


RESEARCH ARTICLE

10.1029/2024JH000161

Key Points:

- A machine learning algorithm based on Gaussian processes can be used to accurately predict (emulate) elastic deformation model output
- We emulate compressibility and surface displacements for pressure and volume changes in vertical spheroidal elastic cavities
- The emulators closely reproduce finite element model output but are orders of magnitude faster to run

Correspondence to:

K. R. Anderson,
kranderson@usgs.gov
Citation:

Anderson, K. R., & Gu, M. (2024). Computationally efficient emulation of spheroidal elastic deformation sources using machine learning models: A Gaussian-process-based approach. *Journal of Geophysical Research: Machine Learning and Computation*, 1, e2024JH000161. <https://doi.org/10.1029/2024JH000161>

Received 23 FEB 2024
Accepted 22 JUN 2024

Published 2024. This article is a U.S. Government work and is in the public domain in the USA. *Journal of Geophysical Research: Machine Learning and Computation* published by Wiley Periodicals LLC on behalf of American Geophysical Union.

This is an open access article under the terms of the [Creative Commons Attribution License](#), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Computationally Efficient Emulation of Spheroidal Elastic Deformation Sources Using Machine Learning Models: A Gaussian-Process-Based Approach

Kyle R. Anderson¹  and Mengyang Gu²
¹U.S. Geological Survey Volcano Science Center, Moffett Field, CA, USA, ²University of California, Santa Barbara, CA, USA

Abstract Elastic continuum mechanical models are widely used to compute deformations due to pressure changes in buried cavities, such as magma reservoirs. In general, analytical models are fast but can be inaccurate as they do not correctly satisfy boundary conditions for many geometries, while numerical models are slow and may require specialized expertise and software. To overcome these limitations, we trained supervised machine learning emulators (model surrogates) based on parallel partial Gaussian processes which predict the output of a finite element numerical model with high fidelity but >1,000× greater computational efficiency. The emulators are based on generalized nondimensional forms of governing equations for finite non-dipping spheroidal cavities in elastic halfspaces. Either cavity volume change or uniform pressure change boundary conditions can be specified, and the models predict both surface displacements and cavity (pore) compressibility. Because of their computational efficiency, using the emulators as numerical model surrogates can greatly accelerate data inversion algorithms such as those employing Bayesian Markov chain Monte Carlo sampling. The emulators also permit a comprehensive evaluation of how displacements and cavity compressibility vary with geometry and material properties, revealing the limitations of analytical models. Our open-source emulator code can be utilized without finite element software, is suitable for a wide range of cavity geometries and depths, includes an estimate of uncertainties associated with emulation, and can be used to train new emulators for different source geometries.

Plain Language Summary Mathematical models are widely used to calculate how an elastic material deforms due to pressure changes in a buried spheroidal cavity. These models have wide application to fields such as volcanology, where they can be used to understand the way the surface of the Earth deforms due to changes in buried magma reservoirs. Some models use relatively straightforward mathematical relations; these can generally make predictions quickly but can become highly inaccurate for some cavity geometries. Other models require sophisticated numerical techniques implemented in computers, but these are generally slow to run and may require specialized expertise or software. Here we use machine learning techniques to train a surrogate model (emulator) based on thousands of numerical computer simulations that can overcome these limitations. The emulator makes predictions that are both fast and accurate and can be applied to a wide range of problems.

1. Introduction

A uniform pressure change applied to the walls of a cavity causes deformation of the surrounding medium, which may be computed using a variety of continuum mechanical models. Models of spheroidal cavities in elastic media, in particular, are commonly used throughout the sciences as first-order representations of more complex structures. In volcanology, such models have been used for more than 60 years to relate displacements measured at the Earth's surface to volume and pressure changes in magma reservoirs (e.g., Mogi, 1958; Segall, 2010; Yang et al., 1988) and to relate reservoir and magma volume changes to one another through the compressibility of the reservoir (e.g., Blake, 1981; Johnson, 1992). Although these models greatly simplify Earth structure, more complex models are often not warranted in practice due in part to observational limitations.

No analytical model can accurately predict deformation induced by pressure changes in a finite spheroidal cavity with arbitrary depth and geometry in an elastic halfspace. Exact solutions for halfspace surface displacements exist only for spheres and horizontal “penny-shaped” circular cracks (equivalent to a spheroid as the vertical dimension goes to zero) and require numerical implementation (Fialko et al., 2001; Zhong et al., 2019). Other

solutions for surface displacements are based on approximations which violate the uniform pressure boundary condition as the ratio of cavity dimension to depth – which we refer to as shallowness – increases (Davis, 1986; McTigue, 1987; Yang et al., 1988). Approximations for cavity volume change or cavity compressibility (also called pore compressibility) are similarly limited (Amoruso & Crescentini, 2009; Cervelli, 2013; Zimmerman, 1985).

Numerical techniques can be used to solve the equations governing linear elasticity to a high degree of accuracy but generally suffer from much higher computational cost, and often a reliance on complex and/or expensive computer code. Approaches have been developed for improving the efficiency of numerical approaches (Charco & Galán Del Sastre, 2014; Masterlark et al., 2012; Ronchin et al., 2017; Trasatti et al., 2008), but these are generally suitable only for limited geometries, lack generality, or result in only modest increases in efficiency. This strongly limits the use of numerical models in many kinds of inverse problems such as Bayesian Markov chain Monte Carlo procedures that seek, through thousands to millions of model simulations, to fully characterize the uncertainties associated with model parameter estimation.

Machine learning approaches are only beginning to be applied to these problems in geophysics and Earth sciences, but show promise for overcoming some of the aforementioned challenges. Emulators (also called surrogate models, metamodels, or approximation models) are efficient model proxies (Gramacy, 2020; Santner et al., 2003). Emulators use machine learning techniques to predict the outputs of numerical models at a lower computational cost in exchange for some reduction in model fidelity (Bayarri et al., 2007; Kovachki et al., 2023). The goal is to obtain a small reduction in fidelity and a large computational acceleration. In previous crustal deformation studies, researchers have accelerated inverse problems by emulating data-model misfit functions (Zhang, 2016) and used artificial neural networks to emulate the output of a numerical solution of the relevant governing equations (DeVries et al., 2017) or to directly ‘learn’ the relevant governing equations themselves (Okazaki et al., 2022). While these approaches have many advantages, they are generally either inflexible – requiring retraining for each new data set, as when emulating a residual (data-model misfit) function – require large amounts of training data, cannot be used independently of the original computer model, or are computationally expensive.

Emulators developed in a probabilistic statistical framework based on Gaussian processes (GPs) may overcome these limitations for certain classes of problems (Sacks et al., 1989). GP emulators (Kriging emulators) can be efficiently trained using only a small number of simulation runs, which is often not the case for artificial neural networks. Furthermore, unlike neural networks or polynomial chaos expansion (PCE) techniques – which approximate simulator outputs by a series of multivariate polynomials (Ghanem & Spanos, 2003; Xiu & Karniadakis, 2003) – the probabilistic model of a GP readily yields prediction uncertainty assessments. Comparisons between GPs, neural networks, and PCEs can be found in Neal (2012), Lee et al. (2018), and Owen et al. (2017).

Previously, Anderson et al. (2019) demonstrated that GP emulators could be used to efficiently and accurately predict surface displacements due to pressure changes in an elastic spheroidal cavity using limited training data, but the model was specialized for data inversion in the context of a particular volcanological problem and lacked generality. Here we develop general and efficient GP emulators based on numerical model simulations which yield scalar cavity compressibility and vector surface displacements for a wide range of spheroid geometry and material properties. These emulators are both fast and accurate, can be applied to a wide range of problem without further training, and require only limited software at runtime.

2. Methods

2.1. Physics of the Problem

We restrict our focus to vertical, axially-symmetric spheroidal cavities embedded in uniform linear elastic half-spaces (Figure 1), and consider displacements only on the free surface. Cavity geometries range from thin horizontal cracks to vertical pipes. We define the spheroid's vertical semi-axis (axis of rotation) length as a , horizontal semi-axis length as b , and centroid depth as d (Figure 1a; Table 1). The elastic medium is parameterized entirely by its shear modulus μ and Poisson's ratio ν .

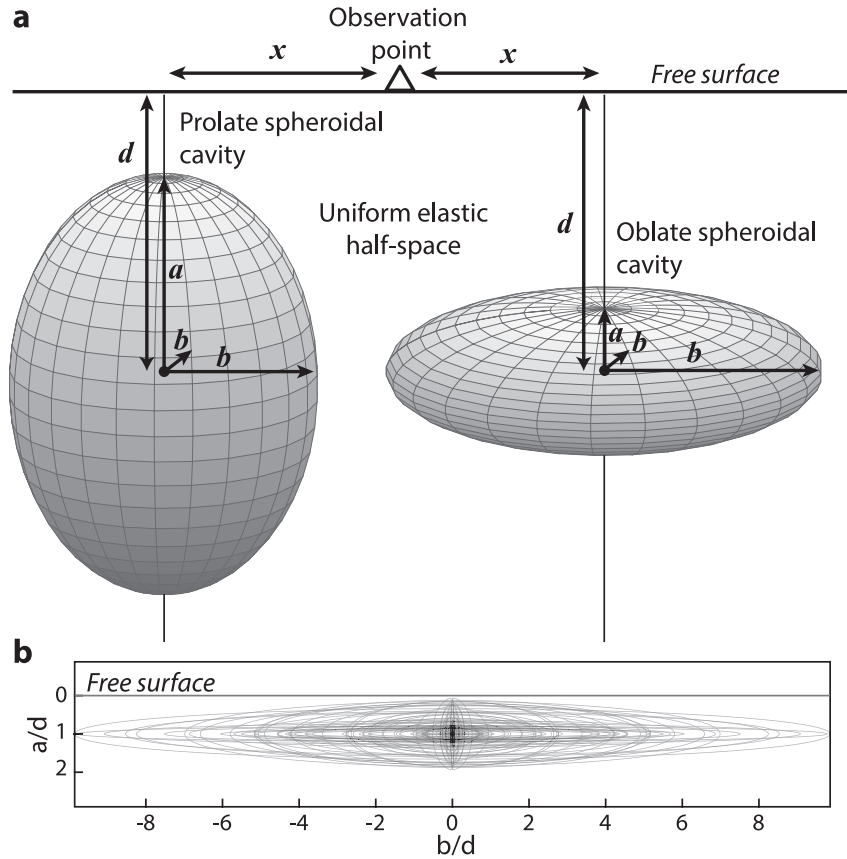


Figure 1. (a) Spheroidal cavity geometry used in this study. The vertical (polar) semi-axis radius is a and the horizontal (equatorial) semi-axis radius is b . (b) 100 typical training geometries for $d = 1$.

A uniform pressure change Δp applied to the cavity walls produces a cavity volume change ΔV which is proportional to the product of the cavity volume V and the applied pressure change. The constant of proportionality is defined as the cavity (pore) compressibility β :

$$\beta \equiv \frac{1}{V} \frac{\Delta V}{\Delta p}. \quad (1)$$

In a halfspace, ΔV is an unknown function of source geometry (including depth) and material properties. However, dimensional analysis and the physics of the problem suggest that β can be written as

$$\beta = \frac{1}{\mu} f_c \left(\frac{a}{b}, \frac{a}{d}, \nu \right), \quad (2)$$

where f_c is an unknown, nonlinear, scalar-output, nondimensional function, and a/b and a/d are nondimensional parameter groups which fully define, respectively, the shape of the cavity (its aspect ratio) and its relative depth (“shallowness”). Other groupings of dimensional parameters a , b , and d are also possible (e.g., a/d and b/d for a more consistent nondimensionalization) but we found in practice that these generally improve emulator performance. Note that since β is an intrinsic property of the cavity it is independent of applied pressure change. Use of nondimensional forms here and elsewhere reduces the number of free parameters and improves emulator performance.

