Algorithm based fault tolerance with wavelets

Roman Andreev

ALGORITHM BASED FAULT TOLERANCE WITH WAVELETS

ROMAN ANDREEV

Abstract. In wavelet Galerkin discretizations of partial differential equations the value of a coefficient of the discrete solution directly translates into its importance. In the context of a parallel iterative computation on faulty computational nodes we propose to use this information to distribute the coefficients in a way that minimizes the expected loss upon hard node failure, thus dramatically increasing the chances of approaching the discretization accuracy.

1. Introduction

A parallel computation may experience hardware or software failure [3]. We focus here on the so-called hard failure of computational nodes. We devise a data allocation scheme that, based on the failure probability of those nodes, distributes the data such as to minimize the expected loss, assuming that every piece of data (*item*) has a certain additive importance (*item value*) associated with it independently of other items. This situation appears in iterative solution algorithms for linear algebraic systems arising from discretizations of partial differential equations. Changing to a wavelet basis allows to interpret each coefficient as an item and to assign an item value to it in a natural way. Thus we propose to dynamically redistribute the coefficients in the course of the iterative solution algorithm across the computational nodes in a way that minimizes the expected accuracy loss due to the potential node failure. This redistribution is an instance of *algorithm based fault tolerance* and complements other fault resilience techniques such as checkpointing.

In the first part we introduce our data allocation model and propose algorithms for optimizing the allocation. In the second part we present proof-of-concept numerical experiments with a wavelet discretization of a simple differential equation.

2. Data allocation model

2.1. **Definitions.** A collection I of *items* is to be stored on N *nodes.* With each item i we associate an *item value* $v_i \geq 0$, and set

$$v_{\text{tot}} := \textstyle\sum_i v_i.$$

All items have the same size of one unit. The n-th node can store an integer number $C_n \geq 0$ of distinct items, called the *capacity* of node n. We assume that each item can be replicated and stored at any node at no cost while within the capacity. We call

$$\tfrac{1}{\#I} \left(\textstyle\sum_n C_n\right) - 1$$

the *redundancy index.* It is nonnegative if and only if the joint capacity of nodes is sufficient to store all items. A *pure allocation map* is a binary matrix $A \in \{0,1\}^{I\times N}$ with the column sum $\sum_i a_{in} \leq C_n$ for each node n. Let

$$\mathcal{A}_{01} := \{A = (a_{in})_{in} \in \{0,1\}^{I\times N} : \textstyle\sum_i a_{in} \leq C_n\} \tag{1}$$

Date: September 4, 2016.

2010 *Mathematics Subject Classification.* 35Q68, 65N30, 65Y05, 68M15.

collect all such pure allocation maps. A *node status vector* is a binary vector $x \in \{0,1\}^N$ with the interpretation that

$$\text{the } n\text{-th node is operational if and only if } x_n = 1.$$

Given a node status vector x and a pure allocation map A we call

$$\phi_x(A) := \textstyle\sum_i (1 - \prod_n (1 - x_n a_{in})) v_i \tag{2}$$

the *accessible value*. Note that $(1 - x_n a_{in})$ is zero if $x_n = 1$ and $a_{in} = 1$, which means that node n is operational *and* stores item i. Thus the product over n equals zero if item i is available on *some* node, and equals one otherwise. The accessible value $\phi_x(A)$ is therefore the total value of available items, where each item is counted at most once. An alternative and sometimes more convenient way to write the product in the definition of $\phi_x(A)$ for binary x and A is $\prod_n (1 - x_n a_{in}) = \prod_n (1 - x_n)^{a_{in}}$, where $0^0 := 1$.

