

# Shear Viscosity of Liquid-Phase Yukawa Plasmas from Molecular Dynamics Simulations on Graphics Processing Units

Á. Budea, A. Derzsi, P. Hartmann, and Z. Donkó\*

Research Institute for Solid State Physics and Optics of the Hungarian Academy of Sciences, P.O.B. 49, H-1525 Budapest, Hungary

Received 14 October 2011, revised 19 January 2012, accepted 26 January 2012

Published online 11 April 2012

**Key words** Dusty plasma, Yukawa liquid, shear viscosity, molecular dynamics.

We have carried out million-particle equilibrium molecular dynamics simulations of 3-dimensional Yukawa liquids in order to determine the shear viscosity coefficient. The computations have been executed on Graphics Processing Unit (GPU) architectures with our largely parallelized code. The results cover the strongly coupled liquid phase, with  $\Gamma$  up to the vicinity of the freezing transition, for the  $1 \leq \kappa \leq 3$  domain of the screening parameter of the Yukawa potential. The good agreement of the present results with those obtained from earlier simulations of significantly smaller systems (consisting of several hundred to several thousand particles) verifies that the viscosity data derived in these smaller scale simulations are also acceptable.

© 2012 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

## 1 Introduction

The shear viscosity and, generally, shear flows [1-4] have been attracting considerable attention. As regards to strongly coupled dusty plasmas [5-7], in the first experiments a sheared velocity profile was created around a laser beam [8, 9]. In the studies reported in [10] two displaced, parallel counter-propagating laser beams were used to realize a planar Couette configuration in a 2D dusty plasma layer. In another experiment the non-Newtonian behavior of a 3D complex plasma in the liquid state was identified [11]. More recently, detailed dusty plasma experiments demonstrated the presence of viscoelastic response [12] and revealed the wave-number dependence of the viscosity in 2D [13]. Experiments in the crystalline phase have identified the slipping of individual crystal lines to be the primary mechanism for relaxing an applied shear stress [14]. Stationary flows at high shear rates, and the complex viscosity, derived from the effect of a periodic shear, were studied in [15].

Calculation of the shear viscosity coefficient  $\eta$  has been the topic of several works [16-21] both for three-dimensional (3D) and two-dimensional (2D) Yukawa liquids. In [22], besides calculations of the “equilibrium” (small shear rate) static viscosity, predictions for the shear-thinning effect (typical for complex molecular liquids) were given at high shear rates. The frequency dependence of the complex shear viscosity, which combines the dissipative and the elastic components of the complex response of matter to oscillating shear stress was computed for 3D Yukawa liquids in [23].

Regarding the dependence of the static viscosity on the coupling parameter,  $\eta(\Gamma)$ , all previous studies give, in agreement, a “U-shape” curve, explained by the prevailing kinetic contribution to the momentum transport at low coupling and a prevailing potential contribution to the momentum transport at high coupling (e.g. [17]). The coupling value, where the minimum of the viscosity occurs, has been found to increase with increasing  $\kappa$ . A recent analysis [20] of the previous simulation results has found that differences as large as a few times 10% exist between the results of different authors. As the Yukawa system [24] is also used as a reference for calculations of viscosity of systems like liquid metals and warm dense matter through mapping of these systems with the Yukawa model [25], verification of the available data and obtaining more accurate values for  $\eta$  are clearly important. All the above mentioned simulation studies concerned systems of several hundred to several thousand particles, which, especially in 3D raises the question whether the results approximate well the properties of “large” systems. Thus simulations on larger systems are definitely needed, and indeed, this is the aim of our present investigations.

\* Corresponding author. E-mail: donko@mail.kfki.hu, Phone: +36 1 392 2222, Fax: +36 1 392 2215

We report viscosity data obtained from million-particle simulations carried out on Graphics Processing Units (GPU) [26-30], which are becoming widespread for scientific computations these days.

