Curriculum Vitae

David Bryan Carpenter

September 7, 1995

Personal Details

Date of Birth: 15th January, 1958

Place of Birth: Birmingham, UK

Marital Status: Single

First Degree: BSc., Physics, from Westfield College, London.
First Class Honours. Degree awarded 1979.

Second Degree: PhD., Theoretical High Energy Physics, also from Westfield.
Supervisor: Prof. E. Leader. Degree awarded 1984.

Thesis: *Lattice and Large N Studies of Two-dimensional QCD*, submitted to University of London, 1983.

Positions:

1983-1984 Research Fellow in Theoretical Physics Department, Edinburgh University.

1985 Held Royal Society Overseas Fellowship at DESY, Hamburg.

1985-1988 Research Fellow in Computational Physics Group, Southampton.

1989 Employed at "Transputer Technology Solutions", Southampton.

1989-91 Research Fellow in Department of Electronics and Computer Science, Southampton (funded by Esprit Project PUMA).

1991-92 Research Fellow jointly in Department of Electronics and Computer Science and Department of Physics, Southampton.

1993 Research Fellow in Department of Electronics and Computer Science, Southampton.

1994-95 Programmer, High Performance Computing Centre, Southampton.

Interests and Experience

Parallel computing, parallel languages, library design, object-oriented programming (C++), scientific programming, functional programming, logic programming, formal methods, UNIX system programming.

Career Summary

Physics

My doctoral research, in theoretical physics, involved applying newly-developed computational techniques for handling lattice fermions to a low-dimensional field theory [1]. For the first two years of my postdoctoral career I worked as a physicist, engaged in more or less theoretical work on lattice gauge theories and field theory [2] [3] [4]. At Edinburgh I worked in David Wallace's group and was exposed to one of the early massively parallel computers—the ICL DAP.

I came to Southampton in 1985, a recruit to Tony Hey's new computational physics group. Initially working as an application programmer, I spent several years implementing physics codes on parallel computers—especially Transputer systems. In close collaboration with *inmos* we wrote some of the first scientific programs to run on Transputers. Between 1989 and 1992 I collaborated with Prof Chris Sachrajda and a group of European physicists on first-principles calculation of parameters of heavy quark systems [13]. Earlier collaborations with physicists involved simulation of solid state [5] [9] and pure gauge theories [10]. During this time, however, my principal interests moved progessively away from physics, towards computational issues.

The Supernode Project

For a number of years I have been particularly interested in design and development of communication libraries to support data-parallel computation (and in the associated language issues).

This interest started during the ESPRIT *Supernode* project, which developed a reconfigurable multi-Transputer supercomputer. I worked on programming paradigms for exploiting Transputer networks in simulation of physical systems [6]. The deliverables for that project included *occam* libraries and harnesses to facilitate distribution of grid-oriented problems [7].

The PUMA Project

During the subsequent *PUMA* (Parallel Universal Message-passing Architectures) project I designed and prototyped a more advanced collective communication library, *ACL*[14]. The PUMA project was centred on the belated T9000 Transputer; when that processor was delayed ACL was prototyped on top of

the Southampton Virtual Channel Router (VCR). The PUMA project was also strongly influenced by the ideas of Bulk Synchronous Parallel (BSP) computation which emerged at that time, and the choice of some of the collective communication primitives in ACL was influenced by the BSP model. Other features of the ACL model are high-level primitives for dynamically spawning multi-dimensional process arrays, and the use of abstract data types to describe process arrays and distributed data. My later work on Adlib design was particularly influenced this latter feature of ACL. ACL was initially implemented in *occam*—although a simplified version was later implemented in Fortran, on top of the Parmacs library.

In the same project John Merlin was also working at Southampton on the *ADAPT* system, a precursor HPF-like system, for semi-automatic parallelisation of Fortran programs with distributed data. He proposed a collective communication library called ADLIB, which his translation tool would target. The ADAPT system was not completed during the project, and ADLIB, as specified then, remained unimplemented.

Computer Science

After the PUMA project finished I returned to physics collaboration for a year. Meanwhile my interests in communication libraries shifted mainly to lower level-support for some of the dynamic channel communication operations used in the VCR implemention of ACL. I worked towards establishing a CSP proof of a particular channel-moving protocol. This work is unpublished.

In late 1992 I worked briefly with Hugh Glaser on the FAST (Functional programming on Arrays of Transputers). We implemented a few scientific codes in the lazy functional language Haskell, with a view, ultimately, to their automatic parallelisation[15].

HPF and Adlib

In 1993 HPF became fashionable, and John Merlin and myself began a collaboration to implement a prototype HPF translation system, recycling components of the ADAPT system to build an HPF-to-Fortran translator, and technology from ACL to implement the run-time library. Some time into this project we obtained formal funding to continue from the UK JISC and EPSRC councils.

