

EFFICIENT TARGET-RESPONSE INTERPOLATION FOR A GRAPHIC EQUALIZER

Jose A. Belloch*

Dept. of Computer Science and Engineering
Universitat Jaume I
Castellón de la Plana, Spain
email: jbelloch@uji.es

Vesa Välimäki

Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland
email: vesa.valimaki@aalto.fi

ABSTRACT

A graphic equalizer is an adjustable filter in which the command gain of each frequency band is practically independent of the gains of other bands. Designing a graphic equalizer with a high precision requires evaluating a target response that interpolates the magnitude response at several frequency points between the command gains. Good accuracy has been previously achieved by using polynomial interpolation methods such as cubic Hermite or spline interpolation. However, these methods require large computational resources, which is a limitation in real-time applications. This paper proposes an efficient way of computing the target response without sacrificing the approximation accuracy. This new approach called Linear Interpolation with Constant Segments (LICS) reduces the computing time of the target response by 55% and has an intrinsic parallel structure. Performance of the LICS method is assessed on an ARM Cortex-A7 core, which is commonly used in embedded systems.

Index Terms—Acoustic signal processing, audio systems, equalizers, interpolation, low power processors.

1. INTRODUCTION

Equalizers correct or enhance signal characteristics in order to meet a desired requirement. In audio technology, equalizers implemented as digital filters are commonly used to correct the magnitude response of loudspeakers and headphones to improve the listening experience [1, 2, 3, 4]. Furthermore, equalizers are widely used in music production and in sound reproduction to control the timbral balance of music [5, 6] and to reduce the effects of room acoustics on the sound quality [7, 8]. A graphic equalizer has the center frequencies fixed together with bandwidth. In the graphic equalizer, the user only controls each band gain by using a set of sliders, which form approximately the desired magnitude response [9, 10, 11, 12]. This paper focuses on the interpolation of a target response based on band gains in graphic equalizer design.

The main challenge in designing a graphic equalizer is the interaction of adjacent band filters with each other [13, 14, 12, 15]. This leads to problems in reaching a desired command gain in a band, which differs much from its neighbor. However, if band filters are made sharp enough so that good separation between all command gains is obtained, producing a flat response becomes difficult, when

the neighboring gains are the same [14]. A graphic equalizer consisting of cascaded high-order band filters has been suggested as one solution [9]. For best performance, an iterative design is needed, and band filters then have a different order from each other [16].

Rämö et al. [12] recently proposed using an optimized parallel filter as a graphic equalizer, since it outperforms other graphic equalizers having non-iterative design and enables the efficient use of multi-core processors, such as GPUs [17]. This design has many desirable properties, such as that the optimization accounts for the interaction between the band filters, making their gain seemingly independent of the other gains. It produces a flat magnitude response when adjacent command gains are the same, such as all commands raised to +12 dB, since the overall gain is mainly adjusted with a direct path gain whereas the parallel filters are largely shut down when they are not required. The computational load of the parallel graphic equalizer is also very modest: its operation count per sample is only 23% larger than that of a basic graphic equalizer consisting of cascaded biquad sections [12].

Nonetheless, the parallel graphic equalizer still has one feature making it less attractive than previous methods [12]: updating its parameters, when a command gain is changed, takes two orders of magnitude more operations than in a basic equalizer. Parameter update is necessary during real-time operation whenever a gain is modified. As part of the parameter update, a target response must be obtained from the gain values. To this end, cubic Hermite interpolation or splines have been used previously, since they connect the command points smoothly without overshooting [13], [12]. However, the use of these methods requires high computational capacity, because they need estimation of derivatives around data points, which is a limitation in a real-time application. The rest of the computing related to the parameter update consists mainly of matrix operations, which are straightforward and parallelizable.

This paper solves the remaining disadvantage of the parallel graphic equalizer by proposing a new efficient way for computing the target response, thus simplifying the coefficient update process. The proposed method is called the Linear Interpolation with Constant Segments (LICS), because it creates a piecewise linear magnitude response with flat regions. It is shown in this paper that the target response interpolation can be replaced with the LICS method without sacrificing much the precision of the parallel graphic equalizer.

