Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Performance Comparison of Neural Network Approaches for Electric Potential Field Generation in Surface Ion Traps

Sergey Denisov

sergeo.denisov@gmail.com

Lebedev Physical Institute

Ilya Gerasin

Lebedev Physical Institute

Ilya Semerikov

Lebedev Physical Institute

Research Article

Keywords: surface ion traps, electric potential fields, neural networks, Fourier Neural Operator, quantum computing

Posted Date: July 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9452458/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Performance Comparison of Neural Network Approaches for Electric Potential Field Generation in Surface Ion Traps

Sergey Denisov[1*], Ilya Gerasin[1] and Ilya Semerikov[1]

[1*]Lebedev Physical Institute, Moscow, Russia.

Contributing authors: igerasin@lebedev.ru;

Abstract

Quantum computing is now one of the most promising areas in modern computational science. Trapped ions are considered one of the best candidates for qubits because they have long coherence times and allow high-fidelity operations. Surface electrode ion traps are especially important because they support scalable quantum charge-coupled device architectures for advanced quantum computations. However, common simulation methods for trap design, such as the finite element method, are computationally expensive and slow down prototyping and optimization. In this paper, we compare three neural network architectures for fast electric potential field generation in surface ion traps: a Convolutional Neural Network, a Fourier Neural Operator, and Fully-Connected Layers. Our results show that the Fourier Neural Operator gives the best overall performance. It reaches an average effective potential field error of only 0.02% and is about 10 times faster than conventional methods. These results suggest that the Fourier Neural Operator is a strong candidate for trap design and ion dynamics simulations.

Keywords: surface ion traps, electric potential fields, neural networks, Fourier Neural Operator, quantum computing

1 Introduction

Quantum computing is one of the most promising directions in modern computational technology because it uses quantum mechanical effects to solve problems that are difficult for classical computers [1]. Among the many physical platforms, trapped ions

are widely seen as one of the strongest options. They offer long coherence times [2], high-fidelity single-qubit [3] and two-qubit gate operations [4], and a high level of uniformity. Surface electrode ion traps [5] have become an important scalable platform. They are made with microfabrication methods on substrates such as fused silica or silicon, usually with gold electrodes. Their feature sizes are typically from a few to several tens of micrometers, and ions are trapped at heights from tens to a few hundred micrometers above the surface. These microfabricated chips make it possible to build quantum charge-coupled device (QCCD) architectures [6], where ions can be moved between different processing zones in a way similar to classical CCD devices.

Accurate calculation of electric potential fields is important for many parts of ion trap operation. These include trap design and optimization [7–10], simulation of ion motion and confinement [11–13], micromotion compensation [14], and the study of trap-induced frequency shifts [15]. Fast field calculation also makes it easier to explore many design parameters and to correct experimental imperfections, such as stray electric fields, by changing voltages on compensation electrodes[16, 17].

Traditional design methods for surface ion traps depend on numerical simulations to calculate electric potential fields. The finite element method (FEM) divides the space into small elements and solves Laplace's equation $\nabla^2\phi = 0$ with boundary conditions. This method is accurate, but it needs a lot of computational resources. The finite difference method (FDM) places the differential operators on a grid, but it is less effective for complicated geometries. The boundary element method (BEM) reduces the problem to the boundaries, which lowers the dimensionality, but it leads to dense matrices. The solid angle method [18] is an efficient boundary-based approach for potential calculation in electrode structures. It works by calculating the solid angle of each electrode at every evaluation point. For a typical 2D evaluation grid of size $N \times N$, the number of evaluation points scales as $n = N^2$, while the number of boundary elements or electrodes scales as $m \sim N$. FEM requires solving a sparse linear system, and its complexity can range from $O(n)$ to $O(n^{1.5})$ depending on the solver [25, 26]. BEM produces a dense linear system of size $m \times m$, which gives a nominal complexity of $O(m^3)$ for direct solvers, although faster algorithms also exist [27]. In contrast, the solid angle method calculates the potential at each point as a direct sum over all electrodes. For a fixed and small number of electrodes, this gives a complexity of $O(nm)$. One of its main advantages is that it handles singularities at electrode edges analytically, without the need for special mesh refinement [18]. This makes it especially suitable for our application.