In the early stages of project I worked on defining a translation scheme for HPF-like languages, with a view to defining a convenient (and complete) interface to the run-time library. Some phases of this translation scheme were prototyped in the logic language Prolog. The initial specification of the new run-time library, *Adlib*, was completed in 1994. Adlib was defined as a C++ class library. Unlike ACL, which was designed primarily to support an *ad hoc* set of common applications and algorithms, Adlib is designed specifically to support translation of HPF. It supports the full HPF distributed data model, providing

high-level gather/scatter and other data remapping operations on HPF arrays. It also provides a set of arithmetic and reduction operations suitable for implementing the Fortran 90 array intrinsics on distributed data.

A prototype implementation of Adlib, built on MPI, was completed in the last year[16]. Adlib is implemented in C++. A restricted Fortran interface, suitable as a target for the modified ADAPT translator, has also been provided.

The subset HPF translation system SHPF (Southampton HPF) has been installed at several beta test sites over the past few months. The major platforms have been SUN and IBM workstations and the IBM SP2, but it should be possible to retarget the system to any platform with a suitable set of compilers which provides MPI.

References

[1] D.B. Carpenter, *Two Dimensional Lattice QCD in Quenched Approximation.* Nucl. Phys. B228 (1983) 365.

[2] D.B. Carpenter and C.F. Baillie, *Free Fermion Propagators and Lattice Finite-Size Effects.* Nucl. Phys. B260 (1985) 103.

[3] D.B. Carpenter, *Blocking Transformations for Lattice Fermions.* DESY Preprint 85-064.

[4] S.J. Hands and D.B. Carpenter, *Lattice Sigma Model and Fermion Doubling.* Nucl. Phys. B266 (1986) 285.

[5] C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, D.A. Nicole and D.J. Pritchard, *Simulation of Statistical Mechanical Systems on Transputer Arrays.* Computer Physics Communications 42 (1986) 21.

[6] C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, M. Moore, D.A. Nicole and D.J. Pritchard, *Monte Carlo Simulation on Transputer Arrays.* Parallel Computing 6 (1988) 247.

[7] D.B. Carpenter and J.H. Merlin, *Gauge Field Theory Simulation on Transputer Arrays.* ESPRIT project P1085, Workpackage 7, 3rd deliverable report (1989).

[8] J.R.A. Allwright and D.B. Carpenter, *A Distributed Implementation of Simulated Annealing for the Travelling Salesman Problem.* Parallel Computing 10 (1989) 335.

[9] D.B. Carpenter and J.T. Chalker, *The Phase Diagram of a Generalised XY Model.* J. Phys: Condens. Matter I (1989) 4907.

[10] B. Carpenter, C. Michael and M.J. Teper, *O^+ and 2^+ Glueball Masses from Large Lattices in SU(2) Lattice Gauge Theory.* Phys. Lett. B198 (1987) 511.

[11] C.R. Allton, D.B. Carpenter, V. Lubicz, L. Maiani, G. Martinelli and C.T. Sachrajda, *Decay Constants of Heavy Mesons.* Nuclear Physics B (Proc. Suppl.) 20 (1991) 504.

[12] A. Abada, C.R. Allton, Ph. Boucard, D.B. Carpenter, M. Crisafulli, J. Garland, S. Güsken, G. Martinelli, O. Pène, C.T. Sachrajda, R. Sarno, K. Schilling and R. Sommer, *Meson Spectroscopy and Decay Constants with Wilson Fermions at* $\beta = 6.4$ Nucl.Phys. B376 (1992) 172.

[13] As. Abada, C.R. Allton, Ph. Boucard, D.B. Carpenter, M. Crisafulli, S. Güsken, P. Hernadez, V. Lubicz, G. Martinelli, O. Pène, C.T. Sachrajda, K. Schilling, G. Siegert and R. Sommer, *Meson Spectroscopy and Decay Constants with Wilson Fermions at* $\beta = 6.4$ Nucl.Phys. B416 (1994) 675.

[14] D.B. Carpenter, *Array Communication Library: Introduction.* ESPRIT project PUMA, Workpackage 5.1, report (1992).

[15] D.B. Carpenter and H. Glaser, *Some Lattice-Based Scientific Problems, expressed in Haskell University of Southampton, Department of Electronics and Computer Science.* Southampton Preprint CSTR 93-06 (1993). To appear, Journal of Functional Programming (1996).

[16] D.B. Carpenter, Adlib*: A Distributed Array Library to Support HPF Translation* presented at 5th Workshop on Compilers on Parallel Computer, Malaga (1995).