We extend $\phi_x(A)$ as defined in equation (2) to a function on $x \in \mathbb{R}^I$ and $A \in \mathbb{R}^{I \times N}$ in the obvious multiaffine way, so that $\phi_x(A)$ is affine in each column of A. By nonnegativity of the item values, $\phi_x(A)$ is nondecreasing in x and in A in the sense that $\phi_x(A) \geq \phi_y(B)$ whenever $A \geq B$ and $x \geq y$ componentwise. Let $\mathcal{A} := \operatorname{conv} \mathcal{A}_{01}$ denote the closed convex hull of $\mathcal{A}_{01} \subset \mathbb{R}^{I \times N}$. A point in $\mathcal{A}$ is called a *mixed allocation map.*

2.2. **Allocation optimization.** We suppose now that x above is a vector of independent binary random variables with probabilities $p_n := \mathbb{P}[x_n = 0]$ and $\hat{p}_n := \mathbb{P}[x_n = 1] = 1 - p_n$. In particular $\mathbb{E}[x_n] = \hat{p}_n$. We refer to p_n as the *failure probability* of node n. Given $A \in \mathcal{A}$,

$$\bar{\phi}(A) := \mathbb{E}[\phi_{(\cdot)}(A)] = \textstyle\sum_i (1 - \prod_n (1 - \hat{p}_n a_{in})) v_i$$

is the *expected accessible value.* Observe that $\bar{\phi}$ is multilinear and nonnegative on $\mathcal{A}$, and $\bar{\phi}(0) = 0$. We can now formulate the allocation problem:

$$\text{Find some} \quad A \in \mathcal{A}_{01} \quad \text{s.t.} \quad \bar{\phi}(A) \geq \bar{\phi}(B) \quad \forall B \in \mathcal{A}.$$

Alternatively, one can minimize the *expected loss*

$$\bar{\psi}(A) := \textstyle\sum_i v_i \prod_n p_n^{a_{in}},$$

which is related to the expected accessible value by $\bar{\phi} + \bar{\psi} = v_{\text{tot}}$ for pure allocation maps $A \in \mathcal{A}_{01}$. Since the solution may not be unique, the problem could be supplemented by the proximity condition that the number of differing entries $|A_{\text{ref}} - A|_0$ for a given $A_{\text{ref}} \in \mathcal{A}_{01}$ be minimal, or relaxed to $\operatorname{argmin}_{A \in \mathcal{A}_{01}} \{\bar{\phi}(A) + \gamma \frac{1}{p} |A_{\text{ref}} - A|_p^p\}$, where $|\cdot|_p$ denotes the vector p-norm of all entries (neither condition guarantees uniqueness).

2.3. **Case of two items.** Suppose $\#I = 2$. We can assume that each node has capacity $C_n = 1$, excluding uninteresting cases. Assume also $v_i = 1$ for the item values. A pure allocation map that minimizes the expected cost is then determined by its first row $a \in \{0,1\}^N$, the second one being the complementary $b := 1 - a$, so that we can write $\bar{\psi}(a) = p^a + p^b$ using the multiindex notation. Setting $\alpha := p^a$ and $\beta := p^b$, observe that the product $\alpha\beta$ is independent of a. This implies that the correspondence $(\alpha + \beta) \mapsto |\log(\alpha/\beta)|$ is nondecreasing. Consequently, minimizing $\bar{\psi}(a) = \alpha + \beta$ is equivalent to minimizing $|\log(\alpha/\beta)| = |S_1 - S_0|$, where $S_1 = \sum_n a_n \log p_n$ and $S_0 = \sum_n (1 - a_n) \log p_n$, over $a \in \{0,1\}^N$. This is the optimization version of the set partition problem, which is NP-hard in N. Hence we will not be aiming at solving the allocation problem exactly.

2.4. **Case of two nodes.** The case $N = 2$ of two nodes can be treated explicitly. By symmetry suppose $0 < p_1 \leq p_2 < 1$ for the failure probabilities. Let $v_{\sigma(1)} \geq v_{\sigma(2)} \geq \ldots \geq v_{\sigma(\#I)}$ be a nonincreasing rearrangement of the item values. If an allocation map $A \in \mathcal{A}_{01}$ is a minimizer of the expected loss $\bar{\psi}$, the capacity $C_n = \sum_i a_{in}$ of each node is exhausted, and $(C_n - C_{12})$ items are stored on node n only, where $C_{12} \geq 0$ is the number of shared items that are stored on both nodes. For any fixed number C_{12} of shared items, the allocation map A with rows

