Force-Chain Finder: A software tool for the recursive detection of force-chains in granular materials via minor principal stress

Citation for published version:
Ejtehadi, O, Gupta, A, Khajepor, S & Haeri, S 2024, 'Force-Chain Finder: A software tool for the recursive detection of force-chains in granular materials via minor principal stress', *Computer Physics Communications*, vol. 297, 109070. https://doi.org/10.1016/j.cpc.2023.109070

Digital Object Identifier (DOI):
10.1016/j.cpc.2023.109070

Published In:
Computer Physics Communications

Force-Chain Finder: A software tool for the recursive detection of force-chains in granular materials via minor principal stress

Omid Ejtehadi, Aashish K Gupta, Sorush Khajepor, Sina Haeri*

Institute for Materials and Processes School of Engineering, University of Edinburgh, Sanderson Building, King's Buildings Robert Stevenson Road, Edinburgh, EH9 3FB, UK

* Corresponding author: shaeri@ed.ac.uk

Abstract

Force transmission in granular media occurs through an inhomogeneous network of inter-particle contacts referred to as force-chains. A thorough understanding of the structure of these chains is indispensable for a better comprehension of the macroscopic signatures they generate. This paper introduces Force-Chain Finder (FCF), an open-source software tool designed for detecting force-chains in granular materials. Leveraging the stress tensor computed for each particle based on its interactions with neighboring particles, the tool effectively identifies the magnitude and direction of the most compressive principal stress. Through a recursive traversal of particles and their neighbors, force-chains are robustly detected based on the alignment of the principal stress directions, which is decided by a parameter α (an angle in radians). The software provides a comprehensive suite of post-processing features, including the exportation of results in different formats, enabling detailed analysis of specific regions and dynamic phenomena. Additionally, the software facilitates the computation of statistical measures pertaining to chain size and population. By streamlining the identification and characterization of force-chains within discrete element method (DEM) simulations, this tool significantly enhances the efficiency and accuracy of force-chain analysis. Thus, the software promotes deeper insights into the behavior of granular materials by enabling researchers to effortlessly detect and analyze force-chains.

Keywords

Force-chain detection, granular flow, discrete element method, recursive search.

PROGRAM SUMMARY

Program Title: Force Chain Finder (FCF)
Licensing provisions: GNU General Public License 3 (GPLv3)
Programming Language: C++, Python
Supplementary material: Illustrative examples

Nature of the Problem: FCF has been developed with the primary objective of efficiently and comprehensively detecting force networks that arise in Discrete Element Modeling (DEM) simulations of granular flow.

Solution Method: The method is based on the work of "J. Peters, M. Muthuswamy, J. Wibowo, and A. Tordesillas, Characterization of force chains in granular material, Physical Review E 72, 041307 (2005)". It utilizes the concept of minor principal stress to identify quasilinear chains, each consisting of at least three particles. However, significant algorithmic modifications have been implemented to enhance the accuracy of the procedure, enabling the detection of chain branching and merging.

1 Introduction

Granular materials, including soils, powders, and sand, are abundant in both natural and industrial settings, attracting significant attention from scientists and engineers due to their intriguing behavior. While the macroscopic or bulk properties of these materials provide valuable information for engineering applications, a more detailed understanding is necessary to comprehend phenomena such as jamming, shear-induced yielding, and mechanical response. This requires a closer examination of the microstructures that emerge at the particle level.

One crucial aspect contributing to the mechanical behavior of granular materials is the formation and propagation of heterogeneous distributions of inter-particle force networks commonly referred to as force-chains. Force-chains represent paths within the granular system that transfer forces between particles, bearing the majority of the load in the system. The identification of force-chains holds significance not only for granular materials but also for the characterization of other material systems, including foams, emulsions, and dense active matter.

The subject of analysis has received significant attention, encompassing various perspectives and objectives. Pioneering investigations utilized photoelastic beads to acquire a qualitative phenomenological understanding [1-5]. Subsequently, Howell, Behringer and Veje [6] quantified this approach in the context of jamming, and the method continues to be further developed to this day [7-11].

Various investigations have been conducted to understand the influence of macroscopic stress on the microscopic arrangement of particles in granular materials. Early experiments and simulations revealed the distribution of forces in confined packings of particles, showing a power-law distribution below the mean force and exponential decay above it and the correlation between contact forces and packing texture was established [12, 13]. Furthermore, some studies examined the distribution of forces during compression, demonstrating near-uniform distribution below the mean force and exponential decay above it [14]. In another study, fragility was linked to the marginal stability of force-chain networks, while jamming transitions were characterized by an increase in mean contact number and pressure [15]. Moreover, forced fields can be applied in the identification of crystallization, as demonstrated by An, Yang, Dong and Yu [16] in the case of mechanical vibrations.

Despite the wide range of applicability and importance of force-chains, there is no universally accepted detection method for identifying force-chains. Some studies involve visualizing individual contact forces using lines, where the thickness of the lines represents the magnitude of the contact force [13, 17]. Alternatively, some methods rely on tracking the path of maximum contact force at each particle [18, 19]. Other detection methods utilize particle properties such as potential energy density [20] or differences in principal stresses [6, 19, 21].