Recent progress in machine learning and neural operators has created new opportunities for speeding up scientific computations [23]. Neural networks have already been used successfully in many physical simulations, including fluid dynamics [20], electromagnetics [21], and quantum systems [22]. For example, Physics-Informed Neural Networks (PINNs) [19] include physical laws directly in the training process, while neural operators such as Fourier Neural Operators (FNOs) [23] learn mappings between function spaces. To the best of our knowledge, there has not yet been a systematic study of neural network methods for electric field approximation in ion traps.

In this work, we compare three neural network architectures for electric potential field generation in surface ion traps. These are Convolutional Neural Networks

(CNNs), which perform direct spatial field mapping using ideas from image-to-image translation; Fourier Neural Operators (FNOs), which use spectral transformations to represent global field behavior; and Fully-Connected Layers (FCL), which predict field values point by point from spatial coordinates.

We compare these approaches with classical numerical simulations and evaluate both their reconstruction accuracy and computational performance on CPU and GPU platforms, including memory usage.

2 Methodology

2.1 Numerical Simulation Baseline

We employ the solid-angle method (SAM) [18] for two main purposes: (a) to generate reference data used for training and validating the neural network approaches, and (b) to serve as a classical baseline for comparison. This method provides an efficient boundary-based technique for computing electrostatic potentials in electrode geometries. It is fast and allows not to cover with the mesh all domains. For a given electrode configuration, the potential ϕ at a point $\mathbf{r}$ can be expressed as

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_i V_i \Omega_i(\mathbf{r}) \tag{1}$$

where V_i is the voltage of electrode i, and $\Omega_i(\mathbf{r})$ is the solid angle subtended by electrode i at point $\mathbf{r}$. For n evaluation points and m electrodes, this method gives accurate potential values with computational complexity $O(nm)$ and handles singularities at electrode edges well.

2.2 FEM baseline

To benchmark in terms of speed our approaches against a standard out-of-the-box method, we compare our calculations with COMSOL FEM simulations. We employ the 2D Electrostatics module to compute the potential generated by a five-rail surface trap. The electrodes are modeled as 1 μm-thick structures placed within a vacuum box that spans the full width of the electrodes and has a height of 200 μm. A surface layer representing an Infinite Element Domain with a thickness of 50 μm is added. The region of interest is discretized with a 50×50 mesh, matching the resolution used in the neural network approach, while the surrounding domain is meshed using an extremely coarse grid. The stationary solver selected for this problem is MUMPS, using the default settings recommended by COMSOL. For performance evaluation, we perform a parametric sweep over the set of widths for the central DC and RF electrodes and compute the average time per simulation run. This scenario closely mimics the optimization process we follow when designing traps in our laboratory.

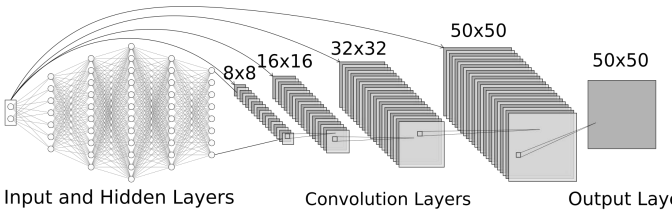


Fig. 1 CNN architecture diagram showing encoder-decoder structure with progressive upsampling from 8×8 to 50×50 resolution. The network takes 2 input parameters and outputs a complete 50×50 potential field. This image was made with the help of Alexander Lenail's tool NN-SVG.

2.3 Neural Network Architectures

2.3.1 Convolutional Neural Network (CNN)

The CNN architecture is inspired by image-to-image translation networks [24]. It uses a multi-resolution approach. The network takes two input parameters, namely the RF electrode widths and the separation distance, and produces a full 50×50 potential field (see Fig. 1). The process includes several steps: first, the two electrode parameters are used as input; next, fully-connected layers map them into a latent space; then convolutional layers with bilinear interpolation gradually upsample the representation from 8×8 to 50×50; finally, the network outputs the full 50×50 potential field. The model has about 15 million parameters. To keep geometric information through the network, the input parameters are added at different resolutions using skip connections.

