difficult to find. We show how to implement adaptive Tikhonov-type regularization to smooth the solution when dealing with noisy data, and we compare the results for noisy test data with a least-squares solver and propose the L-curve method to fine-tune the regularization parameter.

DOI: 10.1103/PhysRevE.103.053306

I. INTRODUCTION

In many areas of modern short pulse spectroscopy, attophysics, and laser optimization knowledge of the precise duration and shape of ultrashort laser pulses is elementary. A direct measurement using electrical detectors is impossible due to their relatively slow response time, and various pulse retrieval schemes exploiting nonlinear optical effects have been developed [1,2]. A common self-referenced technique to measure phase and amplitude of an ultrashort laser pulse is frequency-resolved optical gating (FROG) [3]. A pulse to be investigated is split in two replicas, and then a relative delay is imposed upon the two which are guided into a nonlinear medium where second harmonic generation (SHG) takes place. The upconverted light is measured with a spectrometer for varying delays. This two-dimensional intensity map, the *trace* or spectrogram, encodes all information to retrieve the enveloping electric field of the investigated pulse. Other modern self-referenced pulse retrieval schemes vary the amount of dispersion in the optical path [4] (dispersion scan) or the relative amplitudes of two pulse replicas [5] (amplitude swing) instead of the delay to obtain a spectrogram. To invert the associated autocorrelation-like nonlinear integral the most successful solvers implement the least-squares method using generic optimization or search methods [2,6]. Although the related optimization problem is known to be nonconvex, these solvers are favorable to approaches inspired by the Gerchberg-Saxton algorithm [7] like [8–10], which were the first solvers to be used. They have the tendency to stagnate, in particular, in the presence of noise, which has been shown in a recent comparative study [2].

In this paper we present an algorithm linked to a time-domain representation of the integral and based on yet unexplored (for the present problem), to our best knowledge, numerical methods [11–13] that have been successfully applied to other difficult to solve nonlinear integral equations in physics like the Ornstein-Zernike equations [14,15], describing the direct correlation functions of molecules in liquids, and the Chandrasekhar H-equation [16] arising in radiative transfer theory, to name two classical examples. First, this is Newton's method. Modern Jacobian-free Newton-Krylov methods [17], variants of Newton's method, are the basis of large-scale nonlinear solvers like KINSOL, NOX, and SNES [18–20]. Second, homotopy continuation [21–23], a technique to globalize Newton's method, has proven to be reliable and efficient for computing all isolated solutions of polynomial systems and is the primary computational method for polynomial solvers like Bertini and PHCpack [24,25]. We employ this method here as there are strong guarantees that a global solution can be found without stagnation beginning at arbitrary initial data—a main obstacle for the above mentioned solvers which seek the global minimum of a nonconvex optimization problem.

The way in which the continuation method is implemented here has similarities with techniques in stochastic optimization [26–28]. While path tracking towards the solution we frequently alternate the random matrices, which in any case would be necessary to reduce the over-determined polynomial system to square form through random projection. For each fixed random matrix there is a corresponding homotopy and continuation path, such that the full solution path is partially continuous and partially stochastic, consisting of small linked

[*] jasiulek@mbi-berlin.de

path segments on each of which the error decreases monotonically by construction. This is, to our knowledge, a method for real root finding of polynomial systems [29] that could be used to find real solutions of similar polynomial systems and optimal solutions if noise is present.

For the retrieval with realistic noisy experimental data these methods alone would not be sufficient because Newton's method has certain smoothness assumptions. For that purpose we chose an integral discretization based on grid cell (pixel) surface averages and Tikhonov-type regularization. The Tikhonov factor is adaptively decreased during the solution process to obtain a near-optimal amount of regularization at the solution which can be refined using the L-curve method [30]. The coarsening capability enables fast computations of low-resolution approximants and noise suppression and on a hierarchy of finer-pixelization high-accuracy retrievals. Multilevel approaches for FROG have also been used in [31].

II. NOTATION, INTEGRAL REPRESENTATION, DISCRETIZATION

The nonlinear integral for SHG-FROG is defined as

$$I[E](\omega,\tau) := \left| \int_{-\infty}^{+\infty} E(t)E(t-\tau)e^{-i\omega t}\,dt \right|^2, \tag{1}$$

where $E(t)$ is a complex function, the enveloping electric field (pulse shape) of the pulse which we assume to be nonzero on the interval $t \in [-1, 1]$ and zero elsewhere,[1] with time units such that this interval has length 2. The outcome of the FROG experiment is the FROG *trace* $I_{\text{exp}}(\omega,\tau) \approx I[E_{\text{in}}](\omega,\tau)$ of the pulse to be investigated $E_{\text{in}}(t)$. We obtain $E_{\text{in}}(t)$ by solving the integral equation

$$I[E](\omega,\tau) - I_{\text{exp}}(\omega,\tau) = 0. \tag{2}$$

We bring (1) into a form better suited for polynomial approximation (removing the explicit t, ω dependence) by Fourier transform[2] $\omega \to \sigma$:

$$\begin{aligned} J[E](\tau,\sigma) = \int_{-\infty}^{+\infty} \bigg[\int_{-\infty}^{+\infty} & E(t)E(t-\tau)e^{-i\omega t}\,dt \\ & \times \int_{-\infty}^{+\infty} \bar{E}(s)\bar{E}(s-\tau)e^{i\omega s}\,ds \bigg] e^{i\omega\sigma}\,d\omega\,/\,(2\pi), \\ J[E](\tau,\sigma) = \int_{-\infty}^{+\infty} & E(t)E(t-\tau)\bar{E}(t-\sigma)\bar{E}(t-\tau-\sigma)\,dt. \end{aligned} \tag{3}$$

In the first line we have split the absolute value in (1) into a complex integral and its complex conjugate and then applied the relation $\int_{-\infty}^{+\infty} e^{i\omega[r-(t-\sigma)]}\,d\omega = 2\pi\delta[r-(t-\sigma)]$, where we call $J[E](\tau,\sigma)$ the *double-delay* representation of the SHG-FROG integral and $\bar{E}(t)$ is the complex conjugate of $E(t)$. For $E(t)$ nonzero on $t \in [-1, 1]$ the trace $J[E](\tau,\sigma)$ is nonzero on $\tau,\sigma \in [-2, 2]$. In the following we consider only the first quadrant $\tau,\sigma \in [0, 2]$, as the others are related through discrete symmetries. We introduce two functions for clarity of notation:

$$F_\tau(t) := E(t)E(t-\tau), \quad G_\sigma(t) := E(t)\bar{E}(t-\sigma), \tag{4}$$

now

$$\begin{aligned} J[E](\tau,\sigma) &= \int_{-\infty}^{+\infty} F_\tau(t)\bar{F}_\tau(t-\sigma)\,dt \\ &= \int_{-\infty}^{+\infty} G_\sigma(t)G_\sigma(t-\tau)\,dt. \end{aligned} \tag{5}$$

For fixed $\tau = \text{const}$, the integral $J[E](\tau,\sigma)$ appears to be a one-dimensional autocorrelation of the function $F_\tau(t)$. We discretize the electric field with a piecewise constant function (polynomial of degree zero),[3] in the context of numerical integration often called the *midpoint rule*,

$$E(t) = \begin{cases} 0 & t < -1 \quad \text{or} \quad 1 < t \\ E_k & t \in [t_k, t_{k+1}], \quad k = 0, \dots, N-1 \end{cases}, \tag{6}$$

on a uniform t-grid $t_k = -1 + k\,h$, $k = 0, \dots, N$ with N intervals, where $h = 2/N$ is the grid spacing. In the same way we define the grids along τ and σ: $\tau_i, \sigma_i = i\,h$, $i = 0, \dots, N$. If the delay is equal to an integer multiple of the grid spacing, thus, $\tau = \tau_i$, the product $F_{\tau_i}(t)$ is again piecewise constant [see Fig. 1 (bottom left)], which we abbreviate as ${}^{(i)}F_k = F_{\tau_i}(t_k) = E_kE_{k+i}$. Then the integrand of $J[E](\tau_i,\sigma)$ [see Fig. 1 (bottom right)] is also piecewise constant on small parallelograms[4] and integration over t disassembles into two sums of N subintegrals for the jth σ interval $\sigma \in [\sigma_j, \sigma_{j+1}]$ [jth column in Fig. 1 (bottom right)]:

$$\begin{aligned} J[E](\tau_i,\sigma) = h\int_{\hat{\sigma}-1}^{1} d\hat{t} \sum_{k=1}^{N} {}^{(i)}F_k\, {}^{(i)}\bar{F}_{k+j} + h\int_{-1}^{\hat{\sigma}-1} d\hat{t} \\ \times \sum_{k=1}^{N} {}^{(i)}F_k\, {}^{(i)}\bar{F}_{k+j+1},\ \sigma \in [\sigma_j, \sigma_{j+1}], \end{aligned} \tag{7}$$

$$\begin{aligned} J[E](\tau_i,\sigma) = h(2-\hat{\sigma})\,&\text{corr}({}^{(i)}F_k, {}^{(i)}\bar{F}_k)_j \\ &+ h\,\hat{\sigma}\,\text{corr}({}^{(i)}F_k, {}^{(i)}\bar{F}_k)_{j+1}, \end{aligned} \tag{8}$$

where $\hat{\sigma} \in [0, 2]$ and $\hat{t} \in [-1, 1]$ are local coordinates on the intervals $[\sigma_j, \sigma_{j+1}]$, $[t_k, t_{k+1}]$. The first sum is collecting all small upper triangles per column in Fig. 1 (bottom right) and the second the lower triangles. The expression $\text{corr}({}^{(i)}F_k, {}^{(i)}\bar{F}_k)_j := \sum_{k=1}^{N} {}^{(i)}F_k\, {}^{(i)}\bar{F}_{k+j}$, $i = 0, \dots, N-1$ denotes the list autocorrelations of ${}^{(i)}F_k$ that can be computed with complexity $N[N\log(N)]$.[5]

[1] Setting $E(t)$ on a bounded domain enables clipping of long low-amplitude wings and zooming to the region of interest on the trace, saving computational cost. A bounded domain can be used as the formalism is throughout working in the time domain.

[2] An analogous formulation for the retrieval in frequency domain is possible; see Appendix A.

[3] It is possible to use piecewise linear or more general polynomials or splines; see Appendix B.

[4] The integration boundaries depend on $\hat{\sigma}$; for the upper and lower triangles they are $\int_{\hat{\sigma}-1}^{1} d\hat{t} / \int_{-1}^{\hat{\sigma}-1} d\hat{t}$.

[5] Alternatively, for the relatively small N considered here, the direct method to compute the correlation is more efficient for $N < 1000$ [32] than the FFT-based variant when parallelized on thousands of cores.