The paper is organized as follows. Section 2 analyzes different methods for interpolating the target response based on its command gain values. Section 3 presents and evaluates the LICS method. This section assesses also the computational performance of this method when it is implemented on an ARM Cortex-A7 core. Finally, concluding remarks are given in Section 4.

*This work was conducted in spring 2015 when the first author was a visiting postdoctoral researcher at Aalto University. This research has been partly funded by the TIN2014-53495-R and TIN2011-23283 projects of the Ministerio de Economía y Competitividad and FEDER.

Table 1. Command Frequencies $f_{c,m}$ (Hz) of a Third-Octave Graphic Equalizer.

m	$f_{c,m}$	m	$f_{c,m}$	m	$f_{c,m}$
1	20	12	250	23	3150
2	25	13	315	24	4000
3	31.5	14	400	25	5000
4	40	15	500	26	6300
5	50	16	630	27	8000
6	63	17	800	28	10,000
7	80	18	1000	29	12,500
8	100	19	1250	30	16,000
9	125	20	1600	31	20,000
10	160	21	2000		
11	200	22	2500		

2. TARGET-RESPONSE INTERPOLATION

Graphic equalizer design requires interpolating a target frequency-response based on command gains G_m at center frequencies $f_{c,m}$ for $m = 1, 2, \dots, P$, where P is the number of command gains (sliders). Thus, the m th slider adjusts the contribution of frequency $f_{c,m}$ in the audio signal. A graphic equalizer commonly controls the gain at 31 standard frequencies spaced one third of an octave apart, as listed in Table 1.

The target response of the graphic equalizer is computed by using a suitable interpolation between the command gains G_m , which are manipulated by the user [13, 18]. Hermite and spline interpolation are two potential methods for obtaining a smooth target magnitude response [19]. Both methods fit the interpolating function to the data and its slope at the known points. In MATLAB, the cubic Hermite interpolation is computed by the `pchip` function. This function is preferred over `spline` in MATLAB, since it reduces the overshoot between command points, when the input data are non-smooth [12].

The computation of the `pchip` in real-time systems can be critical since its implementation needs estimation of derivatives around data points. In order to reduce the computational complexity of the target-response interpolation, three common interpolation algorithms that can be implemented with few operations are considered: zeroth-order interpolation, linear interpolation [13], and cubic Lagrange interpolation [19].

2.1. Previous Methods

The zeroth-order interpolation is achieved by rounding the interpolated points to the nearest command gain G_m , leading to a stair-case like target response. Linear interpolation is achieved by connecting adjacent gain values with a straight line [13]. The cubic Lagrange interpolation requires computing four weight factors and using these factors to interpolate between the two center-most gains [20].

The reference for the present work is [12], in which the target response is computed on a logarithmic frequency grid ($10P$ frequency points) on the decibel scale using the cubic Hermite interpolation. Fig. 1(a) compares target responses computed using four methods: zeroth-order, linear interpolation, cubic Lagrange interpolation, and cubic Hermite interpolation. The figure displays the command gains with circles at central frequencies of 315 Hz, 400 Hz, 500 Hz, 630 Hz, 800 Hz, and 1.00 kHz.