Displacements u on the surface of the halfspace are a function of source geometry, material properties, and radial observation (distance) vector $\mathbf{x}$. Dimensional analysis and the physics of the problem suggest that surface

Table 1
Common Symbols Used Throughout

Symbol	Meaning
Geometry, material properties, and physics	
a, b	Cavity dimensions (semimajor axis, semiminor axis)
d	Cavity centroid depth
V	Cavity volume
Δp	Uniform pressure change applied to cavity walls
ΔV	Cavity volume change
μ	Elastic medium shear modulus
ν	Elastic medium Poisson's ratio
β	Cavity compressibility
u_r, u_z	Surface displacements (horizontal [radial], vertical)
Functional form	
$y, \hat{y}$	Model outputs (numerical, emulated)
g	Transformed model output
$f, \hat{f}$	Model functions (numerical, emulated)
$\mathbf{m}, \mathbf{m}^*$	Model parameter vectors (training set, others)
$\mathbf{x}, \mathbf{x}^*$	Observation distance vector on surface (training set, others)
Ω, Ω_x	Parameter space; observation coordinate space
n, n^*	Number of parameter sets (training points, test points)
k	Number of observation points in $\mathbf{x}$
w	Number of model parameters in $\mathbf{m}$

displacements can be written as the product of linear source strength terms S and unknown, nonlinear, nondimensional, vector-output functions f_j (which control the shape of the displacement field) as

$$u_j^{(i)}\left(\frac{\mathbf{x}}{d}\right) = S^{(i)} f_j^{(i)}\left(\frac{a}{b}, \frac{a}{d}, \nu; \frac{\mathbf{x}}{d}\right), \quad (3)$$

where the semicolon is used to separate model parameters from observation coordinates, subscript $j = r, z$ indicates radial and vertical displacements, respectively, and superscript $i = p, v$ indicates pressure and volume change forms, respectively, as.

$$S^{(v)} = \frac{\Delta V}{d^2}, \quad (4)$$

$$S^{(p)} = \frac{V \Delta p}{\mu d^2}. \quad (5)$$

The functional form of Equations 2 and 3 are verified in Appendix B.

We note several features of these expressions. Firstly, outputs are trivially generalizable to obtain three-dimensional surface displacements at any observation point for a cavity at any arbitrary position. Secondly, in the limit that $d \gg (a, b)$ then f_j can be parameterized fully by $(a/b, \nu; \mathbf{x}/d)$ and simple analytical solutions are available (e.g., Segall, 2010; Segall, 2013). Finally, functions $f_j^{(p)}$ and $f_j^{(v)}$ differ only by a scaling factor, equal to the cavity compressibility (note that $S^{(v)}/S^{(p)} = \beta$), which is unknown and varies across parameter space. In the context of an inverse problem, both S terms involve the same number of free parameters since $\Delta p/\mu$ can be treated as a single parameter and V is a dependent function of a and b . However, differences in f_j can affect emulator performance, and parameterizing by cavity volume change or pressure change can be useful in different situations. In this work we therefore trained separate emulators for functions $f_j^{(p)}$ and $f_j^{(v)}$, as well as f_c .

Surface stresses, strains, and tilts are not considered in this work, but we note that the latter may be easily computed by numerically differentiating emulated vertical displacements (Anderson et al., 2019). Future code updates may include these outputs.

2.2. Numerical (Finite Element) Solution

To train the GP emulator we computed f_c and $f_j^{(i)}(\mathbf{x}/d)$ using the finite element method (FEM). The FEM is a numerical technique for solving partial differential equations through discretization of the spatial domain and has been applied to problems in the Earth sciences for decades (e.g., Dieterich & Decker, 1975). FEM models can solve the equations governing linear elasticity to a nearly arbitrary accuracy, including complex geometries and spatially-variable material properties, but are relatively slow to run and rely on specialized software. (Boundary element methods do not discretize the spatial domain and can be faster than FEMs, but are less flexible and still slow enough to limit their use in inverse problems.)

In this study we utilized the commercial FEM software package COMSOL Multiphysics. We implemented a 2D axisymmetric model of a vertical spheroid embedded in a uniform elastic halfspace with shear modulus μ and Poisson's ratio ν (Figure A1). We assumed infinitesimal strains (linear elasticity). The FEM is parameterized using cavity geometry and material properties as follows:

$$[u_j(\mathbf{x}), \Delta V] = FEM(a, b, d, \Delta p, \nu, \mu; \mathbf{x}). \quad (6)$$

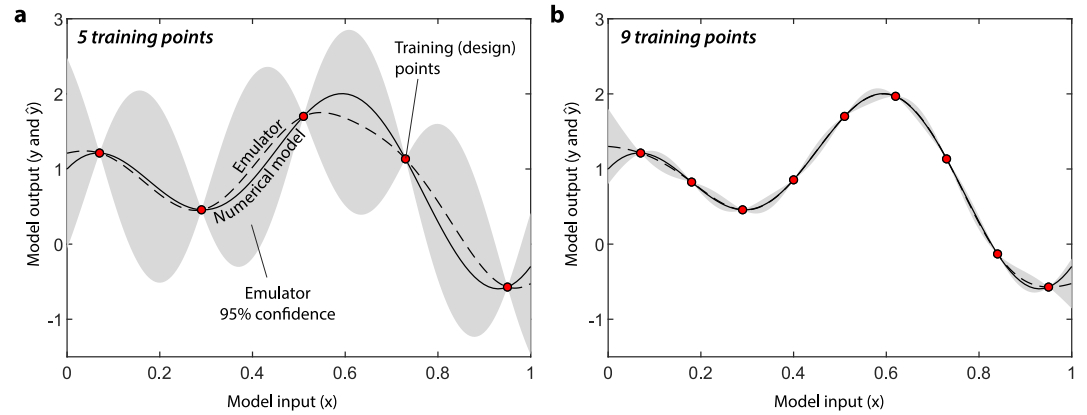


Figure 2. Simple demonstration of the Gaussian Process emulator in Gu (2019) for an example function with scalar output. (a) A computationally expensive numerical model is run at just five input points (red dots) and the outputs used to train a GP emulator. The emulator (dashed line) only roughly approximates the true numerical function (solid black line), but recognizes this uncertainty (gray shaded region, showing 95% credible [confidence] intervals). (b) Adding just four training points greatly increases emulator fidelity and reduces uncertainty.

For all FEM models we fixed $d = 1$, which allowed us to obtain a and b from nondimensional groups a/b and a/d . Primary domain height and width were set to $100 \times \max\{a, b, d\}$, to which we added infinite element domains (which stretch coordinate axes to approximate an infinitely large model domain) extending to $120 \times \max\{a, b, d\}$. Shear modulus was fixed to an arbitrary value μ^* . For boundary conditions we applied a uniform test pressure of Δp^* to the walls of the cavity, and enforced a stress-free halfspace surface and zero displacements on the outer domain boundaries. We experimented to find mesh settings that were suitable for the wide range of cavity geometries employed in this study. The elastic domain was discretized with an unstructured triangular mesh with reduced element size on the free surface and the cavity boundary; rectangular elements were used in the infinite domains. We used quadratic elements throughout. The geometry was remeshed for each set of parameters. We verified FEM output against other models (Appendix A).

For each set of parameters, the FEM directly yielded displacements $u_j(\mathbf{x})$ on the surface of the elastic halfspace. We computed the cavity volume change ΔV by integrating boundary-normal cavity wall displacements using $\int_s (u \cdot \hat{\mathbf{n}}) ds$ where $\hat{\mathbf{n}}$ is the surface normal vector, and computed β using Equation 1. Finally we computed the value of functions f_c and $f_j^{(i)}(\mathbf{x}/d)$ for each set of model parameters using Equations 2 and 3 with μ^* , Δp^* , and ΔV . Note that parameter sets are independent so these FEM calculations are trivially parallelizable.

2.3. Gaussian Process Emulation

Building an emulator typically begins by generating a limited set of training parameters (also called ‘design’ parameters), at which a computer model f is exercised. Denote $\mathbf{y} = f(\mathbf{m}, \mathbf{x})$ where $\mathbf{y} = (y_1, y_2, \dots, y_k) \in \mathbb{R}^k$ is a vector output of function (computer model) f at spatial or temporal coordinates $\mathbf{x} = \{x_1, x_2, \dots, x_k\} \in \Omega_x$ with input model parameter vector $\mathbf{m} = \{m_1, m_2, \dots, m_w\} \in \Omega$ (note that $\mathbf{x}$ is omitted for functions with scalar input and output.) Given training data, emulators represent f by a new, more computationally-efficient mapping between inputs and outputs $\hat{\mathbf{y}} = \hat{f}(\mathbf{m}, \mathbf{x})$ and can then be used to predict output at other, unexplored, inputs $\mathbf{m}^* \in \Omega$ and $\mathbf{x}^* \in \Omega_x$ (Figure 2). Then, $\mathbf{y} = \hat{\mathbf{y}} + \epsilon$, where ϵ is the error introduced by the emulator.

Our goal in this work is to emulate the nonlinear scalar-output function f_c (Equation 2) and nonlinear vector-output functions $f_j^{(i)}(\mathbf{x}/d)$ (Equation 3) using output from a limited number of FEM simulations. Given emulated functions $\hat{f}_c$ and $\hat{f}_j^{(i)}$, compressibility $\hat{\beta}$ and surface displacements $\hat{u}$ may be trivially computed by scaling using linear terms in Equations 2 and 3, respectively.

In this study we used the GP emulator, which treats computer model outputs as a Gaussian process (Rasmussen & Williams, 2006). In this approach a kernel (correlation) function models the relation between inputs and outputs as a function of the distances between the inputs; correlations are generally strongest for more similar inputs. GPs

are parameterized by unknown mean, variance, and correlation parameters, which are estimated using training data. Given the kernel function, new outputs may be predicted using the predictive mean of the GP. An important advantage of GP emulators is that an assessment of the uncertainty associated with any prediction can also be obtained using a closed-form expression for the predictive variance.

GP emulators have been widely used to emulate scalar-output computer models (Bayarri et al., 2007, 2009; Conti & O'Hagan, 2010; Gu & Berger, 2016; Higdon et al., 2008; Sacks et al., 1989; Spiller et al., 2014). To emulate scalar-output function f_c , we use the GP approach with robust estimation of the kernel parameters (Gu et al., 2018). Emulating vector-output function $f_j^{(i)}$ is more challenging. While some approaches are suitable for emulating one-dimensional vectors with a small number of elements k (Conti & O'Hagan, 2010; Farah et al., 2014; Overstall & Woods, 2016), emulating large k is both theoretically and computationally challenging. To overcome these challenges we applied the parallel partial Gaussian process (PP-GP) emulator (Gu & Berger, 2016). The PP-GP utilizes independent emulators for each point in the observation vector $\mathbf{x}$; these emulators have independent means and variances, but share a correlation function between physical inputs. This approach substantially reduces computational expense while producing high-fidelity prediction, and is particularly suitable for models with high-dimensional and spatially heterogeneous vector output. Thus, the PP-GP approach makes possible high-fidelity predictions of computer model output at a large number of spatial coordinates and also provides a closed-form expression for uncertainty assessment. Here we use the MATLAB ROBUSTGASP package (Gu, 2019) to implement both the robust GP emulator for scalar-valued output and the PP-GP emulator for vector output.