2.3.2 Fourier Neural Operator (FNO)

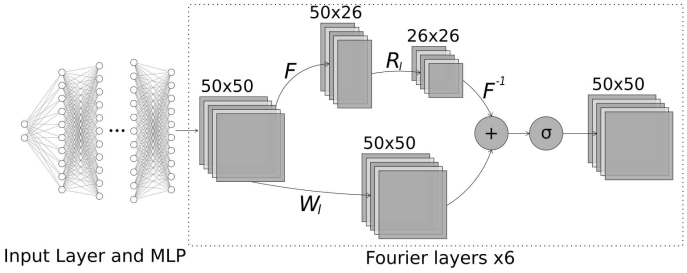


Fig. 2 FNO architecture diagram showing Fourier layers with spectral transformations. The network processes inputs through an encoder, 6 Fourier layers with frequency truncation, and a projection layer to generate 50×50 output fields. This image was made with the help of Alexander Lenail's tool NN-SVG.

The FNO architecture [23] uses spectral transformations to represent global field behavior. It is especially suitable for smooth physical fields such as electric potentials. A Multi-Layer Perceptron (MLP), which is a fully-connected neural network with several hidden layers, is used as an encoder to map the input parameters into a spatial representation (see Fig. 2). Then, six Fourier layers perform spectral convolutions with mode truncation to $(26, 26)$ modes. After that, a projection layer with 1×1

convolutions reduces the channel dimension from 64 to 1 and produces a 50×50 potential field. The full network has about 8 million parameters.

The Fourier layers apply the following operation:

$$z_{l+1} = \sigma\left(\mathcal{F}^{-1}\left(R_l \cdot \mathcal{F}(z_l)\right) + W_l z_l\right) \tag{2}$$

where z_l is the input to layer l, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the forward and inverse Fourier transforms, R_l is the complex spectral weight tensor acting on the truncated Fourier modes, W_l is a local linear transformation implemented by convolution, and σ is the activation function. This structure allows the model to learn global operators efficiently while still keeping local spatial information through the residual connection.

2.3.3 Fully-Connected Layers (FCL)

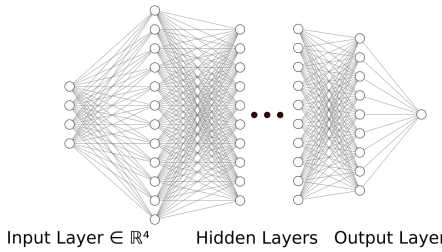


Fig. 3 FCL architecture diagram showing three specialized networks for potential, field magnitude (in log scale), and field sign prediction. The system takes both electrode parameters and spatial coordinates as inputs. This image was made with the help of Alexander Lenail's tool NN-SVG.

The FCL approach uses networks that take spatial coordinates as additional inputs, which allows continuous field representation. The system includes three specialized networks: a 12-layer network that predicts electric potential and has about 4 million parameters, a 13-layer network that predicts $\lg(|\mathbf{E}|)$ and has about 5 million parameters, and a 6-layer network that predicts the sign of the electric field and has about 0.5 million parameters (see Fig. 3).

The electric field is reconstructed as

$$\mathbf{E} = \text{sign} \times 10^{\lg(|\mathbf{E}|)} \tag{3}$$

We predict $\lg(|\mathbf{E}|)$ instead of the direct field value because electric fields in ion traps can vary over several orders of magnitude. This makes direct regression difficult for neural networks. The logarithmic transformation reduces this large dynamic range and makes the learning problem easier.

2.4 Training Methodology

All models were trained using Mean Squared Error (MSE) loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4}$$

where y_i is the ground truth value taken from output of SAM and $\hat{y}_i$ is the prediction made by the neural network.

We also tested a masking strategy and compared masked and unmasked models. Points with pseudopotential magnitude above the threshold $\epsilon = 0.3\ eV$ were excluded from the loss calculation. This threshold corresponds to regions where there is no equilibrium and where useful trap parameters cannot be extracted. As a result, the training focused on the physically relevant region for ion confinement and dynamics, within a radius of about 10 μm around the equilibrium point. This improved the accuracy of parameter extraction.