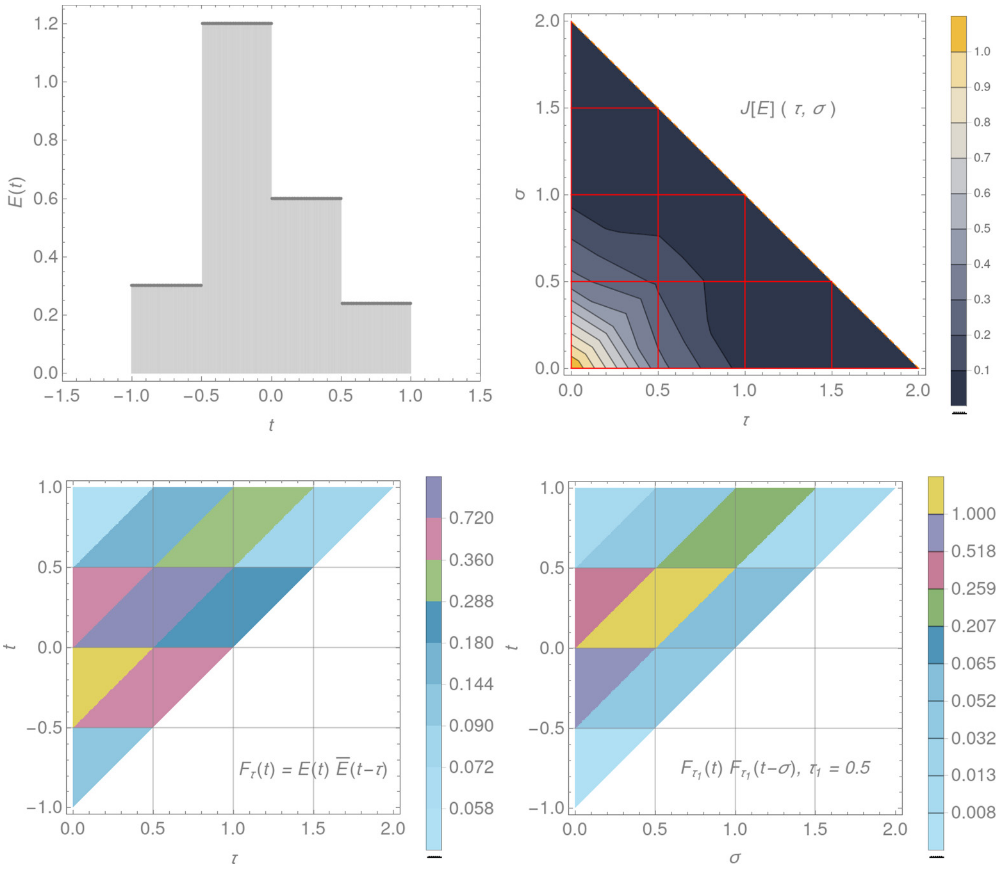

FIG. 1. Illustrating example: $E(t)$ discretized with a piecewise constant function on $N = 4$ intervals (top left). Then the associated products $F_{\tau_i}(t)$ and $F_{\tau_i}(t)F_{\tau_i}(t-\sigma)$ (bottom) are piecewise constant as well on small parallelograms such that the nonlinear integral $J[E](\tau,\sigma)$ (top right) can be computed via list autocorrelations along all grid segments (red lines). We consider J only in quadrand $(+,+)$, $\tau,\sigma \in [0,2]$ as the other quadrands are linked through discrete symmetries. J is nonzero only below the diagonal (dashed line) as $E(t)$ is nonzero on a bounded domain $t \in [-1,1]$ by definition. $E(t)$ is generally complex and normalized such that $\max|J[E](\tau,\sigma)| = 1$.

Equation (7) and the equivalent for $G_\sigma(t)$ gives the nonlinear integral along all grid segments $[\tau_i,\tau_{i+1}]$, $[\sigma_j,\sigma_{j+1}]$, $i,j = 0,\dots,N-1$, the red lines in Fig. 1 (top right). Now N could be chosen such that the τ_i overlap with the points of the experimental data, assuming an equally spaced grid with K points along τ, and the integral equation be solved similarly to what follows. As a measured trace is normally noisy, the better way to go is setting up a pixelwise instead of a pointwise representation of the equation. Moreover, on a coarse-graining hierarchy of smaller grids $N_1 < N_2 < \cdots < K$ the solver is faster and may resolve long- and short-wavelength components successively.

First, we pixelize the integral $J[E](\tau,\sigma)$. For the single pixel with grid coordinates (τ_i,σ_j) (lower left corner) we linearly interpolate the values from the left pixel boundary to the right boundary

$$J[E](\hat\tau,\hat\sigma)_{ij}^{\text{left right}} = J[E](\tau_i,\hat\sigma)(1-\hat\tau/2) + J[E](\tau_{i+1},\hat\sigma)\,\hat\tau/2, \tag{9}$$

to have the integral approximated inside the pixel. As before an over-hat denotes local pixel coordinates $\hat\tau,\hat\sigma \in [0,2]$. Then we integrate $J[E](\hat\tau,\hat\sigma)_{ij}^{\text{left right}}$ over the pixel surface normalized by its area to obtain the dimensionless pixel average

$$\begin{aligned}\langle J[E]_{ij}^{\text{left right}}\rangle &:= \int_0^2\int_0^2 J[E](\hat\tau,\hat\sigma)_{ij}^{\text{left right}}\,d\hat\tau\,d\hat\sigma \Big/ \int_0^2\int_0^2 d\hat\tau\,d\hat\sigma\\ &= \frac{1}{2}h\big[\operatorname{corr}({}^{(i)}F_k,{}^{(i)}\bar{F}_k)_j + \operatorname{corr}({}^{(i+1)}F_k,{}^{(i+1)}\bar{F}_k)_j\\ &\quad + \operatorname{corr}({}^{(i+1)}F_k,{}^{(i+1)}\bar{F}_k)_{j+1} + \operatorname{corr}({}^{(i)}F_k,{}^{(i)}\bar{F}_k)_{j+1}\big].\end{aligned} \tag{10}$$

The analog can be done for the bottom and top boundary and the correlation coefficients $\operatorname{corr}({}^{(j)}G_k,{}^{(j)}G_k)_i$ improving the accuracy[6] of the approximation. Then the total pixel average is

$$\langle J[E]_{ij}\rangle := \big(\langle J[E]_{ij}^{\text{left right}}\rangle + \langle J[E]_{ij}^{\text{bottom top}}\rangle\big)/2, \tag{11}$$

[6] For most applications it is enough to set $\langle J[E]_{ij}\rangle := \langle J[E]_{ij}^{\text{left right}}\rangle$ speeding up the computations by a factor of two, though sacrificing some accuracy. Note the swapping of indices for the coefficients of G.

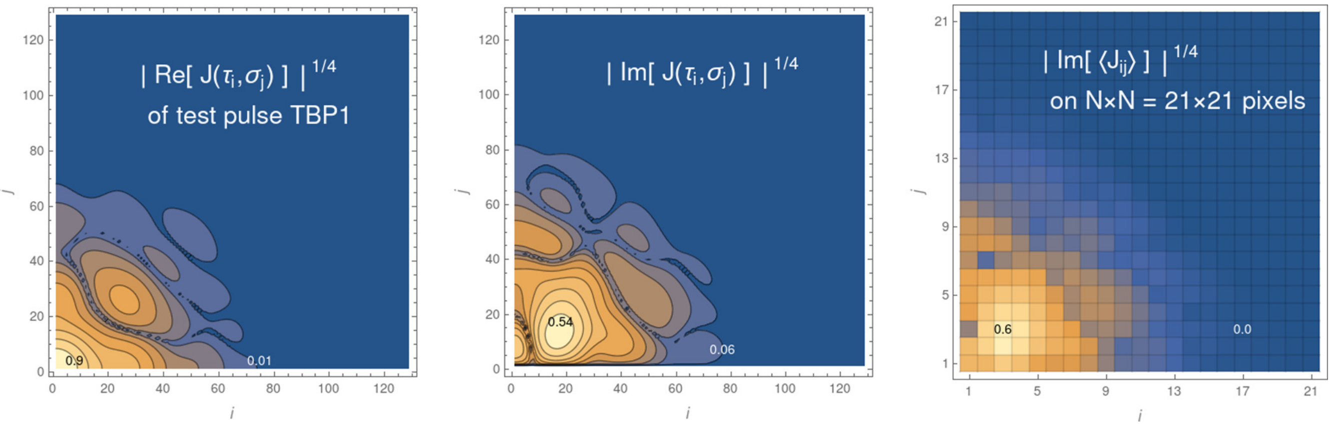

FIG. 2. Illustrating example: Synthetic measurement trace with 129 × 129 data points on $\tau, \sigma \in [0, 2]$. Coarse-grained data (only imaginary part shown) on 21 × 21 pixels (40 data points per pixel) enable fast computation of approximants to initialize refined retrievals. Every pixel of the lower triangular part $(21 \times (21+1)/2)$ is associated with one equation in (16). If E^+ and E^- are real roots, then $\langle J[E^+,E^-]_{ij}\rangle^+$ and $\langle J[E^+,E^-]_{ij}\rangle^-$ are real and equivalent to $\mathrm{Re}[\langle J[E^+,E^-]_{ij}\rangle]$, $\mathrm{Im}[\langle J[E^+,E^-]_{ij}\rangle]$. Note: The exponent 1/4 is convenient for data examination as (12) constitutes a 4th order polynomial system in the components E_k^+, E_k^-.

and the pixel average of the nonlinear integral is given by adding up the list correlation coefficients for each corner square times $\frac{1}{2}h/2$.

Finally, the pixel averages of the Fourier transformed measurement trace $I_{\mathrm{exp}}(\omega,\tau) \to J_{\mathrm{exp}}(\tau,\sigma) \to \langle J_{ij}^{\mathrm{exp}}\rangle$ have to be computed to set up the polynomial system (12) where these values constitute the constant part, see Fig. 2. This can be computed using the trapezoidal rule or simply by averaging all data points within a pixel.

III. SETTING REAL POLYNOMIAL SYSTEM, REAL ROOTS, GAUGE CONDITION

The integral equation (2) in double-delay representation is now discretized,

$$\langle J[E^+,E^-]_{ij}\rangle - \langle J_{ij}^{\mathrm{exp}}\rangle = 0, \tag{12}$$

as a fourth-order polynomial system in the $2N$ (generally complex-valued) new variables E^+, E^- with components E_k^+, E_k^-, where we have replaced

$$E \to E^+ + iE^- \quad \text{and} \tag{13}$$

$$\bar{E} \to E^+ - iE^- \tag{14}$$

in (2) to get rid of the operation of complex conjugation.[7]

Note that the so-created polynomial system has, strictly speaking, no exact solution as computing the pixel averages of the nonlinear integral on the one hand and the pixel averages of the trace come along with numerical and experimental errors limiting the accuracy. For the polynomial solver introduced in Sec. IV we employ methods from stochastic optimization to retrieve an optimal solution.

Clearly, if E^+, E^- are found as real roots of the polynomial system (12), we are dealing with a physical solution. Then $E^+ \to \mathrm{Re}(E), E^- \to \mathrm{Im}(E)$ are nothing but the real and imaginary parts of the electric field, though, more generally, complex solutions exist. In the same fashion we create a polynomial system introducing the linear combinations $\langle J_{ij}\rangle^+ = (\langle J_{ij}\rangle + \langle \bar{J}_{ij}\rangle)/2$ and $\langle J_{ij}\rangle^- = (\langle J_{ij}\rangle - \langle \bar{J}_{ij}\rangle)(-i/2)$:

$$\langle J[E^+,E^-]_{ij}\rangle^+ - \langle J_{ij}^{\mathrm{exp}}\rangle^+ = 0, \tag{15}$$

$$\langle J[E^+,E^-]_{ij}\rangle^- - \langle J_{ij}^{\mathrm{exp}}\rangle^- = 0 \tag{16}$$

such that the system has real coefficients, and as long as E^+ and E^- are real, Eq. (15) and (16) are real and imaginary parts of Eq. (12). The reasons for these rearrangements are the following: starting with a real initial iterate, Newton's method remains real, and we can stick to real arithmetics, which is about five times faster than using complex variables; more importantly, we are interested in finding real roots.

