1 Spherical Harmonics

The most popular, call it standard, definition of the spherical harmonics is

$$Y_\ell^m(\theta,\varphi) \triangleq N_\ell^m\, P_\ell^m(\cos\theta)\, e^{im\varphi}$$

where the normalization is given by

$$N_\ell^m \triangleq \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}$$

2 Numerical considerations

2.1 Normalization

As can be gleaned by looking at the asymptotics of the terms, computing N_ℓ^m and $P_\ell^m(z)$ *separately* causes numerical problems as ℓ becomes large. Empirically, these problems emerge around $\ell = 150$.

Direct computation of *normalized or semi-normalized* associated Legendre functions, which essentially computes $N_\ell^m\, P_\ell^m(z)$ holistically, is a better numerical strategy.

We use the Schmidt semi-normalized Legendre functions because they are implemented in MATLAB as `legendre(l,m,'sch')`. Empirically, if problems emerge then they are for $\ell > 2000$.

2.1.1 Schmidt semi-normalized associated Legendre functions

As defined and implemented in MATLAB, the Schmidt semi-normalized (associated) Legendre functions are:

```
SP(N,M;X) = P(N,X), M = 0
          = (-1)^M * sqrt(2*(N-M)!/(N+M)!) * P(N,M;X), M > 0
```

that is,

$$SP_\ell^m(z) \triangleq \begin{cases} P_\ell^m(z) & m=0 \\ (-1)^m\sqrt{\dfrac{2(\ell-m)!}{(\ell+m)!}}\, P_\ell^m(z) & m>0 \quad (m\le\ell) \end{cases}$$

(available in MATLAB as `legendre(l,m,'sch')`.

In summary, the normalized associated Legendre functions and not the standard associated Legendre functions should be used when computing the spherical harmonics.

2.2 Symmetry

For negative m we note

$$Y_\ell^{-m}(\theta,\varphi) = (-1)^m\, \overline{Y_\ell^m(\theta,\varphi)}$$

so we only need to compute positive orders m. That is, if we have computed $Y_\ell^m(\theta,\varphi)$ for $m>0$ then we can simply determine $Y_\ell^m(\theta,\varphi)$ for $m<0$.

In summary, with a few sign flips we can get the negative order spherical harmonics from the positive ones.

2.3 More robust spherical harmonic computation

Define

$$S_\ell^m(z) \triangleq (-1)^m\sqrt{\frac{2(\ell-m)!}{(\ell+m)!}}\, P_\ell^m(z), \quad m\ge 0 \quad (m\le\ell)$$

and so we have

$$S_\ell^m(z) = \begin{cases} \sqrt{2}\, SP_\ell^0(z) & m=0 \\ SP_\ell^m(z) & m>0 \end{cases} \tag{1}$$

in terms of the Schmidt semi-normalized Legendre functions (available in MATLAB as `legendre(l,m,'sch')`. Further define

$$Q_\ell \triangleq \sqrt{\frac{2\ell+1}{8\pi}}$$

Then we have

$$Y_\ell^m(\theta,\varphi) = \begin{cases} (-1)^m\, Q_\ell\, S_\ell^m(\cos\theta)\, e^{im\varphi} & m\ge 0 \quad (0\le m\le\ell) \\ (-1)^m\, \overline{Y_\ell^{-m}(\theta,\varphi)} & m<0 \quad (-\ell\le m<0) \end{cases}$$

2.3.1 MATLAB implementation

The Schmidt semi-normalized Legendre functions are available in MATLAB so we only need to modify the $m = 0$ case, (**??**). The following code computes $Y_\ell^m(\theta, \varphi)$ for $\ell \geq 0$ and $m \in \{-\ell, \ldots, \ell\}$ on a separable grid in θ and ϕ:

```
function [Ylm,theta,phi]=sphereHarm(l,m,tt,pp)
%sphereHarm spherical harmonic of degree l and order m
   if isvector(tt) && isvector(pp) && abs(m)<=l && l>=0
      tt=tt(:)'; pp=pp(:)'; % ensure they are row vectors
      [theta,phi]=ndgrid(tt,pp);
      Sl=legendre(l,cos(tt),'sch');
      Sl(1,:)=Sl(1,:)*sqrt(2); % m=0 adjustment
      Ql=sqrt((2*l+1)/(8*pi));
      Slm=Sl(abs(m)+1,:)'; % pull out S_l^|m|
      Ylm=(-1)^m*Ql*kron(ones(size(pp)),Slm).*exp(1j*abs(m)*phi);
      if m<0
         Ylm=(-1)^m*conj(Ylm);
      end
   else
      error('** invalid inputs or out-of-range')
   end
end
```