We model the dusty plasma system as a one-component plasma with a polarizable background. The pair interaction potential energy is given as

$$\phi(r) = \frac{Q^2}{4\pi\epsilon_0} \frac{\exp(-r/\lambda_D)}{r}, \quad (1)$$

where  $Q$  is the particle charge and  $\lambda_D$  is the Debye length. The system of interest here is fully characterized by the *coupling parameter*  $\Gamma = Q^2/4\pi\epsilon_0 a k_B T$  and the *screening parameter*  $\kappa = a/\lambda_D$ , where  $a = (3/4\pi n)^{1/3}$  is the Wigner-Seitz (WS) radius,  $n$  is the particle number density, and  $T$  is the temperature (e.g. [24]). Our system is idealized in a way that dynamical effects of the plasma environment (Langevin dynamics) are neglected.

## 2 Molecular dynamics on graphics cards

We have developed a molecular dynamics (MD) simulation code for the NVIDIA Compute Unified Device Architecture (CUDA) [31] that allows massive parallel computing, thereby permitting relatively large systems to be simulated on PC class computers, compared to the traditional Central Processing Unit (CPU) computations. We adapt a classical molecular dynamics simulation code using periodic boundary conditions and finite range interactions (introducing a cutoff radius in the force calculation), to CUDA environment, in order to derive the shear viscosity ( $\eta$ ) of 3D Yukawa liquids via simulating a million-particle system. From the computational methods available for the calculation of  $\eta$  (see e.g. [20]) we chose the equilibrium molecular dynamics approach, because of its fundamental nature.

The most time consuming part of the MD simulations is the calculation and summation of interparticle forces (needed in the integration of the equations of motion), thus designing this part of the code requires careful optimization to achieve a good performance. Our implementation relies on the method described in [32], and utilizes fast, on-chip-cached constant memory to calculate the distances of particle pairs. Constant memory is limited to 64 KB on current graphics cards, and provides optimal performance if multiple threads access the same data at the same time. The simulation cell is divided into subcells, similarly to the chaining mesh approach [33], but in this case particle data are sorted based on the identifier of the subcell it belongs to, therefore particles in the same subcell occupy contiguous global memory blocks, which is required for efficient GPU memory operations. If constant memory is filled with particle positions, and each of the thousands of concurrent computation threads add up forces acting upon a single particle, multiple threads from adjacent subcells can process the same particle concurrently, when forming particle pairs to progress with calculations. Due to the strictly limited size of constant memory (it can only hold positions of  $\approx 2700$  particles, considering that double precision data type has to be used for sufficient precision), all the particles can only be covered in multiple blocks that are first copied to constant memory, and then processed by the specific kernel.

To reduce arithmetic operations, the arithmetically intensive calculation of forces over particle pair distances has been optimized by eliminating redundant floating point operations. Division and square root function are not native operations, but can be translated into composition of different instructions: division consists of reciprocal and multiplication, while square root consists of reciprocal-square-root and reciprocal [34]. Due to the way these operations occur in the original algorithm, after decomposition, several costly reciprocal calculations become redundant, and can be omitted, while other reciprocals involving parameter  $\lambda_D$  can be precalculated. In cases specifically requiring distance of particle pairs (now substituted by its reciprocal), it comes as the result of a single multiplication of the original squared, and previously calculated reciprocal-square-rooted values.

Apart from the force calculation, most tasks of the simulation can be traced back to parallel primitive building blocks, for example reduction is used both in kinetic energy computation, or in calculating stress autocorrelation function components (see later). Conveniently, the CUDA-specific Thrust library [35] provides such algorithms exposed through high level interfaces (including a fast radix sort), similarly to the C++ standard template library, therefore both hiding unnecessary implementation details, and accelerating the development process.