We trained several emulators (Table 2), including some with restricted geometries or material properties in order to improve speed and fidelity at the cost of generality. The performance of the emulators depends on model parameterization, the dimensions of parameter space Ω , the number of training observations, observation coordinates, and the kernel function of the emulator. In this study, several nondimensional parameterizations of Equations 2 and 3 are possible, but we found that $\mathbf{m} = [a/d, \nu]$ offers good performance for spherical cavities, and $\mathbf{m} = [\log_{10}(ab), a/d, b/d, \nu]$ offers good performance for spheroidal cavities. (Although b/d is dependent on ab and a/d so does not mathematically add any new information, we found that including it can improve performance.)

Unless otherwise fixed, we used $\nu \in [0.05, 0.45]$ spanning most common materials, $ab \in [1/50, 50]$ for highly prolate to highly oblate geometries, and $a/d \in [0.005, 0.95]$ for shallow and deep cavities. We also placed limits on dependent parameter b/d in some cases in order to exclude extremely shallow and oblate geometries which are uncommon for most problems.

For each emulator we chose a set of n space-filling training points (parameter sets) and another set of n^* points for testing it, both in Ω . Generally, emulator accuracy can be improved by increasing n , but computational expense increases cubically with n unless approximations are made (e.g., Snelson & Ghahramani, 2005). In our application, n is of order 10^2 to 10^3 (Table 2). We separately generated parameter sets for each emulator using Sobol sequences (Sobol', 1967), which yield sampling points that efficiently fill parameter space. In some cases, to improve emulator performance in the oblate regime we increased training point density proportional to the ellipse flattening parameter $1 - b/a$. Example geometries are shown in Figure 1b.

We ran the FEM for all training and testing points and output to disk. For simplicity we trained all displacement emulators using the same observation coordinates $\mathbf{x}/d \in [0.01, 30]$ with $k = 1,000$ points. We used an observation vector in which point spacing is high close to $x = 0$ and increases toward the maximum $\mathbf{r}$. Note that GP emulators are not suitable for $f_r(\mathbf{x}/d = 0)$ since the output is always exactly zero; we therefore used a minimum $\mathbf{x}/d$ of 0.01 and extrapolate to $\mathbf{x}/d = 0$ as needed during emulation.

Performing transformations g on numerical model output y before emulation can improve performance (Bayarri et al., 2009). After experimentation, we emulated $g(y) = \log(y + C)$ where for f_c we use $C = 0$ and for f_j we use $C = y - \min(y) + A_j \text{mean}[y - \min(y)]$ where A_j is a scaling factor, set here at $A_r = 1$ and $A_z = 0.1$. These transformations are well-defined for $y = 0$ and improve emulator performance by "spreading out" small values of y and de-emphasizing large values (Bayarri et al., 2009). The emulator's predictive median and quantiles (i.e., emulator predictions) are obtained by inverse transformation $y = \exp(g) - C$. We also experimented with subtracting the predictions of approximate analytical models from the FEM output before emulation, but decided that the improvement did not justify the additional complexity.

Table 2
Emulator Performance

Geometry	Set	n	Speed (Sims/s)	Func	NRMSE (%)	mARE (%)	maxARE (%)	RLCI(95%) (%)	PCI(95%) (%)
Nondimensional cavity compressibility									
C1 Sphere	s_1	500	2,800	$\hat{f}_c$	1.4×10^{-4}	6.8×10^{-5}	1.2×10^{-3}	1.5×10^{-3}	100
C2 Spheroid (ν fixed)	s_2	909	2,300	$\hat{f}_c$	4.3×10^{-4}	8.3×10^{-3}	0.29	6.5×10^{-2}	97.0
C3 Spheroid	s_3	989	1,900	$\hat{f}_c$	2.1×10^{-3}	3.1×10^{-2}	0.64	0.19	95.2
Nondimensional surface displacements									
D1 Sphere	s_1	500	1,400	$\hat{f}_r^{(v)}$	4.8×10^{-5}	1.3×10^{-5}	7.0×10^{-3}	1.1×10^{-4}	97.0
				$\hat{f}_z^{(v)}$	2.5×10^{-5}	4.9×10^{-6}	1.5×10^{-3}	5.4×10^{-5}	84.7
				$\hat{f}_r^{(p)}$	5.7×10^{-5}	1.1×10^{-5}	7.2×10^{-3}	5.6×10^{-5}	75.1
				$\hat{f}_z^{(p)}$	2.9×10^{-5}	5.2×10^{-6}	1.9×10^{-3}	4.6×10^{-5}	63.7
D2 Spheroid (ν fixed)	s_2	690	1,300	$\hat{f}_r^{(v)}$	1.0×10^{-2}	2.4×10^{-3}	0.46	3.7×10^{-2}	96.1
				$\hat{f}_z^{(v)}$	1.0×10^{-2}	2.3×10^{-3}	1.2	1.6×10^{-2}	94.7
				$\hat{f}_r^{(p)}$	1.1×10^{-2}	2.7×10^{-3}	0.54	1.6×10^{-2}	93.6
				$\hat{f}_z^{(p)}$	2.7×10^{-2}	3.9×10^{-3}	1.0	2.0×10^{-2}	92.8
D3 Spheroid	s_3	956	1,000	$\hat{f}_r^{(v)}$	1.1×10^{-2}	4.4×10^{-3}	0.63	9.0×10^{-2}	97.0
				$\hat{f}_z^{(v)}$	1.7×10^{-2}	3.5×10^{-3}	1.7	2.6×10^{-2}	94.4
				$\hat{f}_r^{(p)}$	3.6×10^{-2}	4.7×10^{-3}	0.60	3.5×10^{-2}	98.2
				$\hat{f}_z^{(p)}$	1.1×10^{-1}	4.3×10^{-3}	0.92	3.1×10^{-2}	95.2

Note. For each emulator we show the parameter set with approximate bounds (see below), number of training points n , emulator speed (for sequential non-vectorized calculations without uncertainties), emulated function, and statistical tests using held-out numerical test points at observation coordinates used in training. Statistical tests are reported in % and computed for emulators $\hat{f}$ with any log transformations removed. For all $\hat{f}_j$, computational cost includes interpolation for $k = 1,000$ random observation points. Tests were performed using MATLAB R2023b on a desktop workstation computer with an Intel Xeon W-2265 processor. The emulators may be exercised over broader parameter ranges than shown here but with higher uncertainties. Each emulator set prefixed with “D” includes four different emulators. Spheroidal cavity emulators C3 and D3 are the most general but emulators C1–C2 and D1–D2 offer modestly improved performance for spherical cavities and/or a fixed Poisson's ratio. Parameter sets are as follows.

s_1 : $\mathbf{m} = [a/d, \nu]$ $alb = 1$; $a/d \in [0.01, 0.9]$; $b/d \in [0.005, 7]$; $\nu \in [0.05, 0.45]$

s_2 : $\mathbf{m} = [\log_{10}(a/b), a/d, b/d]$ $alb \in [1/40, 40]$; $a/d \in [0.01, 0.9]$; $b/d \in [0.002, 7]$; $\nu = 0.25$

s_3 : $\mathbf{m} = [\log_{10}(a/b), a/d, b/d, \nu]$ $alb \in [1/40, 40]$; $a/d \in [0.01, 0.9]$; $b/d \in [0.002, 7]$; $\nu \in [0.05, 0.45]$

We used the ROBUSTGASP package to train the GP emulators (Gu, 2019). For all emulators we used a product Matérn covariance function with roughness parameter 2.5, which has been empirically found to be accurate for a variety of applications (Gu et al., 2018) and is the default covariance function of the ROBUSTGASP package. Range parameters of the kernel were estimated by the marginal posterior mode with the jointly robust prior (Gu, 2018). We found that in most cases including a nugget parameter (Andrianakis & Challenor, 2012) in the covariance function increased training time and complexity without meaningfully improving performance, so we did not include one. The range parameters were optimized using the limited memory BFGS algorithm (Liu & Nocedal, 1989) handled within the ROBUSTGASP code. Numerically optimizing PP-GP covariance parameters (training the emulator) requires $\mathcal{O}(n^3) + \mathcal{O}(kn^2)$ operations per iteration. Training the emulators in this study required tens of seconds to minutes on a modern desktop computer to yield high predictive accuracy. After training, however, using the emulators to make predictions only requires a fraction of a second. The computational cost of exercising the emulators is discussed in Section 3.3.

The trained emulators can only directly predict output at the observation points used in the training data, so we compute predictions at other coordinates $\mathbf{x}^* \notin \mathbf{x}$ using linear interpolation. Using a large number of observation points k reduces interpolation error, and computational cost increases only linearly with k (Section 3.3). Also, because we found that emulator performance is usually degraded near the edges of parameter space, we generally

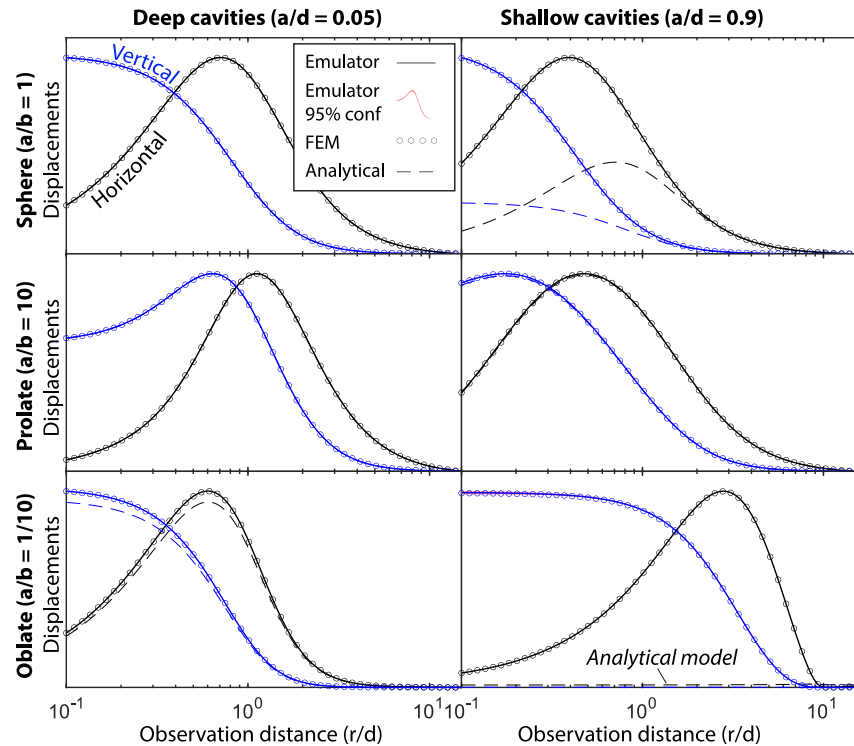


Figure 3. Surface displacements for several cavity geometries computed using emulator D3, a finite element model (FEM), and the Yang-Cervelli analytical model. Cavity depth = 1 km, shear modulus = 20 GPa, Poisson's ratio = 0.25, and the pressure change boundary condition = 10 MPa. 95% emulator confidence bounds are shaded in pink, but are too small to be visible for most geometries. Horizontal and vertical displacements are separately normalized for clarity. Emulator output closely matches FEM output in all cases, confirming its accuracy, but the analytical model deviates substantially for shallow spherical and oblate cavities, reflecting its approximations (analytical models are not visible when they agree closely with emulator predictions). Analytical model deviations are different for volume change boundary conditions, which are not shown here.

exercise the emulators over a reduced range $\Omega_r \in \Omega$; for the spheroid this is generally $a/b \in [1/40, 40]$, $a/d \in [0.01, 0.9]$, and $b/d \leq 7$. However, the emulator is capable of making predictions over the full range of training data.