For the integral (5) the absolute phase as well as the time direction of the electric field are not fixed: for any solution $E(t)$, the product $E(t)\exp(i\,\mathrm{const})$ and $E(-t)$ are also solutions. We fix the rotational symmetry by adding the following equation to the system (15) and (16):

$$\int_{-\infty}^{+\infty} E^+(t)\,dt - \int_{-\infty}^{+\infty} E^-(t)\,dt = 0$$

$$\Rightarrow 2/N \sum_{k=1}^{N} (E_k^+ - E_k^-) = 0, \tag{17}$$

which we call *null gauge condition*;[8] it fixes the absolute complex phase but leaves the overall scaling and shape of

[7] This step may appear confusing at first sight, as we double the number of variables: Newton's method requires the nonlinear function to be Lipschitz continuous to guarantee convergence, which the operations of complex conjugation or taking the absolute value prevent; see, for example, (1.9.1) in [12]. Similar requirements, often overlooked, come in hand with the gradient descent method when applied to least squares.

[8] An alternative null gauge condition with the same properties exists: $\int_{-\infty}^{+\infty} E^+(t)\,dt = 0$.

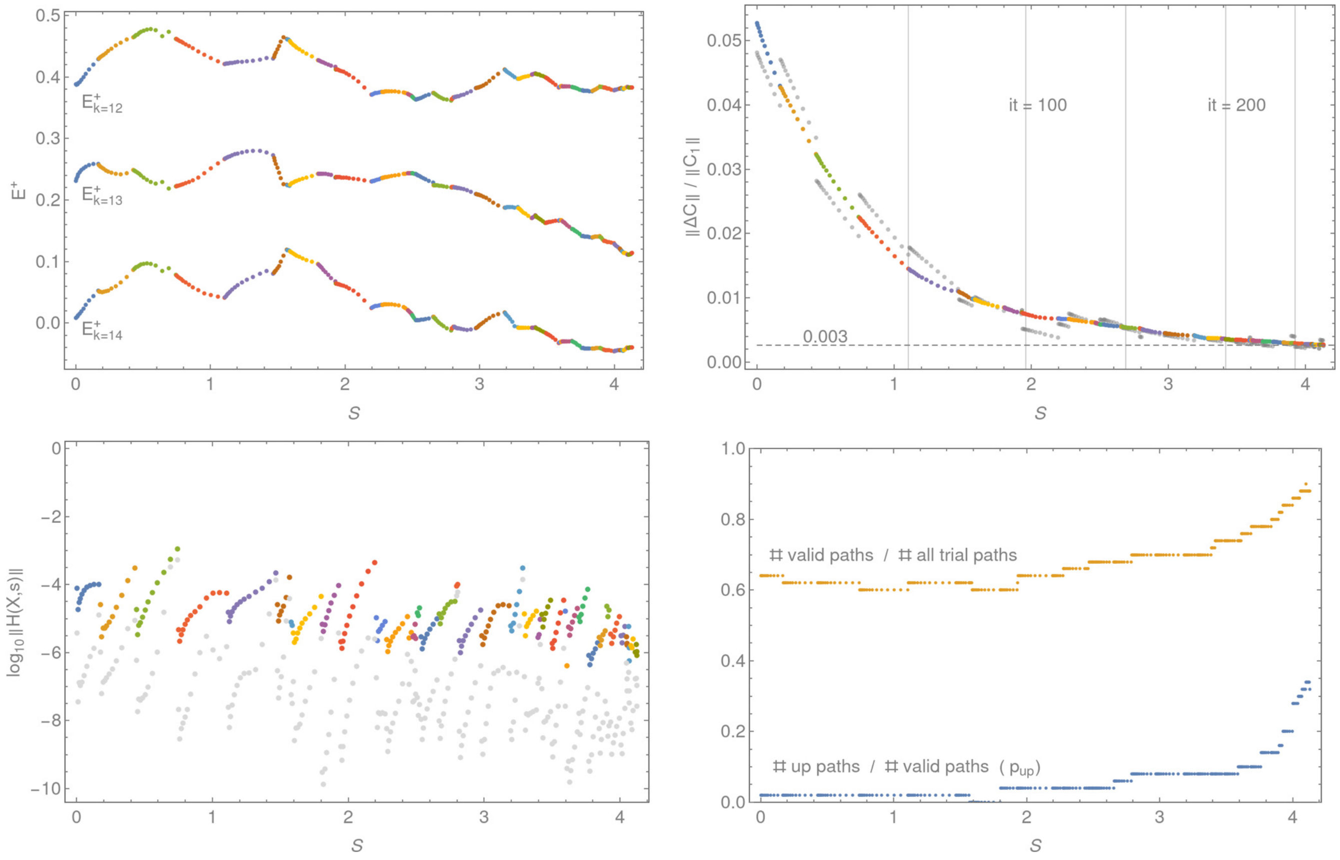

FIG. 3. Solving the polynomial system (18) for a test case on $N = 15$. Top left: Path tracking the solution curve consisting of small path segments each corresponding to a single reduced system with fixed random matrix M. Top right: Global distance to target trace of full (colored) and reduced (gray) system decreases approximately exponentially (on each segment linearly). Bottom left: Local residual after predictor step (colored) and after corrector step (Newton step) (gray). Bottom right: Relative number of up paths (increasing $\|\Delta C\|$) and valid paths (successful Newton step) when doing trial steps at the end of each segment to find a new path (and new M) along which $\|\Delta C\|$ decreases.

$E^+(t)$, $E^-(t)$ free. Moreover, it is a polynomial equation with real coefficients.

IV. SQUARING THE SYSTEM, NEWTON'S METHOD, HOMOTOPY CONTINUATION

The systems (12), (15), and (16) consist each of $(N + 1)N/2$ equations. Only the lower triangular part is nonzero as $E(t)$ is zero beyond the domain $t \in [-1, 1]$, such that (15) and (16) contribute $(N + 1)N$ equations. We denote the total system (15), (16), and (17) with $(N + 1)N + 1$ equations as

$$F(X) - C_1 = 0, \quad F(X) := \begin{pmatrix} F_1(X_1, \dots, X_{2N}) \\ \vdots \\ F_{(N+1)N+1}(X_1, \dots, X_{2N}) \end{pmatrix}, \tag{18}$$

where $X = \{E_k^+, E_k^-\}$ is the list of $2N$ variables and $F(X)$ is the X-dependent part of the set of equations $F_k(X)$, $k = 1, \dots, (N + 1)N$ for the lower triangular part of (12) flattened to a list and, analogously, the constant part of (12) (pixel trace averages) is flattened to the list $C_{1\,k}$, $k = 1, \dots, (N + 1)N$. The last equation $F_{(N+1)N+1}(X) - C_{1\,(N+1)N+1} = 0$ is set to be the gauge condition (17).

The polynomial system (18) is overdetermined. Moreover, due to numerical and experimental errors, it has no exact solution. We multiply[9] the vector of equations with a random matrix M, which can be reshuffled, having dimensions such that the reduced system has as many equations as variables. Then the Jacobian of the reduced system is well defined and by alternating random matrices stochastic optimization can be integrated. Here we choose M with i.i.d. Rademacher random variables (taking values $\{-1, +1\}$ with probability 1/2) to reduce the first $(N + 1)N$ equations to $2N - 1$ and attach the gauge condition as before at the end. We denote the reduced system as

$$F^M(X) - C_1^M = 0. \tag{19}$$

It contains all isolated roots of the original system which Bertini's theorem guarantees (see, for example, Sec. 1.1.4 in [24]) and additional "spurious" roots which do not solve (18) and which will differ, if M is reshuffled. As shown later, at

[9] For Rademacher variables this operation is implemented without any multiplication: 50% of all equations are added up randomly chosen, and the sum of the remaining equations is subtracted to obtain one new equation. Moreover, Rademacher variables do not rescale the noise.

every step of the solution curve $X(S) = \{E_k^+(S), E_k^-(S)\}$ [see Fig. 3 (top left)] the residual of Eq. (19) will be decreasing. Simultaneously, at every step it is tested if the residual of Eq. (18) is decreasing (called the "down path" later), the matrix M reshuffled otherwise, and the corresponding created reduced system (19) solved, shown in the colored path segments in Fig. 3 (top left). Then the solver cannot converge to spurious roots corresponding to any particular M along $X(S)$.

A standard iterative technique for root finding of nonlinear equations is Newton's method. Given an initial iterate $X_{n=0}$[10] and a nearby root X^* the function (19) is linearized at X_0,

$$F^M(X_0) + F'^M(X_0)\Delta X - C_1^M = 0, \tag{20}$$

solved for ΔX, the Newton step, and a step towards the root is taken:

$$X_{n+1} = X_n + \Delta X. \tag{21}$$

For a one-dimensional function X_{n+1} is the point, where the tangent at X_0 crosses the X axes. For vector-valued functions the derivative $F'^M(X_0)$, the Jacobian, is a square matrix and Eq. (21) a linear system for the unknown ΔX. The iteration (21) is known to converge roughly quadratically towards the root $e_{n+1} \sim e_n^2$, if the function is Lipschitz continuous (which polynomial functions satisfy) and the Jacobian nonsingular; see, for example, 1.2.1 in [12]. Here $e_n = \|X^* - X_n\|$ is the error of the nth iteration with $\|\cdot\|$ being the standard Euclidean norm on R^{2N}. The roughly quadratic convergence can be observed monitoring the norm[11] $\|F(X_n) - C_1\|$ and is often called *residual.*

For an arbitrarily chosen initial iterate X_0 there is, in general, no close enough root for the iteration (21) to converge, so then additional tricks are required to globalize Newton's method. The primary computational method for that purpose, polynomial system solvers like Bertini and PHCpack [24,25], employ the *continuation* method, where a homotopy is assembled,

$$\begin{aligned} H^M(X,s) &:= \left[F^M(X) - C_0^M\right](1-s) + \left[F^M(X) - C_1^M\right]s, \\ &\text{with} \quad H^M(X(s=0),0) = 0, \end{aligned} \tag{22}$$

which is connecting two polynomial systems and all roots of them via smooth curves $X(s)$, the start system (first term) at $s = 0$ and the target system (second term) at $s = 1$, where s is the continuation parameter, and, in general, a curve in the complex plane, in the following real $s \in [0, 1]$.

An $X(s = 0)$ is chosen freely (normally a Gaussian) to compute $C_0 := F(X(s = 0))$ in the forward direction. It is guaranteed that beginning at the solution $X(s = 0)$ of the so-created start system and following the curve $X(s)$ to arrive at a solution of the target system, if $H'^M(X(s), s)$ is nonsingular and s an arbitrary complex curve beginning at $s = 0$ and ending at $s = 1$. Though, here, we move along real paths where a finite number of singular points[12] exist, as an arbitrary complex path would render the homotopy to have complex coefficients and the continuation path would, in general, end at an undesirable complex root of the target system.

Starting at $s = s_m$ with $H^M(X(s_m), s_m) = 0$ and taking a step $s_{m+1} = s_m + \Delta s$ with Δs small enough along the $X(s)$ curve, we can guarantee to be close enough to a solution of $H^M(X, s_{m+1}) = 0$ when using the initial iterate $X(s_m)$. This path tracking [see Fig. 3 (top left)] is usually done in a predictor-corrector scheme with adaptive step size control. We use step size parameters as in PHCpack [25], a predictor given by the local tangent, and one Newton step as a corrector (reusing the Jacobian to compute the new local tangent).