To obtain the shear viscosity we first calculate the off-diagonal element of the pressure tensor:

$$P^{xy} = \sum_{i=1}^N \left[ mv_{ix}v_{iy} - \frac{1}{2} \sum_{j \neq i}^N \frac{x_{ij}y_{ij}}{r_{ij}} \frac{\partial \phi(r_{ij})}{\partial r_{ij}} \right], \quad (2)$$

where  $N$  is the number of particles,  $r_{ij} = |r_{ij}| = |r_i r_j| = |(x_{ij}, y_{ij})|$ , and  $m$  is the particle mass. The shear viscosity coefficient is determined from the Green-Kubo integral [36]:

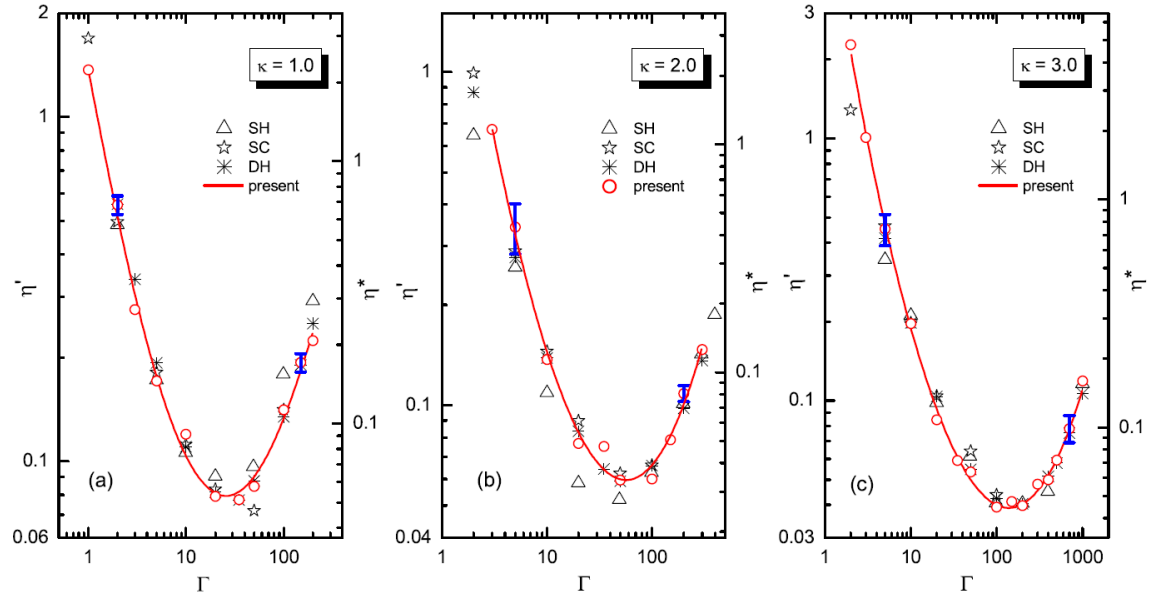
$$\eta = \frac{1}{VKT} \int_0^\infty c_\eta(t) dt, \quad (3)$$

where  $C_\eta(t) = \langle P^{xy}(t)P^{xy}(0) \rangle$  is the stress autocorrelation function and  $V$  is the volume of the system.

The simulation uses periodic boundary conditions. The symmetry of the system in the  $x$ ,  $y$ , and  $z$  directions allows improving the signal-to-noise ratio of the measured quantities: we compute  $C_\eta(t)$  for the other off-diagonal elements ( $P^{yz}$  and  $P^{zx}$ ) as well, and average the resulting  $\eta$  values. The upper limit of integration in equation (3) is replaced by the time when the noisy  $C_\eta(t)$  first crosses zero.

### 3 Results

The results presented here have been obtained from simulations of systems comprising  $N = 10^6$  particles. We used GeForce GTX 590 and Tesla C2050 GPUs for our computations.



**Fig. 1** (a) Shear viscosity of the 3D Yukawa liquid at (a)  $\kappa = 1$ , (b)  $\kappa = 2$ , and (c)  $\kappa = 3$ . SH: Saigo and Hamaguchi [17], SC: Salin and Caillol [18], DH: Donkó and Hartmann [20]. Left scale: viscosity values normalized by the plasma frequency ( $\eta'$ ), right scale: viscosity values normalized by the Einstein frequency ( $\eta^*$ ). The error bars shown for selected ( $\Gamma$ ,  $\kappa$ ) pairs correspond to the standard deviation ( $\sigma$ ) of the results derived from four independent simulations.

