Chapter 12: chiral symmetry on a lattice

Chiral symmetry in the continuum

For the massless Euclidean continuum Dirac operator

$$D = \gamma_\mu D_\mu \tag{1}$$

we have

$$D\gamma_5 + \gamma_5 D = 0 \tag{2}$$

and therefore

$$\psi \to \psi' = \exp(i\alpha\gamma_5)\psi\,, \tag{3}$$
$$\bar\psi \to \bar\psi' = \bar\psi\exp(i\alpha\gamma_5) \tag{4}$$

transforms the fermion Lagrangian

$$\bar\psi D\psi \tag{5}$$

to

$$\bar\psi\exp(i\alpha\gamma_5)D\exp(i\alpha\gamma_5)\psi = \bar\psi\exp(i\alpha\gamma_5)\exp(-i\alpha\gamma_5)D\psi = \bar\psi D\psi\,. \tag{6}$$

Therefore the transformation leaves the action invariant.

If we added a mass term to the action it would break this symmetry. We therefore also call the $m \to 0$ limit the **chiral limit** of the theory.

Consider now a theory with N_f quark flavors described by Lagrangian

$$\sum_f \bar\psi_f(D+m_f)\psi_f\,, \tag{7}$$

which we can write simply as

$$\bar\psi(D+M)\psi \tag{8}$$

by considering a mass matrix M and the fields ψ to live in internal spin, color, and flavor space.

We can then consider unitary rotations in flavor space

$$\psi \to \psi' = U_V\psi = e^{i\alpha_i T_i}\psi\,, \tag{9}$$
$$\bar\psi \to \bar\psi' = \bar\psi U_V^\dagger = \bar\psi e^{-i\alpha_i T_i} \tag{10}$$

with $U_V \in \mathrm{U}(N_f)$ and Hermitian generators T_i. In the massless case, this is a symmetry of the Lagrangian. In the massive case, a symmetry can be preserved if some masses are degenerate since then a sub-space of the algebra commutes with M.

In addition, chiral versions of this transformation also leave the massless theory invariant, i.e.,

$$\psi \to \psi' = U_A\psi = e^{i\alpha_i T_i\gamma_5}\psi\,, \tag{11}$$
$$\bar\psi \to \bar\psi' = \psi^\dagger U_A^\dagger\gamma_3 = \bar\psi e^{i\alpha_i T_i\gamma_5}\,. \tag{12}$$

The algebra of this transformation of ψ is just a direct product of γ_5 and the generators of U(N_f) such that the symmetry group is written as $U_A(N_f)$.

The total flavor symmetry group for N_f massless quarks is then

$$U_V(N_f)\otimes U_A(N_f) \tag{13}$$

where the V and A stand for **vector** and **axial vector** symmetry groups.

This can be re-written as

$$U_V(N_f)\otimes U_A(N_f) = U_V(1)\otimes U_A(1)\otimes SU_V(N_f)\otimes SU_A(N_f) \tag{14}$$

since the global $U(1)$ rotations require special attention. The $U_V(1)$ is responsible for the charge conservation as we will demonstrate below. The $U_A(1)$ is broken by the integration measure, i.e., it is a symmetry of the action but not of the quantized theory. We also say the axial symmetry is **anomalous** and refer to it as the **axial anomaly** of QCD. We demonstrate this explicitly below.

While the mass terms break the $SU_V(N_f)$ and $SU_A(N_f)$ symmetries in general, in reality, the **up** and **down** quarks have almost the same mass (≈ 5 MeV), leaving an approximate $SU_V(2)$ flavor symmetry of the theory. To a much poorer degree, the three light quark flavors **up**, **down**, and **strange** (≈ 100 MeV) are also approximately degenerate yielding a $SU_V(3)$ flavor symmetry of the theory. This has to be understood with respect to typical QCD scales of several hundred MeV. Because of this QCD with realistic values for the quark masses has an approximate flavor SU(2) symmetry.

It turns out, however, that $SU_A(2)$ is spontaneously broken, i.e., the vacuum state does not respect the symmetry. The corresponding **order parameter** is the **chiral condensate**

$$\langle\bar\psi\psi\rangle\,. \tag{15}$$

By **Goldstone's theorem** we therefore expect massless **Nambu-Goldstone** (NG) bosons to exist in the theory. Since the SU(2) flavor symmetry is only approximate, these bosons are also not exactly massless but only very light and are more appropriately referred to as **pseudo** NG bosons. For SU(2) flavor, these are the three light pion states ($\pi^\pm, \pi^0$). These states have mass of ≈ 140 MeV and are therefore indeed much lighter than other mesons or the proton/neutron at ≈ 1000 MeV.

Finally, we remark that by promoting the mass matrix M in the above equation to a field that transforms non-trivially under flavor rotations, one can then construct an effective theory that captures the leading degree of chiral symmetry breaking. This theory is called **chiral perturbation theory** and in the SU(2) case is a theory of pions. The details of this construction are beyond the scope of this chapter.

Ginsparg-Wilson relation on the lattice

Nielsen-Ninomiya theorem: Cannot have continuum chiral symmetry on the lattice and at the same time avoid fermion doublers.

Ginsparg-Wilson: By studying renormalization group transformations find that the Ginsparg-Wilson (GW) relation

$$D\gamma_5 + \gamma_5 D = aD\gamma_5 D \tag{16}$$

is instead consistent with doubler-removal and allows to define an exact chiral symmetry also at finite lattice spacing a through

$$\psi \to \psi' = \exp\Big(i\alpha\gamma_5\left(1-\frac{a}{2}D\right)\Big)\psi\,, \tag{17}$$
$$\bar\psi \to \bar\psi' = \bar\psi\exp\Big(i\alpha\left(1-\frac{a}{2}D\right)\gamma_5\Big)\,. \tag{18}$$

Homework: Show that this transformation is a symmetry of the massless Lagrangian $\bar\psi D\psi$ at non-zero a if D satisfies the Ginsparg-Wilson relation.

While Wilson fermions do not satisfy this symmetry at finite a, we will study actions in this chapter that do.

The spectrum of the Dirac operator

We now study the right-eigenvalues λ of D, i.e., solutions to

$$Dv_\lambda = \lambda v_\lambda \tag{19}$$

with $v_\lambda^\dagger v_\lambda = 1$ without loss of generality.

If $(\gamma_5 D)^\dagger = \gamma_5 D$, then the characteristic polynomial

$$\begin{aligned} P(\lambda) &= \det(D-\lambda 1) = \det(\gamma_5(D-\lambda 1)\gamma_5) = \det(D^\dagger - \lambda 1) \qquad (20)\\ &= \det(D-\lambda^* 1)^* = P(\lambda^*)^*\,. \qquad (21)\end{aligned}$$

Therefore γ_5-Hermiticity implies that if λ is an eigenvalue $(P(\lambda)=0)$, then so is λ^*. So eigenvalues come in complex pairs.

Additionally, we find

$$\lambda v_\lambda^\dagger\gamma_5 v_\lambda = v_\lambda^\dagger\gamma_5 D v_\lambda = v_\lambda^\dagger D^\dagger\gamma_5 v_\lambda = (Dv_\lambda)^\dagger\gamma_5 v_\lambda = \lambda^* v_\lambda^\dagger\gamma_5 v_\lambda\,. \tag{22}$$

This can be re-expressed as

$$\mathrm{Im}(\lambda)v_\lambda^\dagger\gamma_5 v_\lambda = 0\,. \tag{23}$$

Therefore only real eigenvalues can have $v_\lambda^\dagger\gamma_5 v_\lambda \neq 0$ and all imaginary eigenvalues have $v_\lambda^\dagger\gamma_5 v_\lambda = 0$.

Using γ_5-Hermiticity we can re-write the GW relation as

$$D + D^\dagger = aD^\dagger D \tag{24}$$

or alternatively

$$D^\dagger + D = aDD^\dagger\,. \tag{25}$$

From this it follows that for such operators $[D, D^\dagger] = 0$, i.e., the operator is normal and its eigenvectors are orthogonal.

In addition it follows that

$$\begin{aligned} (\lambda+\lambda^*) &= (\lambda+\lambda^*)v_\lambda^\dagger v_\lambda = v_\lambda^\dagger(D+D^\dagger)v_\lambda = v_\lambda^\dagger aD^\dagger D v_\lambda \qquad (26)\\ &= a\lambda\lambda^*\,. \qquad (27)\end{aligned}$$

If we write this explicitly in terms of real and imaginary parts $\lambda = x + iy$, we find

$$2x = a(x^2+y^2) \tag{28}$$

or

$$\left(x-\frac{1}{a}\right)^2 + y^2 = \frac{1}{a^2}\,. \tag{29}$$

The eigenvalues therefore live on a circle with center $1/a$ and radius $1/a$ in the complex plane. We refer to this as the **Ginsparg-Wilson** circle.

This circle touches the real axis on two points, for $x=0$ and $x=2/a$. In both cases $v_\lambda^\dagger\gamma_5 v_\lambda \neq 0$ is possible.

One conveniently parametrizes

$$\lambda = \frac{1}{a}(1-e^{i\phi}) \tag{30}$$

such that

$$\lambda^{-1} = \frac{a}{2} + i\frac{a}{2}\frac{\sin(\phi)}{1-\cos(\phi)}\,. \tag{31}$$

The eigenvalues of D^{-1} therefore are all on a line parallel to the imaginary axis with distance $a/2$.

Finally, we observe that for the **zero modes**

$$Dv_0 = 0 \tag{32}$$

we have

$$\gamma_5 D v_0 = 0\,, \tag{33}$$
$$D\gamma_5 v_0 = (-\gamma_5 D + aD\gamma_5 D)v_0 = 0\,, \tag{34}$$

such that

$$[\gamma_5, D]v_0 = 0$$

and therefore v_0 are also eigenstates of γ_5 with

$$\gamma_5 v_0 = \pm v_0 \tag{35}$$

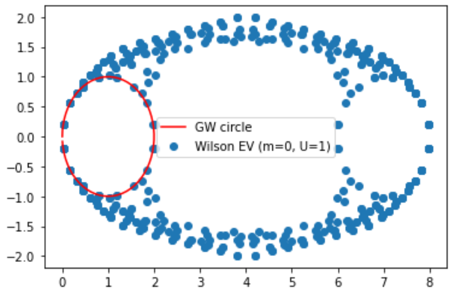
since $\gamma_5^2 = 1$. We call the $+1$ eigenvectors ($\gamma_5 v_0 = v_0$) **right-handed** and the -1 eigenvectors ($\gamma_5 v_0 = -v_0$) **left-handed**.

In the following, we take a look at the eigenvalues of the free Wilson Dirac operator.

