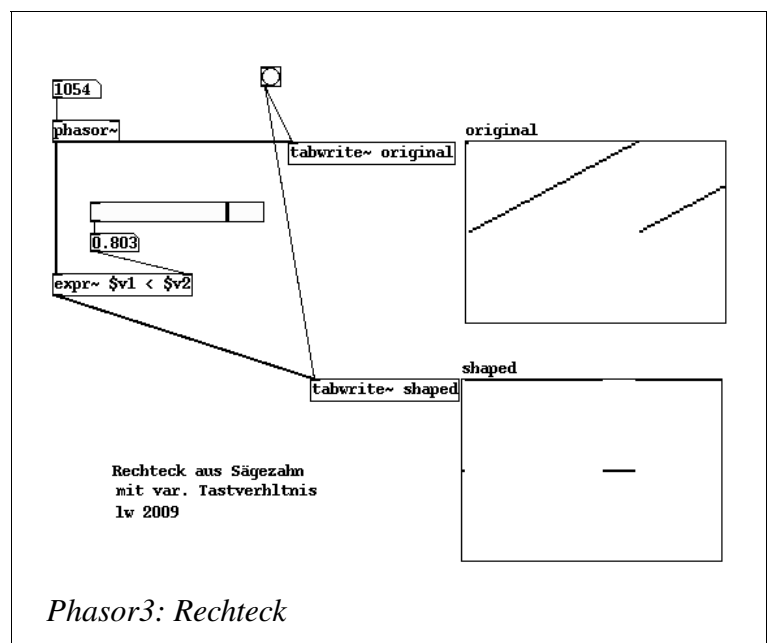
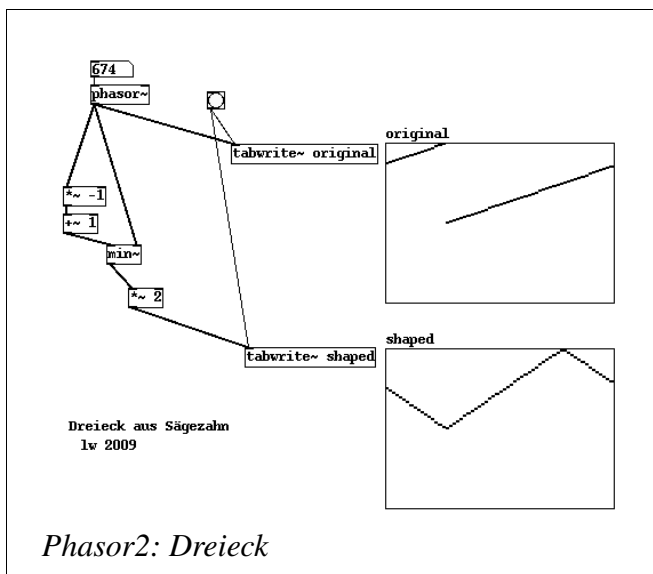
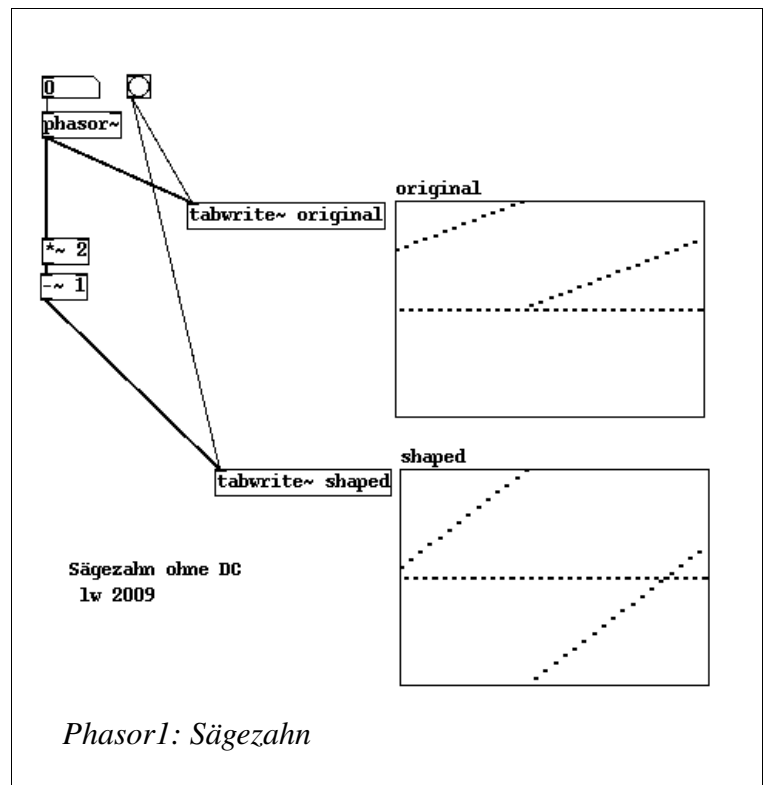