(1) $a_i = (1, 1)$ for $1 \leq \sigma(i) \leq C_{12}$,
(2) $a_i = (1, 0)$ for $C_{12} < \sigma(i) \leq C_1$,
(3) $a_i = (0, 1)$ for $C_1 < \sigma(i) \leq C_1 + C_2 - C_{12}$,
(4) $a_i = (0, 0)$ else,

minimizes the expected loss $\bar{\phi}(A) = \sum_i v_i \prod_n p_n^{a_{in}}$, as can be seen with the rearrangement inequality and $p_1p_2 \leq p_1 \leq p_2 < 1$. Increasing C_{12} by one decreases the expected loss by the amount $(1 - p_2)(p_1 v_{\sigma(C_{12}+1)} - v_{\sigma(C_1+C_2-C_{12})})$, which is nonnegative as long as

$$p_1 v_{\sigma(C_{12}+1)} \geq v_{\sigma(C_1+C_2-C_{12})}. \tag{3}$$

Once this condition is violated, it remains false by the monotonicity of $v_{\sigma(\cdot)}$. We arrive at the following algorithm for finding an optimal pure allocation map for two nodes:

(1) Initialize $C_{12} = \min\{\max\{C_1 + C_2 - \#I, 0\}, \#I\}$.
(2) While $C_{12} < \min\{C_1, C_2\}$ and condition (3) is satisfied: Increase C_{12} by one.
(3) Construct A as above.

We remark that the algorithm does not require $C_n \leq \#I$ or $\#I \leq C_1 + C_2$.

2.5. **Pairwise optimization.** One might attempt to minimize the expected loss $\bar{\psi}(A)$ by optimizing A node by node, but it is more effective to optimize over pairs of nodes:

(1) Let an initial guess $A \in \mathcal{A}$ be given.
(2) Repeat:
    (a) Choose two nodes $n_1 \neq n_2$ uniformly at random.
    (b) Update the columns n_1 and n_2 of A using the two node optimization procedure with the effective item weights $\tilde{v}_i := v_i \prod_{n \notin \{n_1, n_2\}} p_n^{a_{in}}$.

Once each node has been selected, the current allocation map A will necessary be a pure allocation map that exhausts the node capacities, even if the initial guess was not a pure allocation map. The optimization process can be stopped as soon as the value of $\bar{\psi}(A)$ stabilizes, but for simplicity, in the experiments below we always perform exactly 100 iterations (observed to be more than sufficient) starting with the zero allocation map.

In a (near-)optimum solution the most reliable nodes tend to store the most valuable items. The random pair selection step could therefore be replaced by a systematic process which proceeds from the most reliable to the least reliable nodes starting with the zero allocation map. In a similar vein, the likelihood of a random selection of a pair of nodes could be anticorrelated with the probability of their simultaneous failure.

3. Application to wavelet discretization

3.1. **Model problem.** As a model problem we take the elliptic equation $u'' = 1$ on the unit interval $(0, 1)$ with zero Dirichlet boundary conditions. The problem is discretized with P1 finite elements (hat functions) on an equidistant mesh consisting of 2^{10} subintervals. This leads to the algebraic equation $\mathbf{Au} = \mathbf{f}$, where the boundary degrees of freedom have already been eliminated. Now we suppose we have a wavelet basis [2] at hand that forms a Riesz basis in H^1 (we will use the heuristic wavelet construction from

[1] with a slight adaptation to the boundary conditions). Let $\mathbf{T}$ denote the matrix such that $\text{wavelet}_i = \sum_k \mathbf{T}_{ik}\, \text{hat}_k$, or in other words, $\mathbf{T}^\mathsf{T}$ applied to a vector of multiscale coefficients yields a vector of single scale coefficients. By the Riesz basis property, the matrix $\mathbf{TAT}^\mathsf{T}$ is well-conditioned uniformly in the discretization level. Thus we consider the convergent stationary Richardson iteration

$$\mathbf{w}_{\ell+1} = \mathbf{w}_\ell + (\mathbf{Tf} - \mathbf{TAT}^\mathsf{T}\mathbf{w}_\ell), \quad \mathbf{w}_0 := \mathbf{0}, \tag{4}$$

where $\mathbf{w}_\ell$ is the vector of coefficients of the ℓ-th iterate $u_\ell \approx u$ with respect to the wavelet basis. Specifically, we scale the wavelet basis such that the contraction factor is $\rho(\mathbf{I} - \mathbf{TAT}^\mathsf{T}) = 0.95$, which entails an error reduction factor of about $0.95^{100} \approx 0.006$ every 100 iterations, cf. Figure 1. Much lower contraction factors are in fact possible but we wish to simulate here a more general situation.