In [71]:
```python
import gpt as g
import matplotlib.pyplot as plt
import numpy as np

grid = g.grid([8,8,8,16], g.double)

U = g.qcd.gauge.unit(grid)

w = g.qcd.fermion.wilson_clover(U, mass=0.0, csw_r=0.0, csw_t=0.0, nu=1.0, xi_0=1.0,
                                isAnisotropic=False,
                                boundary_phases=[1,1,1,-1])

start = g.vspincolor(w.F_grid)
rng = g.random("evecs")
rng.normal(start)

g.default.set_verbose("arnoldi", False)
a = g.algorithms.eigen.arnoldi(Nmin=700, Nmax=1400, Nstep=100, Nstop=500, resid=1e-6, implicit_restart=True)

evec, evals = a(w, start)
```

```
GPT :  828077.260959 s : Initializing gpt.random(evecs,vectorized_ranlux24_389_64) took 0.00035882 s
```

In [72]:
```python
fig, ax = plt.subplots()

plt.xlim()
ax.scatter([e.real for e in evals], [e.imag for e in evals], label="Wilson EV (m=0, U=1)")

ax.plot([1 - np.cos(phi) for phi in np.arange(0,2*np.pi,0.1)],
           [np.sin(phi) for phi in np.arange(0,2*np.pi,0.1)], c="red", label="GW circle")

plt.legend()
plt.show()
```



The Wilson operator has eigenvalues beyond the GW circle. The corresponding eigenmodes are associated with chiral symmetry breaking, which can only be restored in the continuum for Wilson quarks. We discuss lattice fermions that explicitly preserve chiral symmetry in the next lecture.

Homework: repeat this study for one of the quenched gauge configurations that we previously generated.

Topological charge

The GW circle also allows for heavy real modes with

$$Dv_0 = \frac{2}{a}v_0 \tag{36}$$

for which we also have

$$\gamma_5 D v_0 = \frac{2}{a}\gamma_5 v_0\,, \tag{37}$$

$$D\gamma_5 v_0 = (2D\gamma_5 - D\gamma_5)v_0 = (2D\gamma_5 + \gamma_5 D - aD\gamma_5 D)v_0 = (2D\gamma_5 + \gamma_5\frac{2}{a} - 2D\gamma_5)v_0 = \frac{2}{a}\gamma_5 v_0\,, \tag{38}$$

such that again

$$[D,\gamma_5]v_0 = 0$$

and therefore v_0 are also eigenstates of γ_5 with

$$\gamma_5 v_0 = \pm v_0\,. \tag{39}$$

We therefore have both light and heavy chiral modes with heavy modes decoupling in the $a \to 0$ limit. Interestingly, we can define the **topological charge**

$$Q = \frac{a}{2}\mathrm{Tr}[\gamma_5 D] = \sum_n \frac{a}{2} v_{\lambda_n}^\dagger \gamma_5 D v_{\lambda_n} \tag{40}$$

$$= \sum_{\lambda_n=\frac{2}{a}} v_{\lambda_n}^\dagger \gamma_5 v_{\lambda_n} = n'_+ - n'_- \tag{41}$$

with $n'_\pm$ counting the number of right (+) and left (-) chiral modes with $\lambda = \frac{2}{a}$. We used that D is normal and that therefore the eigenvectors v_λ form a complete basis. Since $\mathrm{Tr}\gamma_5 = 0$, we can also write

$$Q = \frac{a}{2}\mathrm{Tr}[\gamma_5 D] = \frac{a}{2}\mathrm{Tr}[\gamma_5(D-2/a)] \tag{42}$$

$$= -\sum_{\lambda_n=0} v_{\lambda_n}^\dagger \gamma_5 v_{\lambda_n} = n_- - n_+ \tag{43}$$

with $n_\pm$ counting the number of right (+) and left (-) chiral modes with $\lambda = 0$. We find that $n_- - n_+ = n'_+ - n'_-$.

We can also define the **topological charge density** $q(x)$ via

$$Q = a^4 \sum_x q(x) \tag{44}$$

with

$$q(x) = \frac{1}{2a^3}\mathrm{Tr}_{\mathrm{spin,color}}[\gamma_5 D_{x,x}]\,. \tag{45}$$

This is a lattice version of the **Atiyah–Singer index theorem**, relating a property of the gauge field to the integer number $Q = n_- - n_+$. In typical gauge configurations, one finds either $n_- \neq 0 \wedge n_+ = 0$ or $n_+ \neq 0 \wedge n_- = 0$. One contribution to Q originates in field configurations describing local minima of the gauge action, called **instantons**. A detailed description is beyond the scope of this chapter.

In the continuum the Atiyah-Singer index theorem states

$$Q^{\mathrm{cont}} = \frac{1}{32\pi^2}\varepsilon_{\mu\nu\rho\sigma}\int dx \mathrm{Tr}_{\mathrm{color}}[F_{\mu\nu}(x)F_{\rho\sigma}(x)]\,. \tag{46}$$

We can also use a lattice definition of the field-strength tensor and compute a similar object on the lattice. At finite lattice spacing, however, topology is not uniquely defined with ambiguities vanishing as $a \to 0$. In practice, one often performs continuous smoothing operations (c.f. Wilson flow) of the gauge fields that due to their continuous nature cannot change the topological charge, however, they can remove high-frequency modes and hasten convergence for $a \to 0$.

Witten-Veneziano formula: Witten and Veneziano showed that the topological susceptibility

$$\chi_Q = \frac{1}{V}\langle Q^2\rangle$$

in quenched QCD is proportional to the squared mass of η' particle in the chiral limit of QCD. This provides an explanation for the relatively large η' mass that we will observe in a subsequent chapter.

Axial anomaly on the lattice

Let us now re-visit the field transformations that define the chiral symmetry on the lattice. If we consider such a transformation also with generator T in flavor space,

$$\psi \to \psi' = \exp\Big(i\alpha\gamma_5 T\left(1-\frac{a}{2}D\right)\Big)\psi\,, \tag{47}$$

$$\bar\psi \to \bar\psi' = \bar\psi \exp\Big(i\alpha\left(1-\frac{a}{2}D\right)\gamma_5 T\Big)\,. \tag{48}$$

For infinitesimal α, the measure then transforms as

$$d\psi d\bar\psi = d\psi' d\bar\psi' \det\left(1+i\alpha\gamma_5 T\left(1-\frac{a}{2}D\right)\right)\det\left(1+i\alpha\left(1-\frac{a}{2}D\right)\gamma_5 T\right) \tag{49}$$

$$= d\psi' d\bar\psi' \left(1+i\alpha\mathrm{Tr}\gamma_5 T\left(1-\frac{a}{2}D\right)\right)\left(1+i\alpha\mathrm{Tr}\left(1-\frac{a}{2}D\right)\gamma_5 T\right) \tag{50}$$

$$= d\psi' d\bar\psi' \left(1+i\alpha\mathrm{Tr}\left(\gamma_5 T\left(1-\frac{a}{2}D\right)+\left(1-\frac{a}{2}D\right)\gamma_5 T\right)\right) \tag{51}$$

$$= d\psi' d\bar\psi' \left(1-i\alpha a\mathrm{Tr}_{\mathrm{flavor}}[T]\mathrm{Tr}_{\mathrm{spin,color,spacetime}}(\gamma_5 D)\right) \tag{52}$$

$$= d\psi' d\bar\psi' \left(1-i\alpha 2\mathrm{Tr}_{\mathrm{flavor}}[T]Q\right)\,. \tag{53}$$

Therefore $SU_A(N_f)$ is a symmetry also of the quantized theory since its generators are traceless. $U(1)_A$ is anomalous, as already anticipated, i.e., it is only a symmetry of the action but not of the full quantized theory. For this reason, only the three pions are light due to being pseudo-NG bosons of $SU_A(2)$ being spontaneously broken. If $U(1)_A$ was not anomalous, one would expect also the η meson to be light. We note that the anomaly is created by non-trivial topological gauge configurations with $Q \neq 0$.

Chiral condensate

As discussed above, the chiral condensate (in this sub-section in continuum notation)

$$\langle\bar\psi\psi\rangle = -\langle \mathrm{Tr} D^{-1}\rangle \tag{54}$$

is an order parameter of chiral symmetry breaking. In the operator language it is already evident that the chiral condensate is real, however, we can also show this explicitly using γ_5-Hermiticity,

$$\langle\bar\psi\psi\rangle = -\langle \mathrm{Tr} D^{-1}\rangle = -\langle \mathrm{Tr}\gamma_5\gamma_5 D^{-1}\rangle = -\langle \mathrm{Tr}\gamma_5 D^{-1}\gamma_5\rangle = -\langle \mathrm{Tr}(D^{-1})^\dagger\rangle = -\langle \mathrm{Tr} D^{-1}\rangle^* = \langle\bar\psi\psi\rangle^* \,. \tag{55}$$

Banks-Casher relation: Banks and Casher showed that by first going to infinite volume and then going to zero quark mass, one finds

$$\Sigma \equiv -\langle\bar\psi\psi\rangle = \pi\rho(0)\,, \tag{56}$$

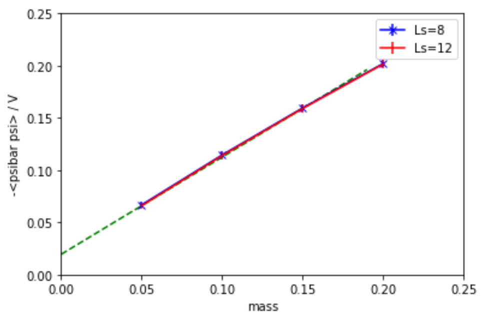
where $\rho(\lambda)$ is the eigenvalue density of D.

As an exercise, we now compute the chiral condensate for domain-wall fermions, a type of GW fermion that we will discuss in detail at the end of this chapter. Domain-wall fermions satisfy the GW relation when taking a parameter $L_s \to \infty$. This limit is approached exponentially in L_s.

In [51]:
```
def chiral_condensate(conf, mass, Ls):
    g.default.push_verbose("io",False)
    U = g.load(conf)
    g.default.pop_verbose()
    grid = U[0].grid

    w = g.qcd.fermion.mobius(U, M5=1.8, mass=mass, Ls=Ls, b=1.5, c=0.5,
                             boundary_phases=[1,1,1,-1])

    inv = g.algorithms.inverter
    pc = g.qcd.fermion.preconditioner
    g.default.set_verbose("cg_convergence", False)
    g.default.set_verbose("cg", False)
    cg = inv.cg({"eps": 1e-6, "maxiter": 1000})
    invD = w.propagator(inv.preconditioned(pc.eo1_ne(), cg))

    src = g.mspincolor(grid)
    g.create.point(src, [0,0,0,0])
    dst2 = g(invD * src)
    return g.trace(dst2[0,0,0,0]).real

def average(x):
    return np.mean(x), np.std(x) / len(x)**0.5

masses = [0.05, 0.1, 0.15, 0.2]
chiral_condensate_beta_5p9_Ls8 = []
for m in masses:
    g.message(m)
    chiral_condensate_beta_5p9_Ls8.append(
        average([chiral_condensate(f"8c8_5.9/su3.{nr}",m,8) for nr in range(100,300,40)])
    )

chiral_condensate_beta_5p9_Ls12 = []
for m in masses:
    g.message(m)
    chiral_condensate_beta_5p9_Ls12.append(
        average([chiral_condensate(f"8c8_5.9/su3.{nr}",m,12) for nr in range(100,300,40)])
    )
```

```
GPT :    32286.519887 s : 0.05
GPT :    32425.447545 s : 0.1
GPT :    32513.509650 s : 0.15
GPT :    32604.220338 s : 0.2
GPT :    32659.508438 s : 0.05
GPT :    32858.499714 s : 0.1
GPT :    32985.510480 s : 0.15
GPT :    33117.060066 s : 0.2
```

In [74]:
```
fig, ax = plt.subplots()

plt.xlim()
plt.xlabel("mass")
plt.ylabel("-<psibar psi> / V")
plt.xlim(0,0.25)
plt.ylim(0,0.25)

x = np.arange(0,0.2,0.01)

ax.plot(x, [chiral_condensate_beta_5p9_Ls12[0][0]
          + (chiral_condensate_beta_5p9_Ls12[2][0]
             - chiral_condensate_beta_5p9_Ls12[0][0])*(xx-0.05)/0.1 for xx in x
         ], "g--")

ax.errorbar(masses,
             [x[0] for x in chiral_condensate_beta_5p9_Ls8],
             [x[1] for x in chiral_condensate_beta_5p9_Ls8], label="Ls=8", c="blue", marker="x")
ax.errorbar(masses,
             [x[0] for x in chiral_condensate_beta_5p9_Ls12],
             [x[1] for x in chiral_condensate_beta_5p9_Ls12], label="Ls=12", c="red")


plt.legend()
plt.show()
```



We note that chiral symmetry is clearly broken in our gauge ensemble for $\beta = 5.9$ as the vacuum expectation value of $\langle\bar\psi\psi\rangle$ is not consistent with zero. We extrapolate the GW fermion to the massless limit and find a non-vanishing condensate. So even in the absence of explicit chiral symmetry breaking due to the quark mass, the vacuum spontaneously breaks the symmetry.