Finally, we emphasize that the predictive accuracy of the emulator depends on numerous choices made during training. We found that the default settings in the ROBUSTGASP package, including the covariance function and roughness parameter, worked well. However, other choices – including how the model itself is parameterized, which parameters are used for training, the distribution of training points, and the transforms applied to the model output before emulation – also play an important role. General principles for good performance include non-dimensionalizing model physics before emulation in order to reduce the number of free parameters and transforming model output to smooth the response surface. We also found that trial-and-error experimentation was invaluable, in some cases improving emulator performance by an order of magnitude. These general concepts should apply also to the development of other emulators in the future (Section 4.4).

2.4. Benchmarking

We first qualitatively compared emulated and FEM-computed surface displacements for six cavity geometries, finding in all cases very close agreement (Figure 3).

For a more quantitative and comprehensive evaluation we compared FEM and emulator output using thousands of semi-random parameter sets which were not used to train the emulator (selected using Sobol sequences).

We evaluated emulator performance using several metrics. To quantify misfit between the output of a model (typically the emulator) and the output of a reference model we used the normalized root mean square error (NRMSE; Appendix C), where we normalized using the mean range of the reference output across observation

coordinates for $\hat{f}_j$ and the mean of all reference output for $\hat{f}_c$. For a more interpretable metric we also defined an absolute relative error

$$\text{ARE}(i,j) = \frac{|\hat{y}(m_i^*;x_j) - y(m_i^*;x_j)|}{\text{range}[y(m_i^*;x)]}, \quad (7)$$

where i and j indicate the model parameter set number and observation point number, respectively. The ARE is normalized by the range of the reference model y along the entire output vector $\mathbf{x}$ (this avoids issues with zero and near-zero output values). For scalar-output functions there is no observation coordinate vector and the range is replaced by the value of the reference function y . We generally computed the mean ARE (mARE) and maximum ARE (maxARE) across i, j , or both (Appendix C). For both NRMSE and ARE, lower values are better.

To evaluate the emulators' formal prediction uncertainties we computed the fraction of reference model outputs falling within the emulators' 95% posterior credible intervals (PCI(95%)) and the average relative length of the emulators' 95% posterior credible intervals (RLCI(95%)) (e.g., Gu & Berger, 2016):

$$\text{PCI}(95\%) = \frac{1}{kn^*} \sum_{j=1}^k \sum_{i=1}^{n^*} 1\{y(m_i^*;x_j) \in \text{CI}_{ij}(95\%)\}, \quad (8)$$

$$\text{RLCI}(95\%) = \frac{1}{kn^*} \sum_{j=1}^k \sum_{i=1}^{n^*} \frac{|\text{length}\{\text{CI}_{ij}(95\%)\}|}{\text{range}[\hat{y}(m_i^*;x)]}, \quad (9)$$

where CI is the formal posterior credible interval of the emulator at the given level (here 95%). PCI measures the accuracy of the emulator's predicted uncertainties; RLCI measures emulator confidence, normalized by the range of the emulator output along the observation coordinates. PCI(95%) should be close to 95% and RLCI(95%) should be as low as possible.

Results are shown in Table 2 and Figure 4. For all scalar-output $\hat{f}_c$ (emulators C1–C3), we found NRMSE and mARE of order $10^{-5}\%$ to $10^{-2}\%$ with no errors exceeding 0.6%, and small formal predictive uncertainties (RLCI(%95) of order $10^{-3}\%$ to $10^{-1}\%$). For vector-output $\hat{f}_j(\mathbf{x}/d)$ (emulators D1–D3) we found NRMSE of order $10^{-5}\%$ to 0.1% and mARE of order $<10^{-3}\%$ or better, with maxARE ranging widely from $<10^{-3}\%$ –1.2%. Predictive uncertainties are small (RLCI(%95) of order $10^{-4}\%$ to $10^{-2}\%$), but formal uncertainties of the emulator D1 (pressure boundary condition formulation) are too optimistic. Fixing Poisson's ratio (models C2 and D2) modestly improves performance. Further limiting the cavity geometry to exclude extremely oblate, prolate, and shallow sources also further improves performance, but we consider this unnecessary for most applications.

We also benchmarked predictions against two exact analytical models. Firstly, we benchmarked the spherical cavity emulators C1 and D1 against the exact (Zhong et al., 2019) series-expansion model calculated to 64th order for a range of a/d , integrating cavity-normal displacements predicted by that model to obtain compressibility as needed. For both we found close agreement, with maxARE of order $<10^{-3}\%$ to $<10^{-1}\%$ (Figure 5 and B1). We note that our compressibilities differ from those reported in Battaglia et al. (2013), we believe due to a higher mesh density employed in our FEM model.

Finally, we benchmarked emulated displacements against a variety of numerical models as a part of a community deformation modeling exercise (Crozier et al., 2023). The accuracy of the tested emulators was comparable to that of the best community finite element and boundary element models (and exceeded some of them when compared against the Zhong et al. (2019) model), but the speed was orders of magnitude faster than the numerical models and even faster than the analytical models.

3. Results

In this section we utilize the computational efficiency of the emulators to characterize in detail how cavity compressibility and surface displacements vary with cavity geometry, compare emulator output with predictions of approximate analytical models, and evaluate the computational expense of the emulators.

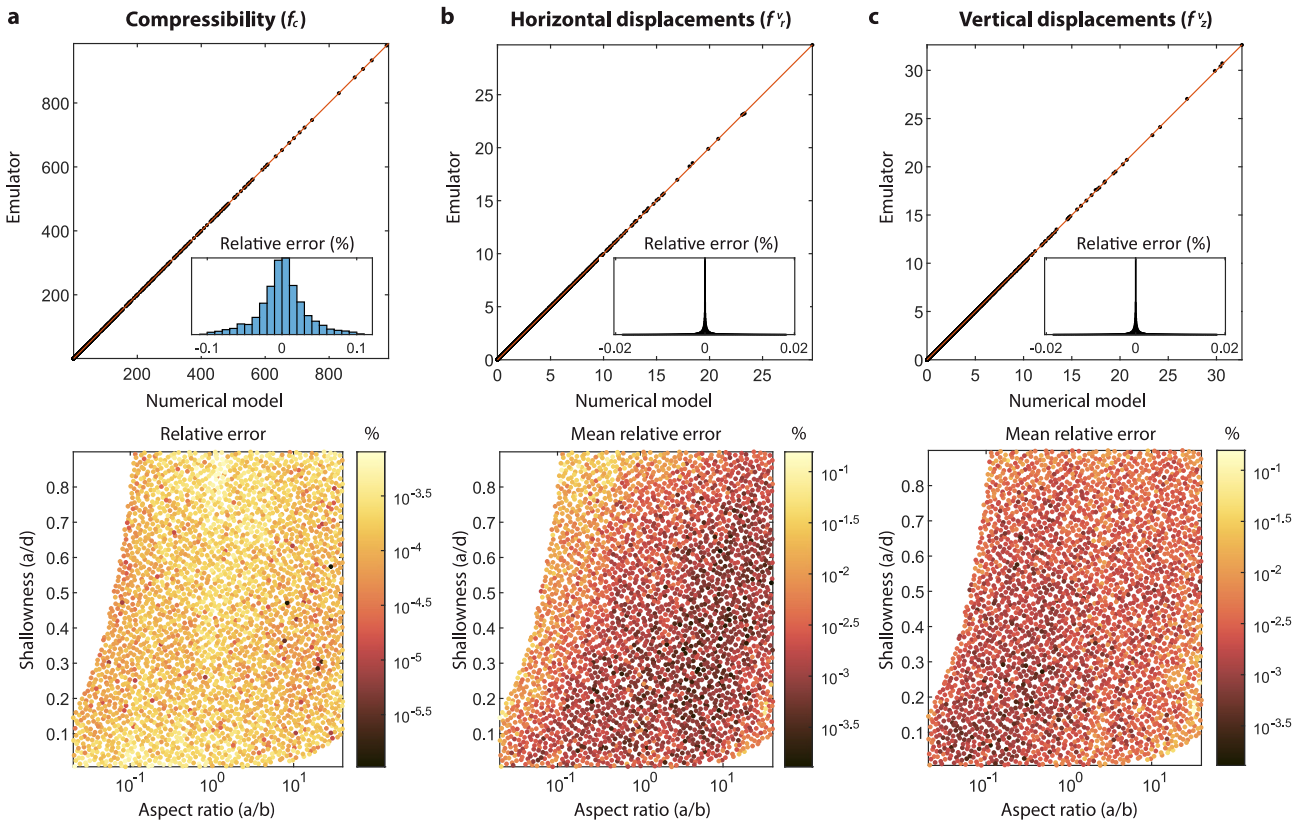


Figure 4. Comparison of spheroidal cavity emulator predictions (emulators C3 and D3) with finite element numerical model output at test points for (a) f_c , (b) f_r^v , and (c) f_z^v . The top row shows FEM output versus emulator output, with histograms showing mean relative errors for all points (as defined in equation (C3) but without the absolute value), with 5% outliers excluded. For (b) and (c) a random 10^5 points were selected across all models and observation coordinates. The bottom row shows mean absolute relative error (mARE; Appendix (c)) along the observation vector as a function of cavity aspect ratio and shallowness. Note different color scales between panels (a) and (b)–(c). Model output is a nonlinear function of input parameters, so emulator performance varies across the parameter space. Errors for other emulators are summarized in Table 2.

Exact solutions for cavity compressibility and half-space surface displacements due to pressure changes in spheroidal elastic cavities are generally unavailable, but a number of approximate solutions have been developed. We compare our results chiefly with the analytical approximation of Cervelli (2013), which extends the Yang et al. (1988) dipping prolate spheroidal source model to oblate geometries. We refer to this model as the Yang-Cervelli model. (The more general Nikkhoo and Rivalta (2023) model agrees closely with the Yang-Cervelli model for appropriate geometries but is more computationally expensive.). This model is known to violate the uniform pressure cavity boundary condition for cavities relatively close to the free surface. For compressibility, we also utilize several approximate series-expansion solutions for end-member geometries derived using full-space relations (Zimmerman, 1985) (see also David and Zimmerman (2011)), which are not well known in the geophysical literature.

3.1. Cavity Compressibility

Cavity compressibility varies by orders of magnitude as a strong function of aspect ratio and depth. For the geometries we study, compressibility can be more than four orders of magnitude larger than predicted by analytical approximations (Figure 6).

In the oblate regime, the Yang-Cervelli model badly under-predicts compressibility except for very deep cavities ($a/d = 0.01$). For strongly oblate cavities, we find that compressibility grows proportionally to b/a , in agreement with fullspace relations (Zimmerman, 1985), but with a constant of proportionality that increases for shallower cavities.