The simulation time covers  $\omega_p t \sim 500 - 5000$ , where  $\omega_p = \sqrt{nQ^2/\epsilon_0 m}$  is the plasma frequency. Fig. 1 shows our simulation results for the shear viscosity  $\phi$ , together with those obtained in previous studies. We use normalization both by the plasma frequency and by the Einstein frequency  $\omega_E$ :

$$\eta' = \frac{\eta}{mn\omega_p a^2} \quad \text{and} \quad \eta^* = \frac{\eta}{mn\sqrt{3}\omega_E a^2}. \quad (4)$$

The use of the Einstein frequency (the oscillation frequency of a test charge in the frozen environment of the particles of the system) has been pointed out to be more appropriate for Yukawa systems, as  $\omega_E$  depends on the screening parameter  $\kappa$  (see e.g. [37]). The values of  $\omega_E$  are taken from [38].

Fig. 1 also displays shear viscosity values obtained in previous studies of Saigo and Hamaguchi [17], Salin and Caillol [18], and Donkó and Hartmann [20]. In [17] and [18] equilibrium MD simulations were used to compute the shear viscosity, while in [20] two different nonequilibrium techniques were applied. In these studies the number of simulation particles was  $N = 250 - 1000$  (typically 250) [17],  $N = 128 - 864$  (typically 500) [18], and  $N = 1000 - 64,000$  (typically 8000) [20]. The number of particles,  $N = 10^6$ , is significantly higher in the present runs. The good agreement between the present data and those given in previous papers, as seen in Fig. 1, confirms that the viscosity data obtained from smaller-scale simulations can be considered acceptable, which is the main conclusion of our work. To test the accuracy of our computations we have carried out four independent simulation runs for selected  $(\Gamma, \kappa)$  pairs. Standard deviations below  $\approx 15\%$  were found for all cases.

Our computations have been limited to the  $\Gamma \geq 1$  range of the coupling parameter. We have also attempted to explore the  $\Gamma < 1$  domain, however, here we have found that the autocorrelation function  $C_\eta(t)$  decays progressively slower, thus much longer simulation times are unavoidable to access this domain of coupling. Extended simulation time would of course help improving the accuracy of the viscosity values over the whole  $\Gamma$  domain.

## 4 Conclusions

In conclusion, we have demonstrated by million-particle Molecular Dynamics simulations the effectiveness of Graphics Processing Units for computing physical properties of strongly interacting many-particle systems. The shear viscosity data derived from our computations have generally been in good agreement with the results of previous simulation studies, which used orders of magnitude smaller systems. Our studies confirm that simulations of relatively small systems already yield meaningful results.

**Acknowledgements** This work has been supported by the HELIOS project (ELI 09-1-2010-0010) and by the Hungarian Fund for Scientific Research (grants OTKA-K-77653 and OTKA-PD-75113). We thank L. Gránágy, T. Pusztai, Gy. Tóth, G. Debreczeni, and G. Barnaföldi for providing access to their Tesla C2050-based GPU systems for tests and simulation runs.