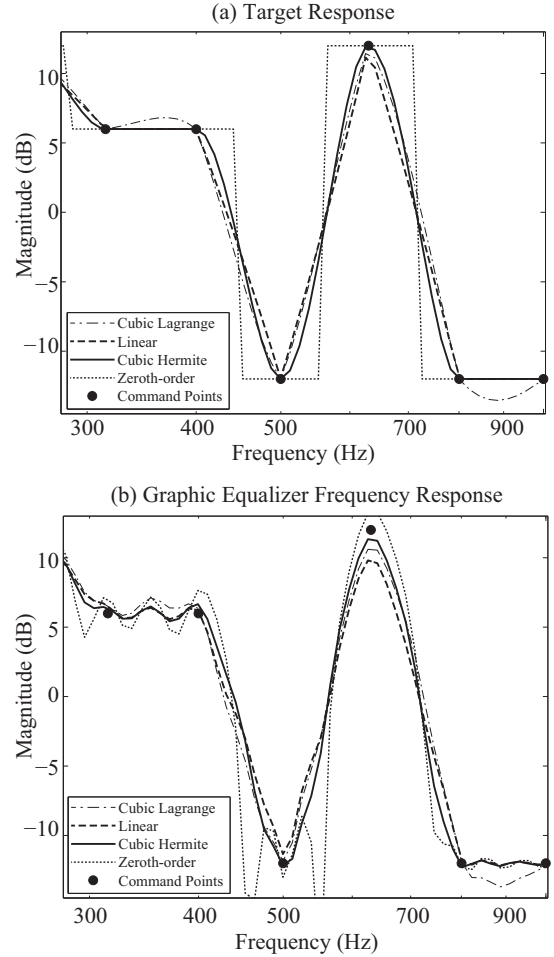


Fig. 1. (a) Target responses and (b) magnitude of the frequency responses obtained from command gains by using different interpolation methods: zeroth-order, linear, cubic Lagrange, and cubic Hermite interpolation.

As can be observed from Fig. 1(a), the zeroth-order interpolation (dotted line) presents sharp transitions and does not fit properly to the response obtained with the cubic Hermite interpolation (solid line). The cubic Lagrange interpolation (dash-dot line) behaves mostly better than the zeroth-order interpolation, but it causes excessive ripple when the neighboring command gains are the same, as seen in Fig. 1(a) around 900 Hz. The reason for this is that the cubic Lagrange interpolation does not account for derivatives of the data, so the interpolated curve becomes discontinuous at the data points [21]. Finally, the curve obtained with linear interpolation (dashed line) resembles the most the target response created using the cubic Hermite interpolation. These two responses never overshoot.

The best way to compare different target-response interpolation methods is to compute the equalizing filter coefficients using each method and evaluate the resulting magnitude responses. In this evaluation, we use the graphic equalizer design based on second-order parallel filters, as in [12]. This method makes use of a fixed-pole design with logarithm frequency resolution [22].

Fig. 1(b) shows the magnitude of the frequency responses for each of the presented interpolation methods. This figure shows

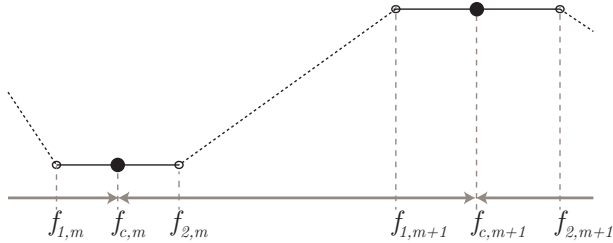


Fig. 2. In the LICS method, frequency points $f_{1,m}$ and $f_{2,m}$ limit the flat portion that surrounds the command gain value.

that both the zeroth-order and the cubic Lagrange interpolation can be discarded, because they lead to large ripples in the response. These conclusions were expected, as the corresponding target responses do not promise anything good. However, designs based on linear interpolation and cubic Hermite interpolation approximate the command gains well, although none of them achieves the exact values. In fact, focusing on the central frequency at 630 Hz, the approximation error with the cubic Hermite interpolation at the command gain is seen to be 0.52 dB, whereas with the linear interpolation it is 2.07 dB. In audio, restricting the maximum error not to be larger than 1 dB is desirable, so it can be concluded that the use of linear interpolation deteriorates the performance of the graphic equalizer too much.

Evidently, the largest errors in the design based on linear interpolation are produced when the target curve becomes too sharp near those command gains which differ much from the neighboring gain values, such as near the 630-Hz points in Fig. 1. In order to improve the linear interpolation, we propose next a new method.

3. LINEAR INTERPOLATION WITH CONSTANT SEGMENTS

We suggest introducing a narrow flat portion in the target response around each command gain and connecting these flat portions to each other with straight lines, as in linear interpolation. The width of the flat segment is made dependent on the frequency distance between command gains. The flat areas are composed of two points that surround the command gain value, as shown in Fig. 2.

The positions of these frequency points (denoted as $f_{1,m}$ and $f_{2,m}$) are related to the frequency distribution of the third-octave graphic equalizer in which the center frequencies are spaced one third of an octave apart:

$$f_{c,m} = 2^{1/3} f_{c,m-1}. \quad (1)$$

The two frequency limits $f_{1,m}$ and $f_{2,m}$ are chosen symmetrically (on the log frequency scale) on each side of a command frequency so that their geometric mean is equal to that command frequency:

$$f_{c,m} = \sqrt{f_{1,m} f_{2,m}}. \quad (2)$$

Thus, we can relate the three frequency points as

$$\begin{aligned} f_{1,m} &= 2^{-a/6} f_{c,m}, \\ f_{2,m} &= 2^{a/3} f_{1,m}. \end{aligned} \quad (3)$$

The question that arises from (3) is how to choose the value of variable a . In case a equals 1, the separation between frequency

Table 2. Optimal and Fixed Values of a and the Resulting Error for the Three Test Cases.