The aforementioned methods are incomplete in either quantifying the force magnitude, recognizing the linearity of the chains, or their applicability to real-world three-dimensional scenarios. Within this field of research, a significant contribution closely related to the present study is the work of Peters, Muthuswamy, Wibowo and Tordesillas [22]. They introduced a method for quantifying force-chains using a well-defined criterion comprising two components. Firstly, a force-chain is defined as a quasi-linear arrangement of three or more particles. Secondly, the vector representing the most compressive principal stress identifies the stress concentration within each particle along the chain. Therefore, the method of Peters, Muthuswamy, Wibowo and Tordesillas [22] becomes a quantitative approach by introducing the maximum principal concepts and also preserves the linearity (quasi-linearity) of the force-chain. In the past decade, attempts to detect force-chains by incorporating concepts from network science [23] or by relying on digital image correlation [24], and more recently, with the application of graph neural networks to predict force-chains in jammed solids [25] are reported.

Among the existing approaches, the method presented by Peters, Muthuswamy, Wibowo and Tordesillas [22] is widely regarded as the most precise and comprehensive. Not only does it provide quantitative measurements, but it also offers insights into the direction of force distribution while preserving the linearity or quasi-linearity of force-chains. Leveraging this algorithm, we have developed our software, which incorporates several new algorithmic enhancements and fixes. Notably, we propose a recursive approach that ensures the inclusion of all relevant particles, including those involved in chain branching and self-intersections—a research direction proposed by Peters, Muthuswamy, Wibowo and Tordesillas [22].

The other notable advantage of our algorithm lies in its ability to analyze specific and moving regions, enabling a more detailed examination of the formation and propagation of force-chains. This capability holds great promise for enhancing our understanding of the mechanical behavior of granular materials in real-world engineering scenarios. The enhanced functionality of our software, known as the Force-Chain Finder, positions it as the first open-source tool capable of efficiently and reliably detecting force-chains in discrete simulations. Its potential applications span a wide range, from predicting landslides [26] to optimizing powder-based additive manufacturing processes [27-29], as well as supporting operations in the pharmaceutical [30] and ceramic production [31] industries.

2 Theoretical background

Peters, Muthuswamy, Wibowo and Tordesillas [22] demonstrated that the particle property to identify both the direction and magnitude of force chains is the most compressive principal stress that can be derived from the particle stress tensor:

$$\sigma_{ij} = \frac{1}{V}\sum_{c=1}^{N^c} f_i^c r_i^c \tag{1}$$

where V, N^c, f^c and r^c are the volume of particles, the number of contacts of the particle, the ith component of the force acting at the contact, and the jth component of the radius vector from the center of the particle to the point of contact, respectively. The principal stresses are given by:

$$\sigma_1 = \frac{\sigma_{11}+\sigma_{33}}{2} + \sqrt{\left(\frac{\sigma_{11}-\sigma_{33}}{2}\right)^2 + \left(\sigma_{13}\right)^2}\ , \tag{2}$$

$$\sigma_3 = \frac{\sigma_{11}+\sigma_{33}}{2} - \sqrt{\left(\frac{\sigma_{11}-\sigma_{33}}{2}\right)^2 + \left(\sigma_{13}\right)^2}\ , \tag{3}$$

where the σ_{ij}, σ_1, and σ_3 are the components of the symmetric part of the particle stress, the major principal stress, and the minor principal stress, respectively. Furthermore, the direction of the major principal stress (i.e., the direction of the most compressive force as a tension-positive convention is adopted), θ is given by:

$$\tan\left(2\theta\right) = \frac{2\sigma_{13}}{\sigma_{11}-\sigma_{33}}\,. \tag{4}$$

In [22], it was suggested that only contact forces with magnitudes greater than the average, or equivalently, particles with stresses greater than the average, should be considered as part of force chains. Consequently, the following criterion is applied for the inclusion of a particle in the force chain:

$$\left|\sigma_3^j\right| > \frac{1}{N}\sum_{i=1}^{N}\left|\sigma_3^i\right| . \tag{5}$$

where N is the number of particles. The above criterion suggests that only those particles with minor stresses greater than the average minor stress of surrounding particles can be part of the force-chains. However, the implementation suggested in the current study allows for relaxing this criterion. It should be noted that even though this check is skipped, the same force network can be obtained by the current algorithm using the provided post-processing filters without any loss of information. The other necessary definitions here are *particle assembly* and *(quasi-)linearity*. Particle assembly states that an assembly of particles with less than three is not recognized as a chain where the quasilinear condition implies the line up of the minor principal stresses with a defined deviation angle (α).

3 Force Chain Finder overview

The software can read LIGGGHTS (LAMMPS improved for general granular and granular heat transfer simulations) [32] output file format containing particle pair interactions. The stress tensor on each particle based on its interactions with other particles is calculated, and then the Eigen Library [33] is used to find the magnitude and direction of the most compressive principal stress. It should be noted that LIGGGHTS is an open-source software package for discrete element method (DEM) simulations of granular materials, including particles of various shapes and sizes, with various interactions between them, such as friction, cohesion, and deformation, and is built on top of the large-scale atomic/molecular massively parallel simulator (LAMMPS), which is a widely used open-source software package for molecular dynamics (MD) simulations. While the Force-chain Finder uses the LIGGGHTS file format as an input, there is no restriction on using any other commercial (e.g. [34, 35]) or in-house (e.g. [36-40]) particle-based code for generating the system under analysis, as long as the outputs are formatted accordingly.

Force-chains are found recursively by traversing a particle, its neighbors, neighbors of those neighbors, and so on. The software also offers post-processing features, including exporting the force-chain results in CSV format, analyzing a specific region, analyzing a moving region, finding the normal vector of a force-chain, and chains statistics such as average, median, minimum, and maximum length, as well as the number of chains.