Overlap fermion operator

In this section we discuss the **overlap** fermion, defined by the Dirac matrix

$$D_{\rm ov} = \frac{1}{a}(1+\gamma_5 {\rm Sign}[H]) \tag{57}$$

with a given Hermitian kernel operator H. The ${\rm Sign}$ function applied to the Hermitian operator acts by replacing all its eigenvalues by their sign. Since H is Hermitian and therefore the operator is diagonalizable with eigenvalues are real, this is a well-defined procedure. It is straightforward to show that any operator of this form satisfies the GW relation since

$$\begin{aligned} aD_{\rm ov}\gamma_5 D_{\rm ov} &= \frac{1}{a}(1+\gamma_5{\rm Sign}[H])\,\gamma_5\,(1+\gamma_5{\rm Sign}[H]) &(58)\\ &= \frac{1}{a}\left(\gamma_5+\gamma_5{\rm Sign}[H]\gamma_5+{\rm Sign}[H]+\gamma_5{\rm Sign}[H]^2\right) &(59)\\ &= \frac{1}{a}(2\gamma_5+\gamma_5{\rm Sign}[H]\gamma_5+{\rm Sign}[H]) &(60)\\ &= D_{\rm ov}\gamma_5+\gamma_5 D_{\rm ov}\,. &(61)\end{aligned}$$

In practice one often uses $H = \gamma_5 D_{\rm wilson}(m=-M_5)$, where the negative mass parameter is also often expressed as $M_5 = 1+s$. This overlap operator is then explicitly gauge covariant and chirally symmetric in addition to having no fermion doublers. It is important to note that the mass M_5 does not correspond to the mass of the resulting fermion. The overlap operator $D_{\rm ov}$ corresponds to a **massless** quark for any M_5.

The fermion operators we have studied so far were all **ultralocal**, i.e., they only coupled a small finite number of spatially separated lattice sites to each other. Concretely, the Wilson fermions only include nearest-neighbor connections through the covariant shift operators.

Due to the sign function the locality properties of the overlap operator are less clear. One can show, however, that the operator is still local in the sense

$$|(D_{\rm ov})_{x,y}| \le \kappa e^{-\gamma|x-y|}\,, \tag{62}$$

where both κ and γ are functions of the chosen kernel H and in particular of M_5 in the standard choice given above. It is important to note that this bound is fixed in lattice units, i.e., the non-locality vanishes in the continuum limit.

Finally a note is in order how to evaluate the sign function. As a first step, we re-write the sign function as

$${\rm Sign}[H] = \frac{H}{|H|} = \frac{H}{\sqrt{H^2}}\,. \tag{63}$$

Since the operators in the numerator and denominator commute, we do not write them in a particular order.

The challenging task is therefore to apply

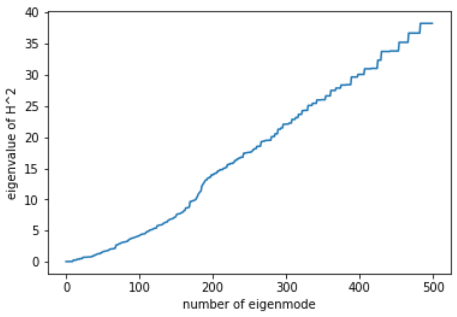
$$(X)^{-1/2}$$

for a positive-definite Hermitian X to a given vector.

To this end, let us first study the spectrum of X. We pick $M_5 = 1.8$ for concreteness, which is a popular choice.

In [73]:
```
grid = g.grid([8,8,8,16], g.double)

U = g.qcd.gauge.unit(grid)

w = g.qcd.fermion.wilson_clover(U, mass=-1.8, csw_r=0.0, csw_t=0.0, nu=1.0, xi_0=1.0,
                                isAnisotropic=False,
                                boundary_phases=[1,1,1,-1])

def H2(dst, src):
    dst @= g.gamma[5] * w * g.gamma[5] * w * src

g.default.set_verbose("arnoldi", False)
a = g.algorithms.eigen.arnoldi(Nmin=700, Nmax=1400, Nstep=100, Nstop=500, resid=1e-6, implicit_restart=True)

evecH2, evalsH2 = a(H2, start)
```

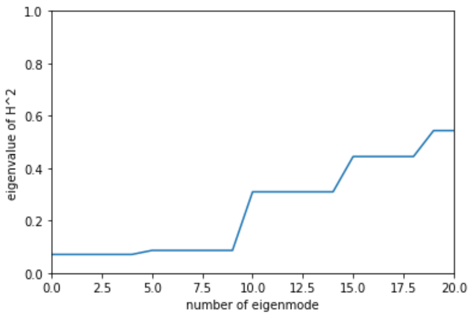
In [74]:
```
fig, ax = plt.subplots()

plt.xlim()
plt.xlabel("number of eigenmode")
plt.ylabel("eigenvalue of H^2")
ax.plot(list(range(len(evalsH2))), sorted([e.real for e in evalsH2]))

plt.show()
```



In [76]:

```
g.message(sorted([e.real for e in evalsH2])[0:10])

fig, ax = plt.subplots()

plt.xlim(0,20)
plt.ylim(0,1)
plt.xlabel("number of eigenmode")
plt.ylabel("eigenvalue of H^2")
ax.plot(list(range(len(evalsH2))), sorted([e.real for e in evalsH2]))

plt.show()
```

```
GPT :  828657.505389 s : [0.07074355135481224, 0.07074355135481682, 0.0707435513548206, 0.07074355135482324, 0.07074355135484864, 0.086115327032
23132, 0.08611532703225581, 0.08611532703225734, 0.08611532703226288, 0.08611532703228077]
```



In the next lecture, we study how calculate the sign function.

It is clear that there may be particularly small eigenvalues, which one then may treat separately, e.g., by computing them and the corresponding eigenvectors as we have demonstrated here. One can then define a subtracted function

$$H^2_{\text{deflated}} = H^2 - \sum_n \lambda_n^2 |n\rangle\langle n| \tag{64}$$

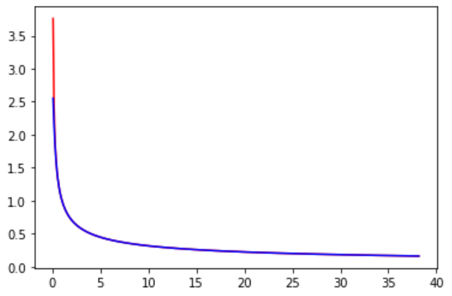
with eigenvalues λ_n and eigenvectors $|n\rangle$ of H that are orthogonalized to each other and the sum being over the smallest eigenvalues up to a given threshold.

Let us demonstrate the combination of such a small-eigenspace-subtraction with a polynomial approximation using Chebyshev polynomials. These are readily available in GPT.

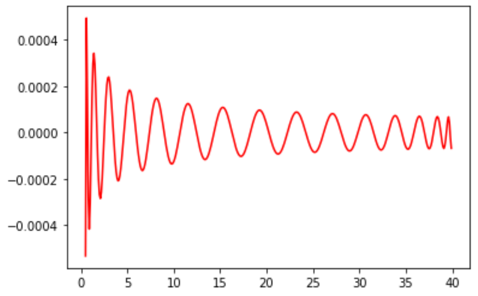
In [81]:

```
c = g.algorithms.polynomial.chebyshev(low=0.5,
                                      high=40,
                                      order=30,
                                      func=lambda x: x**-0.5)

fig, ax = plt.subplots()

g.message(sorted([e.real for e in evalsH2])[-1])

lambdas = np.arange(0.07074355135481224,38.24016691619401,0.1)
ax.plot(lambdas,[1.0 / l**0.5 for l in lambdas], c="red")
ax.plot(lambdas,[c(float(l)) for l in lambdas], c="blue")

plt.show()

fig, ax = plt.subplots()

g.message(sorted([e.real for e in evalsH2])[-1])

lambdas = np.arange(0.5,40,0.1)
ax.plot(lambdas,[c(float(l)) * l**0.5 - 1.0 for l in lambdas], c="red")

plt.show()
```

```
GPT :  829101.817192 s : 38.24016691619401
```



```
GPT :  829101.984571 s : 38.24016691619401
```