## References

- [1] H.M. Wiechen, Phys. Plasmas **13**, 062104 (2006).
- [2] A. Piel, V. Nosenko and J. Goree, Phys. Plasmas **13**, 042104 (2006).
- [3] J. Ashwin and R. Ganesh, Phys. Plasmas **17**, 03706 (2010).
- [4] R. Heidemann, S. Zhdanov, K.R. Stütterlin, H.M. Thomas, and G.E. Morfill, EPL **96**, 15001 (2011).
- [5] D.A. Mendis and M. Rosenberg, Ann. Rev. Astronomy and Astrophysics **32**, 419 (1994).
- [6] P.K. Shukla and B. Eliasson, Rev. Mod. Phys. **81**, 25 (2009).
- [7] M. Bonitz, C. Henning, and D. Block, Rep. Prog. Phys. **73**, 066501 (2010).
- [8] W.-T. Juan, M.-H. Chen, and I. Lin, Phys. Rev. E **64**, 016402 (2001).
- [9] A. Gavrikov, I. Shakhova, A. Ivanov, O. Petrov, N. Vorona, and V. Fortov, Phys. Lett. A **336**, 378 (2005).
- [10] V. Nosenko and J. Goree, Phys. Rev. Lett. **93**, 155004 (2004).
- [11] A.V. Ivlev, V. Steinberg, R. Kompaneets, H. Höfner, I. Sidorenko, and G. E. Morfill, Phys. Rev. Lett. **98**, 145003 (2007).
- [12] C.-L. Chan and Lin I, Phys. Rev. Lett. **98**, 105002 (2007).
- [13] Y. Feng, J. Goree, and B. Liu, Phys. Rev. Lett. **105**, 025002 (2010).
- [14] C. Durniak and D. Samsonov, Phys. Rev. Lett. **106**, 175001 (2011).
- [15] P. Hartmann, M.Cs. Sándor, A. Kovács, and Z. Donkó, Phys. Rev. E **84**, 016404 (2011).
- [16] K.Y. Sanbonmatsu and M.S. Murillo, Phys. Rev. Lett. **86**, 1215 (2001).
- [17] T. Saigo and S. Hamaguchi, Phys. Plasmas **9**, 1210 (2002).
- [18] G. Salin and J.-M. Caillol, Phys. Rev. Lett. **88**, 065002 (2002); Phys. Plasmas **10**, 1220 (2003).
- [19] O.S. Vaulina and I.E. Dranzhevskii, Plasma Phys. Reports **33**, 494 (2007).
- [20] Z. Donkó and P. Hartmann, Phys. Rev. E **78**, 026408 (2008).
- [21] T.S. Ramazanov and K.N. Dzhumagulova, Contrib. Plasma Phys. **48**, 357 (2008).
- [22] Z. Donkó, J. Goree, P. Hartmann, and K. Kutasi, Phys. Rev. Lett. **96**, 145003 (2006).
- [23] Z. Donkó, J. Goree, and P. Hartmann, Phys. Rev. E **81**, 056404 (2010).
- [24] Z. Donkó, G.J. Kalman, and P. Hartmann, J. Phys. Condensed Matter **20**, 413101 (2008).

- [25] M.S. Murillo, *High Energy Density Phys.* **4**, 49 (2008).
- [26] P. Mertmann, D. Eremin, T. Mussenbrock, R. P. Brinkmann, and P. Awakowicz, *Comp. Phys. Commun.* **182**, 2161 (2011).
- [27] H. Bura, R. Widera, W. Hnig, G. Juckeland, A. Debus, T. Kluge, U. Schramm, T.E. Cowan, R. Sauerbrey, and M. Bussmann, *IEEE Trans. Plasma Science* **38**, 2831 (2010).
- [28] T. Reichstein and A. Piel, *Phys. Plasmas* **18**, 083705 (2011).
- [29] I.V. Morozov, A.M. Kazennov, R.G. Bystryi, G.E. Norman, V.V. Pisarev, and V.V. Stegailov, *Computer Phys. Commun.* **182**, 1974 (2011).
- [30] N. Hanzlikova and M.M. Turner, *Proc. 30th Int. Conf. Phenomena in Ionized Gases*, B5-175 Belfast, UK, 28 August - 2 September 2011.
- [31] <http://developer.nvidia.com/cuda-downloads>
- [32] B.G. Levine, J.E. Stone, A. Kohlmeyer, *Journal of Comp. Phys.* **230**, 3556 (2011).
- [33] D. Frenkel, B. Smit, *Understanding Molecular Dynamics Simulation* (Academic Press, 2002).
- [34] *CUDA C Programming Guide*
- [35] <http://code.google.com/p/thrust/>
- [36] J.-P. Hansen and I. R. McDonald, *Theory of simple liquids* (Academic Press, New York, 1976).
- [37] P. Bakshi, Z. Donkó, and G.J. Kalman, *Contrib. Plasma Phys.* **43**, 261 (2003).
- [38] H. Ohta and S. Hamaguchi, *Phys. Plasmas* **7**, 4506 (2000).