Fig. 1 illustrates the file structure of the software. The figure includes only some of the crucial files. Within the example directory, a subdirectory containing the LIGGGHTS output files, as well as sample force-chains detection applications can be found. For instance, "single.cpp" is used for analyzing a single snapshot of the DEM results, "batch.cpp" is employed for processing various snapshots in a transient simulation, "moving.cpp" is utilized in scenarios with moving elements, and "filter.cpp" is designed for filtering the results based on specific criteria. The python directory contains Python applications dedicated to generating visualization data, specifically tailored for use with the Paraview software. The source files for the software are located in the "src" directory. The packages Eigen and Pybind are placed in the src directory. Pybind is an open-source C++ library that offers a seamless bridge between C++ and Python. It enables users to expose C++ code to Python and vice versa, allowing them to call C++ functions from Python and Python functions from C++. Both Eigen and Pybind are header-only libraries, meaning that they do not require a separate installation or compilation. They can be used by simply including the appropriate header files in the C++ code and then compiling the code as part of the project build process.

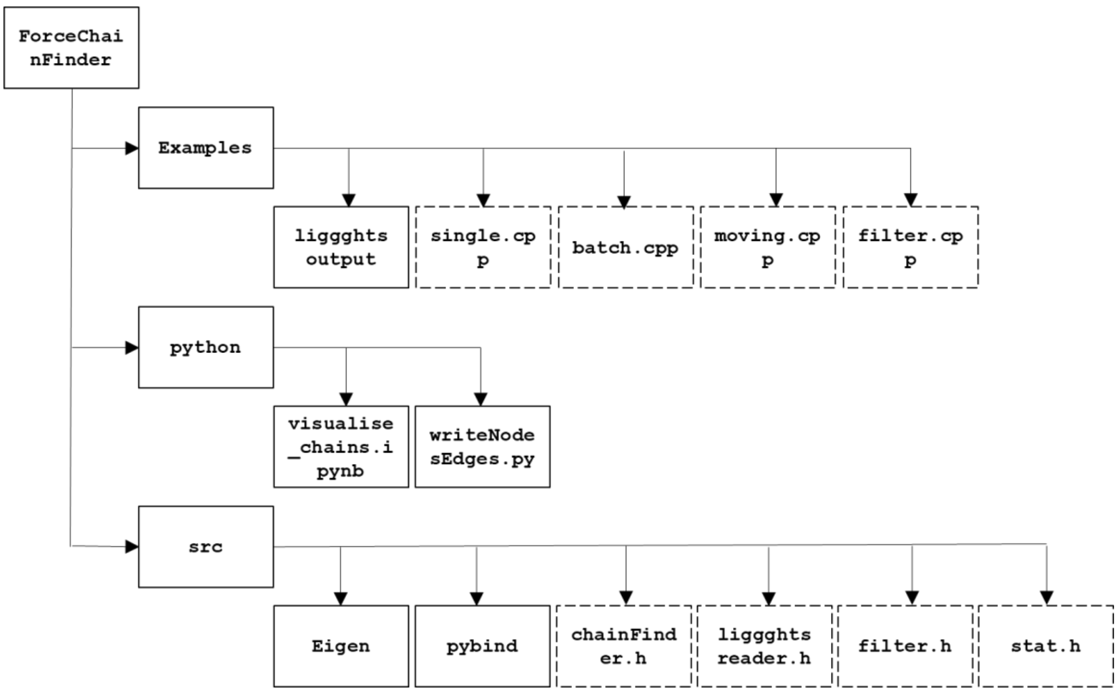

Fig. 1 The file structure of the Force Chain Finder. The solid boxes represent directories and the dashed boxes represent files.

3.1 Software algorithm

The Software implements a class called ‘ChainFinder’, which is responsible for finding chains of particles in a system. The class takes in a vector of particle objects and a maximum angle α (in radians) between neighboring particles that can form a chain. The ChainFinder class has a method called ‘RecursiveFindChains’ which is the entry point for finding all the chains in the system. It iterates across individual particles and if a particle exhibits a non-zero minor stress and is not yet part of a chain, the process of chain identification commences. This is initiated by invoking the ‘RecursiveFindOneSideOfChain’ function in both potential orientations (forward and backward relative to the minor principal stress) that a chain-affiliated particle could assume. As depicted in **Fig. 2**, the propagation of force along the chain is visually represented by a white line. Notably, the bidirectional arrow superimposed on the particle currently under scrutiny (red) signifies the alignment of the predominant compressive stress direction.

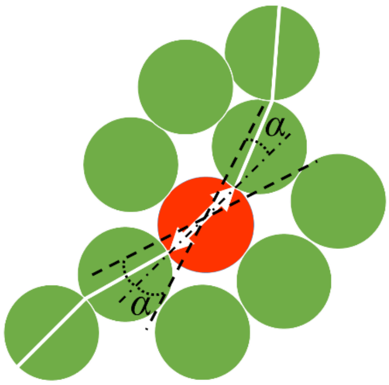

Fig. 2 Representation of particles within a simplified segment of a force chain and the input angle α.

