Beginn: 13:00 Uhr
Ort: Arbeitgruppenraum C2.227
Anwesend: Alle Mitglieder der Projektgruppe
Als Tagespunkte lagen vor:

- Erste Analyseergebnisse der Rechenkern–Gruppe
- Beschluß über das zu verwendende Interface
- Bisherige Ergebnisse der anderen Gruppen, sowie die weitere Vorgehensweise der einzelnen Gruppe

Zu Beginn der heutigen Sitzung wurde ein kleiner Fehler in der Ausarbeitung über Splineinterpolation von Ralf Jungblut korrigiert. Dort steht auf Seite 9

$$c_i = \frac{1}{h_i}(y_{i+1} - y_i) - h_i(y''_{i+1} + \frac{2}{6}y''_i)$$

Korrekt muß es aber lauten:

$$c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{6}h_i(y''_{i+1} + 2y''_i)$$

Daran anschließend stellte die Rechenkern–Gruppe ihre ersten Ergebnisse der Analyse des Spline–Algorithmus', der von Jürgen Hochwald implementiert wurde, vor. An Hand von diesem Algorithmus wurden erste Vermutungen über den eventuell benötigten Speicherplatz und über die benötigten Rechenwerke aufgestellt. Dabei orientierte sich die Gruppe an den Operationen, die während des Ablaufs des Algorithmus' vorkommen. Zuerst wurde den Projektgruppenteilnehmern eine Auflistung des eventuell benötigten Speicherplatzes präsentiert, die im folgenden aufgelistet werden:

- Eine Matrix der Größe 14x13 vom Typ float. Die geringere Größe dieser Matrix kommt dadurch zustande, daß innerhalb der Matrix Zeilen bzw. Spalten gleich Null sind.
- 3x15 Speicherplätze für die berechneten Koeffizienten vom Typ float.
- Weiterhin die Speicherung der 15 X–Werte vom Typ float. Dies wurde so gemacht, damit Register für die eigentlichen Rechenoperationen frei werden.
- Zwischenspeicher vom Typ float, der für den Zeilenaustausch benötigt wird.
- 3 Schleifenvariablen vom Typ integer. Diese Anzahl genügt, da im Verlauf des Algorithmus' maximal 3 Schleifen in einander geschachtelt waren.
- Insgesamt wurde folgender Speicherplatz errechnet:
  - 3 Speicherplätze vom Typ integer
  - 243 Speicherplätze vom Typ float

Zu dem Bedarf des Speicherplatzes wurden die folgenden Anmerkungen gemacht, und zwar, daß es auch möglich sei die Matrix, die durch die Stützstellen gegeben wird, auf eine Drei–Band–Matrix zu reduzieren. Dies hätte dann zur Folge, daß ein erhöhter Aufwand bei der Indexberechnung betrieben werden müsste. Anschließend stellte die Gruppe die Anzahl der benötigten Rechenwerke vor. Es wurde dabei so vorgegangen, daß die wichtigsten C–Funktionen des Algorithmus' auf die Anzahl der gleichzeitig auftretenden Rechenoperationen untersucht wurden. Daraus resultierend wurden die Zahlen für die einzelnen Rechenwerke ermittelt. Die Aufstellung der erwünschten Rechenwerke erfolgte mittels einer Tabelle und der Angabe der jeweiligen Prozeduren. Die folgende Tabelle repräsentiert die so gewonnenen Ergebnisse.

Rechenwerk	FillMat	Loese	Exchange	MatMul	MatAdd	MakeSpline
Add (int)	1	1	1	1	1	1
Add (float)						
Sub (int)		2				1
Sub (float)	1				1	1
Mult (int)						
Mult (float)	1			1-15	1-15	
Div (int)						
Div (float)	1	1			1	3
> (int)	1	1	1	1	1	1
== 0 (float)		1	1			
≠ (int)		1	1			