In [85]:

```
idx = evalsH2.real.argsort()
g.message("First 20 eigenvalues:", evalsH2[idx][0:20].real)

g.message("Spectral inner products (evecs are not orthogonal, so need to orthogonalize on the fly):")
for i in range(4):
    for j in range(4):
        g.message(f"<{i}|{j}> = {g.inner_product(evecH2[idx[i]],evecH2[idx[j]])}")

def Hm0p5(dst, src):
    deflated_src = g.copy(src)
    dst[:] = 0
    for i in range(20):
        v = g.inner_product(evecH2[idx[i]], deflated_src)
        dst += evalsH2[idx[i]]**-0.5 * evecH2[idx[i]] * v
        deflated_src -= evecH2[idx[i]] * v
    dst += c(H2) * deflated_src

# test
for i in range(len(evalsH2)):
    dst = g.copy(evecH2[i])
    Hm0p5(dst, evecH2[i])
    vvv = g.inner_product(evecH2[i], dst)
    vvv_expected = evalsH2[i] ** -0.5
    eps = abs(vvv - vvv_expected) / abs(vvv + vvv_expected)
    assert eps < 1e-3
```

```
GPT :  829240.009220 s : First 20 eigenvalues: [0.07074355 0.07074355 0.07074355 0.07074355 0.07074355 0.08611533
                       :  0.08611533 0.08611533 0.08611533 0.08611533 0.30964862 0.30964862
                       :  0.30964862 0.30964862 0.30964862 0.44447293 0.44447293 0.44447293
                       :  0.44447293 0.54348875]
GPT :  829240.010607 s : Spectral inner products (evecs are not orthogonal, so need to orthogonalize on the fly):
GPT :  829240.011791 s : <0|0> = (1-7.878245644268528e-20j)
GPT :  829240.012920 s : <0|1> = (-0.0050086681752263724+0.00407796094889292j)
GPT :  829240.013934 s : <0|2> = (-0.0074348126494220845-0.013755244561823952j)
GPT :  829240.015989 s : <0|3> = (0.004518939728875274+0.005267604957907369j)
GPT :  829240.017321 s : <1|0> = (-0.0050086681752263724-0.004077960948892921j)
GPT :  829240.019287 s : <1|1> = (0.9999999999999976+2.91000937579722e-20j)
GPT :  829240.020428 s : <1|2> = (-0.05697888813808714+0.021121600764340595j)
GPT :  829240.021904 s : <1|3> = (0.0005929652585921154-0.02159080824829708j)
GPT :  829240.023064 s : <2|0> = (-0.0074348126494220845+0.013755244561823953j)
GPT :  829240.024130 s : <2|1> = (-0.05697888813808714-0.021121600764340595j)
GPT :  829240.025087 s : <2|2> = (0.9999999999999966-2.185349044232908e-22j)
GPT :  829240.026183 s : <2|3> = (-0.006796081156609586+0.0036320547491189123j)
GPT :  829240.027754 s : <3|0> = (0.004518939728875274-0.005267604957907369j)
GPT :  829240.029328 s : <3|1> = (0.0005929652585921154+0.021590808248297073j)
GPT :  829240.030507 s : <3|2> = (-0.006796081156609586-0.003632054749118913j)
GPT :  829240.033083 s : <3|3> = (0.9999999999999956+1.099955426343752e-19j)
```

In [92]:

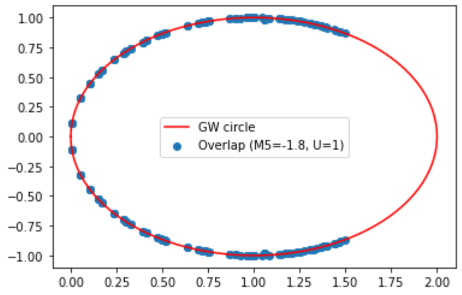
```
def Overlap(dst, src):
    Hm0p5(dst, src)
    dst @= w * dst + src
```

In [95]:

```
g.default.set_verbose("arnoldi", False)
a = g.algorithms.eigen.arnoldi(Nmin=300, Nmax=1400, Nstep=100, Nstop=200, resid=1e-6, implicit_restart=True)

evecOv, evalsOv = a(Overlap, start)
```

In [96]:

```
fig, ax = plt.subplots()

plt.xlim()
ax.scatter([e.real for e in evalsOv], [e.imag for e in evalsOv], label="Overlap (M5=-1.8, U=1)")

ax.plot([1 - np.cos(phi) for phi in np.arange(0,2*np.pi,0.01)],
          [np.sin(phi) for phi in np.arange(0,2*np.pi,0.01)], c="red", label="GW circle")

plt.legend()
plt.show()
```



In practice, one often also uses a rational approximation to the sign function. A complete discussion of optimal algorithms to compute the sign function is, however, beyond the scope of this lecture.

Homework: Compute the eigenvalues for the massless overlap on one of our previous quenched configurations.

Massive overlap fermions

We now consider how to define massive overlap fermions. To this end we first clean up the notation. We begin by defining

$$\hat\gamma_5 = \gamma_5(1-D)\,. \tag{65}$$

We can then show that the GW relation yields

$$\begin{aligned}\hat\gamma_5^2 &= \gamma_5(1-D)\gamma_5(1-D) = 1-\gamma_5 D\gamma_5 + \gamma_5 D\gamma_5 D - D \quad (66)\\ &= 1-\gamma_5 D\gamma_5 - D + \gamma_5(D\gamma_5+\gamma_5 D) = 1\,. \quad (67)\end{aligned}$$

Similarly, one can define

$$\tilde\gamma_5 = (1-D)\gamma_5 = \gamma_5\hat\gamma_5\gamma_5 \tag{68}$$

with $\tilde\gamma_5^2 = 1$.

Using the GW relation, we then find

$$\gamma_5 D + D\hat\gamma_5 = \gamma_5 D + D\gamma_5 - D\gamma_5 D = 0\,. \tag{69}$$

but also

$$\tilde\gamma_5 D + D\gamma_5 = (1-D)\gamma_5 D + D\gamma_5 = 0\,. \tag{70}$$

So to summarize, we have

$$\begin{aligned}\gamma_5 D + D\hat\gamma_5 &= 0\,, \quad (71)\\ \tilde\gamma_5 D + D\gamma_5 &= 0\,, \quad (72)\\ \gamma_5\tilde\gamma_5 - \hat\gamma_5\gamma_5 &= 0\,, \quad (73)\\ \tilde\gamma_5\gamma_5 - \gamma_5\hat\gamma_5 &= 0\,, \quad (74)\\ \hat\gamma_5^2 &= 1\,, \quad (75)\\ \tilde\gamma_5^2 &= 1\,, \quad (76)\\ \gamma_5^2 &= 1\,. \quad (77)\end{aligned}$$

The chiral rotations we have studied so far are then expressed as

$$\begin{aligned}\psi \to \psi' &= \exp\left(i\alpha T\gamma_5\left(1-\frac{1}{2}D\right)\right)\psi = \exp\left(i\alpha T\frac{\gamma_5+\hat\gamma_5}{2}\right)\psi\,, \quad (78)\\ \bar\psi \to \bar\psi' &= \bar\psi\exp\left(i\alpha\left(1-\frac{1}{2}D\right)\gamma_5 T\right) = \bar\psi\exp\left(i\alpha T\frac{\gamma_5+\tilde\gamma_5}{2}\right)\,. \quad (79)\end{aligned}$$

It is easy to show that both

$$\begin{aligned}\psi \to \psi' &= \exp(i\alpha T\hat\gamma_5)\psi\,, \quad (80)\\ \bar\psi \to \bar\psi' &= \bar\psi\exp(i\alpha T\gamma_5) \quad (81)\end{aligned}$$

and

$$\begin{aligned}\psi \to \psi' &= \exp(i\alpha T\gamma_5)\psi\,, \quad (82)\\ \bar\psi \to \bar\psi' &= \bar\psi\exp(i\alpha T\tilde\gamma_5) \quad (83)\end{aligned}$$

define chiral symmetries as well. (They differ only by quantum versions of the equations of motion.)

Using these new operators, we can define chiral projectors

$$P_\pm = \frac{1}{2}(1 \pm \gamma_5)\ , \tag{84}$$
$$\hat{P}_\pm = \frac{1}{2}(1 \pm \hat{\gamma}_5)\ , \tag{85}$$
$$\tilde{P}_\pm = \frac{1}{2}(1 \pm \tilde{\gamma}_5) \tag{86}$$

satisfying

$$P_\pm^2 = P_\pm\,, \tag{87}$$
$$\hat{P}_\pm^2 = \hat{P}_\pm\,, \tag{88}$$
$$\tilde{P}_\pm^2 = \tilde{P}_\pm\,, \tag{89}$$
$$P_+P_- = \hat{P}_+\hat{P}_- = \tilde{P}_+\tilde{P}_- = 0\,, \tag{90}$$
$$P_+ + P_- = \hat{P}_+ + \hat{P}_- = \tilde{P}_+ + \tilde{P}_- = 1\,, \tag{91}$$
$$D\hat{P}_\pm = P_\mp D\,, \tag{92}$$
$$DP_\pm = \tilde{P}_\mp D\,. \tag{93}$$

This now allows for the definition of chiral fields. Two definitions immediately seem sensible. First,

$$\psi_\pm = \hat{P}_\pm\psi\,, \qquad \bar{\psi}_\pm = \bar{\psi}P_\mp \tag{94}$$

for which

$$\bar{\psi}_\pm D\psi_\mp = \bar{\psi}P_\mp D\hat{P}_\mp\psi = \bar{\psi}P_\mp P_\pm D\psi = 0\,. \tag{95}$$

The Dirac operator therefore has the expected chiral structure from chapter 5. Alternatively we define

$$\psi_\pm = P_\pm\psi\,, \qquad \bar{\psi}_\pm = \bar{\psi}\tilde{P}_\mp \tag{96}$$

for which

$$\bar{\psi}_\pm D\psi_\mp = \bar{\psi}\tilde{P}_\mp DP_\mp\psi = \bar{\psi}\tilde{P}_\mp\tilde{P}_\pm D\psi = 0\,. \tag{97}$$

So also in this definition, we find the right chiral structure.

Writing down the mass term now is trivial. We start by using the first definition of projectors,

$$m(\bar{\psi}_+\psi_- + \bar{\psi}_-\psi_+) = \frac{m}{4}\bar{\psi}(1-\gamma_5)(1-\hat{\gamma}_5)\psi + \frac{m}{4}\bar{\psi}(1+\gamma_5)(1+\hat{\gamma}_5)\psi \tag{98}$$
$$= \frac{m}{4}\bar{\psi}(1-\gamma_5)(1-\gamma_5+\gamma_5 D)\psi + \frac{m}{4}\bar{\psi}(1+\gamma_5)(1+\gamma_5-\gamma_5 D)\psi \tag{99}$$
$$= \frac{m}{4}\bar{\psi}(1-\gamma_5)(2-D)\psi + \frac{m}{4}\bar{\psi}(1+\gamma_5)(2-D)\psi \tag{100}$$
$$= m\bar{\psi}\left(1-\frac{1}{2}D\right)\psi\,. \tag{101}$$

The second definition yields a consistent mass term since

$$P_s\hat{P}_r = \tilde{P}_r P_s \tag{102}$$

which follows from Eq. (71) and following.

If we now add this to the massless operator, the massive overlap fermion is

$$D_{\text{ov}}(m) = 1\left(1+\frac{m}{2}\right) + \left(1-\frac{m}{2}\right)\gamma_5\text{Sign}[H]\,. \tag{103}$$

Sometimes, one also uses the more symmetric definition

$$D_{\text{ov}'}(m) = 1\frac{1+m}{2} + \frac{1-m}{2}\gamma_5\text{Sign}[H] \tag{104}$$

for which the modified GW relation $\gamma_5 D_{\text{ov}'}(0) + D_{\text{ov}'}(0)\gamma_5 = 2D_{\text{ov}'}(0)\gamma_5 D_{\text{ov}'}(0)$ holds. We simply have $D_{\text{ov}'}(0) = D_{\text{ov}}(0)/2$.

