Petersen 五行阴阳音乐系统—底层音律说明

Abe.Chua（初稿 C）

2025-08-25

本文件为合并版的底层音律说明（初稿 C），整合先前初稿 A 与初稿 B 的数学定义、参数表、实现建议、导出/集成实践、示例与最小实现脚本。文档为独立可读，旨在支撑工程实现（Python / SuperCollider / DAW 集成）、听觉验证与参数扫参。

1 概览与设计目标

本系统以黄金比例为音高生成基准、以 Petersen 图与五行/阴阳为结构化语法，目标是：

- 参数化、可复现的音律生成（15 方位，扩展至 45 音区）。
- 便于导出至 Scala/.scl、MIDI Tuning (.tun/SysEx) 与 MPE 友好输出。
- 支持和弦评分、搜索、实时合成与听觉评估流程。

2 符号与常量（总览）

主要符号与默认常量：

- 黄金比例：$\varphi = \dfrac{1+\sqrt{5}}{2} \approx 1.6180339887$。
- 基频：F_0（默认 220 Hz）。
- 五行角度基位：$\theta_e = 72^\circ \cdot e,\ e \in \{0,\dots,4\}$（0: 金,1: 木,2: 水,3: 火,4: 土）。
- 阴阳极性：$p \in \{-1, 0, +1\}$；阴阳角度偏移：$\Delta\theta$（默认 5°）。
- 折叠基准比（八度选择）：$R_{\text{oct}} \in \{2, \varphi\}$。
- 其他超参：和弦相容宽度 σ、音行跳跃惩罚 α_e、BPM 基准 B、MPE pitch bend range 等。

3 标准参数表（默认值与可搜索范围）

建议在实现配置中明确列出默认值与搜索范围：

- F_0 = 220 Hz （可试 160–440 Hz）
- $\Delta\theta$ = 5° （可试 0°–18°；特殊：10°,18°）
- R_{oct} = 2（可选 φ）
- σ = 50 cents（可试 10–200 cents）
- α_e = 1.0（可试 0.1–5.0）
- MPE pitch bend range 建议 ±200 cents（按需要调整）

4 角度—频率映射（核心公式）

定义音位角度：

$$\theta = \theta_e + p \cdot \Delta\theta.$$

原始比例因子：

$$r_{\mathtt{raw}}(\theta) = \varphi^{\theta/72}.$$

原始频率：

$$f_{\mathtt{raw}} = F_0 \cdot \varphi^{\theta/72}.$$

折叠到期望区间（单一标准区间）：

$$F = f_{\mathtt{raw}} \cdot R_{\mathtt{oct}}^{k}, \qquad k \in \mathbb{Z},$$

其中可取

$$k = -\left\lfloor \log_{R_{\mathtt{oct}}} \frac{f_{\mathtt{raw}}}{F_0} \right\rfloor$$

以保证

$$F \in [F_0,\ F_0 \cdot R_{\mathtt{oct}}).$$

以 cents 表示：

$$\mathtt{cents}(F) = 1200 \log_2 \frac{F}{F_0}.$$

每度的 cents 增量：

$$\Delta_{1^\circ} = \frac{1200}{72} \log_2 \varphi \approx 11.57 \ \mathtt{cents/deg}.$$

实现注意：

- 使用高精度 double 计算 phi 幂与对数，避免累积误差。
- 若选择 $R_{\mathtt{oct}} = \varphi$（“黄金八度”），需在导出/播放链路中处理非常规八度映射。

5 15 方位与 45 音区

定义：

- 15 方位：五行（5）× 三极性（3）= 15 个音位，索引 (e,p)。
- 45 音区：复制 15 方位到低/中/高区，音区缩放因子 $s \in \{1/\varphi, 1, \varphi\}$（或 $2^{-1}, 1, 2$）生成更丰富的音高集合。

示例（F_0=220Hz，$\Delta\theta = 5^\circ$，$R_{\mathtt{oct}} = 2$）——折叠后频率均落入 [220,440)：