We also explored other approaches, including Physics-Informed Neural Networks (PINNs) with Laplacian constraints ($\nabla^2\phi = 0$) and relative error metrics, but these performed worse than the good old fashion MSE with masking. There are several reasons for this. First, because our simulations use a finite grid resolution, the discrete Laplacian does not become exactly zero. Instead, it approaches a small constant value that depends on grid spacing, which introduces numerical error into the physics constraint. Second, adding physics-based loss terms such as $\nabla^2\phi = 0$ makes the optimization problem more difficult, because the network must fit the data and satisfy the physical constraint at the same time. In practice, this multi-objective optimization was less stable and required careful balancing of the loss weights.

2.5 Trap Parameter Extraction

We extracted several relevant ion trap parameters from the generated fields. These include the equilibrium height (h), defined as the position of minimum effective potential; the secular frequency (freq), obtained from the second derivatives of the potential at the equilibrium point; the trap depth, defined as the maximum confining potential energy; and the stability parameter (q), taken as the Mathieu parameter derived from the field curvature and related to the ratio of the secular frequency to the RF frequency.

These parameters are essential for predicting ion behavior and overall trap performance in quantum operations.

3 Experimental Results

3.1 Field Reconstruction Accuracy

Table 1 summarizes field reconstruction performance. Success rate indicates the percentage of generated fields that produced physically meaningful results. Pushing the limits of these neural networks, they would sometimes yeild generated fields of poor

quality and/or fields that where completely wrong. These fields where not considered for the accuracy comparison results and later the trap parameter error results. The criteria of a "meaningful" result was taken as: the error of all the extracted parameters did not exceed 100%.

Table 1 Field Reconstruction Accuracy Comparison

Model	Potential Error (%)	Field Error (%)	Success Rate (%)
FNO (masked)	0.02	0.15	100
FNO (unmasked)	0.03	0.22	100
CNN (masked)	0.49	3.81	100
CNN (unmasked)	1.30	9.45	90
FCL	0.40	2.00	100

The FNO gives the best overall performance, with an average error of only 0.02% in effective potential field reconstruction. For CNN and FNO, field values were obtained by numerically differentiating the predicted potential fields, while FCL directly predicts the field.

3.2 Trap Parameter Extraction

Table 2 shows the accuracy of the extracted trap parameters, which are critical for real trap operation.

Table 2 Trap Parameter Extraction Error (%)

Model	h	q	freq	depth
FNO (masked)	0.1	1.0	4.1	2.8
FNO (unmasked)	0.5	3.1	10.0	3.1
CNN (masked)	0.9	4.5	23.8	15.6
CNN (unmasked)	1.4	10.2	39.9	33.1
FCL (from E-field)	4.0	11.0	9.0	10.0
FCL (from ϕ)	0.6	1.9	3.3	4.7

The masked FNO achieves very strong accuracy in parameter extraction. The FCL approach gives better frequency accuracy when the parameter is extracted from the potential rather than directly from the predicted electric field.

3.3 Computational Performance

Table 3 compares the inference times of the SAM, COMSOL (we use default settings for multithreading) and the neural models on both CPU (AMD Ryzen 9 7950X, 16 cores) and GPU platforms (NVIDIA GeForce RTX 4060). The reported times are averages over 1000 inference runs.

All neural network approaches provide clear speed improvements on GPU hardware. For FCL, the reported time is per point. Generating a full 50×50 field requires

Table 3 Computational Performance Comparison

Method	GPU Time	CPU Time
SAM	-	7.8 ms
COMSOL	-	980 ms
FNO	0.53 ms	5.7 ms
CNN	2.4 ms	17 ms
FCL (full field)	62 ms	230 ms
FCL (per point)	25 μs	91 μs

2500 separate evaluations. Although FCL is slow for full-field generation, it has the advantage of continuous evaluation at arbitrary coordinates without interpolation.

COMSOL exhibits extremely slow performance. The first reason is that meshing introduces an overhead. One needs to use boundary conditions such as the Infinite Element Domain, which are necessary for correct field calculations. In our calculations for a 50×50 area of interest, COMSOL requires approximately 4,000 elements. The second reason is that the meshing process is included in the evaluation time. Unfortunately, this workflow cannot be avoided—this is the price when using out-of-the-box tools.

4 Discussion

The strong performance of the FNO comes from its ability to capture global field relationships through spectral transformations. Electric potential fields in ion traps are naturally smooth and satisfy Laplace's equation, which makes them well suited for spectral representation. The Fourier layers capture these smooth global variations effectively, while local convolutional parts help handle edge effects near the electrodes.