The ‘IsNeiNextLink’ function is a private member function that checks if a neighbor particle (‘NEIGHP’) can be the next link in a chain (‘LINKEDP’). It considers various conditions such as neighbor particles having minor stress, belonging to the same or different chain, satisfying angle constraints, and range checks. The chain is either stored in the ‘chain’ vector and each particle in the chain is assigned a chain ID (‘CHAIN_ID’) or it is discarded if it has less than 3 particles. Below is a code snippet representing the ChainFinder implementation:

```
#include "vector"
#include "particle.h"
#include <Eigen/Dense>
#include <cmath>
```

```
#include "vectorAngle.h"
namespace ForceChain
{
    class ChainFinder
    {
        std::vector<Particle> &particles;
        double alpha;
    public:
        std::vector<std::vector<size_t>> chains;
        ChainFinder(std::vector<Particle> &particles_, double alpha_)
            : particles(particles_), alpha(alpha_) {}
    protected:
        auto IsNeiNextLink(Particle &link, Particle &nei, int &neiDirSign)
        {
            if (!nei.hasMinorStress || nei.chainId>=0)
                return false;
            auto xln = nei.distanceTo(link);
            if (xln.norm() > 2 * (nei.radius + link.radius))
                xln[1] = -xln[1];
            auto angle_forward = angleBetweenVectors(xln, link.minorDir);
            if (angle_forward >= alpha)
            {
                return false;
            }
            auto angle_back = angleBetweenVectors(-xln, nei.minorDir);
            neiDirSign = +1;
            if (angle_back > M_PI / 2)
            {
                angle_back = M_PI - angle_back;
                neiDirSign = -1;
            }
            if (angle_back >= alpha)
            {
                return false;
            }
            return true;
        }
        void RecursiveFindOneSideOfChain(std::vector<size_t> &chain, Particle *link)
        {
            for (auto inei : link->neighbors)
            {
                auto &nei = particles[inei];
                int neiDirSign;
                if (!IsNeiNextLink(*link, nei, neiDirSign))
                    continue;
                chain.push_back(inei);
                nei.chainId = link->chainId;
                nei.minorDir = -neiDirSign * nei.minorDir;
                RecursiveFindOneSideOfChain(chain, &nei);
            }
        }
    public:
        auto &RecursiveFindChains()
        {
            for (auto &center : particles)
            {
                if (!center.hasMinorStress || center.chainId>=0)
                    continue;
                std::vector<size_t> chain;
```

```
                chain.push_back(center.id);
                center.chainId = chains.size();
                RecursiveFindOneSideOfChain(chain, &center);
                // Trying the other side of center
                center.minorDir = -center.minorDir;
                RecursiveFindOneSideOfChain(chain, &center);
                if (chain.size() < 3)
                {
                    for (auto &particleId : chain)
                    {
                        particles[particleId].chainId = -1;
                    }
                }
                else
                {
                    chains.push_back(chain);
                }
            }
            return chains;
        }
    };
}
```

Fig. 3 illustrates the comprehensive algorithm outlining the flow of the system. The abbreviations DIR, FWD, and BWD correspond to direction, forward, and backward, respectively. Within the algorithm, σ_3 represents the minor principal stress, while the symbol 'P' signifies a particle. This algorithm serves as a valuable reference for understanding the operational steps and relationships within the system, providing a foundation for subsequent analyses and discussions.

3.2 Software functionalities

While the main goal of the FCF is the accurate detection of the force-chains in a granular simulation, a few features are added to the tool that enhances the functionality of the software. The main functionalities are as follows:

1) Automatic detection of the chains based on the work of [22]
2) Versatility for the analysis
3) Filtering the results for specific purposes: This capability allows for deleting the chains that do not satisfy certain conditions.
4) Writing the output in VTP and CSV format for postprocessing via different tools. VTP stands for VTK Polydata where VTK is a powerful open-source library for processing and visualizing scientific data in 3D. Therefore, the results can be analyzed by ParaView, as well as other software that supports VTK file formats. The average, median, minimum and maximum length, and number of chains are provided as well.

5) Capability of finding the force-chain in a batch simulation
6) Capability of finding force-chains in a moving box with arbitrary dimensions.

4 Exemplary test cases

4.1 Fabricated force-chain case

The first example presented in this study involves retrieving a carefully designed force network as shown in **Fig. 4**(a), imposed on a simple configuration of nine particles. The force at every contact point is the same in magnitude except for the contact between particles 5 and 6, where no force exists by design. Moreover, these forces are aligned with the local surface normal vector at the points of contact.

In order to capture all the force-links, irrespective of the minor principal stress alignment, we initially set the value of α to $\pi/2$. **Fig. 4**(b) illustrates a potential configuration of force-chains generated using the algorithm proposed by Peters, Muthuswamy, Wibowo and Tordesillas [22]. It is evident that in this particular scenario, the algorithm fails to detect all the force-links present and gives rise to two distinct force-chains. In contrast, the FCF software successfully identifies the complete imposed configuration, as depicted in **Fig. 4**(c). By selecting a smaller α value, such as $\pi/4$, only the linear chains are retained, as shown in **Fig. 4**(d). It is noteworthy that the force-link between particle 4 and 5 is not picked up in this case due to the restriction of the minimum number of particles in a chain being set to 3. Finally, it is important to emphasize that the Force-Chain Finder (FCF) possesses the capability to accurately identify not just all the particles involved in the force-chains but also all the individual force-links. This is a significant advantage over the existing algorithm since it facilitates the formation of force-chain structures, which better reflect the accompanying physical phenomenon.