Test case	Variable a	Minimum max error
1	0.258	0.77 dB
1	0.20	0.90 dB
2	0.169	0.70 dB
2	0.20	0.88 dB
3	0.090	0.13 dB
3	0.20	0.54 dB

Table 3. Maximum Errors at Command Gain Points. Acceptable Errors Are Highlighted.

Test Case	Linear Interpolation	Cubic Hermite Interpolation	LICS Method
1	2.07 dB	0.63 dB	0.90 dB
2	2.40 dB	0.67 dB	0.88 dB
3	0.61 dB	0.31 dB	0.54 dB

points ($f_{2,m} - f_{1,m}$) corresponds to the bands of a common third-octave graphic equalizer. However, this separation implies that there would exist overlapping bands ($f_{2,m} > f_{1,r}$ for $m < r$) and this would prevent proper interpolation between the command points.

In order to find the optimal value for a , we vary a between 0 and 1.0 in steps of 0.001 and assess the maximum error of the equalizer responses at the command gains in three test cases: 1) A complicated command gain distribution taken from Fig. 2 of [12], a part of which is shown in Fig. 4; 2) A zig-zag formation in which the command gains G_m alternate between ± 12 dB; and 3) a configuration in which every third command is up (+12 dB) and the others are at zero. After selecting the value of a , we configure the limits of constant segments around all command points, and apply linear interpolation to compute the target response on a logarithmic frequency grid of $10P$ points as in [12].

As can be seen in Fig. 3, the value of a achieving the minimum max error is about 0.2 in each case. Since $a = 0.2$ leads to a maximum error of less than 1 dB in all cases, we use this value in the LICS method for all command gain distributions. Table 2 shows the optimal values of a and the resulting smallest errors together with the errors achieved using $a = 0.2$.

For comparison, we have computed the magnitude of the frequency responses using the cubic Hermite interpolation and the linear interpolation. Table 3 shows the maximum errors of the equalizer responses at the command gains for the three test cases. The cubic Hermite interpolation has the minimum error in all cases followed closely by our proposed method, which obtains an acceptable largest maximum error of 0.9 dB.

Fig. 4 shows the target response and the magnitude frequency response when the LICS method is applied to the command gain distribution shown in Fig. 1. The obtained target response does not overshoot, and the response is not as sharp near the distant command points 500 Hz and 630 Hz as with linear interpolation in Fig. 1(a). However, the target curve obtained by the LICS method is not as smooth as that obtained with cubic Hermite method, because the LICS method still produces discontinuities near data points.