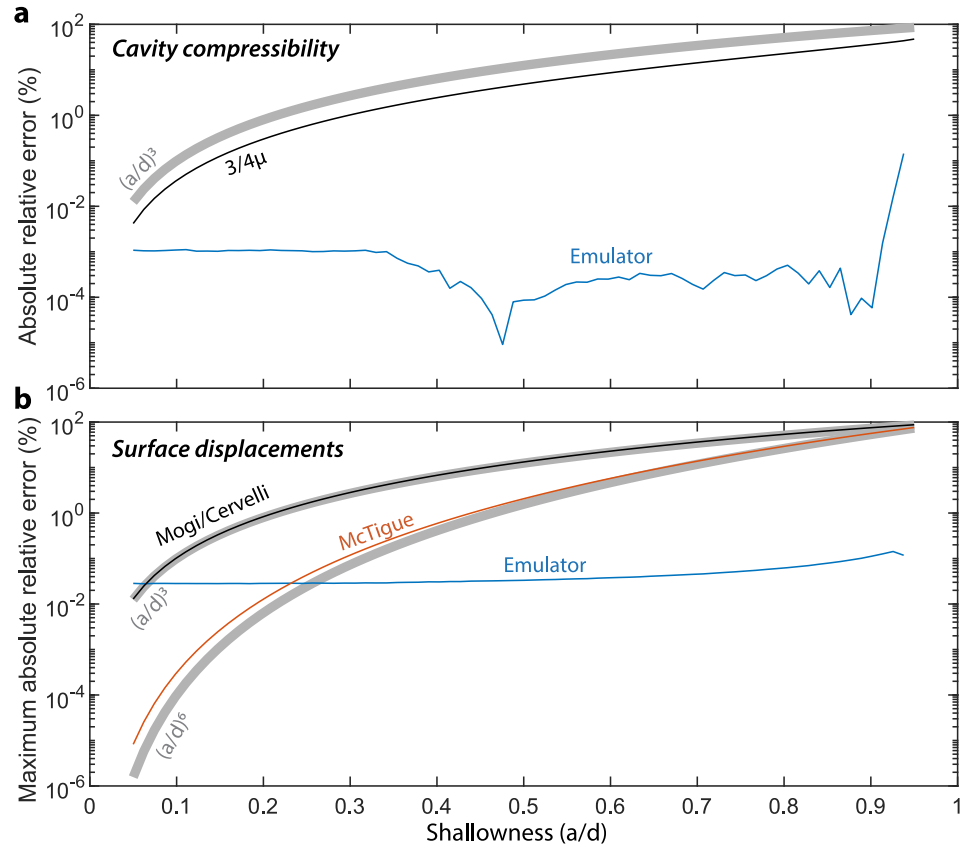


Figure 5. Comparison of emulator predictions with several spherical cavity models for $\nu = 0.25$ with the Zhong et al. (2019) model as the reference solution. (a) Cavity compressibility. Emulator errors (model C1) are very low and do not grow with a/d , whereas errors in the commonly-used $3/4\mu$ relation grow with $(a/d)^3$. (b) Surface displacements, computed using a pressure boundary condition over coordinates $\mathbf{x} \in [0:30d]$. Relative errors are maximums across both radial and horizontal components. Emulator errors (model D2) are very low and do not meaningfully increase as $a/d \rightarrow 1$, unlike analytical model errors which grow approximately with $(a/d)^6$ (McTigue, 1987) and $(a/d)^3$ (Cervelli, 2013; Mogi, 1958) (cf., Crozier et al., 2023). Only for very small a/d do analytical displacement models slightly outperform the emulator (note logarithmic scale).

In the prolate regime, in contrast, compressibility is relatively constant and approaches $1/\mu$ as $a/b \rightarrow \infty$ (Figure 6b), in agreement with fullspace approximations (Zimmerman, 1985). At $a/b = 25$ the emulator predicts compressibility to within 1% of $1/\mu$ with little dependence on a/d or ν . Across the full prolate regime, analytical and emulator results agree reasonably well except for relatively shallow sources for $a/b < \sim 3$.

For a deep sphere ($a/d = 0.1$) our results agree to better than 0.1% with the widely-used approximation $\beta = 3/4\mu$ (McTigue, 1987). However, compressibility increases by $\sim 50\%$ as $a/d \rightarrow 0.9$, and also becomes weakly dependent on ν (Figure 6b), both in contrast to the analytical approximation. We also observe that due to the interaction with the free surface, modestly prolate geometries at shallow depth become more compressible than spherical cavities; thus, the spherical geometry is not always the least compressible.

Numerical results can be used to derive simple empirical relations for certain geometries, which we attempt here only for spherical cavities. We seek compressibility as a function of integer powers of a/d weighted by terms of ν , and with experimentation obtain good fits using

$$\beta_{sph} \approx \frac{3}{4\mu} \left[1 + p_1 \left(\frac{a}{d} \right)^3 + p_2 \left(\frac{a}{d} \right)^{13} \right] \quad (10)$$

with $p_1 = 0.6(1 - \nu)$ and $p_2 = 1.2 - \nu$ for $\nu \in [0.1, 0.4]$ (Figure 6b). Better fits may be obtained with additional a/d terms, but this simple relation should be adequate for many purposes.

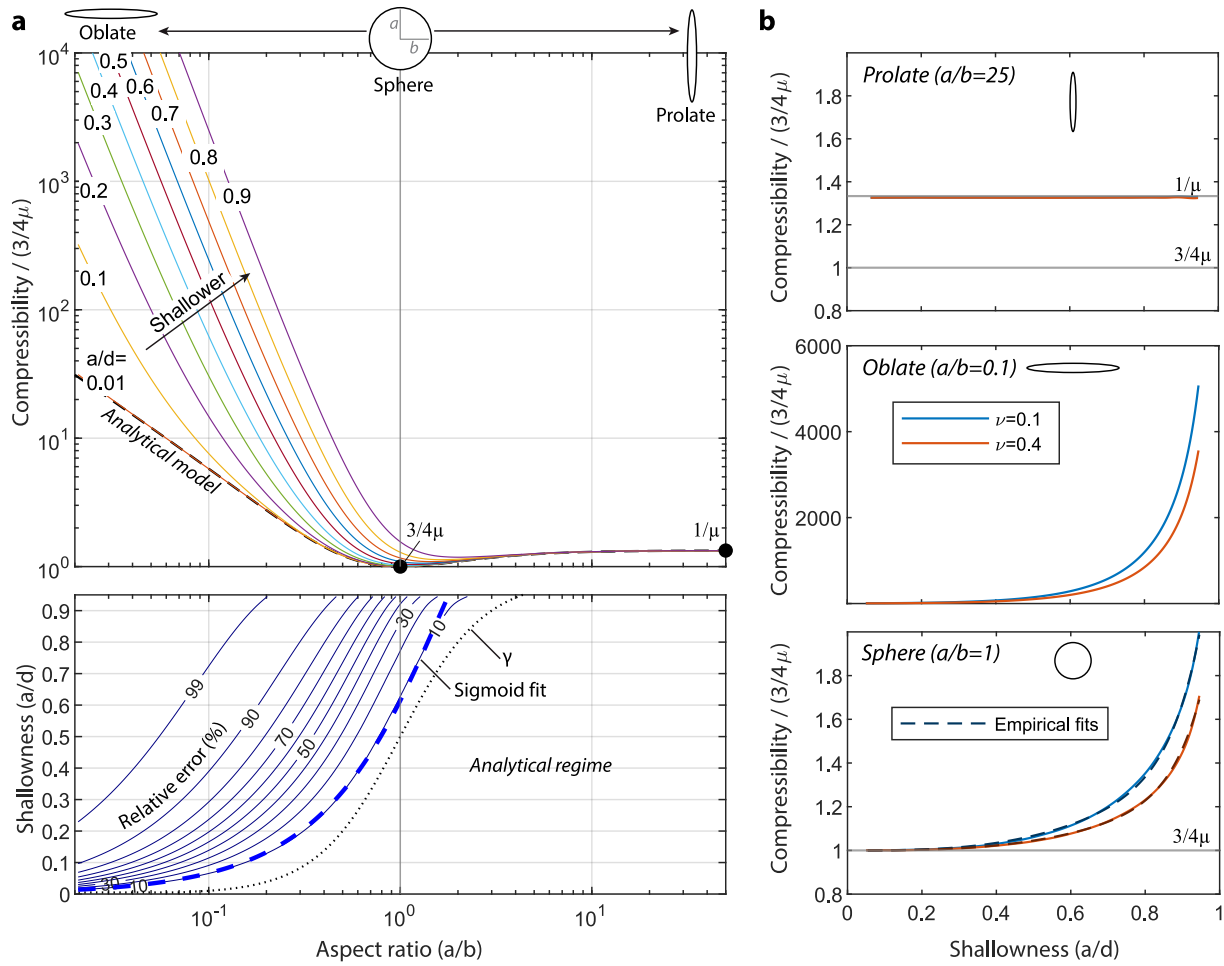


Figure 6. Cavity compressibility. (a) Top: Compressibility normalized by the value for a sphere in a fullspace ($3/4\mu$) as a function of aspect ratio for different values of a/d using emulator D3 for Poisson's ratio $\nu = 0.25$. The black dashed line shows the Yang-Cervelli analytical approximation. Bottom: Absolute relative error (ARE) between the emulator and the Yang-Cervelli model, which is treated as the reference. Since the analytical model underpredicts compressibility, the ARE cannot exceed 100% although the two models differ in some cases by orders of magnitude. The dashed line shows the γ relation of Yang et al. (1988) (Section 4.2) and the dashed blue line shows a sigmoid function fit (Equation 11) to the 10% ARE line. We denote the region of low deviation the "Analytical regime." (b) Compressibility normalized by $3/4\mu$ as a function of shallowness a/d for two values of ν , for prolate, oblate, and spherical geometries. The dashed lines on the panel for the sphere show empirical fits from Equation (10).

3.2. Surface Displacements

Vector surface displacements vary widely as a function of cavity aspect ratio and relative depth (Figure 3). In the prolate regime, analytical approximations are reasonable even for relatively shallow sources. In the oblate regime, in contrast, both the amplitude and spatial pattern of surface displacements can deviate widely from analytical approximations for even relatively deep sources.

Compared with analytical approximations, numerical results show that surface displacements due to shallow cavities are relatively more concentrated above the cavity. For shallow spheres, for instance, vertical displacements become larger and more concentrated directly above the cavity; peak horizontal displacements occur at a reduced surface distance from the cavity and the maximum is modestly increased (Figure 3). This effect could cause spherical-cavity analytical models to underestimate source depths inferred from data (McTigue, 1987).

3.3. Computational Expense

Training the emulators requires of order $\mathcal{O}(n^3) + \mathcal{O}(kn^2)$ calculations (Section 2.3). However, the computational cost of exercising the trained emulator to compute the GP predictive mean value at all k observation points scales as

only $\mathcal{O}(n^2) + \mathcal{O}(nk)$, or as $\mathcal{O}(n^2k)$ if prediction variances (uncertainties) are also required (Gu & Berger, 2016). (Note that the number of model parameters can also be important in the cost of training the emulator, but is usually less important when making predictions.) Because the computational cost of exercising the emulators is linear with k (for $k > n$) it is possible to utilize a dense observation vector without strongly impacting performance. This linearity could reduce the need to downsample large data sets, like interferometric synthetic aperture radar images, before modeling.