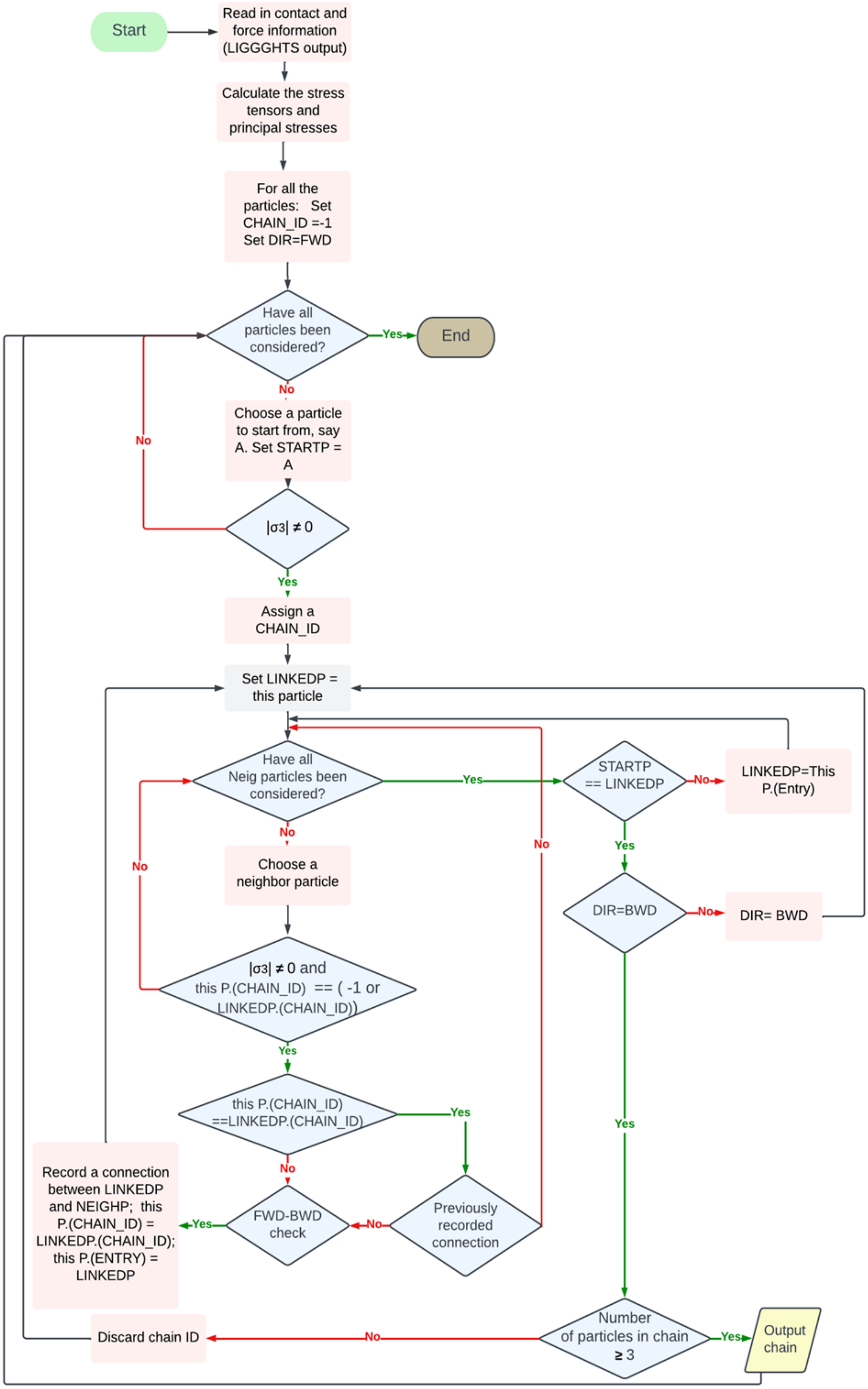

Fig. 3 Flowchart presenting the devised algorithms

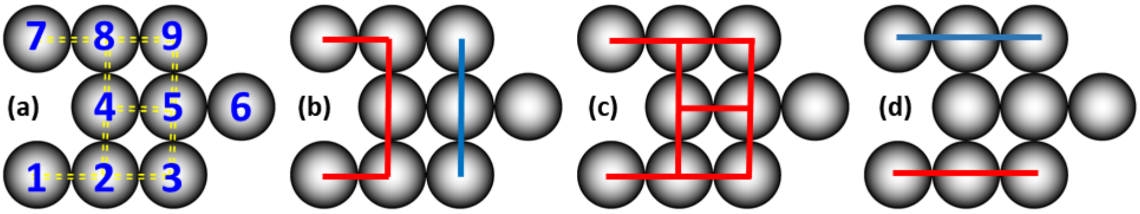

Fig. 4 (a) Representation of an arbitrary fabricated force-chain, (b) the detected force-chain using the algorithm of Peters, Muthuswamy, Wibowo and Tordesillas [22], (c) FCF detected force-chains with $\alpha = \pi/2$, and (d) FCF detected force-chains with $\alpha = \pi/4$.