For any particular M we track the path $X(s)$, hold the tracker, and reshuffle M at a *break point* $s = s_b$:

$$\begin{aligned} H^M(X, s_b) &= F^M(X(s_b)) - C_b^M = 0, \\ \text{where} \quad C_b^M &= C_0^M - s_b(C_0^M - C_1^M), \end{aligned} \tag{23}$$

whenever close to a singular point or if the residual of the full system (18) $\|\Delta C(s)\| = \|F(s) - C_1\|$ increases, also called momentary distance to the target trace. For the randomly reduced system we use the intermediate solution $X(s_b)$ as an initial iterate for the corresponding homotopy beginning at $s = 0$. In this manner, we get a collection of path segments $\{s_{b_i}\}_{i=1,\ldots}$ where $S = s_{b_1} + s_{b_2} + \cdots$ is the total continuation time [see Fig. 3 (top left, colored segments)] with decreasing $\|\Delta C(S)\|$ (top right, colored).

The error of the reduced system (momentary distance to the reduced target trace) (top right, gray) $\|\Delta C^M(S)\| = \|F^M(S) - C_1^M\| = \|C_{b_i}^M - C_1^M\|$ decreases linearly for each path segment:

$$\left\|C_{b_i}^M - C_1^M\right\| > \left\|C_{b_{i+1}}^M - C_1^M\right\| = (1 - s_{b_i})\left\|C_{b_i}^M - C_1^M\right\| \tag{24}$$

$$\begin{aligned} &\Rightarrow \|\Delta C^M(S)\| \approx -\frac{d}{dS}\|\Delta C^M(S)\| \\ &\Rightarrow \|\Delta C^M(S)\| \approx \|\Delta C^M(S=0)\| e^{-S}, \end{aligned} \tag{25}$$

[10] Abuse of notation; this is a variable vector of length $2N$. Here the index $n = 0$ denotes the zeroth Newton iteration.

[11] In the ultrafast optics community instead of the Euclidean norm, typically the rms error is used, often called FROG error or trace error.

[12] The structure of singular points for real homotopies like (22) has been fully characterized [33]: These singularities are quadratic turning points or *simple folds*, where two real and two complex conjugated solution branches meet, rotated by $\pi/2$ in the complex plan and toughening at their turning points, the simple fold. Both branches smoothly transit the turning point, if an arc-length parameter is used instead of s or a pseudo-arc-length continuation [34]. Then it is possible to follow the real curve through the bifurcation point or, alternatively, jump onto the complex solution branch. Following the real branch we simply return to a new real root of the start system; continuing the complex branch we either end up on a complex root (or its complex conjugate) of the target system or eventually flow into another simple fold where a transition to another real branch is possible. We implemented a pseudo-arc-length continuation. Unfortunately, after passing a simple fold along the complex branch it is unlikely that it touches another simple fold and rather ends up on an undesired complex solution of the target system. The same phenomenon has been observed in [35] in the attempt to bypass these singular points towards real roots.

and as the length of each path segment is relatively small $s_b \ll S$, globally, the total error decrease appears like an exponential decay in S.

Initially, the distance $\|\Delta C(S)\|$ is decreasing approximately linearly as well on each segment. We are moving along smooth curves, stepping along any newly created path with decreasing $\|\Delta C(S)\|$, which we call *down paths* as opposed to *up paths*; it is likely that the next step is also decreasing. The number of steps before reaching a break point ($\|\Delta C(S)\|$ increases) is getting smaller as $X(S)$ is getting closer to the optimum, until no significant reduction of the error is possible when reaching the accuracy limit set by numerical errors or noise floor.

At every break point trial predictor-corrector steps are computed for created randomized systems until a down path has been found. If the trail step succeeds (Newton's method converges), the path is called a *valid path*,[13] which can be either an up or down path. Clearly, the probability of finding an up path from the list of all valid paths is an important quantity, which we denote as p_{up}.[14] We found that almost every valid path is a down path before reaching the accuracy limit for noisy systems. Then p_{up} rises steeply [see Fig. 3 (bottom right)]. This intrinsic quantity is, thus, most practical and sensitive to stop the solver by setting a threshold $p_{\text{up}}^{\text{stop}} > p_{\text{up}}$ at $p_{\text{up}}^{\text{stop}} = 50\%–90\%$, for example.

V. ADAPTIVE REGULARIZATION FOR NOISY TRACES

In this section we show how to make the algorithm work for pulse retrieval of noisy experimental data.

One effect of computing the integral (1) in the forward direction given E is smoothing as we are dealing with an autocorrelation-like nonlinear integral. Contrary, the inverse mapping acts as a high-pass filter with the undesirable tendency of noise amplification, which is rather problematic when using Newton's method. We resolve this phenomenon by adding a regularization term to the equations. Tikhonov regularization or ridge regression, when solving an ill-posed least-squares problem, has a long history in statistics; see, for example, [36].

The pixelwise smoothing of the trace (11) is providing noise reduction and thereby implicit regularization. As the pixelization is refined, steep local gradients arise when sampling a set of continuation paths, causing the path tracker to reduce the step size to very small and leading to smaller and smaller path segments, until the smoothness assumptions coming with Newton's method do not apply. Then more direct countermeasures are required.

Analogously to Tikhonov regularization we add a penalty term $K(X,\lambda)$ with components $K_l(X,\lambda)$, $l = 1,\dots,N(N+1)$ to the first $N(N+1)$ equations of the original system (18), which gives preference to solutions with smaller norms, also known as L_2 regularization:[15]

$$F(X) + K(X,\lambda) - C_1 = 0, \quad K(X,\lambda) := \lambda\, M^i_{\text{reg}} \partial_i \|X\|^2, \tag{26}$$

where λ is the Tikhonov factor to scale the penalty term and M^i_{reg} a shuffle matrix which remains constant through out the path tracking and can be used for validation purposes via reshuffling and repeating the tracking.

For piecewise constant approximants (6) to $E(t)$ we get

$$\partial_l \|X\|^2 = \partial_l \sum_{i=0}^{2N-1} (X_{i+1} - X_i)^2 = 0,$$

$$l = 0,\dots,2N-1 \tag{27}$$

$$= (X_{l-1} - 2X_l + X_{l+1})$$

$$= 0, \quad \text{and (optional boundary condition)}, \tag{28}$$

which is nothing but the second-order finite difference of X on a three-point stencil (up to a factor), which means Eq. (26) gives preference to solutions with small mean curvature and implements the desired smoothing effect. An optional boundary condition can be added, for example, $E^{+/-}_{k=-1} = 0$ and $E^{+/-}_{k=N} = 0$, where we added two extra intervals on each end of the grid.

The penalty term alters the solution because as little regularization as necessary is wanted at the target solution, though, while path tracking, λ, can be larger, and this is actually beneficial from a numerical point of view as it improves the conditioning of the Jacobian, and smoother intermediate solutions enable longer path segments and larger steps. By slowly decreasing[16] λ [see Fig. 4 (top right)], we can ensure that smooth initial data are connected by a series of smooth intermediate solutions to the smooth target [Fig. 4 (bottom right)].

Moving along any path segment and the related homotopy of (26), where we hold λ constant, F is deformed slightly towards an improved match with the target trace C_1 for a class of curves with similar mean curvature. Therefore, the difference $\|F + K - C_1\| - \|F - C_1\|$ is growing along each segment, while, of course, both are decreasing simultaneously [see Fig. 4 (top left)], until, to improve the match further, the amount of smoothness must be reduced by decreasing λ (top right). Finally, the match cannot be significantly improved by allowing rougher curves. Even further reducing λ would cause matching F to the noise and the afore-mentioned shortening

[13] This automatically excludes all continuation paths for which the conditioning of the local Jacobian is bad and those with high velocities or curvature. In practice, we first compute the full Jacobian and then try several random projections. In this way we have the cost of computing the full Jacobian at the beginning of every new path segment only once.

[14] An efficient and still accurate enough method to estimate this quantity is, rather than computing many trail steps for every path segment, to keep a running list of the last, say, 20 trials of the preceding path segments. The step size for computing trial steps for each path segment over the whole path has to be the same to make this quantity comparable.

[15] For Tikhonov regularization the penalty term which is $\|X\|$ is added to the least-squares problem. Roots of the first derivative of this sum w.r.t. X correspond to regularized minima.

[16] On the other hand, throttling λ too slowly during the path tracking may cause an undesired prolongation of the path.

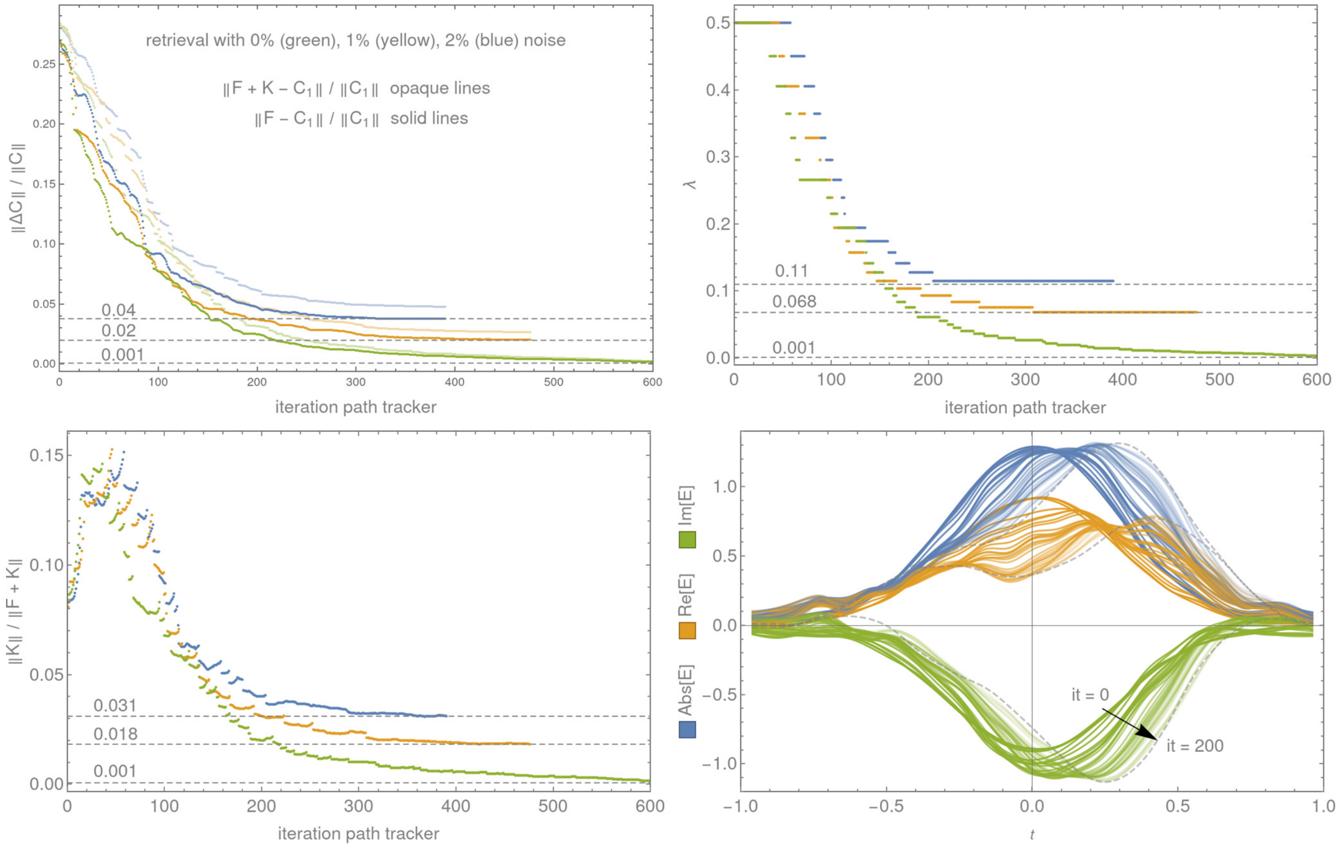

FIG. 4. Solving the polynomial system including a Tikhonov-type penalty term. Top right: The size of λ is constant along each path segment (see Fig. 3) and adaptively decreased at the end of each segment, if a threshold on the relative size of the regularization term is crossed $\delta_{\Delta C}^{\text{regul}} < 20\%$ [Eq. (29)] for three different noise levels 0% (green), 1% (yellow), 2% (blue). Bottom left: The relative size of the penalty term K on each segment is growing and globally adaptively reduced. Top left: The distances to the target trace with (opaque colored) and without (full colored) penalty term are simultaneously decreasing along each path segment, while their difference is increasing because F is deformed slightly towards an improved match with the target trace C_1 for a class of curves with similar mean curvature. Then in the noiseless case (green) [noisy case (yellow, blue)], $\delta_{\Delta C}^{\text{regul}}$ vanishes [settles] where λ is near optimal. Bottom right: Beginning at a zero phase Gaussian $E(s = 0)$, connected by a series of smooth intermediate solutions the path tracker continues towards the target pulse (dashed gray, only first 200 iterations shown).