Bei diesen Puredata-Übungen sollen Möglichkeiten gezeigt werden, auf möglichst einfachem Weg häufig benötigte Signalformen abzuleiten. Als Quelle dient jeweils ein [phasor~]-Objekt, das eine Folge linearer Rampen mit einstellbarer Frequenz erzeugt. [phasor~] ist also eine Art Sägezahngenerator, wobei aber darauf zu achten ist, dass das erzeugte Signal nicht bandbreitenbegrenzt ist: der Rücksprung von 1 auf 0 erfolgt mit unendlicher Steilheit, das Signal besitzt daher sicher Anteile oberhalb der Nyquist-Grenze. Bei Nutzung als Audioquelle muss also mit Artefakten gerechnet werden. Gleiches gilt für die hier abgeleiteten Signale.

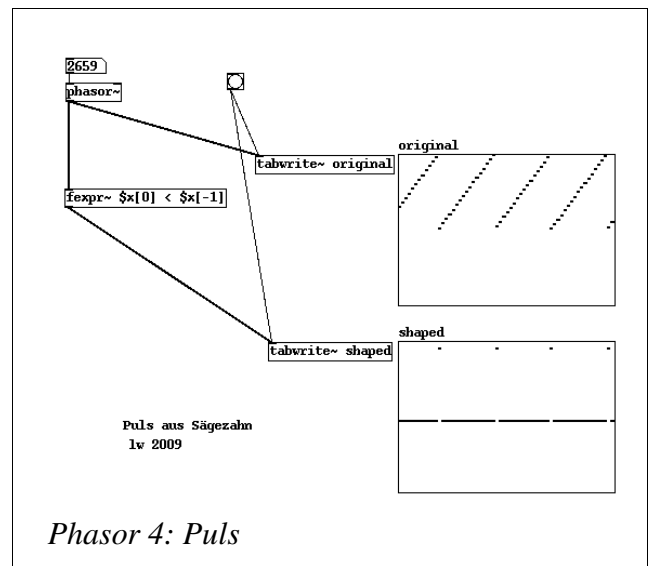
1. Sägezahn: der Ausgang von [phasor~] liefert Rampen im Wertebereich 0 bis 1. Benötigt man eine “gleichspannungsfreie” Rampe, also eine mit zeitlichem Mittelwert Null, muss das Signal verschoben und auf die gewünschte Amplitude gebracht werden. Dazu reichen einfache arithmetische Objekte für den Signalbereich, cf sigbinops-help in der Pd-Hilfe.

2. Dreieck: hierzu benötigen wir eine fallende Rampe, die wir am einfachsten durch Multiplikation mit -1 aus der steigenden ableiten. Diese liegt im Wertebereich [-1, 0], so dass wir 1 addieren und den jeweils kleineren Wert übernehmen. Wir erhalten ein Dreieck im Wertebereich [0, 0.5].



3. Rechteck: durch Vergleich des Rampenwertes mit einem festen Schwellwert erhält man das Rechteck. Das Tastverhältnis (Einschaltzeit/Periode) lässt sich durch Wahl des Vergleichswertes zwischen 0 und 1 festlegen. Vergleichsoperatoren für den Signalbereich stehen in Pd nicht direkt zur Verfügung, daher muss der Vergleich in ein [expr~]-Objekt eingebettet werden.

4. Puls: möchte man ein Signal erhalten, das in jeder Periode genau einen 1-Wert gibt, ansonsten 0, so lässt sich das leider nicht einfach derart erreichen, dass man im vorigen Rechteckpatch den Vergleichswert auf 0 setzt, da der Wert 0 in den meisten Blöcken nicht vorkommt (wovon man sich leicht mittels [print~] überzeugen kann). Man kann aber den Rücksprung des Sägezahns dadurch detektieren, dass nur an dieser Stelle ein kleinerer Wert auf einen größeren folgt. Hierzu benötigt man [fexpr~], das Samplewerte innerhalb eines Datenblocks verarbeiten kann.



5. Frequenzvervielfachung: eine sehr einfache Möglichkeit liefert das [wrap~]-Objekt, das für jeden Eingangswert die Differenz zur nächstkleineren ganzen Zahl angibt, für positive Werte also die Nachkommastellen. Daher reicht es, die Rampe mit dem gewünschten Faktor zu multiplizieren und durch [wrap~] zerstückeln zu lassen. Auch [expr~] bietet die entsprechende Funktion.

*Man überlege sich, wie diese abgeleiteten Funktionen an Stelle der ursprünglichen Rampe einzusetzen sind, z. B. beim Auslesen von Wellenformen.*