4.2 Shear-driven granular flow in a two-dimensional setup

The following example serves to demonstrate the algorithm's efficacy in detecting chains in a two-dimensional shear scenario. Initially, a number of particles are generated randomly inside a box, which extends in the z-direction only by a particle diameter, in order to create a 2D system. These particles are then subjected to bi-axial compression, followed by shearing exerted by two opposing moving walls, made up of particles moving only with a pre-specified velocity. The prevailing consensus is that the dominant force-chains in such systems are formed along the compressive shear direction [15] which in the current case corresponds to a 45º orientation (slope -1 line) in relation to the shear direction. These chains undergo intermittent construction and destruction as the flow progresses. In **Fig. 5**, a snapshot of the simulation at a specific time is presented. **Fig. 5**(a) shows the velocity of the particles, illustrating the propagation of wall velocity throughout the system. **Fig. 5**(b) and (c) display the force-chains through the maximum compressive stress experienced by the particles and their corresponding chain IDs, respectively. This visual examination serves as a straightforward test, and it is evident that the detected force-chains concur with the anticipated behaviour.

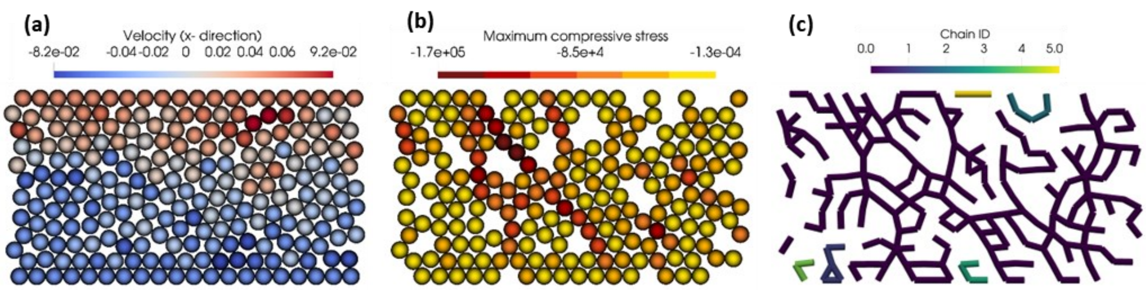

Fig. 5 Snapshot of the sheared particulate system after compression: (a) Particle velocity in the x-direction, (b) Force-chains represented by particles colored according to their maximum compressive stress, (c) Force-chain network visualized with colors indicating chain IDs.

4.3 Isotropic compression of a three-dimensional box filled with particles

The subsequent example demonstrates the evolution of force-chains in a simulation involving 3D isotropic compression. A cubic box, randomly seeded with 5 mm diameter particles at 0.3 solid fraction, is shrunk until a solid fraction of 0.64 is attained. Intuitively, it is expected that as the compression progresses, an increasing number of force-links will form within the system. However, what is unclear beforehand is the size and distribution of the distinct force-chains that appear and vanish in the process as depicted in **Fig. 6**(a) to (e). In these figures, the outer box represents the initial domain while the inner box depicts the region of confinement as the simulation progresses. Initially, the domain contains 354 relatively shorter chains. Subsequently, as the box is further compressed, the number of chains escalates to 579 and then 785 before abruptly decreasing to 139 longer chains. Ultimately, a single persisting force-chain appears, which is not surprising as the angle α dictating the permissible alignment among links was set to the maximum i.e., $\pi/2$. The emergence of a single force-chain however marks a critical transition in the structural arrangement of the particles.

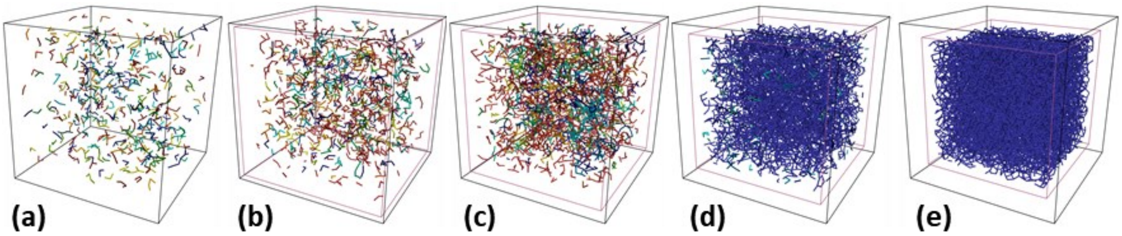

Fig. 6 Formation and merging of force-chains detected by the FCF software in a three-dimensional isotropic simulation case. Panels (a) to (e) display sequential snapshots at varying compression levels, as indicated by the inner box. Panel (f) represents the probability density functions for steps (a) to (d).

Fig. 7 illustrates the tendency for the emergence of distinct cluster sizes denoted by c. These sizes pertain to the total count of particles encompassed within a particular force-chain configuration, observable at various stages of the compression process. The likelihood of encountering a cluster possessing a size c is determined through the subsequent calculation:

$$P(c) = \frac{n_c}{\sum_{c=3}^{c_{\max}} n_c}. \tag{6}$$

where n_c, and $c_{\max}$ are the number of clusters and the maximum cluster size.

The cluster sizes at a given step are first normalized by the largest cluster size at that step, and then a probability density function is constructed. At the beginning of the compression, the force-chains with normalized cluster size at the lower end of the spectrum significantly outnumber those with higher cluster sizes. The effect becomes even more pronounced as the compression proceeds.