We consider the following model scenario. Suppose the iterate $\mathbf{w}_\ell$ is distributed across N computational nodes. We wish to perform one Richardson iteration, but some of the nodes might fail in the process and corrupt the accuracy (measured in the H^1 semi-norm). How should we distribute the coefficients such as to minimize the loss of accuracy? To that end we interpret the coefficients as data items and use the Riesz basis property to interpret $v_i := |(\mathbf{w}_\ell)_i|^2$ as the item value of the i-th coefficient.

3.2. **Numerical experiments.** We use the pairwise optimization algorithm from Section 2.5 in each Richardson iteration to determine an allocation map. The wavelet coefficients which are not covered by the allocation map after a possible simulated node failure are set to zero. The failure probability p_n of the n-th node during one Richardson iteration is fixed at the beginning of the iteration as a random number distributed uniformly between 1% and 2%. We perform 10^5 Richardson iterations for different number of nodes N with different capacities C_n to obtain statistics on the error $e_\ell := |u - u_\ell|_{H^1}$. The capacities are determined through the redundancy index r by setting $C_n := (1+r)\lceil (2^{10}+1)/N \rceil$ for each node. We vary $N = 2, 4, 8$, and $2r = -1, 0, 1, 2, \ldots$. If $N = 2$ then only $r \leq 1$ is meaningful.

In Figure 1 we document the error e_ℓ during the first 300 Richardson iterations. Each plot contains the runs a) without node failure, b) on one faulty node of sufficient capacity, and c) on N nodes with the optimized allocation. These are labeled "ideal", "faulty", and "optim", respectively. We see that the optimization strategy, even if the redundancy index is negative, allows to greatly reduce the loss of accuracy at the cost of a slightly elevated error as $\ell \to \infty$. Figure 2 documents the distribution of the error for the same runs over 10^5 Richardson iterations and shows that the optimized allocation forces the error to hover close to the discretization error, whereas the error is typically much larger in the single faulty node scenario. Increasing the number of nodes (for the same redundancy index) slightly increases the error of the optimized strategy because node failure becomes slightly more likely. Increasing the redundancy index (for the same number of nodes) strongly decreases that error and practically recovers the ideal faultless situation.

4. Conclusions

Under idealized assumptions such as negligible node to node communication cost and uniform data item size, we have developed a framework for data redistribution across faulty computational nodes with the aim of minimizing the expected data loss under potential node failure. We have then proposed and illustrated the notion that wavelet discretizations of partial differential equations naturally fit that framework and thus induce algorithm based fault tolerance.

5. Acknowledgment

Supported by French ANR-12-MONU-0013 and Swiss NSF #164616.

References

[1] Roman Andreev. Wavelet-in-time multigrid-in-space preconditioning of parabolic evolution equations. *SIAM J. Sci. Comput.*, 38(1):A216–A242, 2016.

[2] Wolfgang Dahmen. Wavelet and multiscale methods for operator equations. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 55–228. Cambridge Univ. Press, Cambridge, 1997.

[3] Ifeanyi P. Egwutuoha, David Levy, Bran Selic, and Shiping Chen. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *J. Supercomput.*, 65(3):1302–1326, 2013.