The CNN also performs reasonably well, but it has more difficulty with long-range dependencies. This becomes especially visible in parameters related to dynamics, such as oscillation frequency (23.8% error) and trap depth (15.6% error), because these depend on accurate estimation of field curvature. The progressive upsampling process can introduce artifacts that reduce accuracy in these cases.

The FCL approach shows a different error pattern, even though its number of parameters is similar to the other models. Point-wise evaluation naturally gives continuous output, but because there is no explicit spatial consistency constraint, the predicted fields can be noisier. At the same time, direct field prediction is useful for ion dynamics simulations because it avoids numerical differentiation when calculating forces.

The masking strategy is very important for both training stability and final accuracy. By focusing learning on regions with strong field gradients, where confinement forces act on the ions, masking improves the extraction of physically relevant parameters. For the FNO, masking reduces the depth error from 3.1% to 2.8% and the frequency error from 10.0% to 4.1%. The CNN benefits even more, and unmasked training produces physically unrealistic fields in 10% of cases.

For fast ion trap design and optimization, the FNO offers the best balance between accuracy and speed. It provides about a 10 times speedup over the solid angle method on CPU while keeping the error below 5% for all major parameters. This makes it possible to explore design spaces much faster than with traditional numerical methods. It is also worth noting that many standard differential equation solvers and simulation tools usually run on CPU, while neural network acceleration is especially effective on GPUs.

For ion dynamics simulations, the best choice depends on the task. The FNO is the fastest for generating complete fields, while the FCL allows continuous force evaluation at arbitrary points without interpolation artifacts. The CNN has the fastest inference time, but its lower derivative accuracy makes it less suitable for simulations where precise field curvature is important.

5 Conclusion

This study presents a comparison of neural network methods for electric potential field generation in surface ion traps. Our results show that neural networks can serve as a practical alternative to traditional numerical simulations, and that the Fourier Neural Operator is the most effective architecture among the models we tested.

The FNO reaches very high accuracy, with only 0.02% field error, while also providing major computational speed advantages. This makes it a strong option for fast trap design and optimization. We also show that masking is an important part of the training process, because it focuses the model on physically relevant regions and improves the accuracy of extracted trap parameters.

In future work, it would be useful to extend this approach to 3D field generation for more complex trap geometries, include time-dependent fields for RF simulations, develop hybrid models that combine the strengths of different architectures, and apply these methods to multi-ion crystals and more complicated trapping potentials.

The demonstrated speed and accuracy of neural field generation opens new possibilities for real-time trap optimization, automated design processes, and comprehensive parameter space exploration in quantum computing applications.

Declarations

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

References

[1] Nielsen, Michael A and Chuang, Isaac L (2010) *Quantum computation and quantum information* (Cambridge university press)

[2] Häffner, Hartmut and Roos, Christian F and Blatt, Rainer (2008) Quantum computing with trapped ions *Physics Reports* **469** 155–203

[3] Gaebler, JP and Tan, TR and Lin, Y and Wan, Y and Bowler, R and Keith, AC and Glancy, S and Coakley, K and Knill, E and Leibfried, D and others (2016) High-fidelity universal gate set for $^{9}Be^{+}$ ion qubits *Physical review letters* **117** 060505

[4] Ballance, CJ and Harty, TP and Linke, NM and Sepiol, MA and Lucas, DM (2016) High-fidelity quantum logic gates using trapped-ion hyperfine qubits *Physical review letters* **117** 060504

[5] Seidelin, S and Chiaverini, J and Reichle, R and Bollinger, JJ and Leibfried, D and Britton, J and Wesenberg, JH and Blakestad, RB and Epstein, RJ and Hume, DB and others (2006) Microfabricated surface-electrode ion trap for scalable quantum information processing *Physical review letters* **96** 253003

[6] Kielpinski, David and Monroe, Christopher and Wineland, David J (2002) Architecture for a large-scale ion-trap quantum computer *Nature* **417** 709–711

[7] , Hong, Seokjun and Lee, Minjae and Cheon, Hongjin and Kim, Taehyun and Cho, Dong-il "Dan" (2016) Guidelines for designing surface ion traps using the boundary element method *Sensors* **16** 616