We find that emulators $\hat{f}_c$ can make roughly 2,000–3,000 predictions per second and emulators $\hat{f}_j$ can make >1,000 predictions per second at 1,000 random observation points (Table 2). Computing vertical and horizontal displacements u_z and u_r requires two emulator calls, and computing uncertainties reduces performance by roughly a factor of 5–10. On the other hand, for some applications speed may be improved in several ways, including efficient matrix calculations that are utilized when making predictions from numerous parameter sets simultaneously, and training the emulator using the same observation coordinates required when running the emulator, which eliminates the need for interpolation. Where applicable these techniques may accelerate calculations by more than an order of magnitude.

Comparing the computational cost of the emulator with the numerical (FEM) model is subjective, as both are strongly affected by parameterization (e.g., number of training points for the former and mesh density for the latter), as well as numerical techniques such as parallelization. However, for the parameters chosen in this work, computing a single FEM model required about 5 s (0.2 solutions/s). Thus, the emulators are able to compute surface displacements $\hat{u}$ and cavity compressibility $\hat{\beta}$ roughly 2,000× and 10,000× faster than the FEM, respectively.

The emulators are also much faster than available exact analytical models. They yield surface displacements $\hat{u}$ at least one order of magnitude faster than the published MATLAB implementation of the Zhong et al. (2019) spherical source model, with an advantage that increases with more observation points, and they yield compressibility $\hat{\beta}$ thousands of times faster, depending on the integration accuracy used for the Zhong et al. (2019) model.

For large numbers of observation points the displacement emulators are even faster than the MATLAB implementation of the Cervelli (2013) analytical model. However, the Yang-Cervelli is more general (encompassing arbitrarily dipping geometries), and the implementation used here is not optimized for speed.

These speeds make the emulator suitable for use in intensive parameter sweeps and Monte Carlo-style estimation procedures where thousands or millions of simulations must be computed. We have previously demonstrated the feasibility of this approach using an early version of the GP emulator detailed in this work in Monte Carlo parameter estimation procedures at Kilauea (Anderson et al., 2019), and also as a part of a community volcano deformation model verification exercise (Crozier et al., 2023). These studies confirm both the speed and accuracy of the approach.

4. Discussion

4.1. When Are Emulators Accurate Enough?

We have shown that emulator predictions closely match thousands of numerical simulations, and for many geometries, are far more accurate than analytical approximations. Nonetheless, an important question is the largest error that might plausibly be encountered for other, unexplored input parameters. We found that the highest emulator errors generally occurred for rather extreme geometries near the edges of parameter space and can be avoided accordingly. Relative errors can also be high for far-field observation distances where displacements are very small, but if observation vectors are dense, large relative errors at certain points should not much influence model-data comparisons. Finally, in all cases, the GP emulator's predictive uncertainties can be used to discard poor predictions, albeit at increased computational cost.

We also emphasize that emulator design should optimally depend on the range of inputs and outputs and the fidelity required for a particular problem. Considerably improved performance could be obtained by training emulators specialized for particular problems. We also expect emulator performance to improve in the future with refinement of training points, model output transformations, and other settings.

Finally, it is worth considering the accuracy required for a particular problem. For many applications – such as volcanology – emulator error is likely very small compared with the error introduced by the use of elastic spheroid models as approximations for more complex real-world structures. Thus higher degrees of accuracy may not be meaningful for most applications, excepting those for which absolute accuracy must be guaranteed (e.g., some engineering problems).

4.2. When Are Analytical Models Accurate Enough?

We have argued that GP emulators are fast, accurate, and easy to use. As such, they may be used in place of both analytical and FEM models for a wide range of source geometries, irrespective of considerations regarding the performance of analytical models. Nonetheless, approximate analytical models retain advantages for certain geometries, including (in some cases) simplicity and better speed. Thus, it may be advantageous to know in advance if an analytical model can yield a necessary degree of accuracy.

If a/d is known to be small, we have found that approximate analytical models (e.g., Cervelli, 2013; Yang et al., 1988) are generally adequate for computing both surface displacements and compressibility for a wide range of cavity aspect ratios. For shallower cavities, however, results vary widely across parameter space – analytical models perform relatively well in the prolate regime, but fail badly for all but very small a/d in the oblate regime. For spherical cavities, the Mogi (1958) and Cervelli (2013) models perform poorly for even modestly small a/d , while the higher-order McTigue (1987) model performs better for surface displacements, and the Zhong et al. (2019) model is effectively exact, although at high computational cost (Figure 5). We are aware of no higher-order analytical expressions for half-space cavity compressibility analogous to the McTigue (1987) model for surface displacements, nor any for surface displacements utilizing a volume change rather than pressure change source strength formulation. These quantities may however be computed accurately and rapidly using the emulator (Figure 5).

It is possible to utilize emulator predictions to develop simple empirical relations which can be used to determine when analytical model error exceeds some threshold. For vertical prolate cavities, Yang et al. (1988) used FEM results to argue that the pressure boundary condition on the cavity walls is violated by proximity to the free surface by less than about 10% for $\gamma > 1$, where γ represents the ratio of the depth of cavity's top surface to its minimum radius of curvature and can be written as $\gamma = (a/b)^2[(a/d)^{-1} - 1]$. We test this relation for both prolate and oblate cavities.

For cavity compressibility, we find that the γ metric is only a crude discriminator between the analytical and numerical regimes. A sigmoid function better fits the analytical model error $(\hat{\beta} - \beta_a)/\hat{\beta}$ for $\nu = 0.25$ (Figure 6a), and we find that error is less than 10% for

$$\frac{a}{d} < \frac{2.66}{1 + \exp(-a/b)} - 1.33. \quad (11)$$

This function can be used to roughly discriminate between analytical and numerical parameter regimes for $a/d < 0.9$ and ν for common materials. For a sphere, error is less than 10% for $a/d < 0.6$.

Quantifying the misfit of vector output surface displacements is more challenging because results depend strongly on the statistical metric chosen, channel (vertical vs. radial), length of observation vector, and also model formulation (pressure vs. volume change). Here we examine both the mean and maximum misfits along each observation vector $0 \leq r/d \leq 10$ and across radial and vertical displacements (normalized by the maximum displacement along r/d) (Figure 7). In general, γ poorly follows the 10% curve for both metrics. With experimentation we find that maximum error is $< 10\%$ for a polynomial

$$\frac{a}{d} < N_1 + \left[\frac{a}{b} + N_2 \right]^{N_3} \quad (12)$$

for $\nu = 0.25$ where for the volume change boundary condition $N = [-0.47, 0.046, 0.26]$ and for the pressure change boundary condition $N = [-0.61, 0.20, 0.33]$. We emphasize that these functions are non-unique and depend strongly on the misfit statistic of choice. For a spherical cavity, maxARE is generally less than about 10% for $a/d < 0.45$.

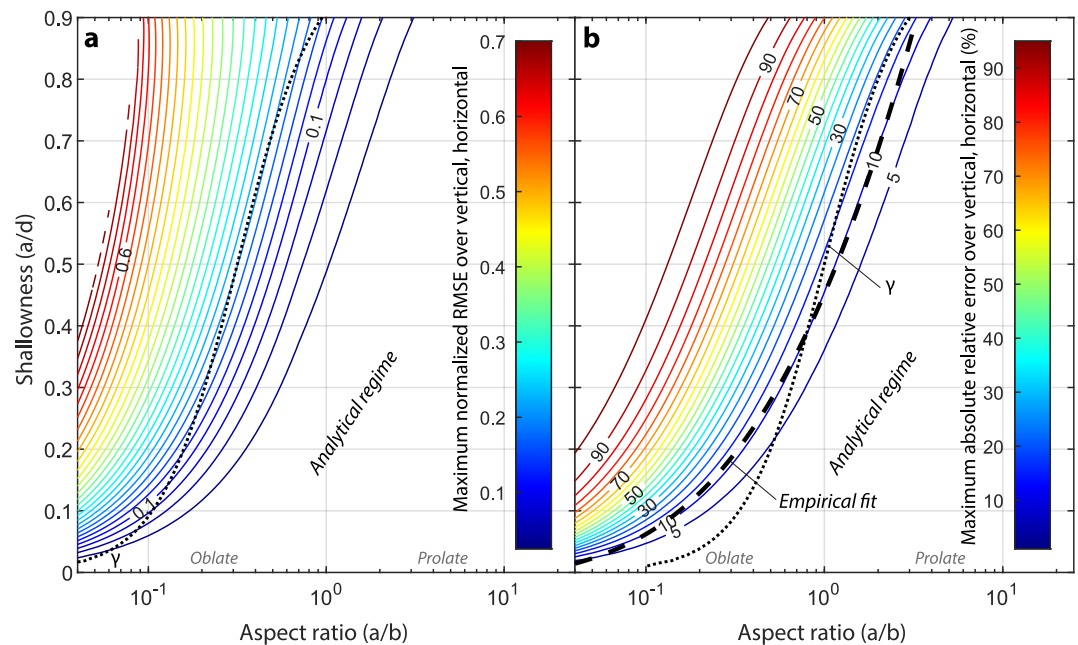


Figure 7. Deviation between predictions of surface displacements from emulator D2 and the Yang-Cervelli analytical approximation, both using the pressure change boundary condition, with the emulator output as the reference. The dotted black lines indicate the γ relation of Yang et al. (1988) (Section 4.2), which is generally a poor discriminator between the analytical and numerical regimes. (a) The maximum of vertical and horizontal normalized root mean square error (NRMSE) along r/d . (b) The maximum of vertical and horizontal absolute relative error (ARE) along r/d . The thick black dashed line indicates the fit of Equation (12).

4.3. Source Strength Formulation (Boundary Condition)

Continuum mechanical cavity models of the type discussed in this work are typically formulated using a uniform pressure boundary condition (Equation 5). A formulation based rather on the cavity's net internal volume change (Equation 4), however, may be of more direct interest for some problems; in volcanology, for instance, magma reservoir volume changes can be related with the volumes of magma injection or withdrawal. Cavity volume changes can also be more efficient to directly estimate in an inverse problem, since pressure changes are generally only weakly separable from the cavity's total volume, with which they appear as a product (Equation 5).

The relation between a cavity's volume and pressure change is controlled by its compressibility (Equation 1), which is a nontrivial function of cavity geometry and material properties (this study; Amoruso & Crescentini, 2009). (Note that a uniform pressure boundary condition does not imply uniform displacement of the cavity walls.) To our knowledge no exact analytical or higher-order approximate halfspace solutions for surface displacements are available that are formulated in terms of cavity volume change, analogous to the McTigue (1987) or Zhong et al. (2019) pressure boundary condition models (the Cervelli (2013) model is formulated in terms of either volume or pressure change but at a lower level of accuracy). Numerical models, such as FEMs, are also formulated in terms of pressure changes, with volume changes computed only as a part of the solution process. This is an important limitation; for instance, inverting geodetic observations for cavity pressure change using the higher-order McTigue (1987) model and converting to volume change using the approximate relation $\beta = 3/4\mu$ can introduce significant error.

In this work we have used emulators to 'learn' the relation between cavity parameters (geometry, material properties) and cavity volume changes computed using a numerical model. This approach sidesteps the need to formulate models using pressure boundary conditions and permits, for the first time, the calculation of surface displacements from a wide range of elastic cavity models as a function of cavity volume change.