(R. Andreev) Université Paris Diderot, Sorbonne Paris Cité, LJLL (UMR 7598 CNRS), F-75205 Paris, France
E-mail address: `roman.andreev@upmc.fr`

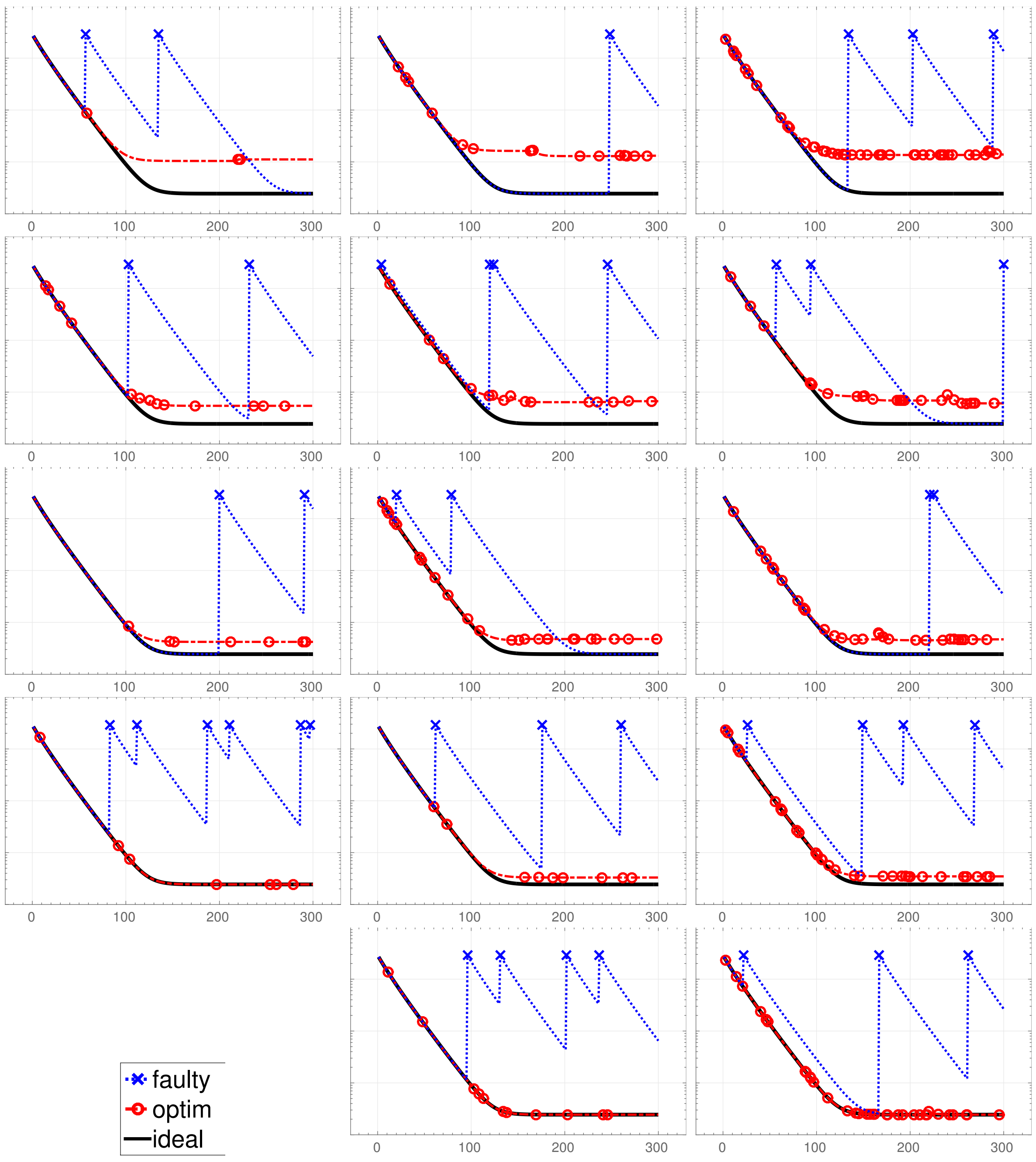

FIGURE 1. Estimated error $|u-u_i|_{H^1}$ as a function of the iteration number $1 \le i \le 300$, for ideal, faulty, and optimized computation. The vertical logarithmic scale ranges from 10^{-4} to 1. Crosses/circles indicate node failure. The optimized computation is performed on $N = 2, 4, 8$ nodes ($\rightarrow$) with redundancy index r given by $2r = -1, 0, 1, 2, 3$ ($\downarrow$).