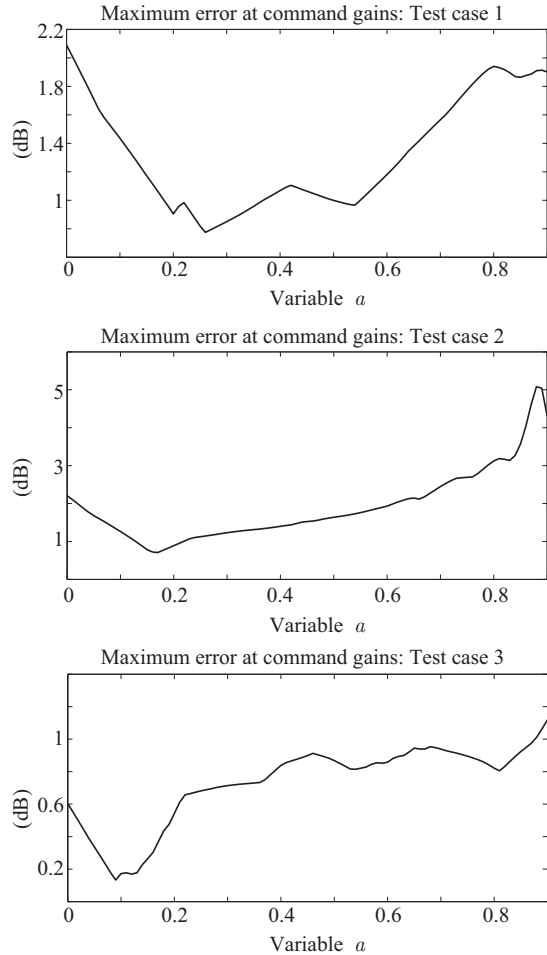


Fig. 3. Maximum error of the equalizer responses at the command points obtained by varying variable a in the three test cases.

The maximum ripple in the magnitude frequency response between 300Hz and 400Hz is 0.7dB for both the cubic Hermite [in Fig. 1(b)] and for the LICS method (in Fig. 4). Thus, the ripples can be seen to have been reduced between command points in comparison to linear interpolation, and the obtained magnitude frequency response connects the command gain values more accurately than most methods in Fig. 1(b).

3.1. Computational Performance

We tested the LICS method and the `pchip` method, which is available in a computational library [23], on a single ARM Cortex-A7 core, running at 250 MHz [24]. We have selected this architecture since it requires a low power [25] and is becoming widespread in several tablet computers and smart phones. The implementation was carried out in C language and the execution times were measured by using the routine `gettimeofday`. Many measurements were taken in order to reduce the effect of cached data. Table 4 indicates that the cubic Hermite interpolation requires approximately 2.3 times more time to compute the target response than the proposed LICS method.

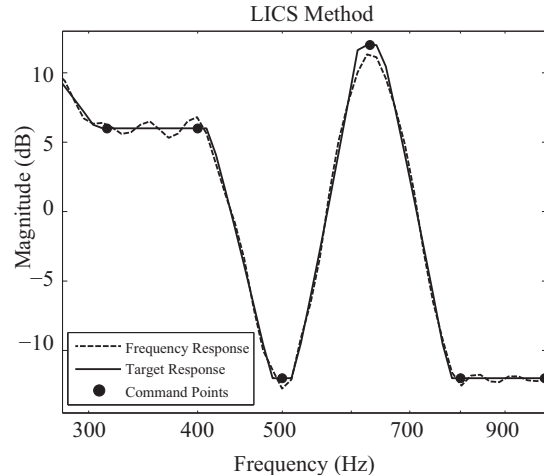


Fig. 4. Target response and magnitude of the frequency response obtained using the LICS method ($a = 0.2$), cf. Fig. 1(b).

Table 4. Execution Time for the Hermite Cubic Interpolation and the LICS Method.

Hermite Cubic <code>pchip</code>	LICS Method
0.108 ms	0.048 ms

4. CONCLUSION

This paper introduced a new approach for computing the target response of a graphic equalizer. It consists of assigning a constant segment around each command gain and connecting them with straight lines. We call this new method Linear Interpolation with Constant Segments or LICS. We devised a rule for choosing the parameters for this method, and found a nearly optimal value for the width of the constant segment. The results in this paper show that this method achieves a maximum error of less than 1 dB in the magnitude of the frequency response approximation for all tested distributions of the command gains. The required time to evaluate the target response using the LICS method speeds up the execution time by a factor of two in comparison to a `pchip` implementation. Besides, the LICS method is totally parallelizable, since the computation of the interpolated points in the bands is independent from one band to another. Thus, it can be efficiently executed in the multi-core architectures, which are used nowadays in tablets and mobile devices.

5. REFERENCES

- [1] M. Karjalainen, E. Piirilä, A. Järvinen, and J. Huopaniemi, “Comparison of loudspeaker equalization methods based on DSP techniques,” *J. Audio Eng. Soc.*, vol. 47, no. 1–2, pp. 15–31, Jan./Feb. 1999.
- [2] G. Ramos and J. J. López, “Filter design method for loudspeaker equalization based on IIR parametric filters,” *J. Audio Eng. Soc.*, vol. 54, no. 12, pp. 1162–1178, Dec. 2006.