- 金 (e=0)
  - p = -1: F ≈ 425.07 Hz, cents ≈ 1140
  - p = 0: F = 220.00 Hz, cents = 0
  - p = +1: F ≈ 227.47 Hz, cents ≈ 58
- 木 (e=1)
  - p = -1: F ≈ 344.28 Hz, cents ≈ 775
  - p = 0: F ≈ 355.99 Hz, cents ≈ 833
  - p = +1: F ≈ 368.20 Hz, cents ≈ 891
- 水 (e=2)
  - p = -1: F ≈ 278.70 Hz, cents ≈ 410
  - p = 0: F ≈ 287.98 Hz, cents ≈ 468
  - p = +1: F ≈ 297.66 Hz, cents ≈ 523
- 火 (e=3)

  - p = -1: F ≈ 225.27 Hz, cents ≈ 42
  - p = 0: F ≈ 232.98 Hz, cents ≈ 99
  - p = +1: F ≈ 242.75 Hz, cents ≈ 170

- 土 (e=4)
  - p = -1: F ≈ 364.25 Hz, cents ≈ 871
  - p = 0: F ≈ 376.98 Hz, cents ≈ 933
  - p = +1: F ≈ 388.78 Hz, cents ≈ 986

45 音区通过对上表分别乘以 $s \in \{1/\varphi, 1, \varphi\}$（或 $2^{-1}, 1, 2$）生成；建议导出 CSV/Scala/.tun 三种格式以便比对与集成。

6 导出格式与 DAW/合成器集成实践

建议同时支持三类输出：

1. Scala (.scl)：用于静态音阶比较、格式通用。
2. MIDI Tuning Standard (.tun) 或 SysEx：用于支持自定义微分音调律的合成器。
3. MPE + per-voice pitch-bend fallback：当合成器支持 MPE 时，每声部使用独立 pitch-bend；否则用 base MIDI note + pitch-bend。策略建议：
   - 计算目标频率与最邻近 MIDI note（12-TET）差值（cents）。
   - 若绝对差值 $\leq$ 100 cents，分配该 MIDI note 并通过 pitch-bend 调整；否则选择不同基准 note/八度组合以压缩到 pitch-bend 范围。
   - 默认 pitch-bend range = ±200 cents；如需更大范围，动态调整或采用 MPE 每声部方案。

7 和弦评分、搜索与约束（实现细化）

两音间相对于小整数比 $(n:m)$ 的差异：

$$\delta_{ij}^{(n:m)} = \left| 1200 \log_2 \frac{F_i}{F_j} - 1200 \log_2 \frac{n}{m} \right|.$$

取

$$\delta_{ij} = \min_{(n:m)\in\mathcal{S}} \delta_{ij}^{(n:m)},$$

常用集合 $\mathcal{S} = \{1:1, 2:1, 3:2, 4:3, 5:4\}$（可扩展）。和弦评分：

$$S(C) = \sum_{i<j} \exp\Big(-\frac{\delta_{ij}^2}{2\sigma^2} \Big).$$

实现建议：

- 在计算 δ_{ij} 时同时记录对应的小整数比（n:m）用于分析与可视化。
- 对多音和弦引入基频对齐惩罚（若两个音接近简单倍频关系则加分）。
- 搜索采取分层策略：先在同一五行/相邻五行中枚举候选，再按和弦分数 S(C) 排序，最后做时间窗口内平滑性二次筛选以避免瞬时 cluster。

8 旋律生成与 Petersen 图语法

系统将 Petersen 图或其它图结构作为音程/转移约束：

- 将 15 方位映射到图节点，使用图的边定义允许的音高转移（例如邻接优先、重复惩罚、五行生克偏好）。
- 概率模型示例：
$$P(\Delta e, \Delta p) \propto \exp\big(-\alpha_e|\Delta e| - \alpha_p|\Delta p|\big) \cdot W(e', p'),$$
其中 W 为音色/和谐权重，α_e, α_p 为跳跃惩罚参数。
- 优先小步移动（同一五行三极性或相邻五行），偶尔长跳以增加张力。

9 实时合成、抗失真与实现注意

实时实现注意点：

- 频率变化平滑（portamento/滑音）：对频率包络应用分段线性或低通滤波，避免瞬时大跳引起 aliasing。
- 采样率与插值：使用至少 48 kHz；对 pitch bend 引起的频率变化使用带线性相位低通滤波器的 resampler。
- 预计算 LUT：在实时环境中预生成 15/45 音频频率表与 pitch bend 值，降低运行时负载。
- 合成器 patch 建议（映射五行到音色）：
  - 金：bell FM / high-q resonant filter, bright harmonic boost。
  - 木：plucked sample / physical model, transient + body noise。
  - 水：pad + portamento, low-pass with subtle chorus。
  - 火：brass/lead with saturation, shorter release。
  - 土：low oscillator, heavy LP filter, long decay。