[8] Gerasin, I and Zhadnov, N and Kudeyarov, K and Khabarova, K and Kolachevsky, N and Semerikov I (2024) Optimized Surface Ion Trap Design for Tight Confinement and Separation of Ion Chains *Quantum Reports* **6** 442-451

[9] Abbasov, T and Zibrov, S and Sherstov, I (2023) Surface-Electrode Ion Trap Development *JETP Letters* **118** 3

[10] Zhang, Chi and Mehta, Karan K and Home, Jonathan P, (2022) Optimization and implementation of a surface-electrode ion trap junction *New Journal of Physics* **24** 073030

[11] Leibfried, D and Blatt, R and Monroe, C and Wineland, D (2003) Quantum dynamics of single trapped ions *Reviews of Modern Physics* **75** 281

[12] Moehring, David Lynn and Highstrete, Clark and Stick, D and Fortier, Kevin Michael and Haltli, R and Tigges, C and Blain, Matthew Glenn (2011) Design, fabrication and experimental demonstration of junction surface ion traps *New Journal of Physics* **17** 075018

[13] Singer, Kilian and Poschinger, Ulrich and Murphy, Michael and Ivanov, Peter and Ziesel, Frank and Calarco, Tommaso and Schmidt-Kaler, Ferdinand (2010) Colloquium: Trapped ions as quantum bits: Essential numerical tools *Reviews of Modern Physics* **82** 2609–2632

[14] Berkeland, DJ and Miller, JD and Bergquist, JC and Itano, WM and Wineland, DJ (1998) Minimization of ion micromotion in a Paul trap *Journal of applied*

physics **83** 5025–5033

[15] Kim, Tony Hyun and Herskind, Peter F. and Chuang, Isaac L. (2011) Surface-electrode ion trap with integrated light source *Applied Physics Letters* **98** 214103

[16] Shu, G and Vittorini, G and Buikema, A and Nichols, CS and Volin, C and Stick, D and Brown, Kenneth R (2014) Heating rates and ion-motion control in a Y-junction surface-electrode trap *Physical Review A* **89** 062308

[17] Charles Doret, S and Amini, Jason M and Wright, Kenneth and Volin, Curtis and Killian, Tyler and Ozakin, Arkadas and Denison, Douglas and Hayden, Harley and Pai, CS and Slusher, Richart E and others (2012) Controlling trapping potentials and stray electric fields in a microfabricated ion trap through design and compensation *New Journal of Physics* **14** 073012

[18] Wesenberg, J. H. (2008) Electrostatics of surface-electrode ion traps *Physical Review A* **78** 063410

[19] Raissi, Maziar and Perdikaris, Paris and Karniadakis, George E (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *Journal of Computational Physics* **378** 686–707

[20] Kochkov, Dmitrii and Smith, Jamie A and Alieva, Ayya and Wang, Qing and Brenner, Michael P and Hoyer, Stephan (2021) Machine learning accelerated computational fluid dynamics *Proceedings of the National Academy of Sciences* **118** e2101784118

[21] Khan, Arbaaz and Ghorbanian, Vahid and Lowther, David (2019) Deep learning for magnetic field estimation *IEEE Transactions on Magnetics* **55** 1–4

[22] Carleo, Giuseppe and Troyer, Matthias (2017) Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–606

[23] Li, Zongyi and Kovachki, Nikola and Azizzadenesheli, Kamyar and Liu, Burigede and Bhattacharya, Kaushik and Stuart, Andrew and Anandkumar, Anima (2020) Fourier neural operator for parametric partial differential equations *arXiv preprint arXiv:2010.08895*

[24] Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A (2017) Image-to-image translation with conditional adversarial networks *Proceedings of the IEEE conference on computer vision and pattern recognition* 1125–1134

[25] George, Alan (1973) Nested dissection of a regular finite element mesh *SIAM Journal on Numerical Analysis* **10** 345–363

[26] Briggs, William L and Henson, Van Emden and McCormick, Steve F (2000) A multigrid tutorial (SIAM)

[27] Greengard, Leslie and Rokhlin, Vladimir (1997) A new version of the Fast Multipole Method for the Laplace equation in three dimensions *Acta Numerica* **6** 229–269