Mobius Domain Wall Fermions

The overlap kernel defined above is just one of many possible choices. We noticed that it has a large ratio of largest to smallest eigenvalues (condition number), which increases the numerical cost to obtain a precise approximation of the sign function. The general Mobius kernel defined by

$$H = \gamma_5 D_{\text{mobius}}\,, \tag{105}$$
$$D_{\text{mobius}} = \frac{(b+c)D_{\text{wilson}}(m=-M_5)}{2+(b-c)D_{\text{wilson}}(m=-M_5)}\,. \tag{106}$$

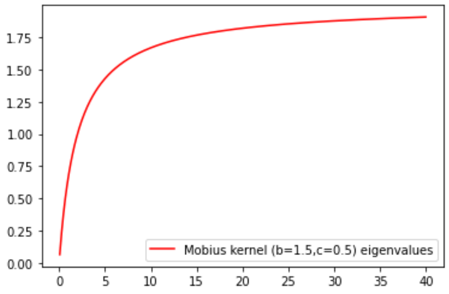
with parameters b and c allows for a substantial reduction of this condition number. It is therefore often used in practice.

Let us demonstrate this for the popular $b = 1.5$ and $c = 0.5$.

In [166…

```
fig, ax = plt.subplots()

plt.xlim()

lr=np.arange(0.07, 40, 0.1)
ax.plot(lr, [(2*l)/(2+l) for l in lr] , c="red", label="Mobius kernel (b=1.5,c=0.5) eigenvalues")

plt.legend()
plt.show()
```

```
lmin=(2*0.07)/(2+0.07)
lmax=(2*40)/(2+40)
print(lmax / lmin)
```



```
28.163265306122444
```

The original condition number of approximately 600 is reduced to 28.

One can show that an approximation to the overlap operator with a general Mobius kernel can be obtained by studying four-dimensional projections of a five-dimensional theory.

We first define

$$D_+ = bD_{\text{wilson}}(-M_5) + 1\,, \tag{107}$$
$$D_- = cD_{\text{wilson}}(-M_5) - 1\,, \tag{108}$$
$$D_{\text{mobius}} = \frac{D_+ + D_-}{D_+ - D_-}\,. \tag{109}$$

and a five-dimensional operator D_{dw5d},

$$\bar\psi D_{\text{dw5d}}(m)\psi = \sum_{s=1}^{L_s} \bar\psi_s D_+ \psi_s + \sum_{s=2}^{L_s} \bar\psi_s D_- P_+ \psi_{s-1} + \sum_{s=1}^{L_s-1} \bar\psi_s D_- P_- \psi_{s+1} - m(\bar\psi_1 D_- P_+ \psi_{L_s} + \bar\psi_{L_s} D_- P_- \psi_1)\,. \tag{110}$$

Then the Lagrange density

$$\mathcal{L} = \bar\psi D_{\text{dw5d}}(m)\psi + \bar\phi_{\text{pv}} D_{\text{dw5d}}(1)\phi_{\text{pv}} \tag{111}$$

generates the proper five-dimensional theory. We need two five-dimensional fields, ψ and the Pauli-Villars fields ϕ_{pv} which are responsible for the operator triviality at $m = 1$, see definition of $D_{\text{ov}'}(m)$. The ψ field is Grassmannian and the ϕ_{pv} field is a bosonic complex field. They generate the determinant ratio

$$\frac{\det(D_{\text{dw5d}}(m))}{\det(D_{\text{dw5d}}(m=1))}\,. \tag{112}$$

We then define four-dimensional projections

$$q_L = P_- \psi_1\,, \qquad q_R = P_+ \psi_{L_s}\,, \qquad q = q_L + q_R \tag{113}$$
$$\bar q_L = \bar\psi_{L_s}(-D_-)P_+\,, \qquad \bar q_R = \bar\psi_1(-D_-)P_- \qquad \bar q = \bar q_L + \bar q_R\,. \tag{114}$$

for which we find

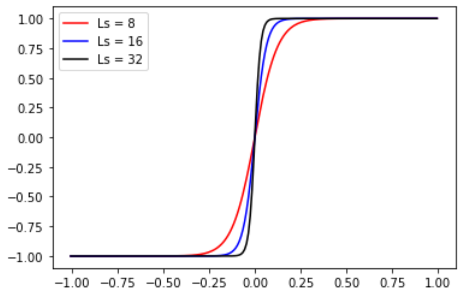
$$\langle q\bar q\rangle_q = \frac{1}{1-m}\Big((D_{\text{ov}'}^{L_s})^{-1} - 1\Big) \tag{115}$$

with

$$T = \frac{1-H}{1+H}\,, \tag{116}$$
$$D_{\text{ov}'}^{L_s} = \frac{1+m}{2} + \frac{1-m}{2}\gamma_5 \frac{1-T^{L_s}}{1+T^{L_s}}\,. \tag{117}$$

For $L_s \to \infty$ the rational function of H converges to the sign function. Let us demonstrate this:

In [108…
```
fig, ax = plt.subplots()

plt.xlim()

def sgn(x, Ls):
    t = (1-x)/(1+x)
    return (1-t**(Ls)) / (1 + t**(Ls))

xr=np.arange(-1.005,1,0.01)
ax.plot(xr, [sgn(x,8) for x in xr] , c="red", label="Ls = 8")
ax.plot(xr, [sgn(x,16) for x in xr] , c="blue", label="Ls = 16")
ax.plot(xr, [sgn(x,32) for x in xr] , c="black", label="Ls = 32")

plt.legend()
plt.show()
```



It may seem odd that we define four-dimensional fields q such that

$$\langle q\bar{q}\rangle_q = \frac{1}{1-m}\Big((D_{\rm ov'}^{L_s})^{-1} - 1\Big) \tag{118}$$

does not correspond just to $(D_{\rm ov'}^{L_s})^{-1}$, however, this definition has significant advantages. It corresponds to a field re-definition that has properties closer to the continuum fields.

One such property is that in the continuum one expects

$$\partial_m(\langle\psi\bar{\psi}\rangle_\psi)^{-1} = 1 \tag{119}$$

since we expect $(\langle\psi\bar{\psi}\rangle_\psi)^{-1} = \gamma_\mu D_\mu + m$ in the continuum. Indeed our definition satisfies this, as

$$\begin{aligned}
D_{\rm ov'}(m) &= D_{\rm ov'}(0) + m(1 - D_{\rm ov'}(0)) = (D_{\rm ov'}(0) - 1)(1-m) + 1\,, && (120)\\
(D_{\rm ov'}(m))^{-1} &= [(D_{\rm ov'}(0) - 1)(1-m) + 1]^{-1}\,, && (121)\\
(D_{\rm ov'}(m))^{-1} - 1 &= [(D_{\rm ov'}(0) - 1)(1-m) + 1]^{-1}[1 - ((D_{\rm ov'}(0) - 1)(1-m) + 1)] && (122)\\
&= -[(D_{\rm ov'}(0) - 1)(1-m) + 1]^{-1}(D_{\rm ov'}(0) - 1)(1-m)\,, && (123)\\
\langle q\bar{q}\rangle_q^{-1} &= (1-m)(D_{\rm ov'}(m))^{-1} - 1)^{-1} && (124)\\
&= -[(1-m) + (D_{\rm ov'}(0) - 1)^{-1}] && (125)\\
&= m1 - (D_{\rm ov'}(0) - 1)^{-1} - 1 && (126)\\
&= m1 + D_{\rm ov'}(0) + D_{\rm ov'}(0)^2 + D_{\rm ov'}(0)^3 + \ldots\,. && (127)
\end{aligned}$$

With this definition, we also obtain the natural definition of the chiral condensate as being the maximally chirally violating term (see discussion of the mass above)

$$\begin{aligned}
\langle\bar{\psi}_+\psi_- + \bar{\psi}_-\psi_+\rangle &= \langle\bar{\psi}(1 - D_{\rm ov'})\psi\rangle = -{\rm Tr}[D_{\rm ov'}^{-1}(1 - D_{\rm ov'})] && (128)\\
&= -{\rm Tr}[D_{\rm ov'}^{-1} - 1] = (1-m)\langle\bar{q}q\rangle\,. && (129)
\end{aligned}$$

Test five-dimensional domain wall operator

In [160…

```
grid = g.grid([8,8,8,16], g.double)

b_test = 1.5
c_test = 0.5
Ls_test = 12
mass_test = 0.1

D_W = g.qcd.fermion.wilson_clover(U, mass=-1.8, csw_r=0.0, csw_t=0.0, nu=1.0, xi_0=1.0,
                                isAnisotropic=False,
                                boundary_phases=[1,1,1,-1])

def D_DWF(dst, src):
    b = b_test
    c = c_test
    mass = mass_test
    src_s = g.separate(src, 0)
    dst_s = [g.lattice(s) for s in src_s]

    Ls = len(src_s)

    src_plus_s = []
    src_minus_s = []
    for s in range(Ls):
        src_plus_s.append(g(0.5 * src_s[s] + 0.5 * g.gamma[5]*src_s[s]))
        src_minus_s.append(g(0.5 * src_s[s] - 0.5 * g.gamma[5]*src_s[s]))
    for d in dst_s:
        d[:] = 0
    for s in range(Ls):
        dst_s[s] += b*D_W* src_s[s] + src_s[s]
    for s in range(1,Ls):
        dst_s[s] += c*D_W * src_plus_s[s-1] - src_plus_s[s-1]
    for s in range(0,Ls-1):
        dst_s[s] += c*D_W * src_minus_s[s+1] - src_minus_s[s+1]
    dst_s[0] -= mass*(c*D_W * src_plus_s[Ls-1] - src_plus_s[Ls-1])
    dst_s[Ls-1] -= mass*(c*D_W * src_minus_s[0] - src_minus_s[0])
    dst @= g.merge(dst_s, 0)

D_dwf = g.qcd.fermion.mobius(U, M5=1.8, mass=mass_test, Ls=Ls_test, b=b_test, c=c_test,
                             boundary_phases=[1,1,1,-1])

rng = g.random("test")
src_test = rng.cnormal(g.vspincolor(D_dwf.F_grid))
dst_test = g(D_dwf * src_test)

dst_test2 = g.vspincolor(D_dwf.F_grid)
D_DWF(dst_test2, src_test)
```

```
eps = (g.norm2(dst_test2 - dst_test) / g.norm2(dst_test)) ** 0.5
g.message(eps)
assert eps < 1e-13
```

```
GPT :  344952.716639 s : Initializing gpt.random(test,vectorized_ranlux24_389_64) took 0.000365019 s
GPT :  344952.921818 s : 1.760513745315208e-16
```

Test overlap propagator

We now test

$$\langle q\bar{q}\rangle_q = \frac{1}{1-m}\left((D^{L_s}_{\mathrm{ov}'})^{-1} - 1\right) \tag{130}$$

by checking

$$(D^{L_s}_{\mathrm{ov}'})((1-m)\langle q\bar{q}\rangle_q + 1)v = v \tag{131}$$

for $b = c = 1$ which corresponds to the overlap operator with the simple Wilson kernel that we implemented above.