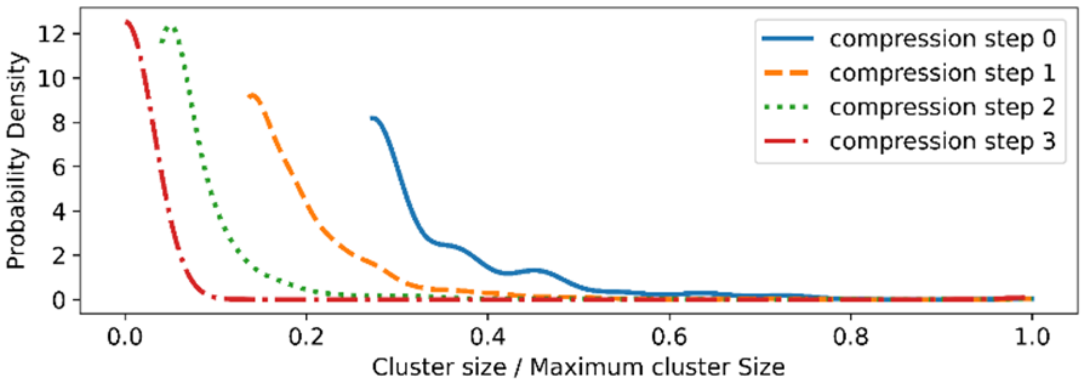

Fig. 7 The propensity of formation of the different cluster sizes for the case of three-dimensional isotropic compression

5 Conclusions

The Discrete Element Method (DEM) has emerged as a widely recognized simulation technique for granular systems, finding extensive applications in both industry and academia. A multitude of commercial and open-source packages have successfully implemented the DEM formulation, enabling efficient simulation of particulate systems. Additionally, significant efforts have been directed toward the development of post-processing and visualization tools [20].

Despite the evident significance of force-chain detection in granular systems, as highlighted in previous sections, the absence of open-source software capable of automatically detecting and outputting force-chains with minimal user input has been a notable limitation. Addressing this gap, our work represents the first endeavor to fulfill this objective. Building upon the well-established methodology proposed by Peters, Muthuswamy, Wibowo and Tordesillas [22], we carefully revisited and refined the approach, incorporating slight modifications. These enhancements were diligently encoded into an accessible software solution, freely available as open-source code.

The development of our software, designed to detect force networks in granular materials, holds substantial implications for the field of granular material physics. Our software provides an accurate and efficient means of identifying and analyzing force-chains, opening new avenues for research in this domain. It enables researchers to explore the intricate relationship between microstructures and the mechanical behavior of granular materials, as well as investigate the role of force-chains in phenomena such as granular flow, deformation, and jamming.

This study has provided a comprehensive discussion of the structure of the developed code and the flow of the software. Moreover, the capabilities of the software, along with the implemented algorithmic fixes, have been successfully demonstrated through the presentation of three illustrative examples. These examples encompassed a wide range of scenarios, ranging from a simple fabricated network to a more intricate three-dimensional compression case involving a significant number of particles. The successful verification of the software's output in these examples further solidifies its reliability and suitability for analysing force-chains in granular materials. Thus, the availability of our software empowers researchers to swiftly and accurately analyze force-chains in granular materials, enabling them to focus on more complex research questions without the burden of developing new tools.

Our software has been met with positive reception among the intended user group and is actively being employed in various projects, including in [29]. Moving forward, our efforts serve as a foundation for future developments and refinements in force-chain analysis. We anticipate that our open-source software will stimulate collaborative research, enabling researchers to expand upon our work and enhance the capabilities of force-chain detection algorithms.

Acknowledgments

This work was funded by UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee EP/X024180/1, and EP/T009128/2.

References

[1] T. Wakabayashi, Photo-elastic method for determination of stress in powdered mass, Journal of the Physical Society of Japan, 5 (1950) 383-385.
[2] G. Schneebeli, Une mécanique pour les terres sans cohésion, Compte rendus des séances de l'Académie des Sciences, 243 (1956) 2647-2673.
[3] P. Dantu, Contribution à l'étude mécanique et géométrique des milieux pulvérulents, Proc. 4th ICSMFE, London, (1957).
[4] C.-h. Liu, S.R. Nagel, D. Schecter, S. Coppersmith, S. Majumdar, O. Narayan, T. Witten, Force fluctuations in bead packs, Science, 269 (1995) 513-515.
[5] A. Drescher, G.D.J. De Jong, Photoelastic verification of a mechanical model for the flow of a granular material, Soil Mechanics and Transport in Porous Media: Selected Works of G. de Josselin de Jong, (2006) 28-43.
[6] D. Howell, R.P. Behringer, C. Veje, Stress fluctuations in a 2D granular Couette experiment: A continuous transition, Physical Review Letters, 82 (1999) 5241.
[7] O. Gendelman, Y.G. Pollack, I. Procaccia, S. Sengupta, J. Zylberg, What determines the static force chains in stressed granular media?, Physical review letters, 116 (2016) 078001.