of path segments and step size due to steeper local gradients—wasted computational cost. The relative difference

$$\delta_{\Delta C}^{\text{regul}} := (\|F + K - C_1\| - \|F - C_1\|)\,/\,\|F + K - C_1\| \tag{29}$$

is, thus, an ideal candidate for setting a threshold to lower λ from one path segment to the other. Moreover, $\delta_{\Delta C}^{\text{regul}}$ is inert to details of the solution and noise model and vanishes if no noise is present. For Fig. 4 the threshold was set at $\delta_{\Delta C}^{\text{regul}} < 20\%$.

When starting from very smooth initial data, like an initial Gaussian, in the coarse initial phase (first 100 iteration) the size of the penalty term can grow undesirably [Fig. 4 (bottom left)] before the above mechanism can set in because the intermediate solutions attain flection. (For the same reason the relative size of the penalty term grows along each path segment.) This is prevented by adding another threshold for decreasing λ, if $\|K\|\,/\,\|F + K\|$ rises above, say, 30%. Of course, if informed initial data are at hand, like a solution from a coarser grid, this is not necessary.

As shown in the following section this adaptation mechanism is steering λ near optimality or close enough for fine-tuning, for example, using the L-curve method or some other tool.

VI. APPLICATION EXAMPLES, CONVERGENCE, PRACTICAL CONCERNS

We implemented the algorithm as a hybrid code in Mathematica (prototyping, pre- and post-processing) and in Fortran90 (core routine path tracker). All simulations were performed on an Intel Core i7-4790 CPU3.6GHz with four cores on 8 GB RAM, Linux OS, and using OpenMP parallelization and the Intel compiler.

As test cases we selected the pulse with index 42 (TBP2) from the database of 101 randomly generated pulses with time-bandwidth product (TBP) equal to 2, which were used in [2] to profile their least-squares solver, another less intricate test pulse (TBP1) using the same generator with TBP = 1 and a third test case (A2908). For other test cases we found the same universal convergence behavior as shown in the following. For every run a different seed is used to initialize the Xorshift random number generator "xoshiro256+" [37] for computing the random matrices M.

To show the applicability to realistic defective traces Gaussian noise is added with $\sigma_{\text{noise}} = 1\%,\ 2\%,\ 3\%$ to the synthetic trace $I_{\text{exp}}(\omega, \tau) = I[E_{\text{in}}](\omega, \tau) + \text{noise}$ before[17] Fourier transforming it to $J_{\text{exp}}(\tau, \sigma)$. We study the effect of varying [see Eq. (29)], the regularization reduction threshold $\delta^{\text{regul}}_{\Delta C} = 5\%,\ 25\%,\ 40\%$ (controlling the decrease of λ while path tracking; see Sec. V), as well as the effect of varying the termination criterion to stop the solver $p^{\text{stop}}_{\text{up}} = 50\%,\ 70\%,\ 90\%$ on the error convergence while fine-graining the pixelization, increasing N. Here we measure the retrieval accuracy or *pulse error* ϵ as

$$\epsilon = \|E - E_{\text{in}}\| \,/\, \sqrt{N} \,/\, \max(|E|) \tag{30}$$

instead of using the relative error norm $\|E - E_{\text{in}}\|/\|E\|$ to make results comparable with the literature, in particular, [2]. The scaling behavior with N is the same for both metrics. To measure the (pixelwise) *trace error* we use the relative Euclidean distance to the target trace $\|\Delta C\| \,/\, \|C\|$ as before. In the literature on ultrafast nonlinear optics often the rms error, also called FROG error, is used.

A. Initialization, first example

As a first example, already discussed above, the retrieval of test pulse A2908 (dashed curves) is shown in Fig. 4 (bottom right) for $N = 25$ with $\sigma_{\text{noise}} = 1\%$, $\delta^{\text{regul}}_{\Delta C} = 25\%$, $p^{\text{stop}}_{\text{up}} = 90\%$. Initial data were set to a Gaussian bell curve with zero phase, where the initial width of the Gaussian is set to minimize the polynomial system including penalty term (26) for some large enough initial value of λ.

The synthetic measurement trace with 129 × 129 points was coarse grained to $N \times N = 25 \times 25$ pixels. In Fig. 4 (top left) the convergence of the trace error while iterating along the solution path is shown, already discussed in a previous section. When averaging over 100 retrievals we get mean trace error/evolution time/within iterations: 0.04/1 s/200, 0.02/2 s/400, 0.019/2.5 s/500; compare with Fig. 4 (top left).

For test pulses without gaps (points or regions with zero amplitude) like TPB1 and A2908 we found no cases of stagnation independent of the noise level, for the retrieval probabilities with noise of the most common solvers (see Fig. 7 [2]). For test pulse TBP2 about 2% of all runs stagnate. In the gap region E^+ and E^- are restricted to be zero, and configurations[18] appear that cannot be freely deformed into one another under the constraint of moving along real “down paths.” Introducing a lifted electrical field by adding a large enough constant $E \rightarrow E + c$ as a new variable which has by construction no gap resolves the problem, which we study in more detail in [38].

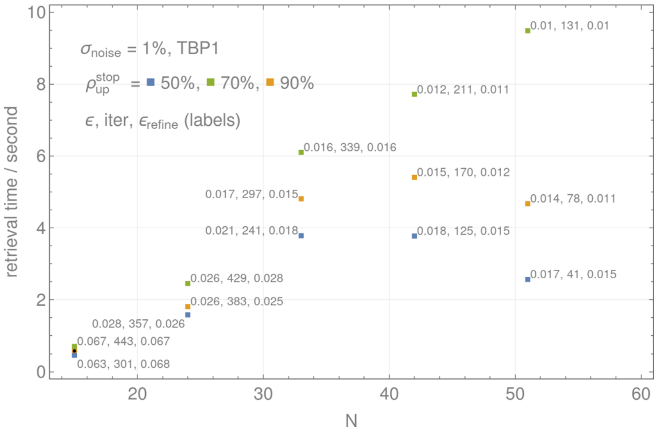

FIG. 5. Performance overview: Retrieval time vs N with 1% additive noise and three different values for the termination criterion $p^{\text{stop}}_{\text{up}}$ and the resulting retrieval accuracy, number of iterations, and retrieval accuracy after applying additional refinement steps all averaged over 10 runs. Initial data for one level is the interpolated result from the next coarser level. The plot implies choosing as smaller $p^{\text{stop}}_{\text{up}}$ while cascading towards the final grid and do refined retrievals there.

At the moment the following practical work-around can be used: first, compute k trial runs ($k \approx 5$) with a zero phase initial Gaussian, zero phase on a coarse grid $N = 15$ to $N = 25$, which takes about 0.3 s to 1 s per run. Then the correct initialization for a fine grid retrieval using this result as initial data has chances $1\text{–}0.10^k$ if the coarse grid retrievals fail in, for example, 10% of the cases.

B. Retrieval timing, computational cost

A more detailed overview of the timing, number of iterations, and the influence of the termination criterion of the implemented algorithm is shown in Fig. 5. Every data point corresponds to an average over 10 runs. As a test case we used the pulse TBP1 with $\sigma_{\text{noise}} = 1\%$. Initial data for one level are the interpolated result from the next coarser level. Then the number of iterations before termination decreases approximately linearly with N for noisy traces and is constant for noiseless cases (not shown here). As mentioned before, a speedup by a factor of two is possible, as any of the summands in Eq. (11) suffice to approximated the total pixel average. Here we use both terms.

A larger value of $p^{\text{stop}}_{\text{up}}$ results in higher accuracy, but the additional costs usually do not justify the small improvement on ϵ at every intermediate level. Figure 5 suggests using $p^{\text{stop}}_{\text{up}} = 50\%$ on coarser grids before reaching the target level and doing refinement there, if required. If speed is a concern, the number of intermediate levels can be optimized. For our purposes a single initial coarse grid and one fine grid

[17] The noise is chosen to have zero mean; if this is not the case, either a background subtraction of $I_{\text{exp}}(\omega, \tau)$ has to be done, or equivalently the zero mode after Fourier transform has to be removed and interpolated for $J_{\text{exp}}(\tau, \sigma)$, which we consider the cleaner choice. The trace $J_{\text{exp}}(\tau, \sigma) \approx J[E_{\text{in}}](\tau, \sigma)$ is initially normalized to have its absolute value maximum equal to one. Then every initial data E should be scaled for the integral $J[E](\tau, \sigma)$ to have the same property.

[18] Taking as an example a double pulse (two pulses separated by a gap) with quadratic phase we found that two false configurations appear which have a small trace error and the correct quadratic phase along each of the two pulses but a phase jump of $+\pi$ or $-\pi$ at the gap. For larger gaps we found a greater likelihood to run into these configurations in numerical experiments.