10 可视化、调试与听觉评估流程

推荐最小可行试验（MVP）流程：

1. 使用默认参数生成 15/45 的 Scala 与 MIDI 文件（见附录最小脚本）。
2. 在受控条件下进行主观 A/B 听测（建议 20–30 位受试者），并记录偏好。
3. 同步记录客观指标：平均偏差（cents）、最大偏差、与常见小整数比的分布。
4. 用热力图显示 Petersen 图上节点被访问频度、和弦评分分布与路径转移矩阵。

11 测试、参数搜索与自动化建议

建议的自动化策略：

- 使用网格或贝叶斯优化搜索 $\Delta\theta$, σ, α_e, R_{oct}。
- 每组参数自动生成一组 MIDI/Audio，记录客观指标并进行批量 A/B（或对照 12-TET）听测。
- 保存所有结果与元数据（参数 JSON + timestamp + render 文件名），便于复现与分析。

12 元数据、引用与许可证

请在项目中添加术语表与参考文献（例如 Bohlen–Pierce, Sevish, microtonal tuning 文献），并声明代码与数据许可证（建议 MIT + CC-BY 组合以兼容科研与艺术共享）。

13 附录 A：实现注意的细节与实践清单

实现时的快速清单：

- 使用 double 精度计算 phi 的幂与对数，避免累积误差。
- 折叠到区间 $[F_0, F_0 \cdot R_{\text{oct}})$ 时使用上文公式计算 k。
- 当导出 cents 时采用基准 F0 的相对 cents：$\text{cents} = 1200 \log_2(F/F_0)$。
- 提前生成 LUT 与 Scala/.tun 的 mapping，测试合成器对非常规八度（若使用 φ 八度）的支持。

14 附录 B：生成与导出最小 Python 脚本（工程化示例）

下列脚本为最小可运行示例：生成 15 音位表，导出 CSV 与 Scala (.scl)。将脚本保存在 tools/generate_tuning.py 并在 macOS 终端运行 python3。

```
# filepath: tools/generate_tuning.py
# 运行: python3 tools/generate_tuning.py
import math, csv

phi = (1+5**0.5)/2
F0 = 220.0
dtheta = 5.0
R_oct = 2.0

elements = ['金','木','水','火','土']
rows = []

def raw_freq(theta):
    return F0 * (phi ** (theta/72.0))

def fold_freq(f):
    k = -math.floor(math.log(f / F0, R_oct))
    return f * (R_oct ** k)

def cents(f):
    return 1200*math.log2(f / F0)

for e_idx, name in enumerate(elements):
    theta_e = 72.0 * e_idx
    for p in (-1,0,1):
        theta = theta_e + p*dtheta
        f_raw = raw_freq(theta)
        F = fold_freq(f_raw)
        rows.append({
            'element': name,
            'e': e_idx,
            'p': p,
            'theta': theta,
            'freq': round(F,6),
            'cents': round(cents(F),3)
        })

# CSV
with open('tools/15_positions.csv','w', newline='') as f:
    w = csv.DictWriter(f, fieldnames=['element','e','p','theta','freq','cents'])
    w.writeheader()
```

```
    w.writerows(rows)

# Scala .scl (single octave relative to F0)
with open('tools/petersen_15.scl','w') as f:
    f.write("! petersen_15.scl\n")
    f.write("Petersen 五行阴阳 15-tone scale (F0=220Hz, dtheta=5deg)\n")
    f.write("15\n")
    for r in rows:
        cents_val = r['cents']
        f.write(f"{cents_val}\n")
print("Generated tools/15_positions.csv and tools/petersen_15.scl")
```

15 结语与下一步建议

本合并稿（初稿 C）整合数学定义、参数表、实现细化、导出策略与最小脚本。下一步建议：

1. 将附录脚本放入 tools 并运行以验证 15/45 导出；
2. 在支持 MPE 或.tun 的 DAW/合成器中进行对照听测；
3. 启动参数扫参并记录结果（自动化保存 metadata）。