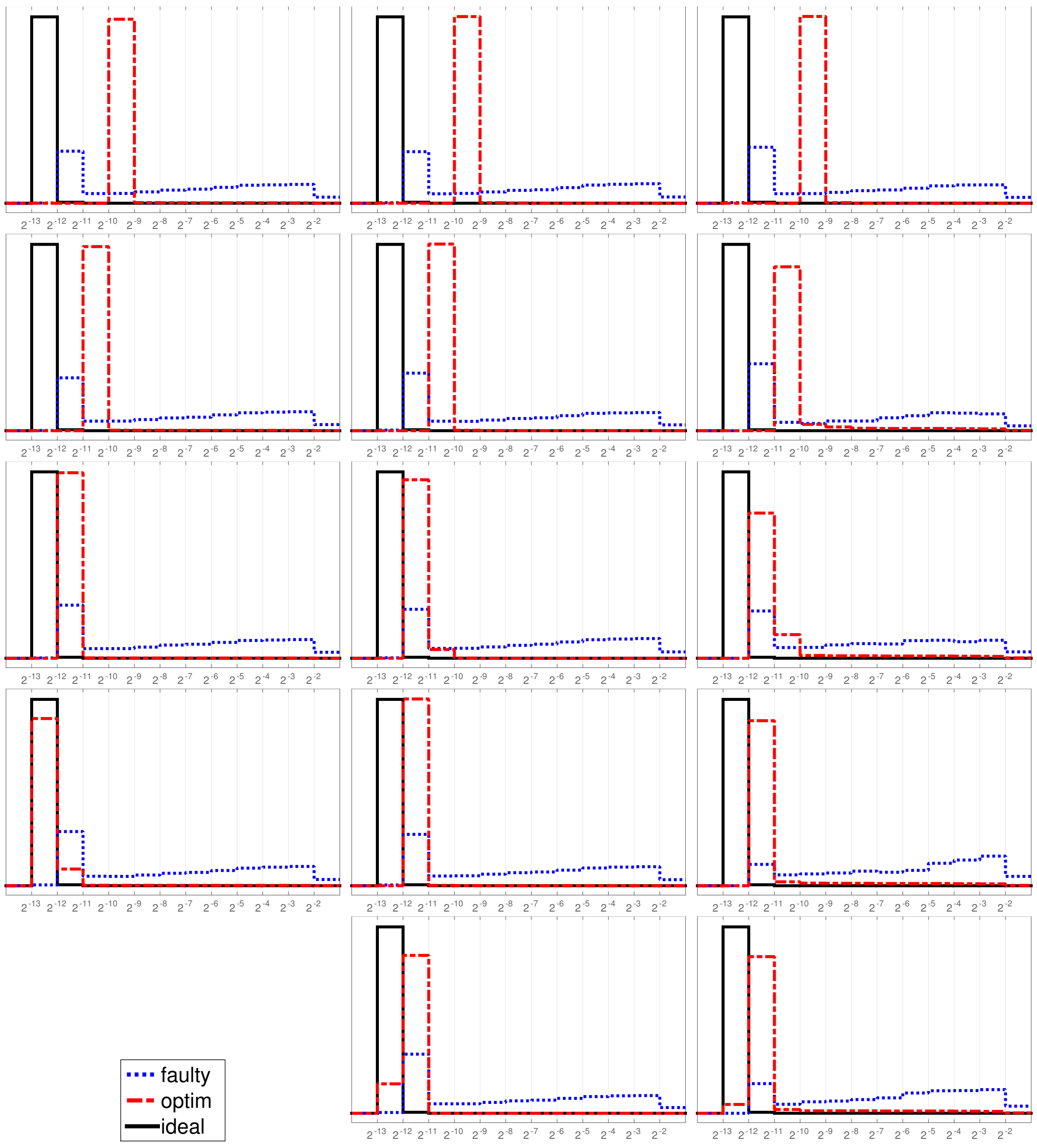

FIGURE 2. Distribution of the error (as in Figure 1) over 10^5 Richardson iterations. The bins are delimited by $2^{-13}, 2^{-12}, \ldots, 2^{-2}$, the area under each curve equals 100%.