- [3] J. Rämö and V. Välimäki, "Digital augmented reality headset," *J. Electr. Comput. Eng.*, vol. 2012, 2012.
- [4] S. Olive, T. Welti, and E. McMullin, "Listener preferences for in-room loudspeaker and headphone target responses," in *Proc. Audio Eng. Soc. 135th Conv.*, New York, USA, Oct. 2013.
- [5] E. Perez Gonzales and J. Reiss, "Automatic equalization of multi-channel audio using cross-adaptive methods," in *Proc. AES 127th Conv.*, New York, Oct. 2009.
- [6] J. Rämö and V. Välimäki, "Live sound equalization and attenuation with a headset.," in *Proc. AES 51st Int. Conf.*, Helsinki, Finland, Aug. 2013.
- [7] A. Mäkivirta, P. Antsalo, M. Karjalainen, and V. Välimäki, "Modal equalization of loudspeaker-room responses at low frequencies," *J. Audio Eng. Soc.*, vol. 51, no. 5, pp. 324–343, May 2003.
- [8] S. Cecchi, L. Romoli, A. Carini, and F. Piazza, "A multichannel and multiple position adaptive room response equalizer in warped domain: Real-time implementation and performance evaluation," *Applied Acoustics*, vol. 82, pp. 28–37, Aug. 2014.
- [9] M. Holters and U. Zölzer, "Graphic equalizer design using higher-order recursive filters," in *Proc. Int. Conf. Digital Audio Effects*, Montreal, Canada, Sept. 2006, pp. 37–40.
- [10] S. Tassart, "Graphical equalization using interpolated filter banks," *J. Audio Eng. Soc.*, vol. 61, no. 5, pp. 263–279, May 2013.
- [11] Z. Chen, G. S. Geng, F. L. Yin, and J. Hao, "A pre-distortion based design method for digital audio graphic equalizer," *Digital Signal Process.*, vol. 25, pp. 296–302, Feb. 2014.
- [12] J. Rämö, V. Välimäki, and B. Bank, "High-precision parallel graphic equalizer," *IEEE/ACM Trans. Audio Speech Language Processing*, vol. 22, no. 12, pp. 1894–1904, Dec. 2014.
- [13] P. H. Kraght, "A linear-phase digital equalizer with cubic-spline frequency response," *J. Audio Eng. Soc.*, vol. 40, no. 5, pp. 403–414, May 1992.
- [14] J. S. Abel and D. P. Berners, "Filter design using second-order peaking and shelving sections," in *Proc. Int. Computer Music Conf.*, Miami, FL, USA, Nov. 2004.
- [15] R. J. Oliver and J.-M. Jot, "Efficient multi-band digital audio graphic equalizer with accurate frequency response control," in *Proc. Audio Eng. Soc. 139th Conv.*, New York, USA, Oct. 2015.
- [16] J. Rämö and V. Välimäki, "Optimizing a high-order graphic equalizer for audio processing," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 301–305, Mar. 2014.
- [17] J. A. Belloch, B. Bank, L. Savioja, A. Gonzalez, and V. Välimäki, "Multi-channel IIR filtering of audio signals using a GPU," in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6692–6696.
- [18] M. Waters, M. Sandler, and A. C. Davies, "Low-order FIR filters for audio equalization," in *Proc. Audio Eng. Soc. 91st Conv.*, New York, USA, Oct 1991.
- [19] F. B. Hildebrand, *Introduction to Numerical Analysis*, Dover Publications, New York, second edition, 1987.
- [20] V. Välimäki, *Discrete-Time Modeling of Acoustics Tubes Using Fractional Delay Filters*, PhD thesis, Helsinki Univ. of Tech., Espoo, Finland, Dec. 1995.
- [21] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [22] B. Bank, "Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing," *J. Audio Eng. Soc.*, vol. 61, no. 1–2, pp. 39–49, Jan./Feb. 2013.
- [23] "Splines Library: pchip," http://people.sc.fsu.edu/~jburkardt/cpp_src/spline/spline.html, accessed Sept. 24, 2015.
- [24] "ARM architecture," <http://www.arm.com/>, accessed Feb. 23, 2015.
- [25] J. I. Aliaga, H. Anzt, M. Castillo, J. C. Fernández, G. León, J. Pérez, and E. S. Quintana-Ortí, "Unveiling the performance-energy trade-off in iterative linear system solvers for multi-threaded processors," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 4, pp. 885–904, 2015.