In [177…

```
inv = g.algorithms.inverter
pc = g.qcd.fermion.preconditioner
g.default.set_verbose("cg_convergence", False)
g.default.set_verbose("cg", False)
cg = inv.cg({"eps": 1e-8, "maxiter": 1000})

D_dwf = g.qcd.fermion.mobius(U, M5=1.8, mass=mass_test, Ls=Ls_test, b=1, c=1,
                             boundary_phases=[1,1,1,-1])

qqbar = D_dwf.propagator(inv.preconditioned(pc.eo1_ne(), cg))

src = g.mspincolor(grid)
rng.cnormal(src)

qqbar_src = g(qqbar * src)
```

In [178…

```
def _Overlap2(dst, src):
    # D_ov'(m) = D_ov'(0) + m(1-D_ov'(0))
    #          = D_ov(0)/2 + m(1-D_ov(0)/2)
    #          = D_ov(0)*(1 - m)/2 / 2 + m
    Overlap(dst, src)
    dst *= 0.5*(1-mass_test)
    dst += mass_test * src

Overlap2 = g.matrix_operator(otype=g.ot_vector_spin_color(4, 3), mat=_Overlap2)

#
one_minus_m_qqbar_src = g((1.0 - mass_test) * qqbar_src)
dst = g(one_minus_m_qqbar_src + src)
dst2 = g.lattice(dst)
Overlap2(dst2, dst)

g.message(g.norm2(dst2-src)/g.norm2(src))
```

```
GPT :  354151.708624 s : 2.1304647219726924e-05
```

Ward–Takahashi identities

Consider the effect of field transformations

$$\psi \to \psi' = \psi + \delta\psi\,, \tag{132}$$
$$\bar\psi \to \bar\psi' = \bar\psi + \delta\bar\psi\,, \tag{133}$$

on an observable $O(\bar\psi,\psi)$. We define

$$O(\bar\psi',\psi') = O(\bar\psi,\psi) + \delta O(\bar\psi,\psi)\,, \tag{134}$$
$$S(\bar\psi',\psi') = S(\bar\psi,\psi) + \delta S(\bar\psi,\psi)\,, \tag{135}$$
$$d\bar\psi' d\psi' = d\bar\psi d\psi - \delta M(\bar\psi,\psi) d\bar\psi d\psi \tag{136}$$
$$= d\bar\psi d\psi (1 - \delta M(\bar\psi,\psi))\,. \tag{137}$$

Then we write the path integral in two identical ways with different names for the integration variables, and compute the difference

$$0 = \int d\bar\psi' d\psi' O(\bar\psi',\psi') e^{-S(\bar\psi',\psi')} - \int d\bar\psi d\psi O(\bar\psi,\psi) e^{-S(\bar\psi,\psi)} \tag{138}$$
$$= \int d\bar\psi d\psi \left((1-\delta M(\bar\psi,\psi)) O(\bar\psi',\psi') e^{-S(\bar\psi',\psi')} - O(\bar\psi,\psi) e^{-S(\bar\psi,\psi)}\right) \tag{139}$$
$$= \int d\bar\psi d\psi \left[\delta O(\bar\psi,\psi) - O(\bar\psi,\psi)(\delta S(\bar\psi,\psi) + \delta M(\bar\psi,\psi))\right] e^{-S(\bar\psi,\psi)} + O(\delta^2)\,. \tag{140}$$

This can be conveniently written as the **Schwinger–Dyson equations**

$$\langle \delta O(\bar\psi,\psi)\rangle_\psi = \langle O(\bar\psi,\psi)(\delta S(\bar\psi,\psi) + \delta M(\bar\psi,\psi))\rangle_\psi\,.$$

It follows that for non-anomalous symmetries of the theory ($\delta S = \delta M = 0$) we find $\langle \delta O\rangle_\psi = 0$. This can be used, e.g., to relate correlation functions after space-time-symmetry transformations to one another.

Other choices will yield quantum analogs to equations of motion or Noether conservation laws. The latter are called **Ward–Takahashi** identities and will be studied next.

As a first step we consider non-anomalous transformations ($\delta M = 0$) of the type

$$\delta\psi(x) = i\varepsilon(x)\lambda\psi(x)\,, \tag{141}$$
$$\delta\bar\psi(x) = i\varepsilon(x)\bar\psi(x)\bar\lambda \tag{142}$$

with $\lambda, \bar\lambda$ acting on internal indices of the fields. For a lattice theory with

$$S = \sum_{x,y} \bar\psi(x) D(x,y)\psi(y) \tag{143}$$

we have

$$\delta S(\bar\psi,\psi) = i\sum_{x,y}(\varepsilon(x)\bar\psi(x)\bar\lambda D(x,y)\psi(y) + \varepsilon(y)\bar\psi(x)D(x,y)\lambda\psi(y)) \tag{144}$$

and therefore

$$\partial_{\varepsilon(x)}\delta S(\bar\psi,\psi) = i\sum_{y}(\bar\psi(x)\bar\lambda D(x,y)\psi(y) + \bar\psi(y)D(y,x)\lambda\psi(x)) \tag{145}$$

For the Wilson Dirac matrix in n_d space-time dimensions with mass matrix M

$$D(x,y) = \sum_\mu \frac{\gamma_\mu - 1}{2} U_\mu(x)\delta_{y,x+\hat\mu} - \sum_\mu \frac{\gamma_\mu+1}{2}U^\dagger_\mu(x-\hat\mu)\delta_{y,x-\hat\mu} + n_d 1 + M \tag{146}$$

and

$$\lambda = \tau_b\,, \tag{147}$$
$$\bar\lambda = -\tau_b \tag{148}$$

with flavor matrices τ_b we find

$$\begin{aligned}
-i\partial_{\varepsilon(x)}\delta S(\bar\psi,\psi) &= \sum_y(\bar\psi(y)D(y,x)\tau_b\psi(x) - \bar\psi(x)\tau_b D(x,y)\psi(y)) && (149)\\
&= \sum_\mu \bar\psi(x-\hat\mu)\frac{\gamma_\mu-1}{2}U_\mu(x-\hat\mu)\tau_b\psi(x) - \sum_\mu\bar\psi(x)\frac{\gamma_\mu-1}{2}U_\mu(x)\tau_b\psi(x+\hat\mu)\\
&\quad - \sum_\mu\bar\psi(x+\hat\mu)\frac{\gamma_\mu+1}{2}U^\dagger_\mu(x)\tau_b\psi(x) + \sum_\mu\bar\psi(x)\frac{\gamma_\mu+1}{2}U^\dagger_\mu(x-\hat\mu)\tau_b\psi(x-\hat\mu)\\
&\quad + \bar\psi(x)[M,\tau^b]\psi(x) && (150)\\
&= \bar\psi(x)[M,\tau^b]\psi(x) - \sum_\mu(J^b_\mu(x) - J^b_\mu(x-\hat\mu)) && (151)\\
&= \bar\psi(x)[M,\tau^b]\psi(x) - \sum_\mu \overleftarrow{\partial}_\mu J^b_\mu(x) && (152)
\end{aligned}$$

with

$$J^b_\mu(x) = \bar\psi(x)\frac{\gamma_\mu-1}{2}U_\mu(x)\tau_b\psi(x+\hat\mu) + \bar\psi(x+\hat\mu)\frac{\gamma_\mu+1}{2}U^\dagger_\mu(x)\tau_b\psi(x)\,. \tag{153}$$

For $O=1$ we then have the equation

$$\langle\sum_\mu \overleftarrow{\partial}_\mu J^b_\mu(x)\rangle_\psi = \langle\bar\psi(x)[M,\tau^b]\psi(x)\rangle_\psi\,. \tag{154}$$

For a τ^b diagonal in flavor space the right-hand term vanishes and the current J_μ is conserved. Let us identify the case $\tau^b = 1$ with J_μ, i.e., suppressing the b index. We can define the charge

$$Q(x_4) = \sum_{x_1,x_2,x_3} J_4(x) \tag{155}$$

for which

$$\begin{aligned}
Q(x_4) - Q(x_4-1) &= \sum_{x_1,x_2,x_3}\overleftarrow{\partial}_4 J_4(x) && (156)\\
&= -\sum_{i=1,2,3}\sum_{x_1,x_2,x_3}\overleftarrow{\partial}_i J_i(x) = 0 && (157)
\end{aligned}$$

due to periodic boundary conditions of J_i, i.e., the charge Q is time-independent.

In the case $O \neq 1$ we can use our formalism again and find current conservation with additional **contact terms**. We leave this discussion for subsequent lectures.

We first test the vector current conservation numerically.

In [11]:
```
import gpt as g
import numpy as np
inv = g.algorithms.inverter
pc = g.qcd.fermion.preconditioner
g.default.set_verbose("cg_convergence", False)
g.default.set_verbose("cg", False)

grid = g.grid([8,8,8,16],g.double)
rng = g.random("t")
U_rnd = g.qcd.gauge.random(grid, rng)

w = g.qcd.fermion.wilson_clover(U_rnd, mass=-0.5, csw_r=0.0, csw_t=0.0, nu=1.0, xi_0=1.0,
                                isAnisotropic=False,
                                boundary_phases=[1,1,1,-1])
```

```
cg = inv.cg({"eps": 1e-8, "maxiter": 1000})
invD = w.propagator(inv.preconditioned(pc.eo1_ne(), cg))

src = g.mspincolor(U_rnd[0].grid)
g.create.point(src, [1,0,0,0])
prop = g(invD * src)
```

```
GPT :     513.889909 s : Initializing gpt.random(t,vectorized_ranlux24_389_64) took 0.00026083 s
```

In [12]:

```
def divergence(f, current):
    resN = g.lattice(f)
    resN[:] = 0

    b = g(g.gamma[5] * g.adj(f) * g.gamma[5])
    for mu in range(4):
        c_mu = current(f, b, mu)
        resN += c_mu - g.cshift(c_mu,mu,-1)

    return g.norm2(resN)

def local_current(f, b, mu):
    return g(g.gamma[mu] * f * b)

g.message("div(J_local)",
          divergence(prop, local_current))
```

```
GPT :     516.032148 s : div(J_local) 0.4504373871226864
```

In [14]:

```
def conserved_current(psi, bar_psi, mu):
    s = g.covariant.shift(U_rnd, boundary_phases=w.params["boundary_phases"])
    return g(
        + 0.5 * bar_psi * (g.gamma[mu].tensor() - g.gamma["I"].tensor()) * s.forward[mu] * psi
        + 0.5 * g.adj(s.forward[mu](g.adj(bar_psi))) * (g.gamma[mu].tensor() + g.gamma["I"].tensor()) * psi
    )

g.message("div(J_conserved)",
          divergence(prop, conserved_current))

g.message("div(J_conserved_gpt)",
          divergence(prop, w.conserved_vector_current))
```

```
GPT :     594.806758 s : div(J_conserved) 3.0420761637370467e-19
GPT :     594.972310 s : div(J_conserved_gpt) 3.0420761637370467e-19
```

For domain-wall fermions, the construction of the four-dimensional conserved currents proceeds in the same manner. One first finds a five-dimensional conserved current

$$j_\mu^{\rm DWF}(x,s)$$

with space-time position x and fifth-dimension coordinate s that satisfies

$$\sum_{\mu=1,2,3,4} \overleftarrow{\partial}_\mu j_\mu^{\rm DWF}(x,s) + \overleftarrow{\partial}_5 j_5^{\rm DWF}(x,s) = 0\,.$$