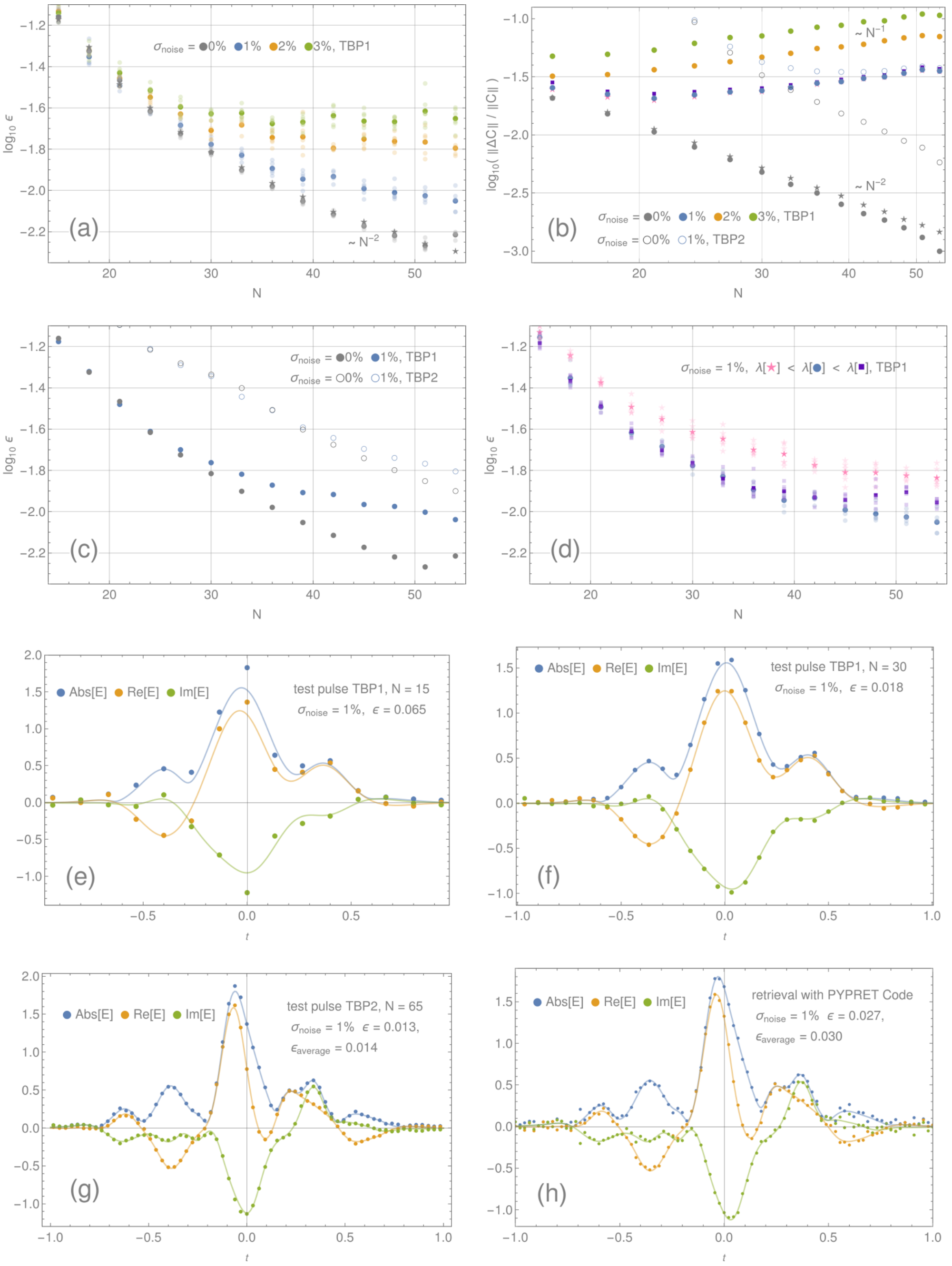

FIG. 6. Error convergence vs N: Average (full colored points) final pulse error ϵ (a, c, d) and trace error $\|\Delta C\|/\|C\|$ (b) on 14 pixelizations for the test cases TBP1 and TBP2 with and without noise. Two examples of the retrieved electric field on a coarse (e) and medium (f) pixelized grid for TBP1. (g) vs (h): Comparison of our solver with the result of a least-squares solver without regularization. For more details see Sec. VI C. Note: Here more intermediate grids then necessary are used. Normally, a cascade like $N \rightarrow 20 \rightarrow 40 \rightarrow 65$ is sufficient.

was enough. As an example $N_{\text{initial}} = 15$, $N_{\text{target}} = 51$, we got $\epsilon_{\text{target}} \approx 0.015$, iterations ≈ 600, retrieval time ≈ 30 s.

Another practical concern is, if the investigated pulse has large low-amplitude wings, a large part of the computational domain and cost are spent on this low-amplitude region, and a clipping or zoom of the experimental data to the region of interest is suggestive, setting the wings to zero at first. Then, in a follow-up retrieval a larger domain could be included using the result as initial data, if required.

C. Convergence, scaling behavior, comparison

In Figs. 6(a)–6(d) the convergence of pulse and trace error are studied on 14 grid levels $N = 15, 18, \ldots, 53$ with six retrievals per N (opaque colored points) and their averages (full colored points) for test pulses TBP1 and TBP2. In Figs. 6(e)–6(g) the input pulse shapes are shown (solid lines) and a retrieval result (points) on $N = 15$ [Fig. 6(e)], $N = 30$ [Fig. 6(f)], $N = 51$ [Fig. 6(g)]. We set $p_{\text{up}}^{\text{stop}} = 90\%$ and do an

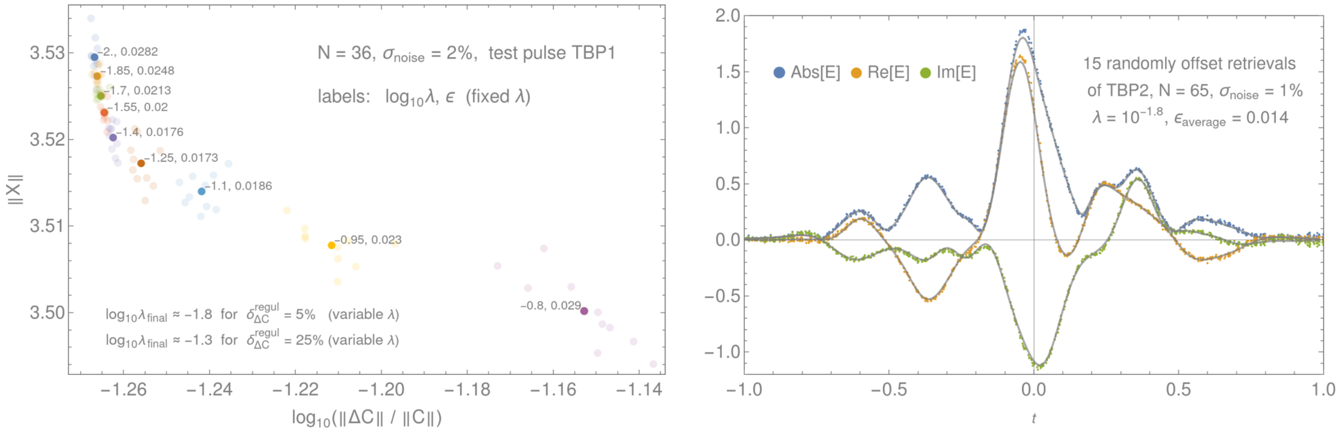

FIG. 7. Left: Applying the L-curve method for nine different values of λ, 10 retrievals each (opaque colored), mean value (full colored). Noisy oscillations begin to increase the length of the pulse $\|X\|$ when decreasing λ; the trace error $\|\Delta C\|$ improves through overfitting. Too smooth solutions (too large λ) show deviations from the original solution and have larger trace errors. The optimal amount of λ is within the corner of the L-curve where the pulse error ϵ is small. Right: 15 randomly offset retrievals of pulse TBP2 (gray lines) on $N = 65$ intervals. As the position of the pulse is not fixed relative to the numerical grid; sampling on arbitrary intergrid locations is possible.

additional centering and two refinement steps at every level; see Sec. VI D.

The retrieval accuracy ϵ without noise (gray dots and stars) [Fig. 6(a)] as well as the trace error converge with $\sim 1/N^2$ (gray dots and stars) [Fig. 6(b)] as the dominant numerical error is stemming from the interpolating of $J[E](\tau,\sigma)$ from boundary values to the pixel interior (9), which is of the order $O(h^2)$. This scaling behavior is overlayed with a $\sim N$ increase of the (pixelwise) relative trace error for noisy traces J_{exp}, as the noise suppression scales with $\sim 1/\sqrt{N^2}$ when averaging over less data points per pixel area for larger N. Still ϵ may decrease while $\|\Delta C\|$ is increasing as more details of the pulse are being resolved when increasing N, until ϵ becomes approximately constant at the accuracy limit [see Fig. 6(c)] for TBP1 with 1% noise (solid circles) at about $\epsilon = 10^{-2}$, $N = 50$ for TBP2 with 1% (empty circles) at about $\epsilon = 10^{-1.9}$, $N = 65$. Resolving more details beyond this point is possible, if fewer noisy traces are input. The dependence of the accuracy limit on the noise level is apparent in Fig. 6(a) (blue, yellow, green). Analogously, for larger enough N the minimal trace error should depend only on the amount of noise and not the particular pulse. This is apparent in Fig. 6(b) [empty (filled) gray and blue points].

To make the numerical integration error apparent when computing pixel surface averages from the trace $J_{\text{exp}}(\tau,\sigma) \rightarrow \langle J_{ij}^{\text{exp}} \rangle$ we chose two different resolutions 129 × 129 points (gray dots) versus 257 × 257 (gray stars) in Figs. 6(a) and 6(b). The integration[19] error is relevant only if the noise level is low enough to reach this high accuracy.

An interesting effect becomes apparent when varying the amount of regularization by varying the threshold $\delta_{\Delta C}^{\text{regul}} = 5\%, 25\%, 40\%$ (pink, violet, blue) [zoom into Fig. 6(b)]. However, the trace error is larger for larger λ (violet, blue vs pink), and the corresponding pulse error, as shown in Fig. 6(d), is smaller (does not apply to even larger λ). This is a reminder that retrieval techniques or optimization codes that aim only at minimizing the trace error without regularization are prone to fitting the noise; the right balance between overfitting and oversmoothing has to be found. Consider this when comparing a retrieval of our algorithm with the result of the least-squares solver (no regularization)[20] used in [2] [see Fig. 6(g) versus Fig. 6(h)].[21] As this effect on the trace error is relatively small, one could also express it differently: there are many possible pulse shapes of varying smoothness which have approximately the same trace error but a rather different pulse error. Through regularization and coarse graining an optimal pulse shape can be computed.

Due to Tikhonov regularization the presented solver provides a smoother and more accurate solution in the pulse tails as well as a smaller total retrieval error[22] [see Fig. 6(g) versus Fig. 6(h), or Fig. 7(right) versus Fig. 6(h)] in comparison with the least-squares solver of [2]; the computational time compares to ≈30 s versus ≈10 s. This solver had the lowest retrieval error for noisy data in comparison with the most common existing solvers; see Fig. 7 in [2]. Another advantage of the presented solver is that the retrieval probability does not

[19] We use the trapezoidal rule for numerical integration and first-order interpolation, if the pixel boundaries do not automatically lie on data grid. First-order interpolation was used to preserve the additive nature of the noise. For realistic traces, where other sources of error dominate, higher order interpolation could be used.

[20] An unpublished version (private repository) of this solver including regularization is available by now.

[21] More grid points (parameters) do not necessarily imply higher resolution and accuracy. An increase beyond the number of significant parameters (effective DOF) for least squares can cause overfitting. In Fig. 6(g) for TBP2 with 1% noise the pulse error did improve beyond $N = 65$, $\epsilon = 0.013$, and similar to TBP1 with 1% [see Fig. 6(c)], the accuracy limit is in both cases at about 10^{-2}. As the position of the pulse relative to the grid is not fixed, the result can be sampled on arbitrary intergrid locations; see next section.

[22] The relatively larger error of the presented solver at the pulse's peak near $t = 0$ disappears when refining the resolution.

depend on the noise level and that the cases of stagnation can be fully resolved; see Sec. VI A and [38].