[8] C. Liang, J.A. Wang, M.M. Wang, X.G. Han, Identification and Analysis of Force Chain Structures in Photoelastic Experiments, in: X. Li, Y. Feng, G. Mustoe (Eds.) Proceedings of the 7th International Conference on Discrete Element Methods, Springer Singapore, Singapore, 2017, pp. 1441-1445.
[9] R.P. Behringer, B. Chakraborty, The physics of jamming for granular materials: a review, Reports on Progress in Physics, 82 (2018) 012601.
[10] D. Wang, J. Ren, J.A. Dijksman, H. Zheng, R.P. Behringer, Microscopic origins of shear jamming for 2d frictional grains, Physical review letters, 120 (2018) 208004.
[11] D. Fischer, R. Stannarius, K. Tell, P. Yu, M. Sperl, Force chains in crystalline and frustrated packing visualized by stress-birefringent spheres, Soft Matter, 17 (2021) 4317-4327.
[12] F. Radjai, M. Jean, J.-J. Moreau, S. Roux, Force Distributions in Dense Two-Dimensional Granular Systems, Physical Review Letters, 77 (1996) 274-277.
[13] F. Radjai, D.E. Wolf, M. Jean, J.-J. Moreau, Bimodal character of stress transmission in granular packings, Physical review letters, 80 (1998) 61.
[14] D.M. Mueth, H.M. Jaeger, S.R. Nagel, Force distribution in a granular medium, Physical Review E, 57 (1998) 3164.
[15] M. Cates, J. Wittmer, J.-P. Bouchaud, P. Claudin, Jamming, force chains, and fragile matter, Physical review letters, 81 (1998) 1841.
[16] X. An, R. Yang, K. Dong, A. Yu, DEM study of crystallization of monosized spheres under mechanical vibrations, Computer Physics Communications, 182 (2011) 1989-1994.
[17] F. Radjai, S. Roux, J.J. Moreau, Contact forces in a granular packing, Chaos: An Interdisciplinary Journal of Nonlinear Science, 9 (1999) 544-550.
[18] H.A. Makse, D.L. Johnson, L.M. Schwartz, Packing of compressible granular materials, Physical review letters, 84 (2000) 4160.
[19] N. Lačević, S.C. Glotzer, Dynamical heterogeneity and jamming in glass-forming liquids, The Journal of Physical Chemistry B, 108 (2004) 19623-19633.
[20] S. Luding, Stress distribution in static two-dimensional granular model media in the absence of friction, Physical Review E, 55 (1997) 4720.
[21] R. Behringer, D. Howell, L. Kondic, S. Tennakoon, C. Veje, Predictability and granular materials, Physica D: Nonlinear Phenomena, 133 (1999) 1-17.
[22] J. Peters, M. Muthuswamy, J. Wibowo, A. Tordesillas, Characterization of force chains in granular material, Physical Review E, 72 (2005) 041307.
[23] D.S. Bassett, E.T. Owens, M.A. Porter, M.L. Manning, K.E. Daniels, Extraction of force-chain network architecture in granular materials using community detection, Soft Matter, 11 (2015) 2731-2744.
[24] F. Chen, Q. Zhuang, H. Zhang, Mechanical analysis and force chain determination in granular materials using digital image correlation, Appl. Opt., 55 (2016) 4776-4783.
[25] R. Mandal, C. Casert, P. Sollich, Robust prediction of force chains in jammed solids using graph neural networks, Nature Communications, 13 (2022) 4424.
[26] Z.-M. Shi, H.-C. Zheng, S.-B. Yu, M. Peng, T. Jiang, Application of CFD-DEM to investigate seepage characteristics of landslide dam materials, Computers and Geotechnics, 101 (2018) 23-33.
[27] S. Haeri, Optimisation of blade type spreaders for powder bed preparation in Additive Manufacturing using DEM simulations, Powder Technology, 321 (2017) 94-104.
[28] S. Haeri, Y. Wang, O. Ghita, J. Sun, Discrete element simulation and experimental study of powder spreading process in additive manufacturing, Powder Technology, 306 (2017) 45-54.
[29] S. Khajepor, O. Ejtehadi, S. Haeri, The effects of interstitial inert gas on the spreading of Inconel 718 in powder bed fusion, Additive Manufacturing, (2023) 103737.
[30] S. Sarkar, R. Mukherjee, B. Chaudhuri, On the role of forces governing particulate interactions in pharmaceutical systems: A review, International Journal of Pharmaceutics, 526 (2017) 516-537.
[31] S. Jiang, X. Li, L. Zhang, Y. Tan, R. Peng, R. Chen, Discrete element simulation of SiC ceramic containing a single pre-existing flaw under uniaxial compression, Ceramics International, 44 (2018) 3261-3276.
[32] C. Kloss, C. Goniva, A. Hager, S. Amberger, S. Pirker, Models, algorithms and validation for opensource DEM and CFD–DEM, Progress in Computational Fluid Dynamics, an International Journal, 12 (2012) 140-152.

[33] G. Guennebaud, B. Jacob, Eigen, URl: http://eigen. tuxfamily. org, 3 (2010).
[34] S.-C. CD-adapco, User's Guide-Version 9.04. 011, CD-adapco, 2014.
[35] D. Solutions, EDEM user guide, Edinburgh, UK, (2018).
[36] J. Kozicki, F.V. Donze, YADE-OPEN DEM: An open-source software using a discrete element method to simulate granular material, Engineering Computations, (2009).
[37] D.K. Weatherley, V.E. Boros, W.R. Hancock, S. Abe, Scaling benchmark of esys-particle for elastic wave propagation simulations, 2010 IEEE Sixth International Conference on e-Science, IEEE, 2010, pp. 277-283.
[38] N. Govender, D.N. Wilke, S. Kok, Blaze-DEMGPU: Modular high performance DEM framework for the GPU architecture, SoftwareX, 5 (2016) 62-66.
[39] M. Dosta, V. Skorych, MUSEN: An open-source framework for GPU-accelerated DEM simulations, SoftwareX, 12 (2020) 100618.
[40] H.R. Norouzi, PhasicFlow: A parallel, multi-architecture open-source code for DEM simulations, Computer Physics Communications, 291 (2023) 108821.