One then defines a four-dimensional conserved current by summing over the fifth dimension. This is similar to the construction of the conserved charge above, where we reduced the dimension of the conserved quantity from four to one by summing over the orthogonal dimensions. Concretely, we have

$$J_\mu^{\rm DWF,4d}(x) = \sum_s j_\mu^{\rm DWF}(x,s)$$

with

$$\sum_{\mu=1,2,3,4} \overleftarrow{\partial}_\mu J_\mu^{\rm DWF,4d}(x) = -\sum_s \overleftarrow{\partial}_5 j_5^{\rm DWF}(x,s) = 0\,. \tag{158}$$

Finally for domain-wall fermions the left and right modes are located at different parts of the fifth dimension. One can then define left- and right-handed 4d currents by

$$J_\mu^{\rm DWF,4d,left}(x) = \sum_{s=1}^{L_s/2} j_\mu^{\rm DWF}(x,s)\,, \tag{159}$$

$$J_\mu^{\rm DWF,4d,right}(x) = \sum_{s=L_s/2}^{L_s} j_\mu^{\rm DWF}(x,s)\,, \tag{160}$$

$$J_\mu^{\rm DWF,4d,5}(x) = J_\mu^{\rm DWF,4d,left}(x) - J_\mu^{\rm DWF,4d,right}(x)\,. \tag{161}$$

From this follows the domain-wall axial ward identity

$$\overleftarrow{\partial}_\mu J_\mu^{\rm DWF,4d,5}(x) = -2 j_5^{\rm DWF}(x,L_s) + 2 j_5^{\rm DWF}(x,L_s/2)\,. \tag{162}$$

The first current on the right-hand side corresponds to

$$j_5^{\rm DWF}(x,L_s) = -m\bar{q}(x)\gamma_5 q(x)\,. \tag{163}$$

The second current measures the residual chiral symmetry breaking for finite L_s. This corresponds to the **residual mass** of domain-wall fermions. For modern calculations these residual masses are of order 10^{-4} or 10^{-5}.

In [60]:

```
D_w = g.qcd.fermion.wilson_clover(U_rnd, mass=-1.8, csw_r=0.0, csw_t=0.0, nu=1.0, xi_0=1.0,
                                  isAnisotropic=False,
                                  boundary_phases=[1,1,1,-1])

D_dwf = g.qcd.fermion.mobius(U_rnd, M5=1.8, mass=0.1, Ls=12, b=1.5, c=0.5,
                             boundary_phases=[1,1,1,-1])
```

```
src = g.mspincolor(U_rnd[0].grid)
g.create.point(src, [1,0,0,0])
cg = inv.cg({"eps": 1e-9, "maxiter": 1000})
invD = D_dwf.bulk_propagator(inv.preconditioned(pc.eo1_ne(), cg))

prop = g(invD * src)
#prop = g(invD * D_dwf.ImportPhysicalFermionSource * src)
#inv.preconditioned(pc.eo1_ne(), cg)
# TODO: first discuss 5d Gamma-5 hermiticity
```

$$\begin{aligned} q_L &= P_-\psi_1\,, & q_R &= P_+\psi_{L_s}\,, & q &= q_L + q_R \qquad (164)\\ \bar q_L &= \bar\psi_{L_s}(-D_-)P_+\,, & \bar q_R &= \bar\psi_1(-D_-)P_- & \bar q &= \bar q_L + \bar q_R\,. \qquad (165)\end{aligned}$$

$$\langle\psi\bar q\Gamma q\bar\psi\rangle = \langle\psi\bar\psi\rangle(-D_-)\bar P\Gamma P\langle\psi\bar\psi\rangle \qquad (166)$$