D. Optimal λ, refined solution, oversampled solution

The most natural choice to test and fine tune the amount of regularization for optimality in the presents of additive noise would be a chi-square test while varying λ because the pulse error is not available *a priori*. Considering modern measurement devices and pulse retrieval setups (see, for example, [39]), for the problem at hand, where (difficult to quantify) systematical errors and multiplicative noise are the most relevant sources of error, a goodness-of-fit test of this type does not seem applicable yet to a measured trace. A popular practical solution that we recommend in this case is the so-called *L-curve* method [30] until more sophisticated techniques are required.

An application of the L-curve method for test case TBP1 with $\sigma_{\text{noise}} = 2\%$, $N = 36$ is shown in Fig. 7, where for fixed λ (adaptive decrease of λ turned off) 10 retrievals (opaque colored points) and the average (full colored points) are shown. For smallest λ the retrieved pulses have smallest trace error (overfitting region), but they do not have the smallest pulse error. The length $\|X\|$ of each pulse is extended by noisy wiggling about some smoother solution, which is reached for increasing λ. In the L's corner, the amount of regularization is optimal, and the pulse error is minimal.

Note that the overfitted solutions also reveal themselves by having the smallest spread (opaque colored points) in the trace error as they differ only by a new random sample of the reverse amplified noise, all samples having on average about the same length.

The implemented mechanism to adaptively decrease λ while path tracking (see Sec. V) steers λ towards optimality. Here are two examples: $\log_{10} \lambda_{\text{final}} = -1.8$ for $\delta_{\Delta C}^{\text{regul}} = 5\%$ and for $\delta_{\Delta C}^{\text{regul}} = 25\%$ $\log_{10} \lambda_{\text{final}} = -1.3$, which is optimal.

As the ratio of up paths relative to all valid paths p_{up} is estimated from a finite sample, the number of iterations before crossing the threshold set by $p_{\text{up}}^{\text{stop}}$ differ slightly. To ensure the solver cannot improve the solution within this threshold we perturb the solution sightly by shifting it one (or more) grid points to the left or right and use it as initial data for a new retrieval. This refinement step does not improve ϵ much if the threshold was already set high, $p_{\text{up}}^{\text{stop}} \approx 90\%$, as shown in Sec. VI B (Fig. 5) but yields a small improvement otherwise.

There is a translational symmetry which has not been discussed yet: $E(t) \rightarrow E(t + \delta t)$. This symmetry is, strictly speaking, broken as the $E(t)$ is defined on a bounded domain $t \in [-1, 1]$ and zero elsewhere. But as we are dealing with finite accuracy solutions and as $E(t)$ models a physical light pulse with low-amplitude wings, there are actually infinitely many similar solutions within a given error bound which differ only by a small shift δt of the pulse relative to the numerical grid. As a consequence, in particular on coarser grids any solution should be centered on the numerical grid before transferring the result. As a benefit, on finer grids, if we reprocess a solution shifted by some small random intergrid distance $\delta t < h$ via interpolation, the result is a shifted solution, sampled on slightly different points $E(t + \delta t)$. With this technique the pulse can be sampled on arbitrary intergrid locations, as shown in Fig. 7 (right), where the above procedure was applied for $N = 65$ for TBP2 with 1% noise. To be more precise, "the" oversampled solution is rather an error band as for noisy traces a spread of near-optimal solutions within some error bound exists. A small deviation from the original pulse is apparent when looking at its peak value, which is caused by the discretization error of $E(t)$ on $N = 65$ intervals that should disappear as the grid is refined. We did the same as above for the pulse TBP2 but with $\lambda = 10^{-1.7}$ (not shown). The average pulse error and oscillations in the error band were slightly less, though small deviations of the mean curve through the error band from the original pulse are visible in some regions. This implies that care has to be taken when fine-tuning λ at the corner of the L-curve, rather, choosing solutions with slightly less λ, slightly bigger $\|X\|$ and averaging to obtain a mean curve. This phenomenon is also apparent in Fig. 7 (left, compare green, red, violet, brown) were all four points have small nearby errors; the smallest for brown, though, is not the optimal choice.

VII. CONCLUSION AND OUTLOOK

An algorithm has been developed having in mind the applicability to real experimental data for common pulse retrieval schemes in ultrafast nonlinear optics like FROG, d-scan, or amplitude swing [3–5]. The employed numerical techniques have been successfully applied in other fields of physics, where nonlinear integral equations of similar type appear, techniques used in polynomial system solvers and stochastic optimization. It has been shown how to implement Tikhonov-type regularization into the polynomial equations, how to adaptively decrease it while path tracking the solution, and how to fine-tune λ at the solution when dealing with noisy, defective experimental data. The integral equation was discretized to a polynomial system such that each equation corresponds to a grid cell (pixel) surface average of the original integral equation, rather than a pointwise representation. This coarsening capability enables fast computations of approximants, noise suppression, and high-accuracy retrievals on fine pixelizations.

The full solution path is a collection of small linked continuation paths, along each the error decreases monotonically by construction, where these path segments are associated with different random projections of the full polynomials system. This constitutes an alternative method for path tracking real solutions of similar overdetermined polynomial systems through these partially continuous and stochastic solution paths. Each movement along a single path segment causes a deformation of the momentary solution in parts in the direction of the global solution and in parts towards some random perturbation. These perturbations appear to cancel when path tracking over a collection of many path segments. Similarly, perturbations due to added noise seem to compensate each other, and an optimal solution can be computed. A link to the theoretical framework developed for the Newton-sketch method [26,27], or in randomized numerical linear algebra [28,40], or in other areas of stochastic optimization seems plausible. The main difference is that the continuation method is implemented here. This may also be advantageous when computing the global minimum of nonconvex optimization problems.

If speed is a concern, there are two ways to accelerate the solver: The most immediate solution is parallelizing the list autocorrelations in (9) on GPUs. Second, it stands to reason trying Jacobian-free Newton-Krylov methods [17] or other quasi-Newton methods [12,41] to replace Newton's method in the algorithm, which could accelerate the computations by a factor of N. Third, when reducing the full system to random linear combinations the Hadamard transform is the method of choice for dimensional reduction [42,43] for many other applications in stochastic optimization.

For realistic FROG or d-scan traces additional frequency dependent functions can be introduced to the nonlinear integral, which model frequency-dependent systematical errors in the nonlinear medium and the experimental setup which are otherwise neglected. These functions could be added to the list of unknowns and retrieved with the presented algorithm, similar to [44].

For refining a retrieved pulse further the pointwise representation of the integral equation could be used as well [see Eq. (7)] if the quality of the measured data admits this. Then the numerical integration errors could be avoided with a small gain in accuracy.

Finally, there are other similar phase retrieval problems in science and engineering [45,46] and in optics and nonlinear optics [47–50] where an application of this algorithm seems plausible.

ACKNOWLEDGMENTS

I would like to thank Günter Steinmeyer, Esmerando Escoto, and Lorenz von Grafenstein from the MBI Berlin, Peter Staudt from APE GmbH, as well as Carl T. Kelley, Michael H. Henderson, Jan Verschelde, and Alex Townsend for helpful comments and discussion. In particular, I thank Nils C. Geib for carefully reading and commenting on the first version of this article. This work was financially support by the European Union through the EFRE program (IBB Berlin, ProFit grant, Contract No. 10164801, project OptoScope) in collaboration between the Max-Born Institute, Berlin, and the company APE GmbH.

APPENDIX A: MODIFICATIONS FOR SIMILAR INTEGRALS

The algorithm is applicable to other similar pulse retrieval schemes when replacing the nonlinearity in the integral (2), $E(t)E(t-\tau) \rightarrow E(t)H(t-\tau)$. For example, for polarization-gate FROG or third-harmonic-generation FROG we have

$$H(t) = E(t)\bar{E}(t),\ E(t)^2,$$

and the integral (3) becomes

$$\begin{aligned} J[E,\bar{E}](\tau,\sigma) \;&=\; \int_{-\infty}^{+\infty} E(t)H(t-\tau)\bar{E}(t-\sigma) \\ &\quad\times \bar{H}(t-\tau-\sigma)\,dt, \\ F_\tau(t) &:= E(t)H(t-\tau), \\ G_\sigma(t) &:= E(t)\bar{H}(t-\sigma). \end{aligned}$$

For the pulse retrieval scheme d-scan [4], again applying the convolution theorem, we get

$$\begin{aligned} I[E](z,\omega) \;&=\; \left|\int_{-\infty}^{+\infty} F_z(t)^2 e^{-i\omega t}\,dt\right|^2 \\ \rightarrow J[E](z,\sigma) \;&=\; \int_{-\infty}^{+\infty} F_z(t)^2 \bar{F}_z(t-\sigma)^2\,dt, \\ F_z(t) &:= \int_{-\infty}^{+\infty} \hat{E}(\omega)\hat{\phi}(\omega,z)e^{i\omega t}\,d\omega \\ \rightarrow F_z(t) \;&=\; \int_{-\infty}^{+\infty} E(t)\phi(s-t,z)\,ds, \end{aligned} \tag{A1}$$

where the function $\hat{\phi}(\omega,z) := e^{izk(\omega)}$ [23] is known and given by z the thickness of the material in the beam path, $k(\omega)$ the dispersion of the material, and $\hat{E}(\omega)$ and $\hat{\phi}(\omega,z)$ are the Fourier transforms of $E(t)$ and $\phi(t,z)$. As before [see Eq. (5)], along lines of constant z the integral $J[E](z,\sigma)$ is a one-dimensional autocorrelation of the function $F_z(t)^2$, and the pixel average can be computed with (11). To compute $F_z(t)$ an additional complex convolution with the material function $\phi(s,z)$ has to be evaluated instead of a multiplication. When using piecewise-constant approximants for $E(t)$ and $\phi(s,z)$ the corresponding list convolution can be computed as before with Eq. (7).

Another useful formula is given by an equivalent form of the nonlinear integral (1) using the frequency domain representation $\hat{E}(\omega)$ of the enveloping electric field

$$I[E](\omega,\tau) := \left|\int_{-\infty}^{+\infty} \hat{E}(\omega-\Omega)\hat{E}(\Omega)e^{i\Omega\tau}\,d\Omega\right|^2; \tag{A2}$$

applying a Fourier transform as before but in the variable $\tau \rightarrow \rho$ and with an opposite sign, the frequency domain representation of the integral $\hat{J}[E](\rho,\omega)$ appears to be an autoconvolution instead of an autocorrelation in the auxiliary variable $\hat{F}_\omega(\Omega)$

$$\begin{aligned} \hat{J}[E](\rho,\omega) \;&=\; \int_{-\infty}^{+\infty} \hat{F}_\omega(\Omega)\bar{\hat{F}}_\omega(\rho-\Omega)\,d\Omega, \\ \hat{F}_\omega(\Omega) &:= \hat{E}(\Omega)\hat{E}(\omega-\Omega). \end{aligned} \tag{A3}$$

Therefore, instead of retrieving $E(t)$ the whole formalism can be rephrased to retrieve $\hat{E}(\omega)$. The same applies to the other pulse retrieval schemes from above.