4.4. Opportunities for Further Development

The work described here represents only an early step toward the efficient and accurate computation of deformation models using machine learning techniques. Future work may include the emulation of new output parameters including stresses and ground tilt; computation of parameters within the elastic body itself; generalization to more complex source geometries; and consideration of heterogeneous materials, surface topography, and non-elastic rheologies. In the context of volcanology, heterogeneous rock properties and significant topography can cause analytical models to fail badly (e.g., Masterlark, 2007), motivating the development of specialized three-dimensional numerical models (e.g., Hickey et al., 2015; Ronchin et al., 2013; Silverii et al., 2021), informed for instance by seismic, gravity, and topographic data. These models could be used to train custom emulators that could be used by observatories to interpret data at a particular volcano.

Many of these developments would require higher-dimensional parameter and observation spaces, which could prove challenging for emulators trained on a relatively small number of computer simulations. Techniques such as hierarchical emulation (Han & Görtz, 2012), adaptive selection of optimal training points (Gramacy & Lee, 2009), combining numerical and analytical model output, or utilizing different emulators for different parameter ranges, may help to mitigate these challenges. In some cases, however, it may be necessary to use emulator techniques that are suitable for larger numbers of training points, such as artificial neural networks.

Of course, emulators based on continuum mechanical models must suffer the same limitations as these models – namely, they are not suitable for non-elastic (e.g., plastic) deformation (Bertelsen et al., 2021), large (non-infinitesimal) displacements, or fracturing or damage of the host material (e.g., Got et al., 2019). The importance of these effects will depend on the problem at hand; large displacements, for instance, may not be common for a deeply-buried magma reservoir, but may be much more likely for intrusion into a near-surface body. Although speculative, we suggest that emulator techniques will someday facilitate the broader application of computationally expensive models of these processes where they are important.

5. Conclusions

In this work we have investigated using a machine learning technique to accelerate calculations of cavity compressibility and surface displacements due to pressure and volume changes in vertical spheroidal cavities. We have shown that emulators trained on of order 1,000 numerical finite element simulations using parallel partial Gaussian processes (Gu & Berger, 2016) can closely reproduce computationally-intensive numerical output but orders of magnitude more efficiently. The emulators offer nearly the accuracy of a numerical model at a computational cost comparable to that of some analytical models. They do not require finite element software, and because they utilize a nondimensional model formulation, can be used for spheroids of any size (e.g., mm-scale rock pores to km-scale magma reservoirs). We use the emulators to demonstrate that analytical models can fail badly – especially for oblate to mildly prolate cavities at shallow depths – and we derive a simple empirical approximation for the compressibility of a spherical cavity. The emulators can generally be used in place of other analytical or numerical approaches and are suitable for a wide range of forward and inverse problems including parametric sweeps and Bayesian Monte Carlo analyses.

The use of machine learning techniques for rapidly and accurately computing elastic deformations holds great promise and should open new opportunities for analyzing real-world data. The approach detailed herein can be customized for particular problems and should generally be extensible in the future to more complex geometries and material properties.

Appendix A: Finite Element Model

Figure A1 shows the finite element geometry and mesh used for this study.

We verified FEM output against analytical models under appropriate limiting conditions. For a shallow spherical cavity with $a/d = 0.8$ the mean absolute relative error (Equation 7) between the FEM model and Zhong et al. (2019) model over an observation distance 0 to $30d$ is better than $10^{-4}\%$. We also compared FEM-calculated displacements with those from a variety of numerical models as a part of a community exercise (Crozier et al., 2023) and found close agreement.

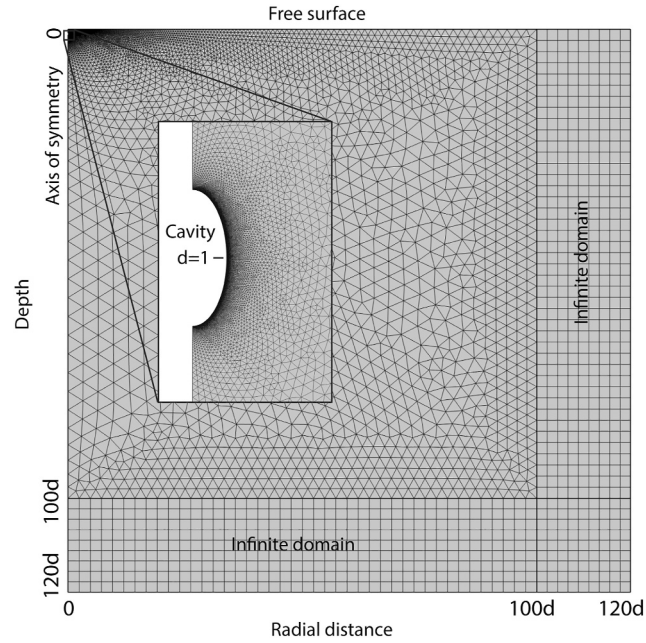


Figure A1. Example finite element geometry and mesh. Infinite-element domains use rectangular mesh elements. Distances are normalized by cavity centroid depth d (fixed to $d = 1$).

Appendix B: Verification and Benchmarking

To confirm the form of Equations 2 and 3 we showed that

$$\beta(a, b, d, \nu, \mu) = \frac{1}{\mu} f_c \left(\frac{a}{b}, \frac{a}{d}, \nu \right), \text{ and} \quad (\text{B1})$$

$$u_j(a, b, d, \Delta p, \nu, \mu; \mathbf{x}) = S^{(i)} f_j^{(i)} \left(\frac{a}{b}, \frac{a}{d}, \nu; \frac{\mathbf{x}}{d} \right). \quad (\text{B2})$$

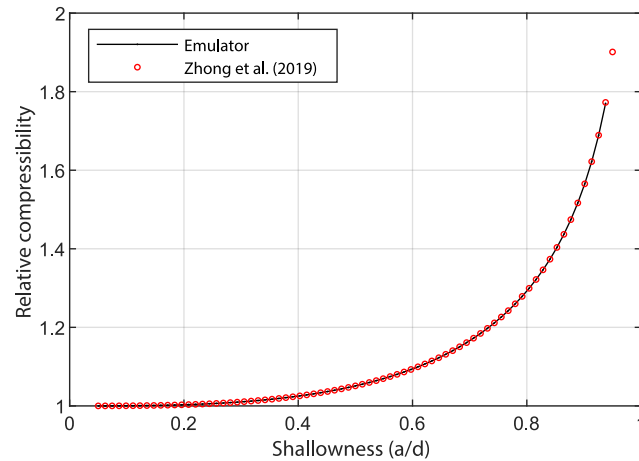


Figure B1. Comparison of compressibility for a spherical cavity calculated using emulator C1 to that calculated using the Zhong et al. (2019) model for a range of relative burial depths. An arbitrary Poisson's ratio of $\nu = 0.25$ is used. Compressibility is normalized by the compressibility of a cavity in a fullspace ($3/4\mu$). Absolute relative error between the two models is shown in Figure 5.

To do so we ran the FEM using $\sim 4,000$ random combinations of dimensional input parameters (a , b , d , Δp , ν , and μ) and computed ΔV , β and $u_j(\mathbf{x})$. Using the same input parameters but nondimensionalized with $d = 1$ and with μ and Δp fixed to arbitrary test values, we similarly ran the FEM to compute f_c and $f_j(\mathbf{x}/d)$ as described in the main text. Finally, we scaled f_c and $f_j^{(i)}$ using the appropriate linear scaling terms and compared with dimensional output. In all cases median differences were $< 0.02\%$.

Benchmarking the emulator against analytical models, including the exact Zhong et al. (2019) model for spherical sources, further confirms results (Section 3 and Figure B1).

Appendix C: Statistical Tests

We used several statistical tests to evaluate emulator performance:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^k \sum_{i=1}^{n^*} [\hat{y}(m_i^*; x_j) - y(m_i^*; x_j)]^2}{kn^*}}, \quad (\text{C1})$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\text{mean}\{\text{range}[y(\mathbf{m}^*; \mathbf{x})]\}}, \quad (\text{C2})$$

$$\text{mARE} = \frac{1}{kn^*} \sum_{i=1}^{n^*} \sum_{j=1}^k \text{ARE}(i, j), \quad (\text{C3})$$

$$\text{maxARE} = \max_i \max_j [\text{ARE}(i, j)], \quad (\text{C4})$$

where the range in equation (C2) is taken along all observation coordinates for each reference model output. For scalar-output functions the range is omitted and we take simply the mean of y . Dropping operations over i and/or j yields metrics that are a function of observation coordinate, model parameter set, or both (for scalar-output functions there are no summations over observation coordinate). Note that maxARE indicates the maximum error over the entire output vector so it may be strongly influenced by a small number of points.

Data Availability Statement

Finite element calculations were performed using the commercial software package COMSOL Multiphysics. For training and exercising the emulators we used version 1.0 of SPHEROID90GP, which is public domain software available at the USGS code repository (Anderson, 2024a). SPHEROID90GP uses version 2.0 of the parallel partial Gaussian process emulator (Gu, 2019), which is freely available at <https://doi.org/10.5281/zenodo.10397797>. SPHEROID90GP may be used to train new emulators or exercise pre-trained emulators; those developed in this work are available in Anderson (2024b).

Acronyms

FEM	Finite element method
GP	Gaussian process
PP-GP	Parallel partial Gaussian process

Acknowledgments

Fabio Pulvirenti assisted with COMSOL volume change calculations. Josh Crozier and three anonymous reviewers provided constructive comments. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. government. M. Gu acknowledges support from the National Science Foundation under Award No. 2053423.

References

- Amoruso, A., & Crescentini, L. (2009). Shape and volume change of pressurized ellipsoidal cavities from deformation and seismic data. *Journal of Geophysical Research*, 114(B2), B02210. <https://doi.org/10.1029/2008JB005946>
- Anderson, K. R. (2024a). spheroid90gp: Gaussian process emulation of vertical spheroidal elastic cavity models (Version 1.0.0) [Software]. U.S. Geological Survey. <https://doi.org/10.5066/P1NEG8BH>
- Anderson, K. R. (2024b). Trained emulators from the spheroid90gp software package [Data Release]. U.S. Geological Survey ScienceBase. <https://doi.org/10.5066/P9KAX1QP>
- Anderson, K. R., Johanson, I. A., Patrick, M. R., Gu, M., Segall, P., Poland, M. P., et al. (2019). Magma reservoir failure and the onset of caldera collapse at Kīlauea Volcano in 2018. *Science*, 366(6470), eaaz1822. <https://doi.org/10.1126/science.aaz1822>