Die Namen der einzelnen Spalten bezeichnen jeweils den zugehörigen Namen der Funktion des C–Programmes. Zu der Tabelle sei angemerkt, daß einige Werte unrealistisch sind, da es nicht möglich sein wird 15 Multiplizierer auf unserem Coprozessor unterzubringen. Als mögliche Anzahl wurde hier 2–3 genannt. In diesem Zusammenhang fiel auch der Begriff des kombinatorischen Multiplizierers, von dem aber dann nur einer realisiert werden kann. Der Vorteil dieses Multiplizierers wäre, daß er Berechnungen schneller durchführen kann. Weitere Anmerkungen zu obenstehender Tabelle waren, daß einige Rechenwerke durch andere simuliert werden könnten, so z.B. kann eine Multiplikation vom Typ float mit einer konstanten Zahl durch eine entsprechende Mehrfach–Addition vom Typ float realisiert werden. Ebenso könnte die Subtarktion der Form $i - 2$ durch einen Zähler realisiert werden. Bei Multiplikationen bzw. Divisionen mit 2, könnten dieser Berechnungen mit Hilfe von shift–Operationen schneller gelöst werden, wobei der Exponent auf einen möglichen Overflow untersucht werden muß. Bei dieser Analyse stellte sich heraus, daß es an manchen Stellen auch noch Optimierungsmöglichkeiten gibt, vor allem was die Anzahl der einzelnen Rechenwerke betrifft. Hier wurde insbesondere Synopsis angesprochen, das die Anzahl der Rechenwerke optimiert. In einem anschließenden Schritt könnte man jedoch versuchen "in eigener Regie" weitere Rechenwerke hinzu zu nehmen.

Es folgte nun eine allgemeine Diskussion zu der Kommunikation zwischen Rechenkern und Interface, sowie welches Interface im weiteren Verlauf des Projektes benutzt werden soll. Als wichtigste Punkte kristallisierten sich hier heraus, daß die Kommunikation zwischen Rechenkern und Interface so ablaufen soll, wie im P30–Paper der Rechenkern–Gruppe beschrieben wurde. Ferner wurde besprochen, daß unser Hauptaugenmerk in Sachen Interface auf das von Wolfram Hardt liegt. Im Rahmen dieser Diskussion wurde von der Interface–Gruppe angemerkt, daß es nötig sei eine eigene Controll–Unit mit Zugriff auf den Bus zu realisieren. Des weiteren kann es zu einer Erhöhung der Anzahl der Register auf 128 anstatt der bisher veranschlagten 32 kommen. Diese Zahl setzt sich dabei wie folgt zusammen:

- je 32 Register für Input und Output
- 2x32 Register für den Rechenkern

Am Ende dieser Diskussion wurde folgender Beschluß gefaßt:
Beschluß: Es wird das Interface von Wolfram Hardt übernommen, simuliert und entsprechend unseren Anforderungen angepaßt.

Es folgten die Berichte der einzelnen Gruppen zum Stand ihrer Arbeiten, sowie dem weiteren Vorgehen für die nächste Zeit:

1. **Rechenkern**: Für eine konkretere Analyse müsste der Gruppe einen genaueren Algorithmus vorliegen haben. Als nächstes Ziel wird jedoch erst einmal angestrebt, daß die Gruppe sich darüber klar wird, wie die einzelnen arithmetischen Rechenwerke zu realisiert werden können.
2. **Algorithmus**: Es erfolgt eine weitere Einarbeitung in Speedchart. Ferner sollen Source–Codes von verschiedenen Spline–Interpolationsprogrammen gesichtet werden.
3. **Interface**: Diese Gruppe hat sich bereits mit der in vorangegangenen Sitzungen besprochenen Taktverzögerung beim Anhalten der IU befaßt und man hat sich auch hierzu eine Realisierung überlegt. Außerdem wurde vorgeschlagen, daß diese Verzögerung in unseren Coprozessor eingebaut werden soll, da es dort leichter ist, als in der IU, die wir nach Möglichkeit unberührt lassen wollen. Für die nächste Zeit ist eine weitere Vertiefung in diese Problemstufe vorgesehen.

Ende: ca. 14:15 Uhr