APPENDIX B: HIGHER ORDER POLYNOMIALS, SPLINES

It is possible to use polynomials of higher order, like splines in each interval; compare with Eq. (6):

$$E(t) = \begin{cases} 0 & t < -1 \quad \text{or} \quad 1 < t \\ \sum_{A=0}^{p} E_k^A\,\hat{t}^A & t \in [t_k, t_{k+1}], \quad k = 0, \ldots, N-1 \end{cases}$$

[23] For the pulse retrieval scheme amplitude-swing $\hat{\phi} := \hat{\phi}(\omega,\theta)$ is a function of ω and the angle θ, the relative orientation of a rotating birefringent material and a linear polarizer [5].

with polynomial order p and $\hat{t}$ is again the local interval coordinate. Then the function $F_\tau(t)$

$$
\begin{aligned}
{}^{(i)}F_k(\hat{t}) &= E_k^0 E_{k+i}^0 + (E_k^0 E_{k+i}^1 + E_k^0 E_{k+i}^1)\hat{t} + \cdots,\\
{}^{(i)}F_k(\hat{t}) &= {}^{(i)}C_k^0 + {}^{(i)}C_k^1\,\hat{t} + \cdots + {}^{(i)}C_k^{2p}\,\hat{t}^{2p},\\
{}^{(i)}\bar{F}_k(\hat{t}) &= {}^{(i)}D_k^0 + {}^{(i)}D_k^1\,\hat{t} + \cdots + {}^{(i)}D_k^{2p}\,\hat{t}^{2p}
\end{aligned}
$$

discretizes into a polynomial of order $2p$ for which the integration in Eq. (7) becomes

$$
J[E](\tau_i,\sigma) = h \sum_{m=0}^{(2p+1)^2} \int_{\hat{\sigma}-1}^{1} \hat{t}^{A(m)}(\hat{t}-\hat{\sigma})^{B(m)} d\hat{t} \sum_{k=1}^{N} {}^{(i)}C_k^{A(m)}\,{}^{(i)}D_{k+j}^{B(m)} + \cdots, \tag{B1}
$$

where we use an index vector $(A(m), B(m))$ in degree lexicographical order with lowest order first:

$$
\begin{aligned}
\langle J[E]_{ij}^{\text{left right}}\rangle = \frac{h}{4}\sum_{m=0}^{(2p+1)^2} v_m\big(&\text{corr}({}^{(i)}C^{A(m)}{}_k,\,{}^{(i)}D^{B(m)}{}_k)_j + \text{corr}({}^{(i+1)}C^{A(m)}{}_k,\,{}^{(i+1)}D^{B(m)}{}_k)_j\big)\\
+\frac{h}{4}\sum_{m=0}^{(2p+1)^2} w_m\big(&\text{corr}({}^{(i)}C^{A(m)}{}_k,\,{}^{(i)}D^{B(m)}{}_k)_{j+1} + \text{corr}({}^{(i+1)}C^{A(m)}{}_k,\,{}^{(i+1)}D^{B(m)}{}_k)_{j+1}\big).
\end{aligned}
$$

As an example, consider a polynomial chain $p = 1$, then ${}^{(i)}C_k^A$, $A = 0, \ldots, 2$ and $(2p+1)^2 = 9$ list correlations that have to be computed on every interval as compared to 1 for $p = 0$. The index vector is $(A(m), B(m)) = ((0,0), (0,1), (1,0), (0,2), (1,1), (2,0), (1,2), (2,1), (2,2))$. The canonical integrals over $\hat{t}$ in Eq. (B1) turn into a list of integers when integrating over $\hat{t}$ and $\hat{\sigma}$:

$$
\begin{aligned}
v_m &= (2,\ -2/3,\ +2/3,\ 2/3,\ 0,\ 2/3,\ +2/15,\ -2/15,\ 2/9),\\
w_m &= (2,\ +2/3,\ -2/3,\ 2/3,\ 0,\ 2/3,\ -2/15,\ +2/15,\ 2/9).
\end{aligned}
$$

[1] I. A. Walmsley and C. Dorrer, Characterization of ultrashort electromagnetic pulses, Adv. Opt. Photon. **1**, 308 (2009).

[2] N. C. Geib, M. Zilk, T. Pertsch, and F. Eilenberger, Common pulse retrieval algorithm: A fast, and universal method to retrieve ultrashort pulses, Optica **6**, 495 (2019).

[3] R. Trebino and D. J. Kane, Using phase retrieval to measure the intensity, and phase of ultrashort pulses: Frequency-resolved optical gating, JOSA A **10**, 1101 (1993).

[4] M. Miranda, T. Fordell, C. Arnold, A. L'Huillier, and H. Crespo, Simultaneous compression, and characterization of ultrashort laser pulses using chirped mirrors, and glass wedges, Opt. Express **20**, 688 (2012).

[5] B. Alonso, W. Holgado, and Í. J. Sola, Compact in-line temporal measurement of laser pulses with amplitude swing, Opt. Express **28**, 15625 (2020).

[6] J. Hyyti, E. Escoto, and G. Steinmeyer, Pulse retrieval algorithm for interferometric frequency-resolved optical gating based on differential evolution, Rev. Sci. Instrum. **88**, 08 (2017).

[7] R. W. Gerchberg and W. Saxton, A practical algorithm for the determination of phase from image, and diffraction plane pictures, Optik **35**, 237 (1971).

[8] K. DeLong, D. Fittinghoff, R. Trebino, B. Kohler, and K. Wilson, Pulse retrieval in frequency-resolved optical gating based on the method of generalized projections, Opt. Lett. **19**, 2152 (1994).

[9] D. J. Kane, Real-time measurement of ultrashort laser pulses using principal component generalized projections, IEEE J. Sel. Top. Quantum Electron. **4**, 278 (1998).

[10] P. Sidorenko, O. Lahav, Z. Avnat, and O. Cohen, Ptychographic reconstruction algorithm for frequency-resolved optical gating: Super-resolution, and supreme robustness, arXiv:1606.09622 (2016).

[11] C. T. Kelley, Numerical methods for nonlinear equations, Acta Numerica **27**, 207 (2018).

[12] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Vol. 1 (SIAM, Raleigh, North Carolina, 2003).

[13] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance, and Adaptive Algorithms*, Springer Series in Computational Mathematics Vol. 35 (Springer Science & Business Media, 2011).

[14] L. S. Ornstein and F. Zernike, Accidental deviations of density, and opalescence at the critical point of a single substance, Proc. Akad. Sci. **17**, 793 (1914).

[15] C. Kelley and B. M. Pettitt, A fast solver for the Ornstein-Zernike equations, J. Comput. Phys. **197**, 07 (2004).

[16] S. Chandrasekhar, *Radiative Transfer* (Dover, New York, 1960).

[17] D. A. Knoll and D. E. Keyes, Jacobian-free Newton–Krylov methods: A survey of approaches, and applications, J. Comput. Phys. **193**, 357 (2004).

[18] A. Collier, A. Hindmarsh, R. Serban, and C. Woodward, User documentation for KINSOL v2.7.0, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, UCRL-SM-208116 (2012).

[19] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps,*et al.*, An overview of the Trilinos project, ACM Trans. Math. Softw. **31**, 397 (2005).

[20] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. F. Smith, and H. Zhang, PETSc users manual, v. 3.11, Oxford University Research Archive (2019).

[21] E. Allgower and K. Georg, Continuation, and path following, Acta Numerica **2**, 1 (1993).

[22] A. Morgan, *Solving Polynominal Systems Using Continuation for Engineering, and Scientific Problems* (SIAM, 2009).

[23] A. J. Sommese and C. W. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science* (World Scientific, University of Notre Dame du Lac, USA, 2005).

[24] D. J. Bates, A. J. Sommese, J. D. Hauenstein, and C. W. Wampler, *Numerically Solving Polynomial Systems with Bertini* (Society for Industrial and Applied Mathematics, Philadelphia, 2013).

[25] J. Verschelde, Polynomial homotopy continuation with PHCpack, ACM Commun. Comput. Algebra **44**, 217 (2011).

[26] M. Pilanci and M. J. Wainwright, Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence, SIAM J. Optim. **27**, 205 (2017).

[27] A. S. Berahas, R. Bollapragada, and J. Nocedal, An investigation of Newton-sketch, and subsampled Newton methods, Optimization Methods, and Software **35**, 661 (2020).

[28] P.-G. Martinsson and J. Tropp, Randomized numerical linear algebra: Foundations & algorithms, arXiv:2002.01387 (2020).

[29] V. Y. Pan and A.-L. Zheng, New progress in real, and complex polynomial root-finding, Comput. Math. Appl. **61**, 1305 (2011).

[30] P. C. Hansen, The L-curve, and its use in the numerical treatment of inverse problems, Adv. Comput. Bioeng. **5**, 119 (2001).

[31] R. Jafari and R. Trebino, High-speed "multi-grid" pulse-retrieval algorithm for frequency-resolved optical gating, Opt. Express **26**, 2643 (2018).

[32] J. O. Smith, *Spectral Audio Signal Processing*, http://ccrma.stanford.edu/~jos/sasp/, online book, 2011 edition (2020).

[33] T. Li and X. Wang, Solving real polynomial systems with real homotopies, Math. Comput. **60**, 669 (1993).

[34] H. B. Keller, Lectures on numerical methods in bifurcation problems, Appl. Math. **217**, 50 (1987).

[35] M. Henderson and H. Keller, Complex bifurcation from real paths, SIAM J. Appl. Math. **50**, 04 (1990).

[36] H. W. Engl, M. Hanke, and, A. Neubauer, *Regularization of Inverse Problems*, Mathematics and Its Applications, Vol. 375 (Springer Science & Business Media, 1996).

[37] S. Vigna, Further scramblings of Marsaglia's xorshift generators, J. Comput. Appl. Math. **315**, 04 (2014).

[38] M. Jasiulek and G. Steinmeyer, Robust numerical path tracking for laser pulse retrieval (unpublished).

[39] N. C. Geib, R. Hollinger, E. Haddad, P. Herrmann, F. Légaré, T. Pertsch, C. Spielmann, M. Zürch, and F. Eilenberger, Discrete dispersion scan setup for measuring few-cycle laser pulses in the mid-infrared, Opt. Lett. **45**, 5295 (2020).

[40] P. Drineas and M. W. Mahoney, Randnla: randomized numerical linear algebra, Commun. ACM **59**, 80 (2016).

[41] C. T. Kelley, *Iterative Methods for Linear, and Nonlinear Equations* (SIAM, 1995).

[42] N. Ailon and B. Chazelle, The fast Johnson-Lindenstrauss transform, and approximate nearest neighbors, SIAM J. Comput. **39**, 302 (2009).

[43] C. Boutsidis and A. Gittens, Improved matrix algorithms via the subsampled randomized Hadamard transform, SIAM J. Matrix Anal. Appl. **34**, 1301 (2013).

[44] M. Miranda, J. Penedones, C. Guo, A. Harth, M. Louisy, L. Neoričić, A. L'Huillier, and C. L. Arnold, Fast iterative retrieval algorithm for ultrashort pulse characterization using dispersion scans, JOSA B **34**, 190 (2017).

[45] E. J. Candes, X. Li, and M. Soltanolkotabi, Phase retrieval via Wirtinger flow: Theory, and algorithms, IEEE Trans. Inf. Theory **61**, 1985 (2015).

[46] A. Fannjiang and T. Strohmer, The numerics of phase retrieval, Acta Numer. **29**, 125 (2020).

[47] J. R. Fienup, Phase retrieval algorithms: A comparison, Appl. Opt. **21**, 2758 (1982).

[48] A. Walther, The question of phase retrieval in optics, Opt. Acta: Int. J. Opt. **10**, 41 (1963).

[49] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, Phase retrieval with application to optical imaging: A contemporary overview, IEEE Signal Proc. Mag. **32**, 87 (2015).

[50] Y. Mairesse and F. Quéré, Frequency-resolved optical gating for complete reconstruction of attosecond bursts, Phys. Rev. A **71**, 011401(R) (2005).