

```

/**
 * MiniPRCoords. Public Domain.
 * WGS (GPS) → GCJ (China Obfs 2002) → BD (Baidu 2009)
 * (35, 105) → (34.999093, 105.003286) → (35.005398, 105.009667)
 * (34, 106) → (33.998424, 106.003994) → (34.004217, 106.010579)
 * https://github.com/Artoria2e5/PRCoords */
let {sqrt, abs, sin, cos, hypot, atan2, PI} = Math

function wgs_to_gcj({lat, lon}) {
  /* 常数 / Constants: */
  let A = 6378245 // Krasovsky 1940/SK-42: a
  let _F = 1 / 298.3 // Krasovsky 1940/SK-42: f
  let EE = 2*_F - _F**2 // e^2 = 2*f - f^2

  /* 偏移 / Shifts: */
  // noise origin: (35, 105); default shift: <300, -100>
  let y = lat - 35, x = lon - 105
  let spi = ((z) => sin(z*PI))
  let Δ_N = -100+2*x+3*y+0.2*y*y+0.1*x*y+0.2*sqrt(abs(x))
    +20/3*(2*spi(6*x)+2*spi(2*x)+2*spi(y)
    +4*spi(y/3)+16*spi(y/12)+32*spi(y/30))
  let Δ_E = 300+1*x+2*y+0.1*x*x+0.1*x*y+0.1*sqrt(abs(x))
    +20/3*(2*spi(6*x)+2*spi(2*x)+2*spi(x)
    +4*spi(x/3)+15*spi(x/12)+30*spi(x/30))

  /* 经纬 / Lat-lon packing: */
  let lat_r = lat * PI / 180
  let common = 1 - EE * sin(lat_r)**2

  // meter → °
  let lat1 = (PI / 180) * (A * (1 - EE)) * common**1.5
  let lon1 = (PI / 180) * (A * cos(lat_r)) / common**0.5

  return {lat: lat + Δ_N/lat1, lon: lon + Δ_E/lon1}
}

```

```

let B_F = 3000/180*PI
let B_DLAT = 0.0060, B_DLON = 0.0065
function gcj_to_bd({lat, lon}) {
  let r = hypot(lat, lon) + 2e-5 * sin(B_F * lat)
  let θ = atan2(lat, lon) + 3e-6 * cos(B_F * lon)

  return {lat: r*sin(θ) + B_DLAT, lon: r*cos(θ) + B_DLON}
}

df = (a, b) => ({lat: a.lat - b.lat, lon: a.lon - b.lon})
function gcj_to_wgs(a) {
  return df(a, df(wgs_to_gcj(a), a))
}

function bd_to_gcj(a) {
  let {lat, lon} = df(a, {lat: B_DLAT, lon: B_DLON})
  let r = hypot(lat, lon) - 2e-5 * sin(B_F * lat)
  let θ = atan2(lat, lon) - 3e-6 * cos(B_F * lon)

  return {lat: r*sin(θ), lon: r*cos(θ)}
}

itery = (fwd, rev) => function(bad) {
  let curr = rev(bad)
  let diff = {lat: Infinity, lon: Infinity}

  while (abs(diff.lat) > 1e-5 || abs(diff.lon) > 1e-5) {
    diff = df(fwd(curr), bad)
    curr = df(curr, diff)
  }

  return curr
}

gcj_to_wgs_exact = itery(wgs_to_gcj, gcj_to_wgs)

```