In [70]:

```
rhs = g.vspincolor(D_dwf.F_grid)
lhs = g.vspincolor(D_dwf.F_grid)
rng.cnormal([rhs,lhs]);

g.default.set_verbose("cg_convergence", True)
inverter = inv.cg({"eps": 1e-9, "maxiter": 1000})

def _Gamma5(dst, src):
    dst @= g.gamma[5] * g.merge(list(reversed(g.separate(inverter(g.adj(D_dwf.Dminus)*D_dwf.Dminus) * g.adj(D_dwf.Dminus) * src, 0))),0)

def _Gamma5_inv(dst, src):
    dst @= D_dwf.Dminus * g.merge(list(reversed(g.separate(g.gamma[5] * src, 0))),0)

Gamma5 = g.matrix_operator(mat=_Gamma5, inv_mat=_Gamma5_inv)

a = g.inner_product(rhs, D_dwf * lhs)
b = g.inner_product(lhs, Gamma5 * D_dwf * g.inv(Gamma5) * rhs)

# Gamma5^-1 D^dag = D Gamma5^-1

# D = Gamma5^-1^dag D^dag Gamma5^dag

# Gamma5^dag D Gamma5^dag^-1 = D^dag

# ⟨ψψ¯⟩(−D−)
# ⟨ψψ¯⟩^dag = Gamma5 ⟨ψψ¯⟩(−D−) inv(−D−) inv(Gamma5)

g.message((a - b.conjugate())/(a + b.conjugate()))
```

```
GPT :   58211.897432 s : cg: iteration 0: 2.202411e+07 / 1.418691e-10
GPT :   58211.923757 s : cg: iteration 1: 5.312684e+06 / 1.418691e-10
GPT :   58211.956349 s : cg: iteration 2: 1.726483e+06 / 1.418691e-10
GPT :   58211.988393 s : cg: iteration 3: 6.888980e+05 / 1.418691e-10
GPT :   58212.024988 s : cg: iteration 4: 2.804072e+05 / 1.418691e-10
GPT :   58212.062706 s : cg: iteration 5: 1.315746e+05 / 1.418691e-10
GPT :   58212.098438 s : cg: iteration 6: 5.991755e+04 / 1.418691e-10
GPT :   58212.137956 s : cg: iteration 7: 3.040094e+04 / 1.418691e-10
GPT :   58212.173858 s : cg: iteration 8: 1.577698e+04 / 1.418691e-10
GPT :   58212.210679 s : cg: iteration 9: 8.833351e+03 / 1.418691e-10
GPT :   58212.248622 s : cg: iteration 10: 4.913932e+03 / 1.418691e-10
GPT :   58212.286033 s : cg: iteration 11: 2.788112e+03 / 1.418691e-10
GPT :   58212.322470 s : cg: iteration 12: 1.645698e+03 / 1.418691e-10
GPT :   58212.358847 s : cg: iteration 13: 9.688578e+02 / 1.418691e-10
GPT :   58212.394665 s : cg: iteration 14: 5.970286e+02 / 1.418691e-10
GPT :   58212.432387 s : cg: iteration 15: 4.113252e+02 / 1.418691e-10
GPT :   58212.471221 s : cg: iteration 16: 2.813187e+02 / 1.418691e-10
GPT :   58212.506407 s : cg: iteration 17: 1.913160e+02 / 1.418691e-10
GPT :   58212.545371 s : cg: iteration 18: 1.249638e+02 / 1.418691e-10
GPT :   58212.580200 s : cg: iteration 19: 8.633711e+01 / 1.418691e-10
GPT :   58212.616595 s : cg: iteration 20: 6.557036e+01 / 1.418691e-10
GPT :   58212.652321 s : cg: iteration 21: 5.542382e+01 / 1.418691e-10
GPT :   58212.688536 s : cg: iteration 22: 4.706772e+01 / 1.418691e-10
GPT :   58212.724626 s : cg: iteration 23: 3.916843e+01 / 1.418691e-10
GPT :   58212.760521 s : cg: iteration 24: 3.544855e+01 / 1.418691e-10
GPT :   58212.795659 s : cg: iteration 25: 3.510840e+01 / 1.418691e-10
GPT :   58212.832404 s : cg: iteration 26: 3.331810e+01 / 1.418691e-10
GPT :   58212.869145 s : cg: iteration 27: 3.017491e+01 / 1.418691e-10
GPT :   58212.906480 s : cg: iteration 28: 2.920309e+01 / 1.418691e-10
GPT :   58212.947149 s : cg: iteration 29: 2.815355e+01 / 1.418691e-10
GPT :   58212.983665 s : cg: iteration 30: 2.315154e+01 / 1.418691e-10
GPT :   58213.023056 s : cg: iteration 31: 1.924410e+01 / 1.418691e-10
GPT :   58213.057017 s : cg: iteration 32: 1.557772e+01 / 1.418691e-10
GPT :   58213.097012 s : cg: iteration 33: 1.184432e+01 / 1.418691e-10
GPT :   58213.131327 s : cg: iteration 34: 9.021179e+00 / 1.418691e-10
GPT :   58213.167859 s : cg: iteration 35: 7.013720e+00 / 1.418691e-10
GPT :   58213.203530 s : cg: iteration 36: 5.270697e+00 / 1.418691e-10
GPT :   58213.238931 s : cg: iteration 37: 4.038425e+00 / 1.418691e-10
GPT :   58213.275632 s : cg: iteration 38: 3.243681e+00 / 1.418691e-10
GPT :   58213.309869 s : cg: iteration 39: 2.309687e+00 / 1.418691e-10
GPT :   58213.346049 s : cg: iteration 40: 1.699295e+00 / 1.418691e-10
GPT :   58213.382716 s : cg: iteration 41: 1.304594e+00 / 1.418691e-10
GPT :   58213.417793 s : cg: iteration 42: 9.469796e-01 / 1.418691e-10
GPT :   58213.452606 s : cg: iteration 43: 7.012050e-01 / 1.418691e-10
GPT :   58213.486875 s : cg: iteration 44: 5.095387e-01 / 1.418691e-10
GPT :   58213.525305 s : cg: iteration 45: 3.952718e-01 / 1.418691e-10
GPT :   58213.560326 s : cg: iteration 46: 3.267909e-01 / 1.418691e-10
GPT :   58213.599509 s : cg: iteration 47: 2.670515e-01 / 1.418691e-10
GPT :   58213.638362 s : cg: iteration 48: 2.283742e-01 / 1.418691e-10
GPT :   58213.674204 s : cg: iteration 49: 2.116531e-01 / 1.418691e-10
GPT :   58213.710890 s : cg: iteration 50: 2.217771e-01 / 1.418691e-10
GPT :   58213.743966 s : cg: iteration 51: 2.421812e-01 / 1.418691e-10
GPT :   58213.781328 s : cg: iteration 52: 2.255819e-01 / 1.418691e-10
GPT :   58213.814368 s : cg: iteration 53: 2.087119e-01 / 1.418691e-10
GPT :   58213.847386 s : cg: iteration 54: 1.844820e-01 / 1.418691e-10
GPT :   58213.882616 s : cg: iteration 55: 1.516327e-01 / 1.418691e-10
GPT :   58213.918145 s : cg: iteration 56: 1.219690e-01 / 1.418691e-10
GPT :   58213.955135 s : cg: iteration 57: 9.449011e-02 / 1.418691e-10
GPT :   58213.991276 s : cg: iteration 58: 6.964766e-02 / 1.418691e-10
GPT :   58214.029355 s : cg: iteration 59: 4.827851e-02 / 1.418691e-10
GPT :   58214.063642 s : cg: iteration 60: 3.277311e-02 / 1.418691e-10
GPT :   58214.098110 s : cg: iteration 61: 2.301075e-02 / 1.418691e-10
GPT :   58214.135938 s : cg: iteration 62: 1.598068e-02 / 1.418691e-10
GPT :   58214.169653 s : cg: iteration 63: 1.114899e-02 / 1.418691e-10
GPT :   58214.208316 s : cg: iteration 64: 8.167370e-03 / 1.418691e-10
GPT :   58214.242803 s : cg: iteration 65: 6.024267e-03 / 1.418691e-10
GPT :   58214.281762 s : cg: iteration 66: 4.578524e-03 / 1.418691e-10
GPT :   58214.316637 s : cg: iteration 67: 3.487379e-03 / 1.418691e-10
GPT :   58214.352724 s : cg: iteration 68: 2.699488e-03 / 1.418691e-10
GPT :   58214.388757 s : cg: iteration 69: 2.284561e-03 / 1.418691e-10
GPT :   58214.422407 s : cg: iteration 70: 2.134252e-03 / 1.418691e-10
GPT :   58214.459761 s : cg: iteration 71: 2.091098e-03 / 1.418691e-10
GPT :   58214.491021 s : cg: iteration 72: 2.136099e-03 / 1.418691e-10
GPT :   58214.529751 s : cg: iteration 73: 2.174970e-03 / 1.418691e-10
GPT :   58214.563128 s : cg: iteration 74: 1.965991e-03 / 1.418691e-10
GPT :   58214.596433 s : cg: iteration 75: 1.787866e-03 / 1.418691e-10
GPT :   58214.632552 s : cg: iteration 76: 1.668386e-03 / 1.418691e-10
GPT :   58214.666049 s : cg: iteration 77: 1.340445e-03 / 1.418691e-10
GPT :   58214.704940 s : cg: iteration 78: 1.058356e-03 / 1.418691e-10
GPT :   58214.740962 s : cg: iteration 79: 8.534024e-04 / 1.418691e-10
GPT :   58214.777667 s : cg: iteration 80: 6.117287e-04 / 1.418691e-10
GPT :   58214.815179 s : cg: iteration 81: 4.414577e-04 / 1.418691e-10
GPT :   58214.849175 s : cg: iteration 82: 3.333651e-04 / 1.418691e-10
GPT :   58214.886008 s : cg: iteration 83: 2.422032e-04 / 1.418691e-10
GPT :   58214.919892 s : cg: iteration 84: 1.717283e-04 / 1.418691e-10
GPT :   58214.959015 s : cg: iteration 85: 1.242395e-04 / 1.418691e-10
GPT :   58214.993537 s : cg: iteration 86: 9.372097e-05 / 1.418691e-10
GPT :   58215.034999 s : cg: iteration 87: 7.330767e-05 / 1.418691e-10
GPT :   58215.076451 s : cg: iteration 88: 5.821452e-05 / 1.418691e-10
GPT :   58215.112985 s : cg: iteration 89: 4.362652e-05 / 1.418691e-10
GPT :   58215.154050 s : cg: iteration 90: 3.145243e-05 / 1.418691e-10
GPT :   58215.196569 s : cg: iteration 91: 2.513932e-05 / 1.418691e-10
GPT :   58215.230340 s : cg: iteration 92: 2.203589e-05 / 1.418691e-10
GPT :   58215.271288 s : cg: iteration 93: 1.903159e-05 / 1.418691e-10
GPT :   58215.317139 s : cg: iteration 94: 1.660660e-05 / 1.418691e-10
GPT :   58215.357441 s : cg: iteration 95: 1.496033e-05 / 1.418691e-10
GPT :   58215.391757 s : cg: iteration 96: 1.444024e-05 / 1.418691e-10
GPT :   58215.426384 s : cg: iteration 97: 1.450029e-05 / 1.418691e-10
GPT :   58215.463675 s : cg: iteration 98: 1.341707e-05 / 1.418691e-10
GPT :   58215.496824 s : cg: iteration 99: 1.219021e-05 / 1.418691e-10
GPT :   58215.531658 s : cg: iteration 100: 1.134627e-05 / 1.418691e-10
GPT :   58215.568427 s : cg: iteration 101: 1.036746e-05 / 1.418691e-10
GPT :   58215.600954 s : cg: iteration 102: 8.374553e-06 / 1.418691e-10
GPT :   58215.636285 s : cg: iteration 103: 6.139290e-06 / 1.418691e-10
GPT :   58215.672240 s : cg: iteration 104: 4.503940e-06 / 1.418691e-10
GPT :   58215.707880 s : cg: iteration 105: 3.242411e-06 / 1.418691e-10
GPT :   58215.744595 s : cg: iteration 106: 2.181438e-06 / 1.418691e-10
GPT :   58215.778360 s : cg: iteration 107: 1.460245e-06 / 1.418691e-10
GPT :   58215.815683 s : cg: iteration 108: 1.059289e-06 / 1.418691e-10
GPT :   58215.849074 s : cg: iteration 109: 7.567011e-07 / 1.418691e-10
GPT :   58215.885919 s : cg: iteration 110: 4.922791e-07 / 1.418691e-10
GPT :   58215.921771 s : cg: iteration 111: 3.507766e-07 / 1.418691e-10
GPT :   58215.956850 s : cg: iteration 112: 2.731455e-07 / 1.418691e-10
GPT :   58215.995727 s : cg: iteration 113: 2.185314e-07 / 1.418691e-10
GPT :   58216.038138 s : cg: iteration 114: 1.865807e-07 / 1.418691e-10
GPT :   58216.078073 s : cg: iteration 115: 1.582089e-07 / 1.418691e-10
GPT :   58216.121324 s : cg: iteration 116: 1.437802e-07 / 1.418691e-10
GPT :   58216.163979 s : cg: iteration 117: 1.461678e-07 / 1.418691e-10
GPT :   58216.197931 s : cg: iteration 118: 1.446564e-07 / 1.418691e-10
GPT :   58216.230854 s : cg: iteration 119: 1.357581e-07 / 1.418691e-10
GPT :   58216.264952 s : cg: iteration 120: 1.277107e-07 / 1.418691e-10
GPT :   58216.300345 s : cg: iteration 121: 1.171745e-07 / 1.418691e-10
GPT :   58216.332755 s : cg: iteration 122: 1.042156e-07 / 1.418691e-10
GPT :   58216.365923 s : cg: iteration 123: 9.019872e-08 / 1.418691e-10
GPT :   58216.398912 s : cg: iteration 124: 7.122664e-08 / 1.418691e-10
GPT :   58216.431416 s : cg: iteration 125: 5.169487e-08 / 1.418691e-10
GPT :   58216.465844 s : cg: iteration 126: 3.785981e-08 / 1.418691e-10
GPT :   58216.498837 s : cg: iteration 127: 2.789881e-08 / 1.418691e-10
GPT :   58216.532891 s : cg: iteration 128: 2.000441e-08 / 1.418691e-10
GPT :   58216.565243 s : cg: iteration 129: 1.497193e-08 / 1.418691e-10
GPT :   58216.600145 s : cg: iteration 130: 1.113561e-08 / 1.418691e-10
GPT :   58216.632918 s : cg: iteration 131: 7.661028e-09 / 1.418691e-10
GPT :   58216.665907 s : cg: iteration 132: 5.563641e-09 / 1.418691e-10
GPT :   58216.698311 s : cg: iteration 133: 4.356843e-09 / 1.418691e-10
GPT :   58216.733328 s : cg: iteration 134: 3.356765e-09 / 1.418691e-10
GPT :   58216.771571 s : cg: iteration 135: 2.584895e-09 / 1.418691e-10
GPT :   58216.809073 s : cg: iteration 136: 2.066487e-09 / 1.418691e-10
GPT :   58216.853396 s : cg: iteration 137: 1.795182e-09 / 1.418691e-10
GPT :   58216.892171 s : cg: iteration 138: 1.618207e-09 / 1.418691e-10
GPT :   58216.935180 s : cg: iteration 139: 1.327603e-09 / 1.418691e-10
GPT :   58216.977170 s : cg: iteration 140: 1.159867e-09 / 1.418691e-10
GPT :   58217.015602 s : cg: iteration 141: 1.058597e-09 / 1.418691e-10
GPT :   58217.058742 s : cg: iteration 142: 9.784659e-10 / 1.418691e-10
GPT :   58217.103906 s : cg: iteration 143: 9.714124e-10 / 1.418691e-10
GPT :   58217.149447 s : cg: iteration 144: 9.651181e-10 / 1.418691e-10
GPT :   58217.187425 s : cg: iteration 145: 8.291485e-10 / 1.418691e-10
GPT :   58217.228496 s : cg: iteration 146: 7.115612e-10 / 1.418691e-10
GPT :   58217.265219 s : cg: iteration 147: 6.191710e-10 / 1.418691e-10
GPT :   58217.304433 s : cg: iteration 148: 5.079246e-10 / 1.418691e-10
GPT :   58217.348382 s : cg: iteration 149: 3.895019e-10 / 1.418691e-10
GPT :   58217.381376 s : cg: iteration 150: 3.022021e-10 / 1.418691e-10
GPT :   58217.416667 s : cg: iteration 151: 2.275574e-10 / 1.418691e-10
GPT :   58217.449164 s : cg: iteration 152: 1.576711e-10 / 1.418691e-10
GPT :   58217.484652 s : cg: iteration 153: 1.067313e-10 / 1.418691e-10
GPT :   58217.630415 s : (-7.891172705659715e-09+2.657322559449351e-09j)
```

In [ ]:

```python
def conserved_current_mobius(psi, bar_psi, mu):
    psi_s = g.separate(psi, 0)
    bar_psi_s = g.separate(bar_psi, 0)
    Ls = len(psi_s)
    b = D_dwf.params["b"]
    c = D_dwf.params["c"]
    m = D_dwf.params["mass"]
    Pp = (g.gamma["I"].tensor() + g.gamma[5].tensor()) * 0.5
    Pm = (g.gamma["I"].tensor() - g.gamma[5].tensor()) * 0.5
    dst = g.lattice(psi_s[0])
    dst[:] = 0
    for s in range(Ls):
        dst += b*D_w.conserved_vector_current(psi_s[s], bar_psi_s[s], mu)
    for s in range(1, Ls):
        dst += c*D_w.conserved_vector_current(g(Pp*psi_s[s-1]), bar_psi_s[s], mu)
    for s in range(Ls-1):
        dst += c*D_w.conserved_vector_current(g(Pm*psi_s[s+1]), bar_psi_s[s], mu)
    dst -= m*c*(
        D_w.conserved_vector_current(g(Pp*psi_s[Ls-1]), bar_psi_s[0], mu)
        + D_w.conserved_vector_current(g(Pm*psi_s[0]), bar_psi_s[Ls-1], mu)
    )
    return dst

def divergence(f, b, current):
    resN = None

    for mu in range(4):
        c_mu = current(f, b, mu)
        res = g(c_mu - g.cshift(c_mu,mu,-1))
        if resN is None:
            resN = res
        else:
            resN += res

    g.message(g.norm2(resN[1,0,0,0]))
    g.message(g.norm2(resN[1,0,0,3]))
    return g.norm2(resN)

g.message("div(J_conserved)",
          divergence(prop, prop_bar, conserved_current_mobius))
```