- Andrianakis, I., & Challenor, P. G. (2012). The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56(12), 4215–4228. <https://doi.org/10.1016/j.csda.2012.04.020>
- Battaglia, M., Cervelli, P. F., & Murray, J. R. (2013). *Modeling crustal deformation near active faults and volcanic centers - a catalog of deformation models* (pp. 13–B1). U.S. Geological Survey Techniques and Methods. <https://doi.org/10.3133/tm13B1>
- Bayarri, M. J., Berger, J. O., Calder, E. S., Dalbey, K., Lunagomez, S., Patra, A. K., et al. (2009). Using statistical and computer models to quantify volcanic Hazards. *Technometrics*, 51(4), 402–413. <https://doi.org/10.1198/TECH.2009.08018>
- Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., et al. (2007). A framework for validation of computer models. *Technometrics*, 49(2), 138–154. <https://doi.org/10.1198/004017007000000092>
- Bertelsen, H. S., Guldstrand, F., Sigmundsson, F., Pedersen, R., Mair, K., & Galland, O. (2021). Beyond elasticity: Are Coulomb properties of the Earth's crust important for volcano geodesy? *Journal of Volcanology and Geothermal Research*, 410, 107153. <https://doi.org/10.1016/j.jvolgeores.2020.107153>
- Blake, S. (1981). Volcanism and the dynamics of open magma chambers. *Nature*, 289(5800), 783–785. <https://doi.org/10.1038/289783a0>
- Cervelli, P. F. (2013). Analytical expressions for deformation from an arbitrarily oriented spheroid in a half-space. In *AGU Fall Meeting Abstracts*. (Abstract V44C-06).
- Charco, M., & Galán Del Sastre, P. (2014). Efficient inversion of three-dimensional finite element models of volcano deformation. *Geophysical Journal International*, 196(3), 1441–1454. <https://doi.org/10.1093/gji/ggt490>
- Conti, S., & O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3), 640–651. <https://doi.org/10.1016/j.jspi.2009.08.006>
- Crozier, J., Karlstrom, L., Montgomery-Brown, E., Angarita, M., Cayol, V., Bato, M. G., et al. (2023). Understanding the drivers of volcano deformation through geodetic model verification and validation. *Bulletin of Volcanology*, 85(12), 74. <https://doi.org/10.1007/s00445-023-01687-4>
- David, E., & Zimmerman, R. (2011). Compressibility and shear compliance of spheroidal pores: Exact derivation via the Eshelby tensor, and asymptotic expressions in limiting cases. *International Journal of Solids and Structures*, 48(5), 680–686. <https://doi.org/10.1016/j.ijsolstr.2010.11.001>
- Davis, P. M. (1986). Surface deformation due to inflation of an arbitrarily oriented triaxial ellipsoidal cavity in an elastic half-space, with reference to Kilauea volcano, Hawaii. *Journal of Geophysical Research*, 91(B7), 7429–7438. <https://doi.org/10.1029/JB091iB07p07429>
- DeVries, P. M. R., Thompson, T. B., & Meade, B. J. (2017). Enabling large-scale viscoelastic calculations via neural network acceleration. *Geophysical Research Letters*, 44(6), 2662–2669. <https://doi.org/10.1002/2017GL072716>
- Dieterich, J. H., & Decker, R. W. (1975). Finite element modeling of surface deformation associated with volcanism. *Journal of Geophysical Research*, 80(29), 4094–4102. <https://doi.org/10.1029/JB080i029p04094>
- Farah, M., Birrell, P., Conti, S., & Angelis, D. D. (2014). Bayesian emulation and Calibration of a dynamic Epidemic model for A/H1N1 Influenza. *Journal of the American Statistical Association*, 109(508), 1398–1411. <https://doi.org/10.1080/01621459.2014.934453>
- Fialko, Y., Khazan, Y., & Simons, M. (2001). Deformation due to a pressurized horizontal circular crack in an elastic half-space, with applications to volcano geodesy. *Geophysical Journal International*, 146(1), 181–190. <https://doi.org/10.1046/j.1365-246X.2001.00452.x>
- Ghanem, R. G., & Spanos, P. D. (2003). Stochastic finite elements: A spectral approach. *Courier Corporation*.
- Got, J., Amiran, D., Stefanou, I., Brothelande, E., & Peltier, A. (2019). Damage and strain Localization Around a pressurized shallow-level magma reservoir. *Journal of Geophysical Research: Solid Earth*, 124(2), 1443–1458. <https://doi.org/10.1029/2018JB016407>
- Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.
- Gramacy, R. B., & Lee, H. K. H. (2009). Adaptive design and analysis of Supercomputer experiments. *Technometrics*, 51(2), 130–145. <https://doi.org/10.1198/TECH.2009.0015>
- Gu, M. (2018). Jointly robust prior for Gaussian stochastic process in emulation, variable selection and Calibration. *Bayesian Analysis*, 14(3). <https://doi.org/10.1214/18-BA1133>
- Gu, M. (2019). RobustGaSP: Robust Gaussian stochastic process emulation in MATLAB [Software]. *Zenodo*. <https://doi.org/10.5281/zenodo.3370575>
- Gu, M., & Berger, J. O. (2016). Parallel partial Gaussian process emulation for computer models with massive output. *Annals of Applied Statistics*, 10(3). <https://doi.org/10.1214/16-AOAS934>
- Gu, M., Wang, X., & Berger, J. O. (2018). Robust Gaussian stochastic process emulation. *Annals of Statistics*, 46(6A). <https://doi.org/10.1214/17-AOS1648>
- Han, Z.-H., & Görtz, S. (2012). Hierarchical Kriging model for variable-fidelity surrogate modeling. *AIAA Journal*, 50(9), 1885–1896. <https://doi.org/10.2514/1.J051354>
- Hickey, J., Gottsmann, J., & Mothes, P. (2015). Estimating volcanic deformation source parameters with a finite element inversion: The 2001–2002 unrest at Cotopaxi volcano, Ecuador. *Journal of Geophysical Research: Solid Earth*, 120(3), 1473–1486. <https://doi.org/10.1002/2014JB011731>
- Higdon, D., Gattiker, J., Williams, B., & Rightley, M. (2008). Computer model Calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482), 570–583. <https://doi.org/10.1198/016214507000000888>
- Johnson, D. J. (1992). Dynamics of magma storage in the summit reservoir of Kilauea Volcano, Hawaii. *Journal of Geophysical Research*, 97(B2), 1807–1820. <https://doi.org/10.1029/91JB02839>
- Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2023). Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89), 1–97.
- Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., & Sohl-Dickstein, J. (2018). *Deep neural networks as Gaussian processes*. Published as a conference paper at ICLR 2018.
- Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1), 503–528. <https://doi.org/10.1007/BF01589116>
- Masterlark, T. (2007). Magma intrusion and deformation predictions: Sensitivities to the Mogi assumptions. *Journal of Geophysical Research*, 112(B6), B06419. <https://doi.org/10.1029/2006JB004860>
- Masterlark, T., Feigl, K. L., Haney, M., Stone, J., Thurber, C., & Ronchin, E. (2012). Nonlinear estimation of geometric parameters in FEMs of volcano deformation: Integrating tomography models and geodetic data for Okmok volcano, Alaska. *Journal of Geophysical Research*, 117(B2). <https://doi.org/10.1029/2011JB008811>
- McTigue, D. F. (1987). Elastic stress and deformation near a finite spherical magma body: Resolution of the point source paradox. *Journal of Geophysical Research*, 92(B12), 12931–12940. <https://doi.org/10.1029/JB092iB12p12931>
- Mogi, K. (1958). *Relations between the eruptions of various volcanoes and the deformation of the ground* (Vol. 36, pp. 99–134). Bulletin of the Earthquake Research Institute, University of Tokyo.

- Neal, R. M. (2012). *Bayesian learning for neural networks* (Vol. 118). Springer Science & Business Media.
- Nikkhoo, M., & Rivalta, E. (2023). Surface deformations and gravity changes caused by pressurized finite ellipsoidal cavities. *Geophysical Journal International*, 232(1), 643–655. <https://doi.org/10.1093/gji/ggac351>
- Okazaki, T., Ito, T., Hirahara, K., & Ueda, N. (2022). Physics-informed deep learning approach for modeling crustal deformation. *Nature Communications*, 13(1), 7092. <https://doi.org/10.1038/s41467-022-34922-1>
- Overstall, A. M., & Woods, D. C. (2016). Multivariate emulation of computer simulators: Model selection and Diagnostics with application to a Humanitarian Relief model. *Journal of the Royal Statistical Society - Series C: Applied Statistics*, 65(4), 483–505. <https://doi.org/10.1111/rssc.12141>
- Owen, N. E., Challenor, P., Menon, P. P., & Bennani, S. (2017). Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1), 403–435. <https://doi.org/10.1137/15m1046812>
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press. (OCLC: ocm61285753).
- Ronchin, E., Masterlark, T., Dawson, J., Saunders, S., & Marti Molist, J. (2017). Imaging the complex geometry of a magma reservoir using FEM-based lineations modeling of InSAR data: Application to Rabaul caldera, Papua New Guinea. *Geophysical Journal International*, 209(3), 1746–1760. <https://doi.org/10.1093/gji/ggx119>
- Ronchin, E., Masterlark, T., Molist, J. M., Saunders, S., & Tao, W. (2013). Solid modeling techniques to build 3D finite element models of volcanic systems: An example from the Rabaul Caldera system, Papua New Guinea. *Computers & Geosciences*, 52, 325–333. <https://doi.org/10.1016/j.cageo.2012.09.025>
- Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4. <https://doi.org/10.1214/ss/1177012413>
- Santner, T. J., Williams, B. J., & Notz, W. I. (2003). *The design and analysis of computer Experiments* (Vol. 1). Springer.
- Segall, P. (2010). *Earthquake and volcano deformation*. Princeton University Press.
- Segall, P. (2013). Volcano deformation and eruption forecasting. *Geological Society, London, Special Publications*, 380(1), 85–106. <https://doi.org/10.1144/SP380.4>
- Silverii, F., Pulvirenti, F., Montgomery-Brown, E., Borsa, A., & Neely, W. (2021). The 2011–2019 Long Valley Caldera inflation: New insights from separation of superimposed geodetic signals and 3D modeling. *Earth and Planetary Science Letters*, 569, 117055. <https://doi.org/10.1016/j.epsl.2021.117055>
- Snelson, E., & Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18.
- Sobol, I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4), 86–112. [https://doi.org/10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9)
- Spiller, E. T., Bayarri, M. J., Berger, J. O., Calder, E. S., Patra, A. K., Pitman, E. B., & Wolpert, R. L. (2014). Automating emulator construction for geophysical Hazard maps. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1), 126–152. <https://doi.org/10.1137/120899285>
- Trasatti, E., Giunchi, C., & Agostinetti, N. P. (2008). Numerical inversion of deformation caused by pressure sources: Application to Mount Etna (Italy). *Geophysical Journal International*, 172(2), 873–884. <https://doi.org/10.1111/j.1365-246X.2007.03677.x>
- Xiu, D., & Karniadakis, G. E. (2003). Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1), 137–167. [https://doi.org/10.1016/s0021-9991\(03\)00092-5](https://doi.org/10.1016/s0021-9991(03)00092-5)
- Yang, X.-M., Davis, P. M., & Dieterich, J. H. (1988). Deformation from inflation of a dipping finite prolate spheroid in an elastic half-space as a model for volcanic stressing. *Journal of Geophysical Research*, 93(B5), 4249–4257. <https://doi.org/10.1029/JB093iB05p04249>
- Zhang, H. (2016). *A General Approach for Solving Inverse Problems in geophysical systems by applying finite element method and metamodel techniques (Unpublished doctoral dissertation)*. Technischen Universität München.
- Zhong, X., Dabrowski, M., & Jamtveit, B. (2019). Analytical solution for the stress field in elastic half-space with a spherical pressurized cavity or inclusion containing eigenstrain. *Geophysical Journal International*, 216(2), 1100–1115. <https://doi.org/10.1093/gji/ggy447>
- Zimmerman, R. W. (1985). Compressibility of an Isolated spheroidal cavity in an isotropic elastic medium. *Journal of Applied Mechanics*, 52(3), 606–608. <https://doi.org/10.1115/1.3169108>